

## Análisis de Tráfico

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?

Al correr los tres ejecutables por primera vez (server.py, conecta4.go, client.py), vemos que Wireshark detecta tres conexiones mediante puertos TCP. Podemos notar que primero el puerto 46268 se comunica con el puerto 8000, este último le responde y finalmente el 46268 le envía un último mensaje:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	127.0.0.1	127.0.0.1	TCP	74	46268 → 8000 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM
2	0.000013810	127.0.0.1	127.0.0.1	TCP	74	8000 → 46268 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495
3	0.000024800	127.0.0.1	127.0.0.1	TCP	66	46268 → 8000 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=39502850

  

Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
Ethernet II, Src: 00:00:00:00:00:00 (00:00:00:00:00:00), Dst: 00:00:00:00:00:00 (00:00:00:00:00:00)
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
Transmission Control Protocol, Src Port: 46268, Dst Port: 8000

La jugada posterior muestra las siguientes conexiones en Wireshark. En donde las primeras conexiones TCP van desde y hacia el puerto 8000 y 46268. En relación a las UDP, vemos que van desde y hacia el puerto 8001, 60092, 44934 y 43619:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	127.0.0.1	127.0.0.1	TCP	74	46268 → 8000 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM
2	0.000013810	127.0.0.1	127.0.0.1	TCP	74	8000 → 46268 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495
3	0.000024800	127.0.0.1	127.0.0.1	TCP	66	46268 → 8000 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=39502850
4	363.039619922	127.0.0.1	127.0.0.1	TCP	67	46268 → 8000 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=1 TSval=395
5	363.039641052	127.0.0.1	127.0.0.1	TCP	66	8000 → 46268 [ACK] Seq=1 Ack=2 Win=65536 Len=0 TSval=39506481
6	363.040386292	127.0.0.1	127.0.0.1	UDP	43	60092 → 8001 Len=1
7	363.040448323	127.0.0.1	127.0.0.1	UDP	47	8001 → 60092 Len=5
8	363.040736997	127.0.0.1	127.0.0.1	UDP	43	44934 → 43619 Len=1
9	363.040806598	127.0.0.1	127.0.0.1	UDP	43	43619 → 44934 Len=1
10	363.041111182	127.0.0.1	127.0.0.1	TCP	67	8000 → 46268 [PSH, ACK] Seq=1 Ack=2 Win=65536 Len=1 TSval=395
11	363.041122392	127.0.0.1	127.0.0.1	TCP	66	46268 → 8000 [ACK] Seq=2 Ack=2 Win=65536 Len=0 TSval=39506481

  

Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
Ethernet II, Src: 00:00:00:00:00:00 (00:00:00:00:00:00), Dst: 00:00:00:00:00:00 (00:00:00:00:00:00)
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
Transmission Control Protocol, Src Port: 46268, Dst Port: 8000

Podemos notar que es consistente con lo que los programas reportan:

```

5 • • • • •
Jugada Player-> Columna (0)
. 0 | 1 | 2 | 3 | 4 | 5
0 • • • • •
1 • • • • •
2 • • • • •
3 • • • • •
4 • • • • •
5 ○ • • • • •
Recibido Puerto SOCKET UDP2 :43619
3
recibida Jugada desde SOCKET UDP2 3
Jugada CPU-> Columna (3)
=====
. 0 | 1 | 2 | 3 | 4 | 5
0 • • • • •
1 • • • • •
2 • • • • •
3 • • • • •
4 • • • • •
5 ○ • • ○ • •

```

La consola anterior reporta lo que muestra el servidor intermediario por pantalla, podemos notar que es consistente con lo que Wireshark lee.

```

Conexion UDP creada en el puerto: 8001
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[1025/1025]0xc0000b596fPuerto UDP2 Aleatorio : 43619
Recibi el msg del Server intermedio
Envio movimiento aleatorio
Recibir msg UDP2
Tablero: [0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
columna: 0
direccion: 127.0.0.1:44934
=====
Puerto UDP2 Aleatorio : 30513

```

Esta consola reporta lo que muestra el BOT de conecta4, la cual tambien calza con lo visto en Wireshark. Asi que es correcto afirmar que la cantidad de mensajes reportados es igual a la que se leen.

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

TCP y UDP. Lo cual es consistente en base a lo que vimos en Wireshark.

3. ¿El contenido de los mensajes dentro de Wireshark son legibles?, ¿Por Qué si? o ¿Por Qué no?

El contenido de los mensajes no es directamente legible, ya que se codificaron en formato binario, esto se debe al protocolo que se adopto para la tarea (TCP y UDP).