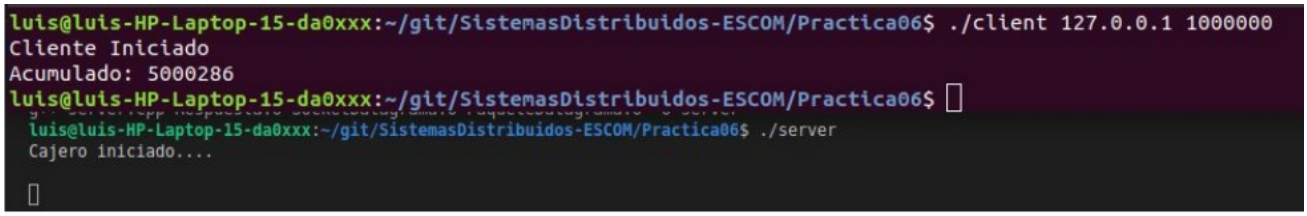
Protocolo Solicitud-Respuesta confiable

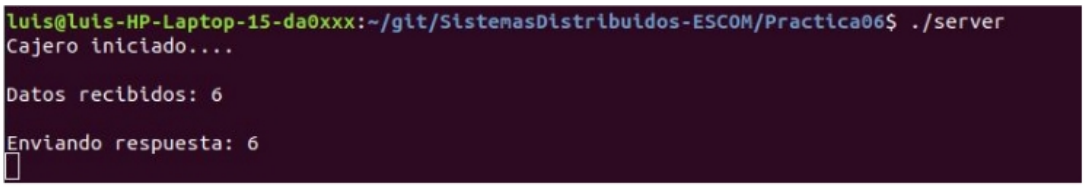
Prueba del primer ejercicio:

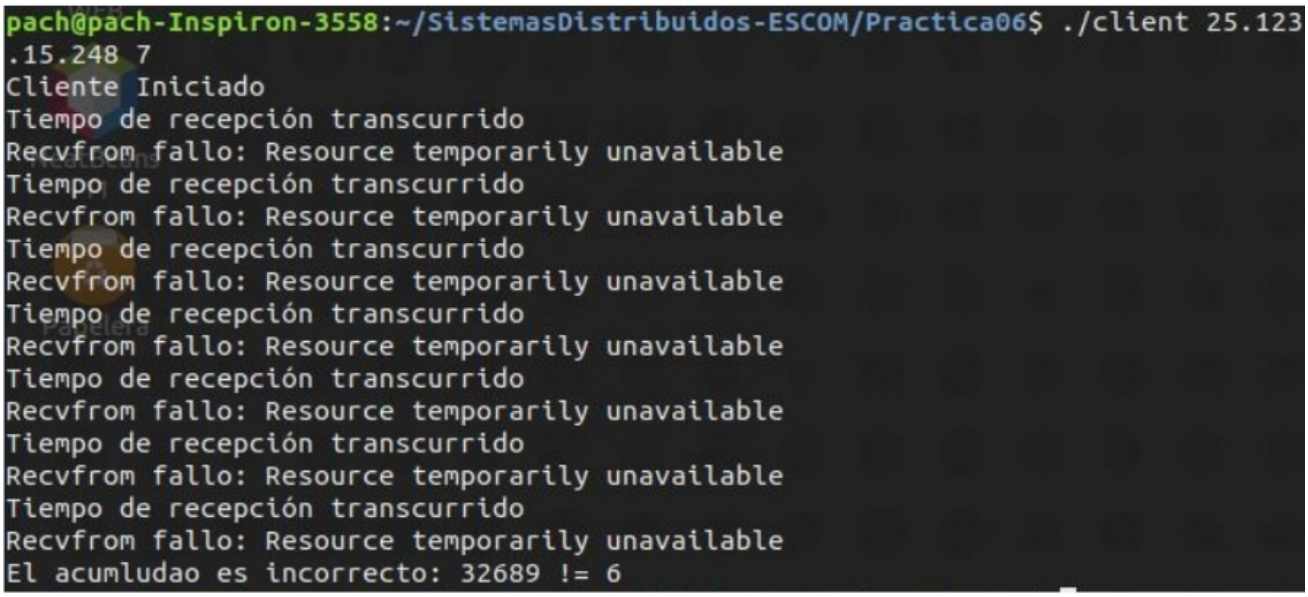
Al probar en un solo equipo funciona, incluso si usamos números grandes como 1000000. Donde el cliente imprime el valor total del estado de la cuenta.



Captura de pantalla mostrando la ejecución del cliente y del servidor en una misma computadora con n igual a 1000000.

Al probar dos equipos, utilizando hamachi, no funciona, incluso si usamos números pequeños como 7, en este caso, probamos con 7, de los cuales el servidor solo recibió 1.



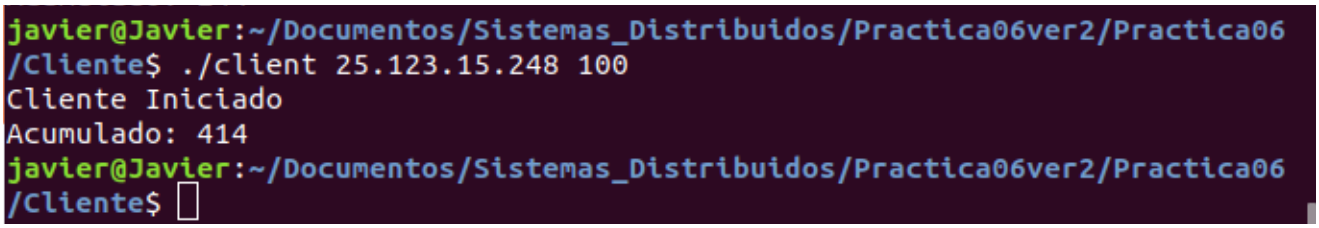


Esto se debe a que al implementamos los tres primitivos (doOperation, getRequest y sendReply) a través de datagramas UDP, así que sufren de los mismos errores de comunicación. Es decir:

- Sufren de fallas de omisión.
- No se garantiza que los mensajes se entreguen en orden del remitente.

Prueba del Segundo ejercicio:

Para compensar la posibilidad de mensajes perdidos, doOperation envía el mensaje de solicitud repetidamente hasta que recibe una respuesta o está razonablemente seguro de que el retraso se debe a la falta de respuesta del servidor en lugar de a mensajes perdidos.
También utilizamos hamachi para este ejercicio. El número más grande que podíamos utilizar sin que recibiéramos un error era 100.



Captura de pantalla de ejecución del cliente con n igual a 100.

Captura de pantalla del servidor, devolviendo la respuesta.

Para resolver el problema, se optó por realizar un ciclo en la función doOperation, cuya función se encuentra en el programa Solicitud.cpp, que se se repitiera hasta que la función enviar devolviera un número distinto de -1, confirmando por parte de esta función, que el cliente realizo el envío correcto del paquete.
Para el caso del archivo Respuesta.cpp, se modificaron las funciones getRequest y sendReply. Para la primera función, esta fue modificada, mediante un ciclo que nos indicara que el servidor recibe mensajes por parte del cliente.
Para la función sendReply, esta también modificada, mediante un ciclo que nos indica que el servidor envía mensajes exitosamente al cliente.