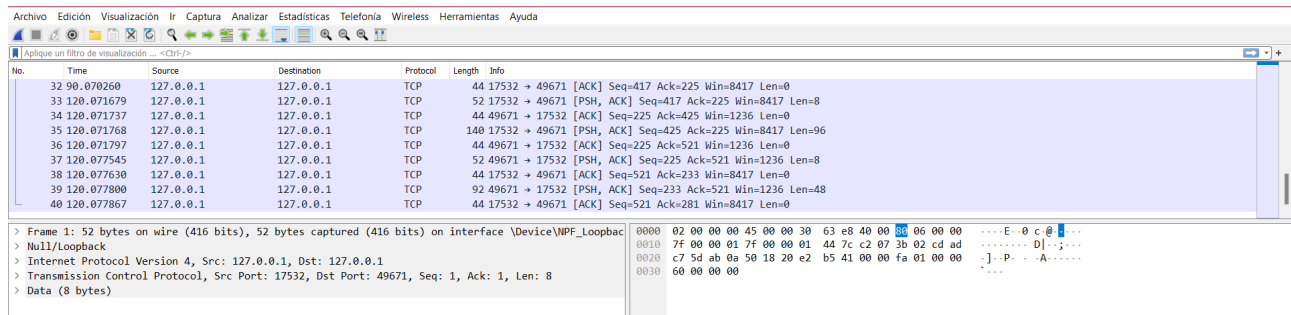


## Laboratorio 1

### Análisis de tráfico

Si se analiza el número de los mensajes enviados dentro de la aplicación. ¿Cuántos son los que logra detectar Wireshark? Y comparando en base al código, ¿es la misma cantidad?, si no lo es, ¿a qué se debería?



The screenshot shows the Wireshark network protocol analyzer. The top pane displays a list of captured packets. The bottom pane shows the details of the selected packet (No. 40), which is an Internet Protocol Version 4 packet. The packet list shows several TCP packets between 127.0.0.1 and 127.0.0.1. The packet details pane shows the structure of the selected packet, including the Ethernet II, Internet Protocol Version 4, and Transmission Control Protocol headers, followed by the application data.

No.	Time	Source	Destination	Protocol	Length	Info
32	90.070260	127.0.0.1	127.0.0.1	TCP	44	17532 → 49671 [ACK] Seq=417 Ack=225 Win=8417 Len=0
33	120.071679	127.0.0.1	127.0.0.1	TCP	52	17532 → 49671 [PSH, ACK] Seq=417 Ack=225 Win=8417 Len=8
34	120.071737	127.0.0.1	127.0.0.1	TCP	44	49671 → 17532 [ACK] Seq=225 Ack=425 Win=1236 Len=0
35	120.071768	127.0.0.1	127.0.0.1	TCP	140	17532 → 49671 [PSH, ACK] Seq=425 Ack=521 Win=8417 Len=96
36	120.071797	127.0.0.1	127.0.0.1	TCP	44	49671 → 17532 [ACK] Seq=225 Ack=521 Win=1236 Len=0
37	120.077545	127.0.0.1	127.0.0.1	TCP	52	49671 → 17532 [PSH, ACK] Seq=225 Ack=521 Win=1236 Len=8
38	120.077630	127.0.0.1	127.0.0.1	TCP	44	17532 → 49671 [ACK] Seq=521 Ack=233 Win=8417 Len=0
39	120.077800	127.0.0.1	127.0.0.1	TCP	92	49671 → 17532 [PSH, ACK] Seq=233 Ack=521 Win=1236 Len=48
40	120.077867	127.0.0.1	127.0.0.1	TCP	44	17532 → 49671 [ACK] Seq=521 Ack=281 Win=8417 Len=0

Frame 1: 52 bytes on wire (416 bits), 52 bytes captured (416 bits) on interface \Device\NPF\_{...} Loopback

Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1

Transmission Control Protocol, Src Port: 17532, Dst Port: 49671, Seq: 1, Ack: 1, Len: 8

Data (8 bytes)

0000 02 00 00 00 45 00 00 30 63 e8 40 00 06 00 00 .....E..0..c..@...  
0010 7f 00 00 01 7f 00 00 01 44 7c c2 07 3b 02 cd ad .....D].....  
0020 c7 5d ab 0a 50 18 20 e2 b5 41 00 00 fa 01 00 00 ...].P...A.....  
0030 60 00 00 00

Cuando analizamos todo el tráfico que nos muestra la aplicación WireShark y lo comparamos con la cantidad de texto que generan nuestras terminales, podemos ver que los números no calzan a la perfección, lo cual está bien, ya que estos mensajes indican tanto envío como recepción de mensajes, mientras que WireShark parece solo registrar los mensajes enviados, además, pareciera que cíclicamente se envían mensajes de `length = 0`, lo cual frente a nuestra mejor explicación, esto serían los handshakes que hacen los servidores antes de mandar la “data” y nuestro código que muestra los mensajes enviados no toma eso en consideración.

¿Cuál es el protocolo que se debiese ver a la hora de revisar el intercambio de mensajes en Wireshark? ¿Y cuáles encontró?

Durante la codificación de este laboratorio se estableció que mientras que el Cliente y el servidor Intermediario usarían conexiones con protocolos TCP, el servidor Intermediario con el servidor Conecta4 se comunicarán por medio de conexiones con protocolo UDP, debido a esto se esperaba poder encontrar ambos tipos de protocolos en el análisis de tráfico de WireShark, y finalmente luego de utilizar el software, pudimos notar tanto protocolos TCP como UDP en el tráfico.

¿El contenido de los mensajes dentro de Wireshark son legibles?, ¿por qué sí? o ¿por qué no?

Los mensajes en Wireshark muestran la “data” como una serie de números en hexadecimal, pero que al ser interpretados no son legibles, esto se debe a la encriptación que estos presentan durante el proceso de las conexiones TCP y UDP, en el código esto es fácilmente identificable, ya que en muchas ocasiones usamos el proceso `encode()`, el cual convierte un string en una serie de bytes según un cierto patrón de codificación.