Lab 1 Wireshark

Rodrigo Mansilla 22611
Nelson Escalante – 22046
Redes

Introducción:

Este laboratorio tiene como objetivo introducir los conceptos fundamentales de la comunicación y transmisión de información mediante dispositivos. Para lograrlo, se realizaron dos actividades: la simulación de esquemas de comunicación utilizando códigos como Morse y Baudot, y una introducción al uso de la herramienta Wireshark, la cual nos permitirá analizar paquetes de red en tiempo real.

A través de estas actividades podremos reflexionar sobre los retos que existen a la hora de la transferencia de datos y la comunicación de dispositivos en entornos de red, así como las implicaciones que tiene el hacer esta comunicación de manera segura, rápida y eficiente.

Descripción:

Esta práctica está dividida en dos partes: Una simulación de comunicación codificada y el análisis de redes digitales utilizando Wireshark.

Para la primera parte se utilizaron dos métodos de codificación que han sido importantes a lo largo de la historia: el código Morse y el código de Baudot. Se realizó una simulación de emisión y recepción de mensajes codificados utilizando nuestras voces mediante una llamada de zoom. Luego de esto, realizamos el envío de mensajes “empaquetados” utilizando notas de voz de WhatsApp. Por último, se realizó una simulación de conmutación de mensajes en conjunto con otro grupo, donde un compañero fue asignado el rol de conmutador y los demás enviaban y recibían mensajes a través de él.

Parte 1 (En clase)

1.1 Transmisión de códigos:

- Código Morse:
  - Mensaje 1: "hola mundo"
    - .... --- .-.. .- / -- ..- -. -.. ---
  - Mensaje 2: "universidad"
    - ..- -. .. ...- . .-. ... .. -.. .- -..
  - Mensaje 3: "computadora"
    - -.-. --- -- .--. ..- - .- -.. --- .-. .-
- Código Baudot:
  - Mensaje 1: "hola mundo"
  - 10101 01110 10110 10001 00111 11010 01110 10100 10110 10011
  - Mensaje 2: "universidad"
  - 00111 01100 00110 11110 00001 01010 00101 00110 01001 00011 01001
  - Mensaje 3: "computadora"
  - 01110 11000 11100 10110 00111 10000 00011 01001 11000 01010 00011
  -

  ¿Qué esquema (código) fue más fácil de transmitir y por qué?¿Qué esquema (código) fue más difícil de transmitir y por qué?
  ¿Qué esquema tuvo menos errores (incluir datos que lo evidencien)?

El código Morse fue más fácil de recordar y transmitir, especialmente porque es más conocido y es más intuitivo asociar puntos y rayas con letras. El código Baudot fue más difícil, ya que requiere memorizar la secuencia de bits, y es más fácil confundirse al pronunciar varias secuencias de ceros y unos.

1.2 Transmisión “empaquetada”:

- Mensaje 1: "wifi seguro"
  - .-- .. ..-. .. / ... . --. ..- .-. ---
- Mensaje 2: "laboratorio"
  - .-...- -... --- .-. .- - --- .-. .. ---
- Mensaje 3: "ingenieria"
  - .. -. --. . -. .. . .-. .. .-

¿Qué dificultades involucra el enviar un mensaje de esta forma “empaquetada”?
El receptor no podía hacer preguntas o pedir aclaraciones, lo que generó más errores de interpretación. Se perdió la retroalimentación inmediata y hubo que esperar a que el receptor escuchara la nota de voz y respondiera.

Conmutación de mensajes: Estructura de Conmutación

Estructura de Conmutación

```
R
M
C
B/J T
B/J F NO RECIBO MENSAJES B/J T RECIBO MENSAJES
B/J R MENSAJE C
0 O
1 A
– OOO
– o
```

- Mensajes: Persona 1
  - Mensaje
    - OOAAO AOAAO AAOAO AAAOO AAOAA AOOOO OAOAA
  - Codificación en Baudot:
    - 00110 10110 11010 11100 11011 1000 01011
  - Traducción
    - BHOLAM
  - Persona 2
  - Mensaje
    - AOOAA OOAAO
  - Codificación en Baudot:
    - 10011 00110
  - Traducción
    - KB

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

La introducción de un conmutador permite segmentar la red, reduciendo colisiones y mejorando el rendimiento. Además, el conmutador filtra el tráfico enviándolo solo al destinatario correcto, optimizando el uso del ancho de banda y facilitando la

expansión y el control del tráfico.

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

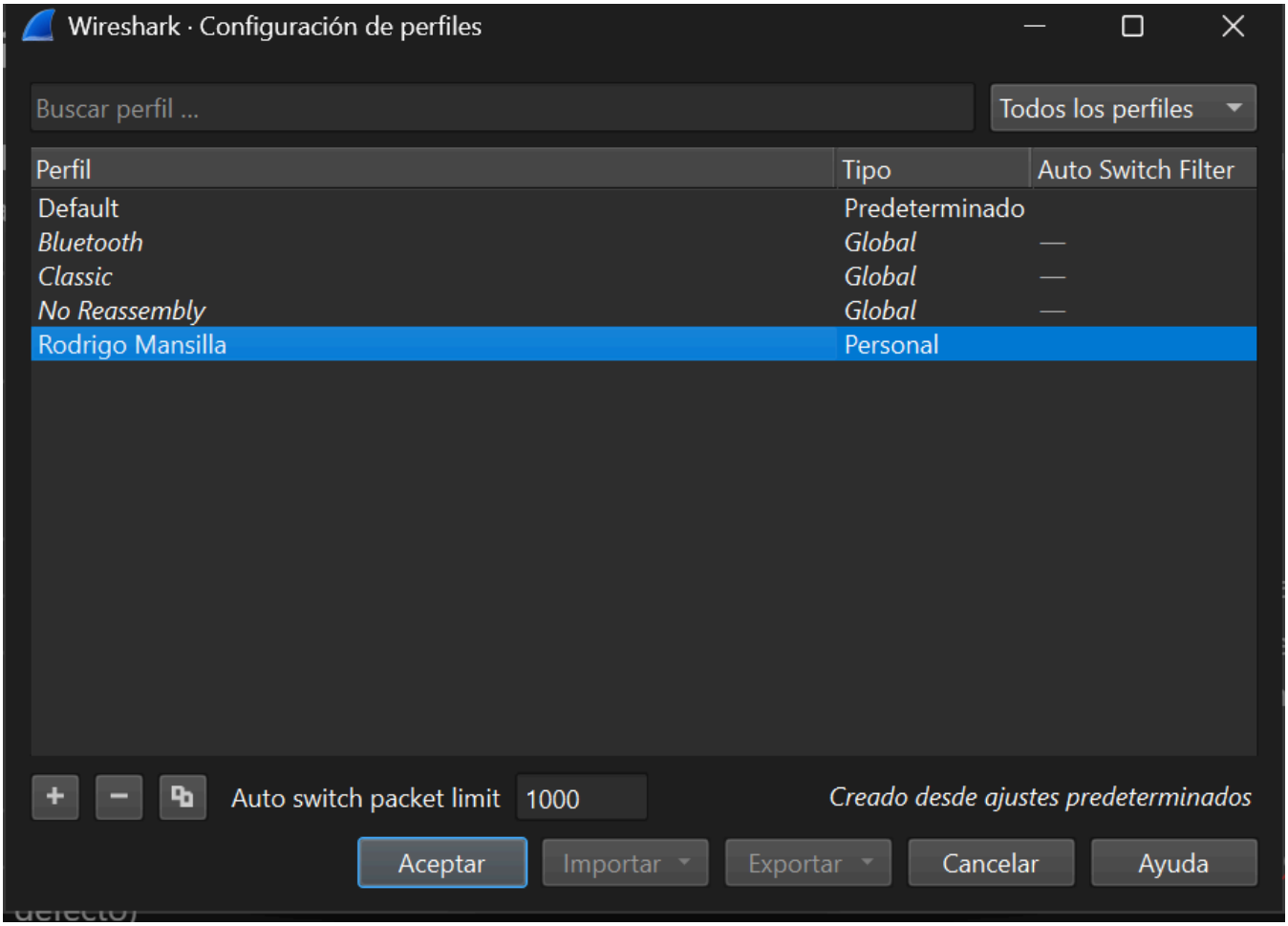
Agregar más conmutadores incrementa la capacidad para conectar dispositivos y mejora la organización de la red, ayudando a reducir la congestión al limitar el tráfico dentro de segmentos específicos. Sin embargo, también aumenta la complejidad de la gestión y puede generar retrasos si no se configura adecuadamente. En nuestro proceso, a pesar de tener una estructura clara, la traducción y transmisión de mensajes no se realizó correctamente, causando pérdida de información y errores frecuentes. Esto dificultó que los mensajes llegaran de forma clara al destinatario, resaltando la importancia de una configuración precisa cuando se trabajan múltiples conmutadores.

Parte 2

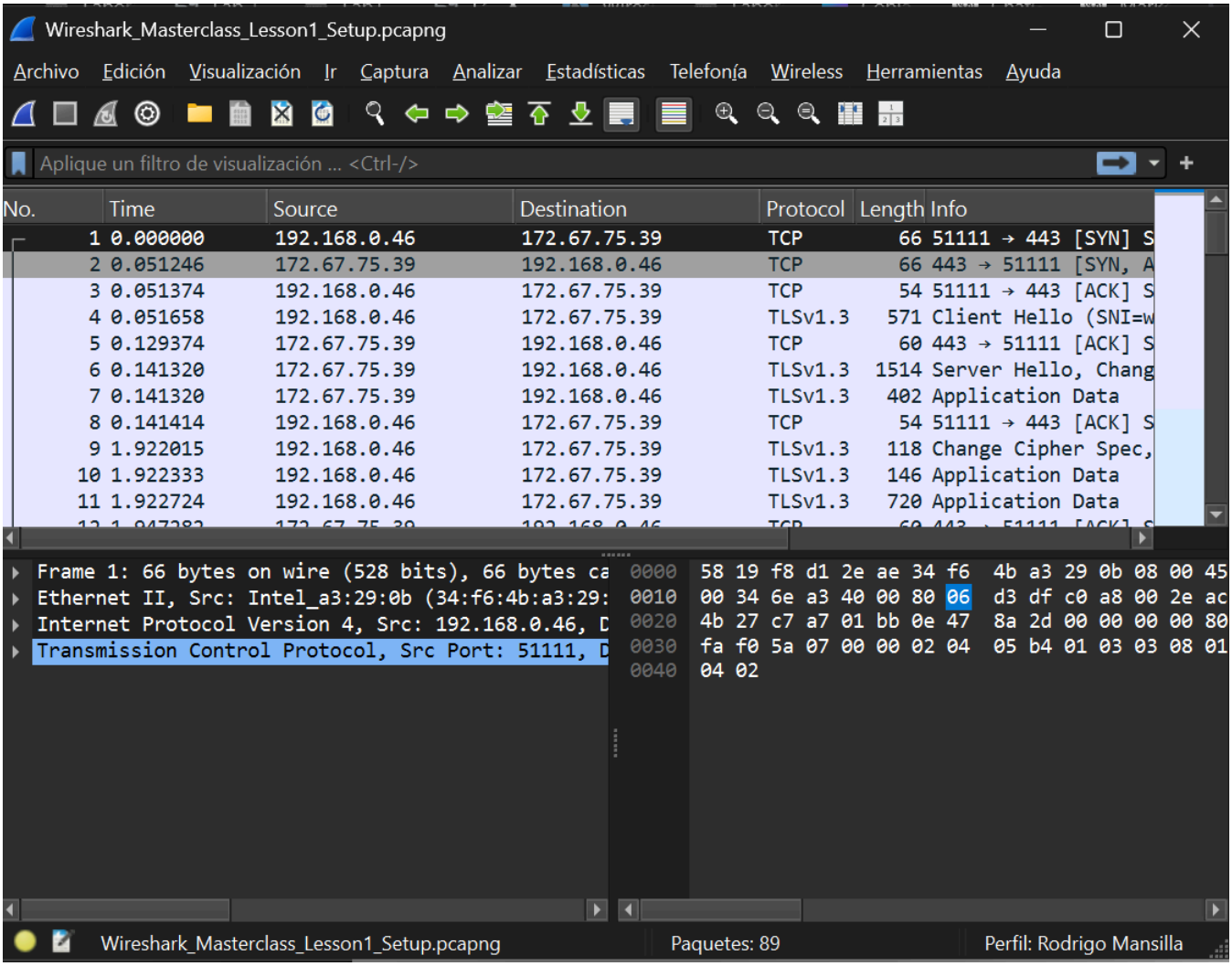
Primera parte: personalización del entorno

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

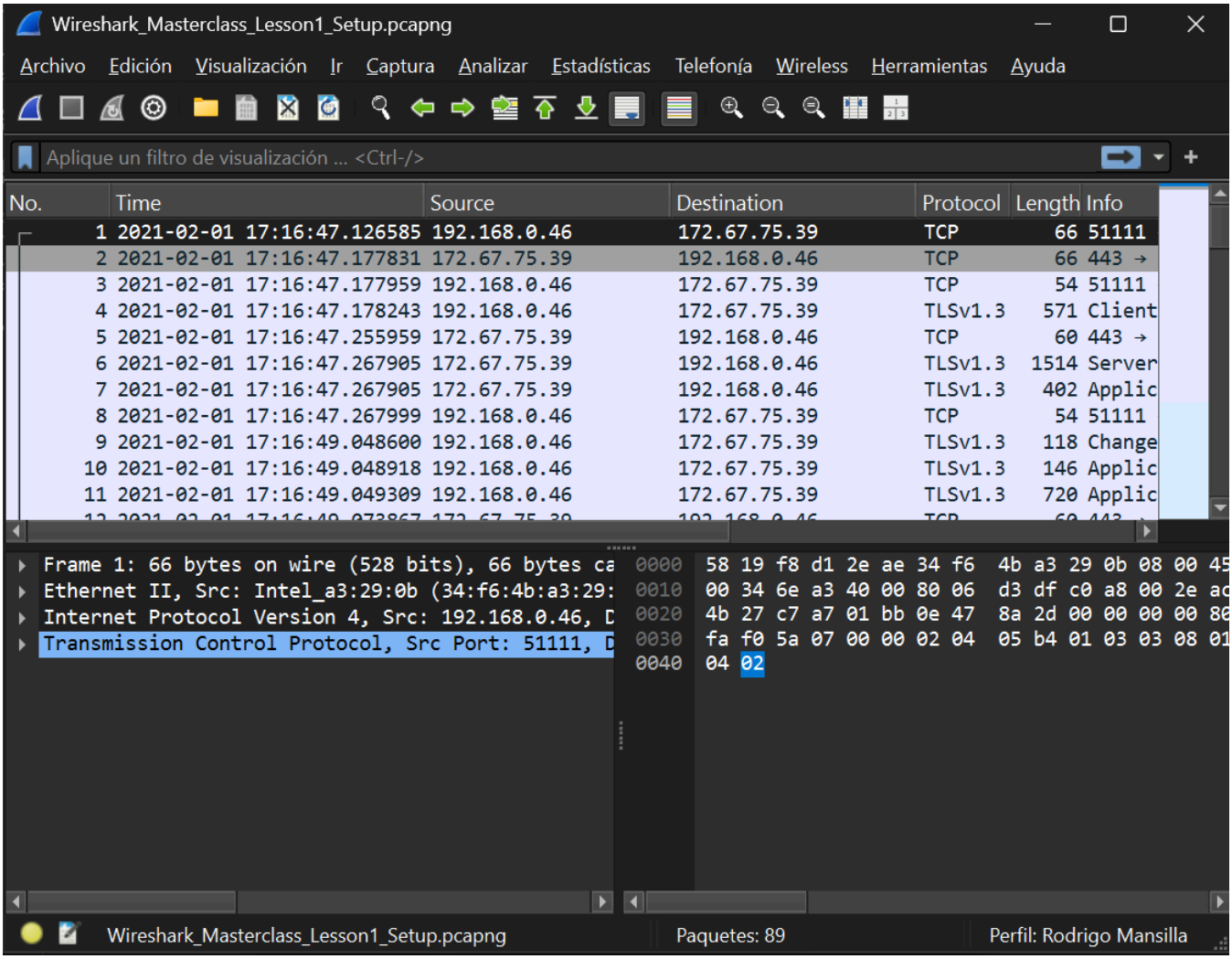


3. Abra el archivo descargado (File -> Open)

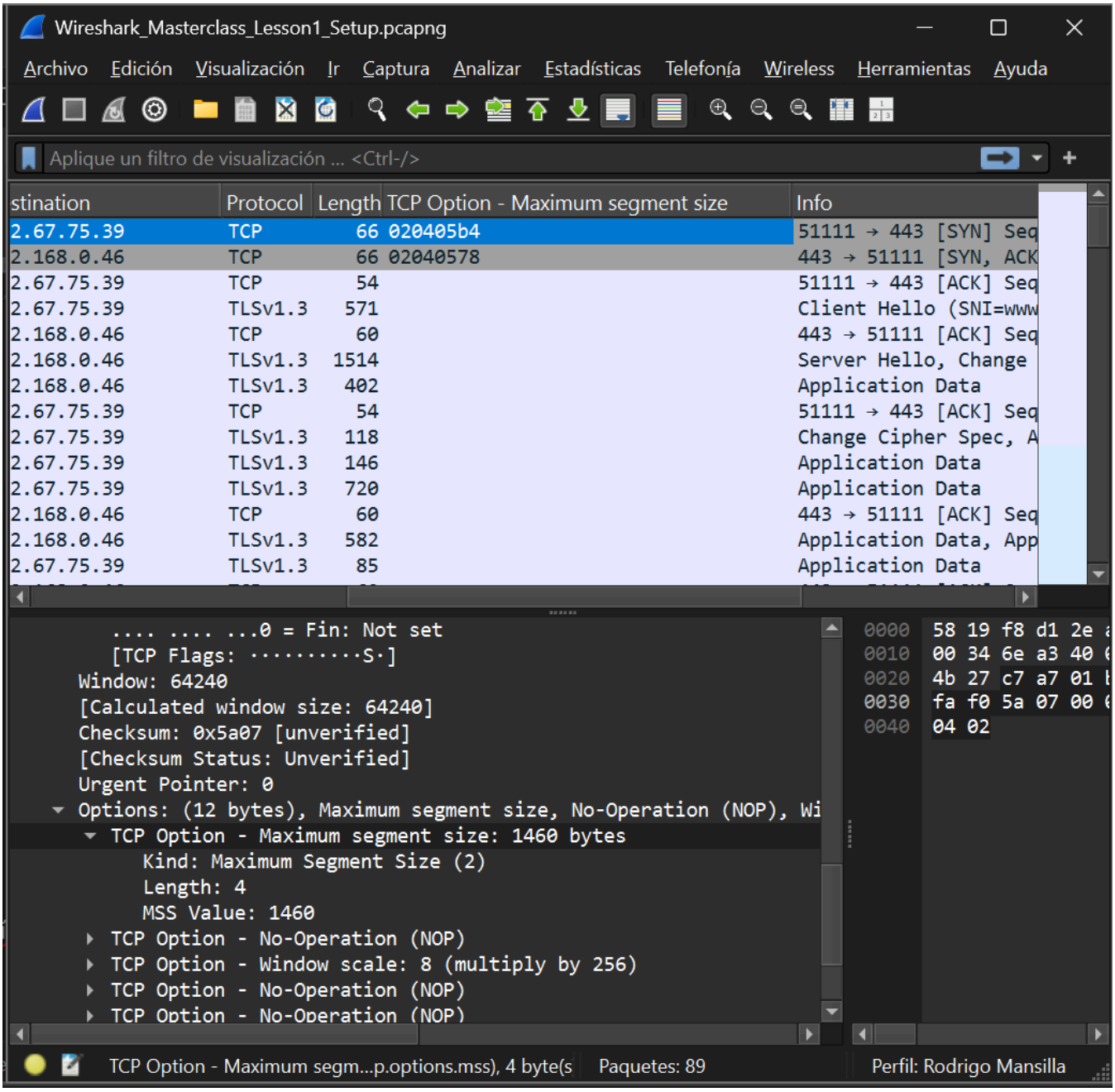

Wireshark_Masterclass_Lesson1_Setup.pcapng
Archivo Edición Visualización Ir Captura Analizar Estadísticas Telefonía Wireless Herramientas Ayuda
Aplique un filtro de visualización ... <Ctrl-/>
No. Time Source Destination Protocol Length Info
1 0.000000 192.168.0.46 172.67.75.39 TCP 66 51111 → 443 [SYN] S
2 0.051246 172.67.75.39 192.168.0.46 TCP 66 443 → 51111 [SYN, A
3 0.051374 192.168.0.46 172.67.75.39 TCP 54 51111 → 443 [ACK] S
4 0.051658 192.168.0.46 172.67.75.39 TLSv1.3 571 Client Hello (SNI=w
5 0.129374 172.67.75.39 192.168.0.46 TCP 60 443 → 51111 [ACK] S
6 0.141320 172.67.75.39 192.168.0.46 TLSv1.3 1514 Server Hello, Chang
7 0.141320 172.67.75.39 192.168.0.46 TLSv1.3 402 Application Data
8 0.141414 192.168.0.46 172.67.75.39 TCP 54 51111 → 443 [ACK] S
9 1.922015 192.168.0.46 172.67.75.39 TLSv1.3 118 Change Cipher Spec,
10 1.922333 192.168.0.46 172.67.75.39 TLSv1.3 146 Application Data
11 1.922724 192.168.0.46 172.67.75.39 TLSv1.3 720 Application Data
Frame 1: 66 bytes on wire (528 bits), 66 bytes ca
Ethernet II, Src: Intel_a3:29:0b (34:f6:4b:a3:29:
Internet Protocol Version 4, Src: 192.168.0.46, D
Transmission Control Protocol, Src Port: 51111, D
0000 58 19 f8 d1 2e ae 34 f6 4b a3 29 0b 08 00 45
0010 00 34 6e a3 40 00 80 06 d3 df c0 a8 00 2e ac
0020 4b 27 c7 a7 01 bb 0e 47 8a 2d 00 00 00 00 80
0030 fa f0 5a 07 00 00 02 04 05 b4 01 03 03 08 01
0040 04 02
Wireshark_Masterclass_Lesson1_Setup.pcapng
Paquetes: 89
Perfil: Rodrigo Mansilla

4.

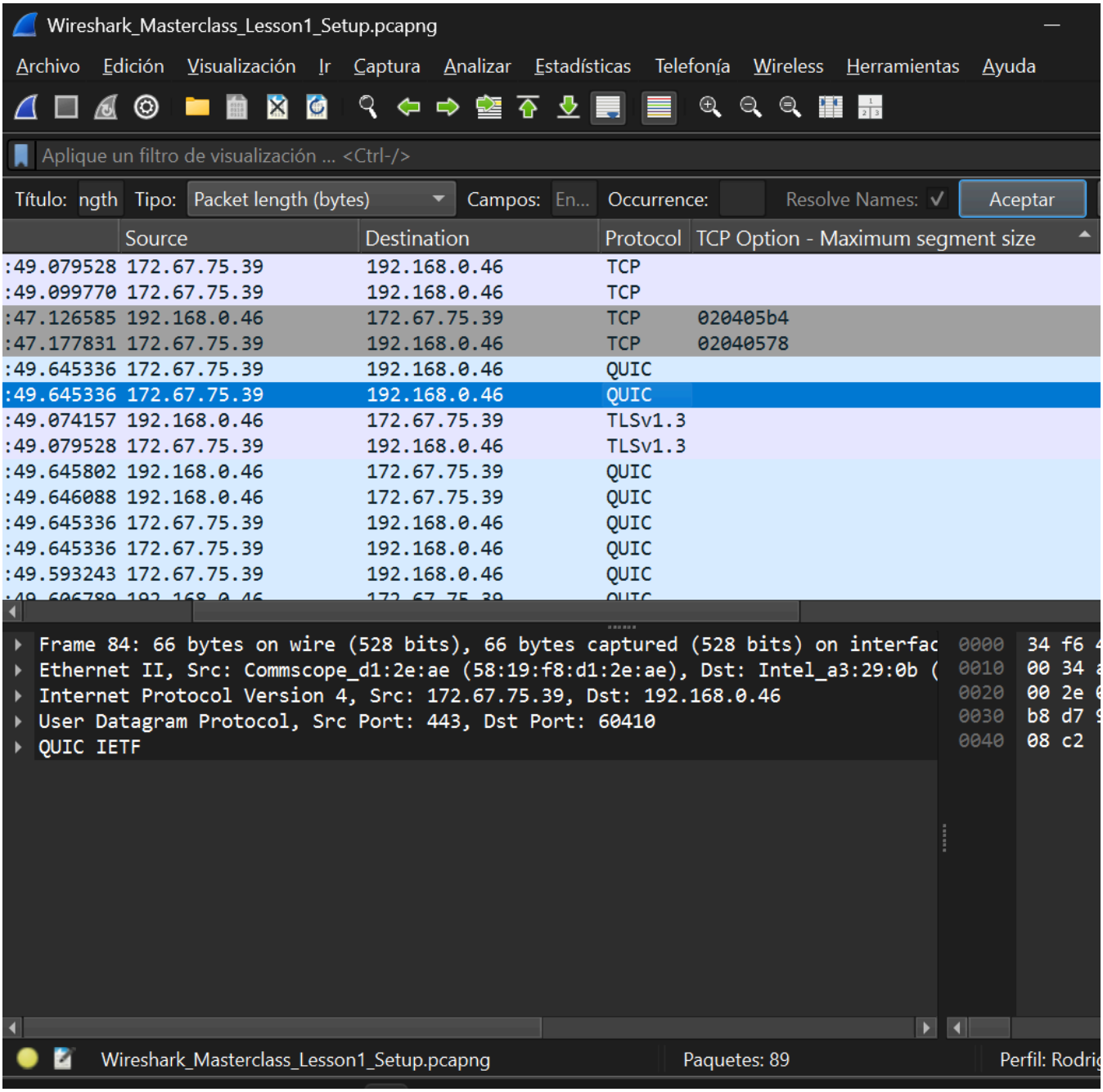
5. Aplique el formato de tiempo Time of Day a la columna Tiempo (View -> Time Display)



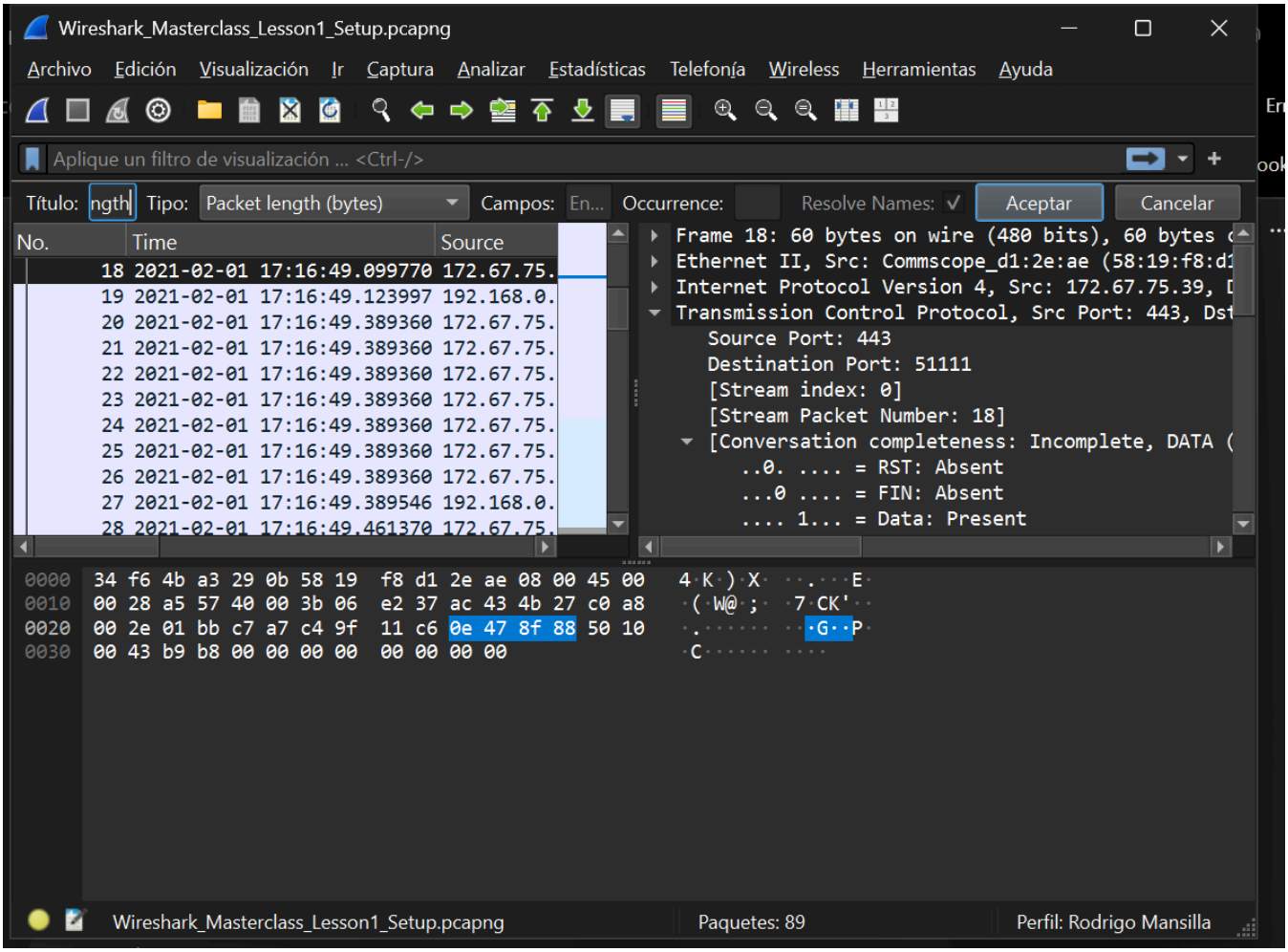
6. Agregue una columna con la longitud del segmento TCP (Selecciona la primera fila, en el panel inferior despliegue Transmission Control Protocol, seleccione TCP Segment Len y apliquelo como una columna)


Wireshark_Masterclass_Lesson1_Setup.pcapng
Archivo Edición Visualización Ir Captura Analizar Estadísticas Telefonía Wireless Herramientas Ayuda
Aplique un filtro de visualización ... <Ctrl-/>
stination Protocol Length TCP Option - Maximum segment size Info
2.67.75.39 TCP 66 020405b4 51111 → 443 [SYN] Seq
2.168.0.46 TCP 66 02040578 443 → 51111 [SYN, ACK
2.67.75.39 TCP 54 51111 → 443 [ACK] Seq
2.67.75.39 TLSv1.3 571 Client Hello (SNI=www
2.168.0.46 TCP 60 443 → 51111 [ACK] Seq
2.168.0.46 TLSv1.3 1514 Server Hello, Change
2.168.0.46 TLSv1.3 402 Application Data
2.67.75.39 TCP 54 51111 → 443 [ACK] Seq
2.67.75.39 TLSv1.3 118 Change Cipher Spec, A
2.67.75.39 TLSv1.3 146 Application Data
2.67.75.39 TLSv1.3 720 Application Data
2.168.0.46 TCP 60 443 → 51111 [ACK] Seq
2.168.0.46 TLSv1.3 582 Application Data, App
2.67.75.39 TLSv1.3 85 Application Data
....0 = Fin: Not set
[TCP Flags: ··········S·]
Window: 64240
[Calculated window size: 64240]
Checksum: 0x5a07 [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
Options: (12 bytes), Maximum segment size, No-Operation (NOP), Wi
TCP Option - Maximum segment size: 1460 bytes
Kind: Maximum Segment Size (2)
Length: 4
MSS Value: 1460
TCP Option - No-Operation (NOP)
TCP Option - Window scale: 8 (multiply by 256)
TCP Option - No-Operation (NOP)
TCP Option - No-Operation (NOP)
0000 58 19 f8 d1 2e
0010 00 34 6e a3 40
0020 4b 27 c7 a7 01
0030 fa f0 5a 07 00
0040 04 02
TCP Option - Maximum segm...p.options.mss), 4 byte(s
Paquetes: 89
Perfil: Rodrigo Mansilla

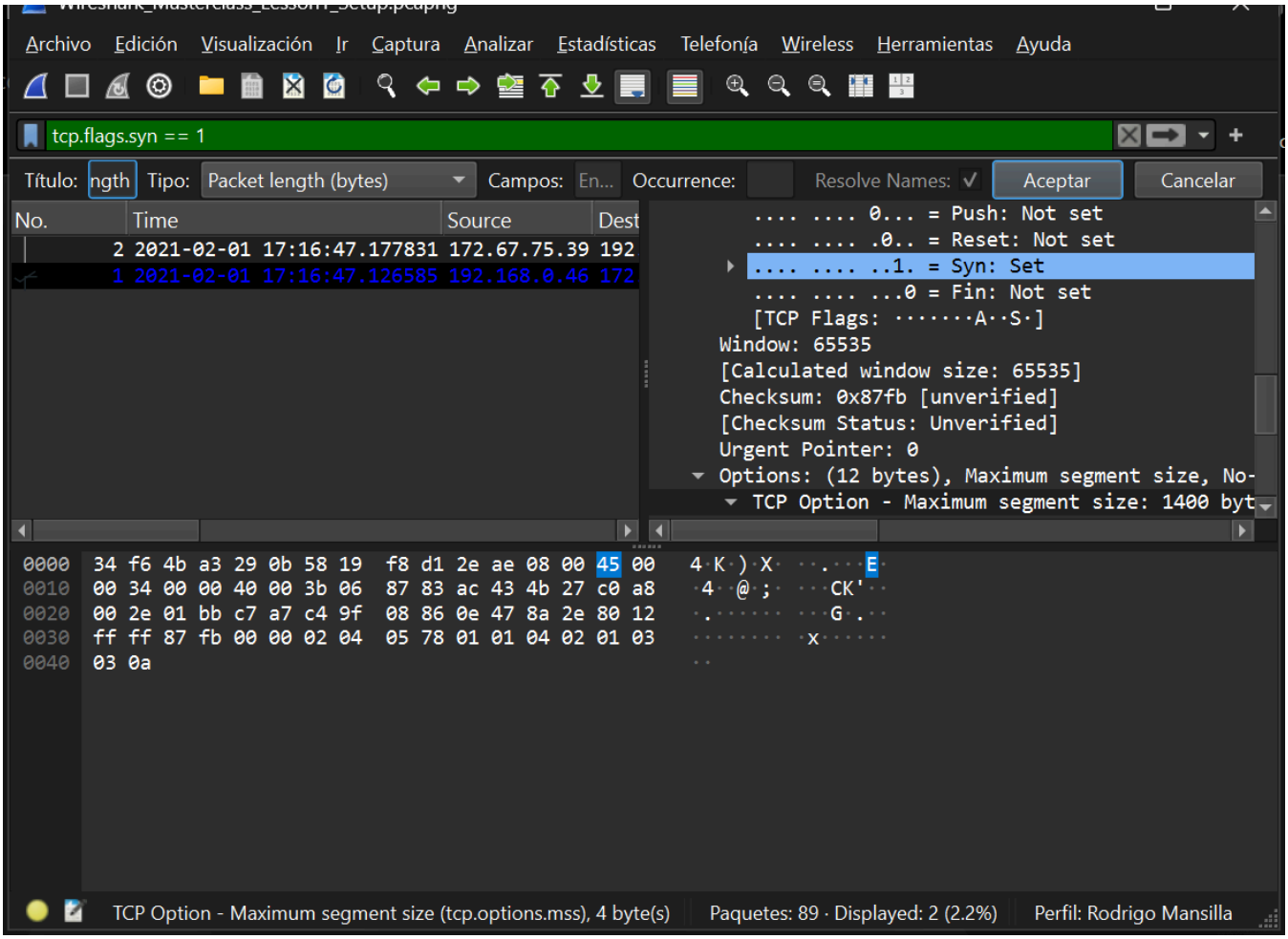
7. Elimine u oculte la columna Longitud



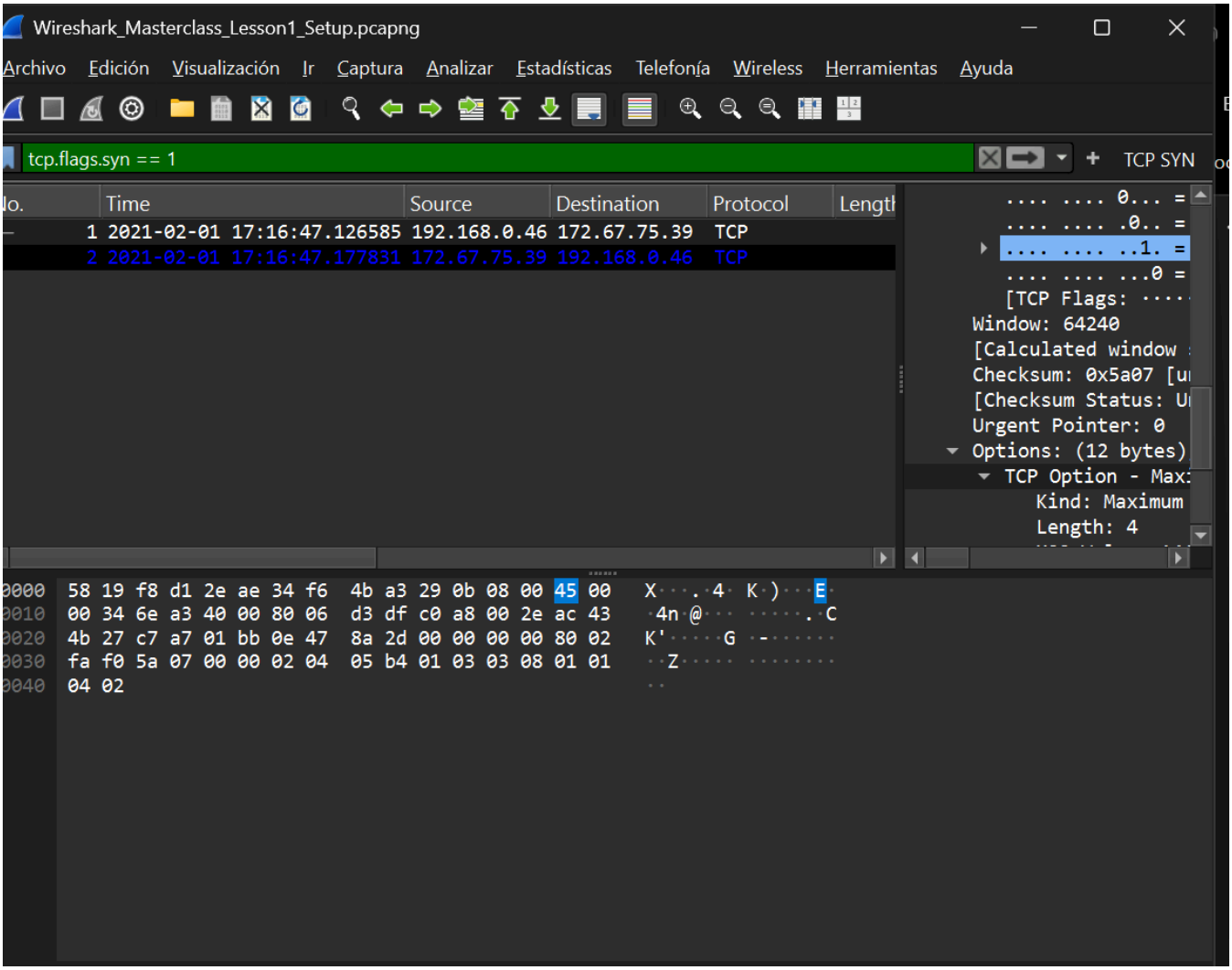
8. Aplique un esquema de paneles que sea de su preferencia (que no sea el esquema por defecto)



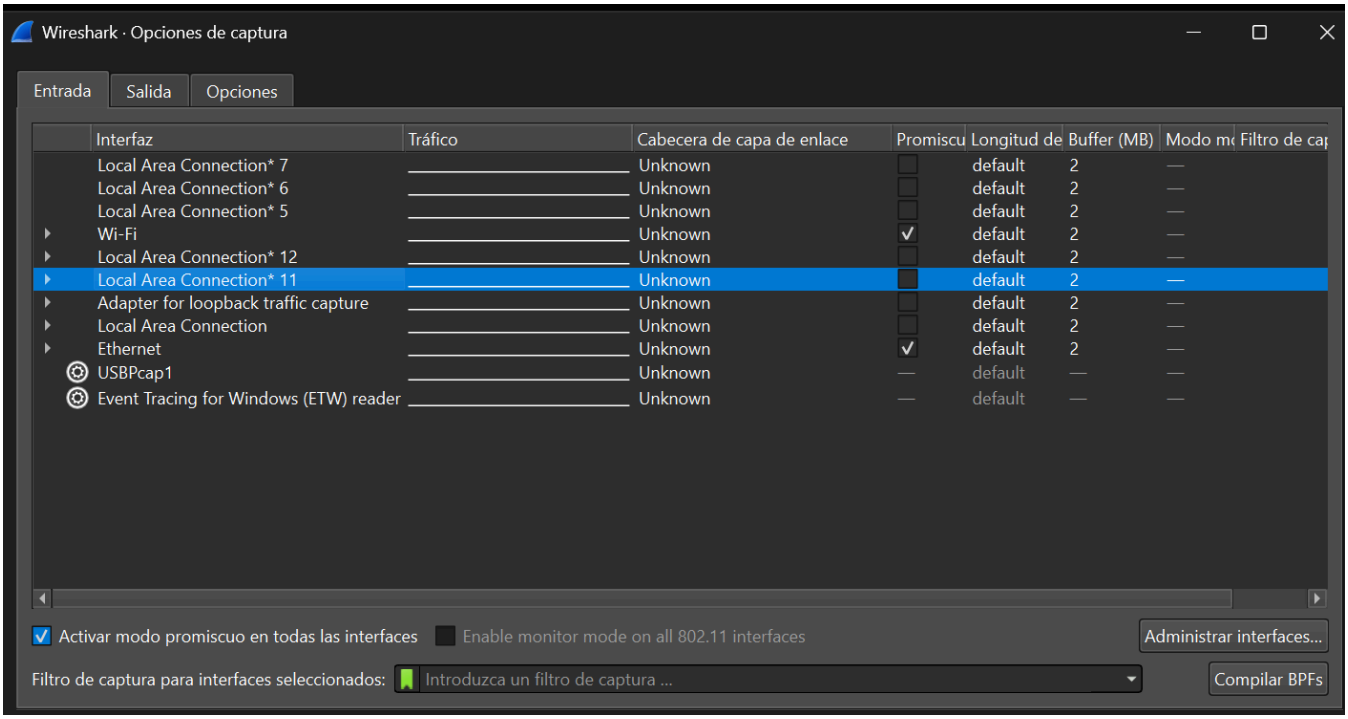
9. Aplique una regla de color para los paquetes TCP cuyas banderas SYN sean iguales a 1, y coloque el color de su preferencia (View -> Coloring Rules)


Archivo Edición Visualización Ir Captura Analizar Estadísticas Telefonía Wireless Herramientas Ayuda
tcp.flags.syn == 1
Título: ngth Tipo: Packet length (bytes) Campos: En... Occurrence: Resolve Names: Aceptar Cancelar
No. Time Source Dest
2 2021-02-01 17:16:47.177831 172.67.75.39 192
1 2021-02-01 17:16:47.126585 192.168.0.46 172
.... 0... = Push: Not set
.... .0.. = Reset: Not set
....1. = Syn: Set
....0 = Fin: Not set
[TCP Flags: ·····A··S·]
Window: 65535
[Calculated window size: 65535]
Checksum: 0x87fb [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
Options: (12 bytes), Maximum segment size, No-
TCP Option - Maximum segment size: 1400 byt
0000 34 f6 4b a3 29 0b 58 19 f8 d1 2e ae 08 00 45 00
0010 00 34 00 00 40 00 3b 06 87 83 ac 43 4b 27 c0 a8
0020 00 2e 01 bb c7 a7 c4 9f 08 86 0e 47 8a 2e 80 12
0030 ff ff 87 fb 00 00 02 04 05 78 01 01 04 02 01 03
0040 03 0a
TCP Option - Maximum segment size (tcp.options.mss), 4 byte(s)
Paquetes: 89 · Displayed: 2 (2.2%)
Perfil: Rodrigo Mansilla

10. Cree un botón que aplique un filtro para paquetes TCP con la bandera SYN igual a 1.



11. 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 las 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.

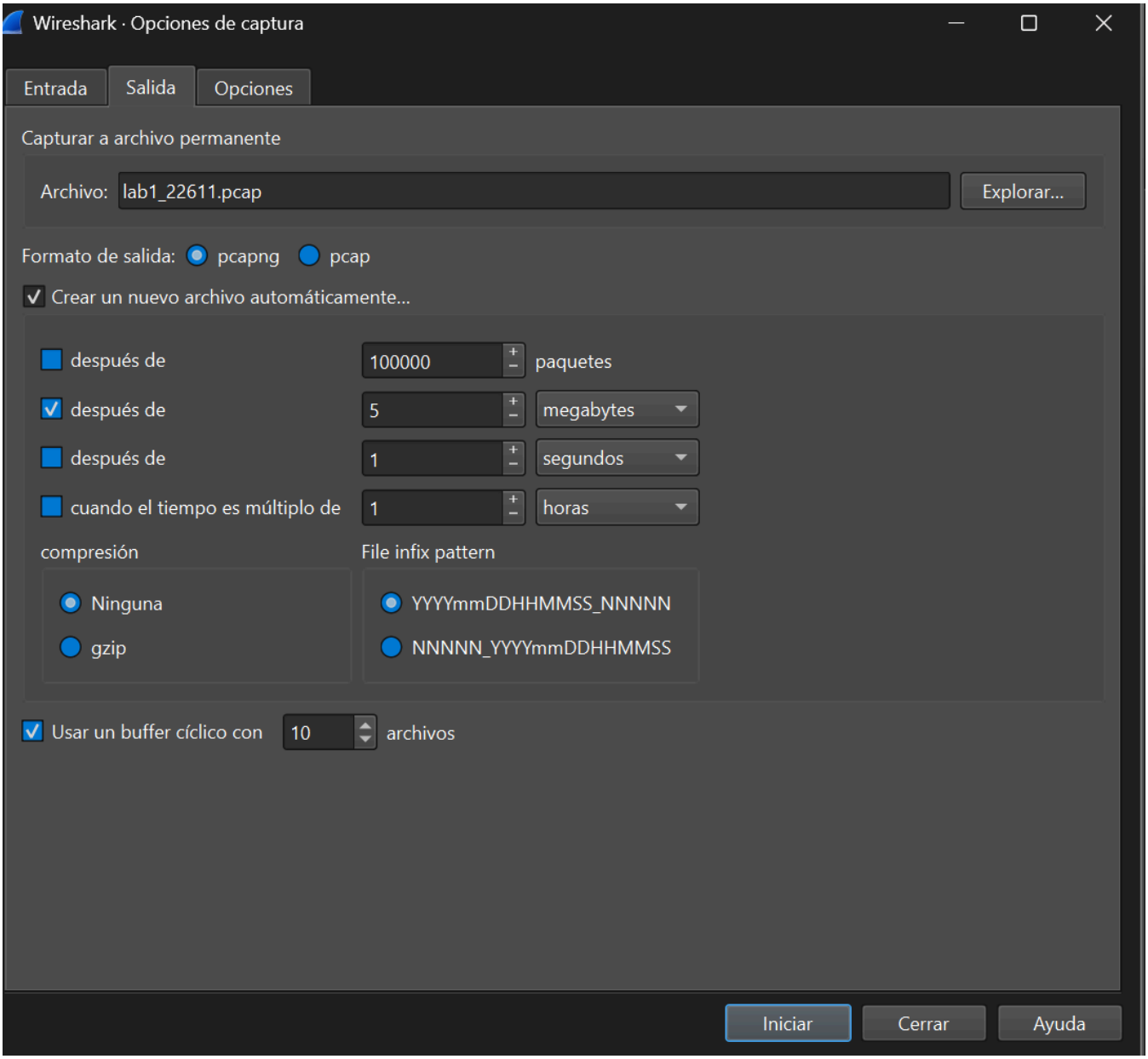
1. Abra una terminal y ejecute el comando ifconfig/ipconfig (dependiendo de su SO). Detalle y explique lo observado, investigue (i.e.: 'man ifconfig', documentación) de ser necesario. ¿Cuál es su interfaz de red?

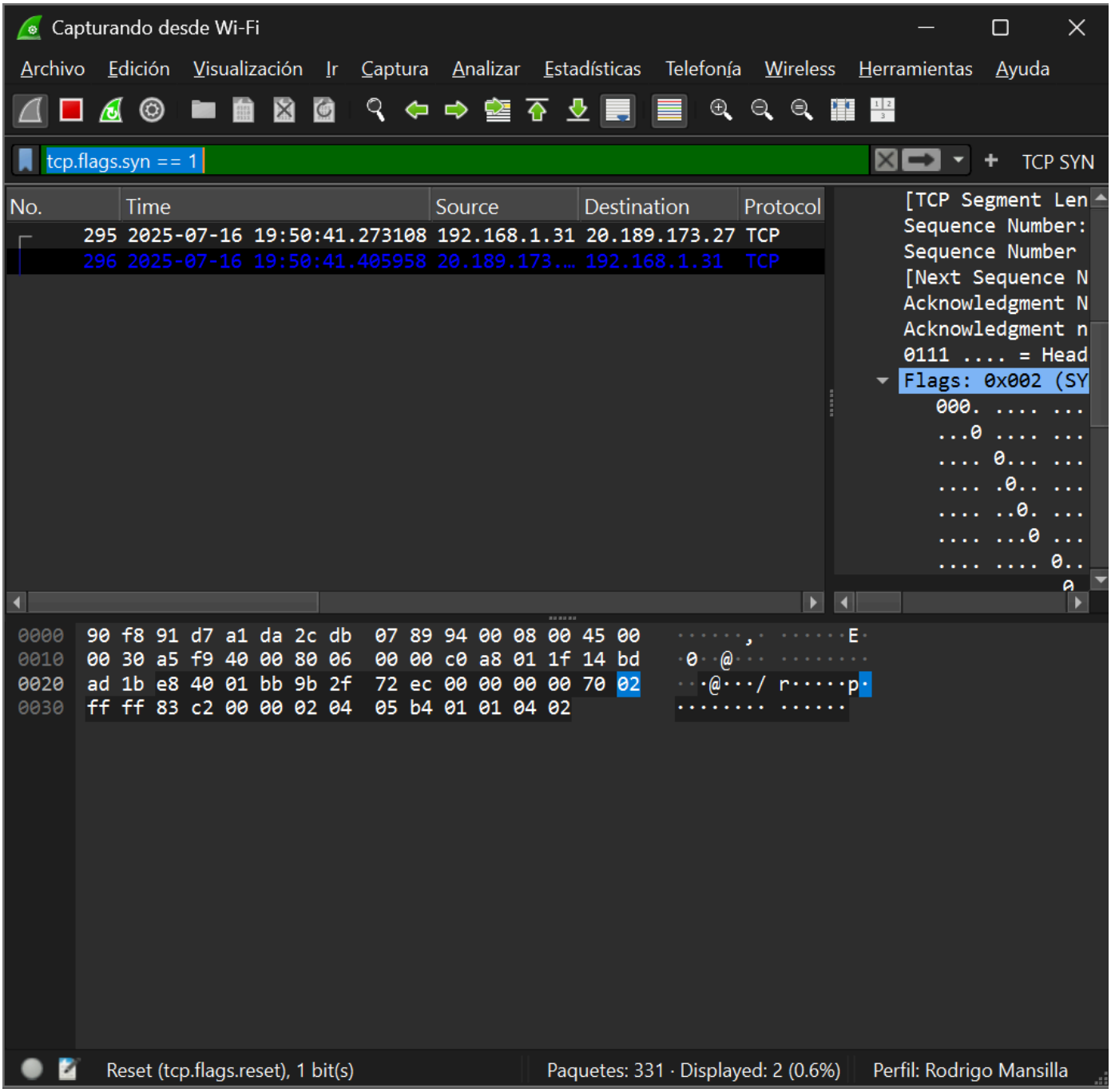
```
C:\Users\rodri>IPCONFIG

Windows IP Configuration


Ethernet adapter Ethernet:

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

Unknown adapter Local Area Connection:

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

Wireless LAN adapter Local Area Connection* 11:

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

Wireless LAN adapter Local Area Connection* 12:

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

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   IPv6 Address. . . . . . . . . . . : 2800:98:1117:1101::15
   IPv6 Address. . . . . . . . . . . : 2800:98:1117:1101:80a9:a739:16e1:bd64
   Temporary IPv6 Address. . . . . . : 2800:98:1117:1101:1d57:850e:13b8:a739
```

- **Ethernet adapter Ethernet:**
  Adaptador de red física por cable (Ethernet).
  - `Media disconnected` : no está conectada a una red actualmente.
- **Unknown adapter Local Area Connection:**
  Adaptador de red virtual o residual, creado por software de VPN o máquinas virtuales.
  - `Media disconnected` : tampoco está en uso.
- Wireless LAN adapter Local Area Connection *11 y 12:*
  Interfaces virtuales creadas por software de red inalámbrica.
  - `Media disconnected` : no están activas.
- **Wireless LAN adapter Wi-Fi:**
  Interfaz de red activa.
  - Direcciones IPv6 asignadas, lo que indica que está conectada y operativa. Esta es la 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.

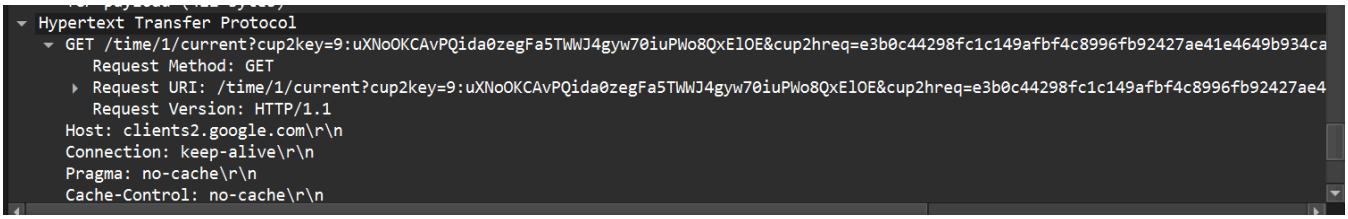




1.3 Tercera parte: análisis de paquetes

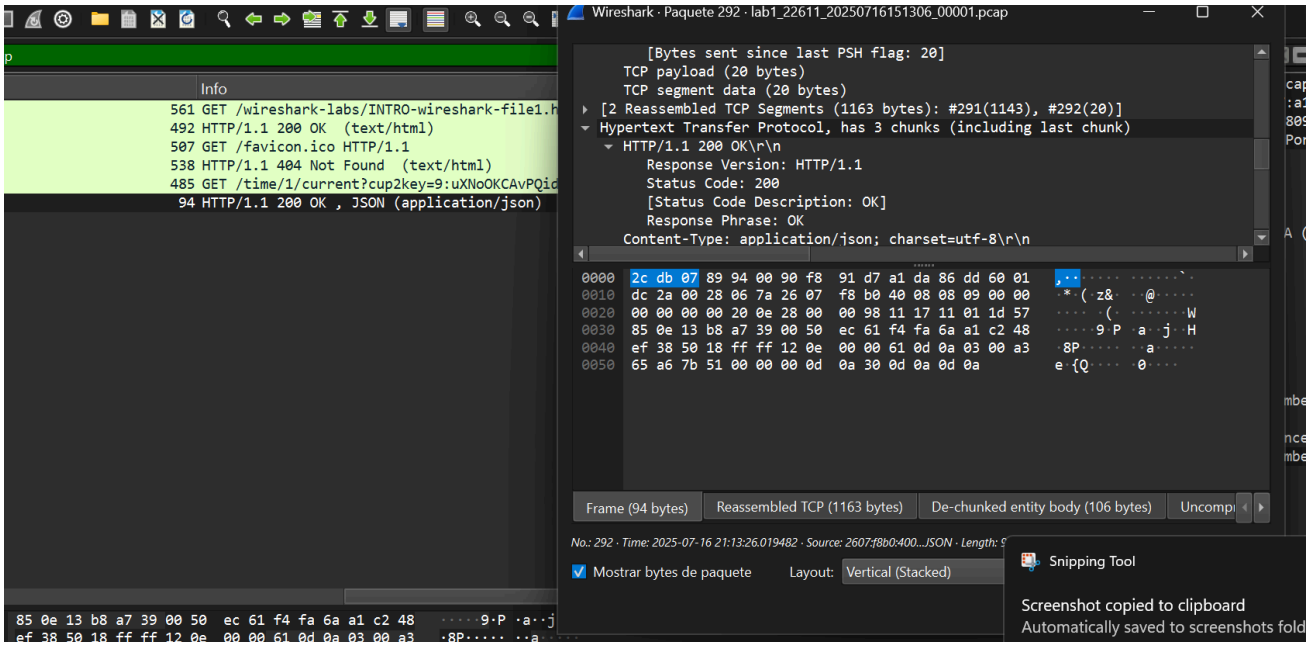
1. Abra su navegador, inicie una captura de paquetes en Wireshark (sin filtro) y acceda a la siguiente direccion (Si por alguna razón debe repetir el paso, borre su caché o utiliza el modo incógnito de su navegador): http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html
2. Detenga la captura 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:

    - ¿Qué versión de HTTP está ejecutando su navegador?



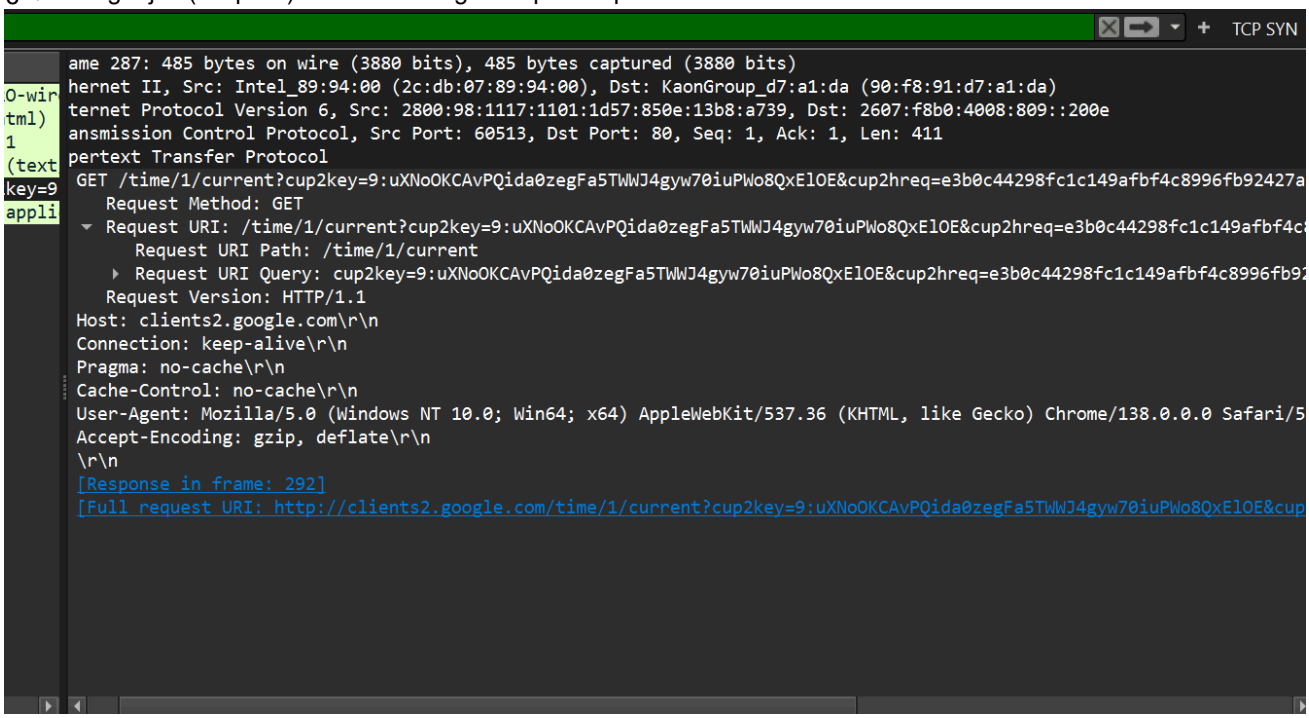
Usa 1.1

- ¿Qué version de HTTP está ejecutando el servidor?



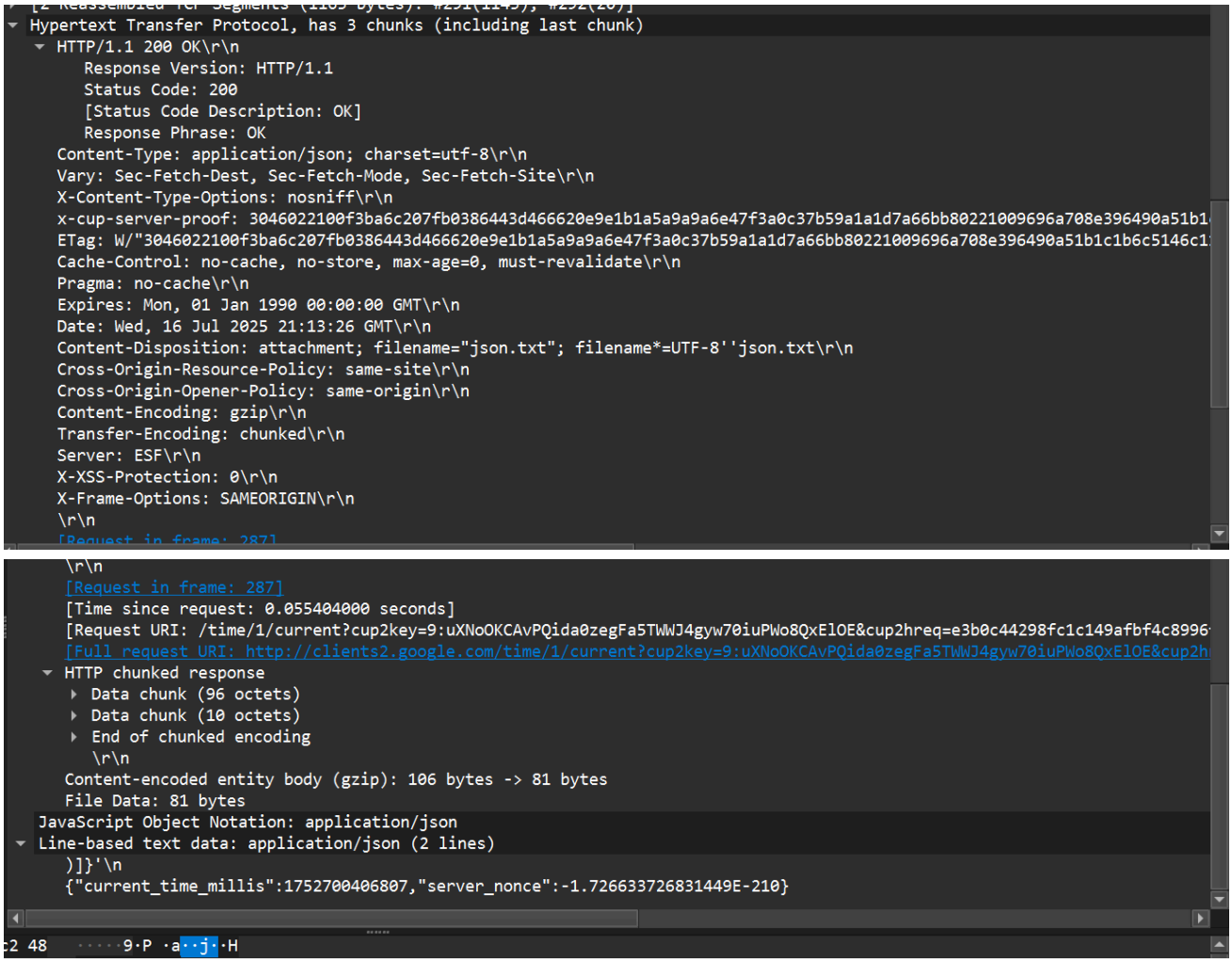
Usa http 1.1

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



En esta solicitud, el navegador **no envió el header “Accept-Language”**.

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



El servidor devolvió un total de **81 bytes de contenido** como cuerpo de la respuesta HTTP. El contenido fue enviado en formato comprimido (gzip) y utilizando transferencia “chunked”.

- 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.

  Para diagnosticar problemas de rendimiento es más eficiente capturar el tráfico en el cliente o en un punto intermedio de la red, donde se pueda aislar el flujo relevante sin impactar la operación del servidor ni manejar volúmenes excesivos de datos que dificulten el análisis.

Discusión sobre la actividad, experiencia y hallazgos

La actividad permitió comprender el funcionamiento y la importancia de los conmutadores en una red. Durante el proceso, se evidenció que la segmentación mejora el rendimiento y el control del tráfico. Sin embargo, la experiencia mostró que, aunque la estructura estaba clara, la transmisión y traducción de los mensajes tuvo errores frecuentes, lo que causó pérdida de información y dificultó la comunicación efectiva. Resaltando que la configuración y sincronización correcta son fundamentales para evitar fallas en redes con múltiples conmutadores.

Referencias

- Kurose, J.F. & Ross, K. W. Computer Networking: A Top-Down Approach. Pearson Education. ISBN: 978-0-13-285620-1.
- Tanenbaum, A. & Wetherall, D. Computer Networks. Seattle: Prentice Hall. ISBN: 978-0-13212695-3