

- 1. 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?

Para los mensajes enviados por el protocolo UDP, Wireshark logra detectar todos los mensajes enviados pues coincide con la cantidad real enviada de los mensajes entre el servidor gato y el servidor intermedio.

En el caso de los mensajes enviados por el protocolo TCP, pese a ser una cantidad mayor de mensajes enviados entre el servidor intermedio y el cliente, tampoco hay ninguna pérdida de los mensajes, ya que la cantidad detectada es la misma que la que fue realmente enviada.

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

Uno de los protocolos es UDP. Este protocolo permite el envío de mensajes entre el servidor intermedio y el servidor gato. El segundo protocolo es TCP. Este protocolo permite el envío de mensajes entre el servidor intermedio y el cliente. Ambos protocolos fueron encontrados al utilizar la herramienta Wireshark.

- 3. ¿El contenido de los mensajes dentro de Wireshark son legibles?, ¿Por qué sí? o ¿Por qué no?

Usualmente, la mayor parte de las veces en las que se hizo una prueba de detección de mensajes, este no era legible. Sin embargo, en pocas ocasiones sí fue posible detectar en cierto sector el contenido de los mensajes en Wireshark, y de hecho era legible, y contenía la totalidad del mensaje enviado. Un ejemplo puede verse en la IMAGEN1. También se pudo apreciar que existían palabras que deberían verse unidas y sin embargo están separadas, por ejemplo, en vez de mostrar “Bienvenido” y muestra: “Bie nvenido” como se ve en la IMAGEN 2).

Esto podría explicarse asumiendo que Wireshark es capaz de decodificar algunas veces el mensaje y mostrar por pantalla su “traducción”.

Por otro lado, dado que casi siempre la detección de los mensajes no fue legible, podemos concluir que esto se debe a que Wireshark es capaz de detectar todos o casi todos los mensajes enviados, sin embargo, estos viajan a través de paquetes y por lo tanto no se puede conocer a simple vista su contenido, puesto que se le añaden headers y otros elementos para permitir este envío.

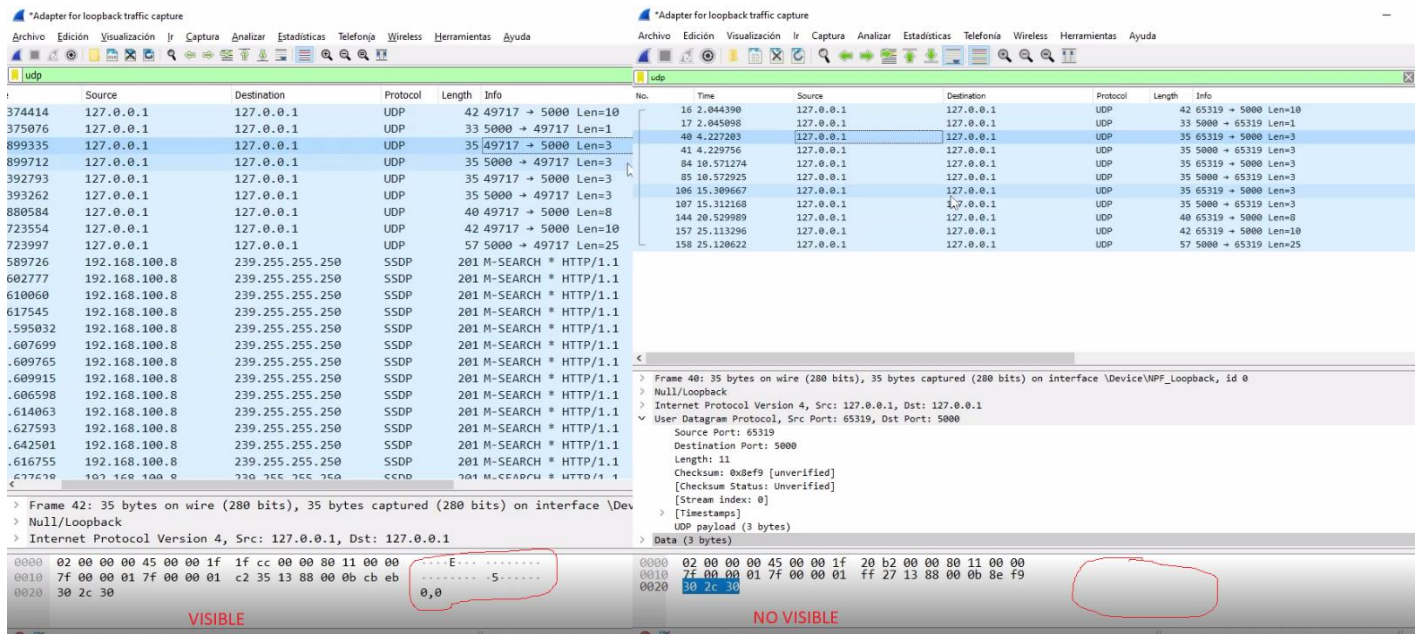


IMAGEN 1: "Comparación dos pruebas de envío de mensajes"

6	30.654974	127.0.0.1	127.0.0.1	TCP	82	5001 → 59283 [PSH, ACK] Seq=1 Ack=1
7	30.654990	127.0.0.1	127.0.0.1	TCP	44	59283 → 5001 [ACK] Seq=1 Ack=39
8	30.655007	127.0.0.1	127.0.0.1	TCP	68	5001 → 59283 [PSH, ACK] Seq=39 Ack=1
9	30.655015	127.0.0.1	127.0.0.1	TCP	44	59283 → 5001 [ACK] Seq=1 Ack=63
10	30.655028	127.0.0.1	127.0.0.1	TCP	52	5001 → 59283 [PSH, ACK] Seq=63 Ack=1
11	30.655037	127.0.0.1	127.0.0.1	TCP	44	59283 → 5001 [ACK] Seq=1 Ack=71
12	30.655073	127.0.0.1	127.0.0.1	TCP	52	5001 → 59283 [PSH, ACK] Seq=71 Ack=1
13	30.655082	127.0.0.1	127.0.0.1	TCP	44	59283 → 5001 [ACK] Seq=1 Ack=79
14	33.370130	127.0.0.1	127.0.0.1	TCP	1068	59283 → 5001 [PSH, ACK] Seq=1 Ack=1
15	33.370184	127.0.0.1	127.0.0.1	TCP	44	5001 → 59283 [ACK] Seq=79 Ack=102
16	33.370234	127.0.0.1	127.0.0.1	TCP	45	59283 → 5001 [PSH, ACK] Seq=1025
17	33.370256	127.0.0.1	127.0.0.1	TCP	44	5001 → 59283 [ACK] Seq=79 Ack=102
20	33.375201	127.0.0.1	127.0.0.1	TCP	79	5001 → 59283 [PSH, ACK] Seq=79 Ack=1
21	33.375224	127.0.0.1	127.0.0.1	TCP	44	59283 → 5001 [ACK] Seq=1026 Ack=1
22	33.375245	127.0.0.1	127.0.0.1	TCP	62	5001 → 59283 [PSH, ACK] Seq=114 Ack=1
23	33.375253	127.0.0.1	127.0.0.1	TCP	44	59283 → 5001 [ACK] Seq=1026 Ack=1
24	33.375266	127.0.0.1	127.0.0.1	TCP	56	5001 → 59283 [PSH, ACK] Seq=132 Ack=1
25	33.375274	127.0.0.1	127.0.0.1	TCP	44	59283 → 5001 [ACK] Seq=1026 Ack=1

Frame 6: 82 bytes on wire (656 bits), 82 bytes captured (656 bits) on interface \Device\NPF\_{Loopback, id 0

Null/Loopback

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

```

0000 02 00 00 00 45 00 00 1f 1f a8 40 00 80 06 00 00  ....E..N..@....
0010 7f 00 00 01 7f 00 00 01 13 89 e7 93 3b 08 4b ec  ....K.
0020 0b 4d c0 4e 50 18 27 f9 1d ee 00 00 2d 2d 2d 2d  .M.NP.....
0030 2d 2d 2d 2d 20 42 69 65 6e 76 65 6e 69 64 6f 20  ....Bie nvenido

```

IMAGEN 2: "Vista de envío de mensaje separado"