Cliente y Servidor con protocolo UDP

Contenido

[Solución de los Requerimientos 1](#_Toc527558937)

[Servidor 1](#_Toc527558938)

[Cliente 2](#_Toc527558939)

[Tablas y análisis de resultados 3](#_Toc527558940)

[Cotización en servidores Cloud (Microsoft Azure) 5](#_Toc527558941)

## Solución de los Requerimientos

### Servidor

El servidor recibe conexiones de clientes a través de un paquete que le llega de un cliente que quiere recibir los datos. Este informa que el cliente esta listo para recibir. Cuando están todos los clientes, el servidor empieza a enviar los datos para el envío del archivo. Antes de empezar a escuchar las solicitudes de los clientes, es pedido la cantidad de clientes que se van a conectar, el archivo que se va a enviar, el numero de la prueba que se están realizando y el tamanio de los paquetes a enviar:

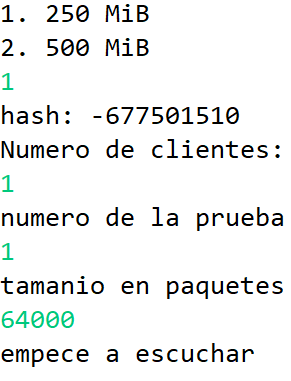
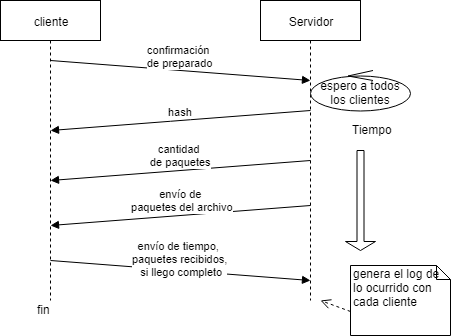


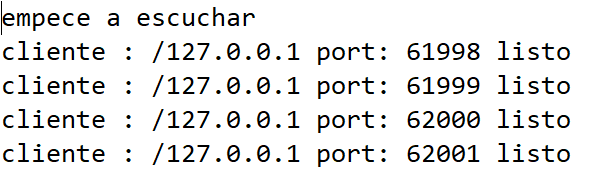
Ilustración consola pidiendo datos antes de iniciar

El comportamiento del cliente y servidor se comportan de la manera como se muestra en el siguiente diagrama de secuencia:

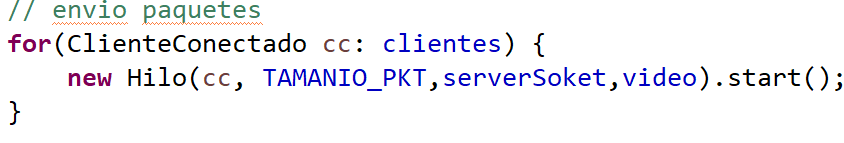


Los datos a enviar son datos de texto plano (.txt) y hay 2 archivos: uno de 250 MB y otro de 500 MB. Una vez el proceso acaba, el servidor se reinicia y vuelve al comienzo.

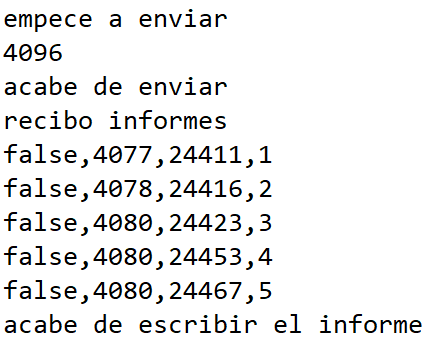
Cada ves que se conecta un cliente se hace un log que informa que ese cliente (con IP y puerto) esta listo para recibir los datos.



Cuando se conectan todos los clientes, se genera un hilo por cada uno y se empieza el proceso de descarga de archivo:



El servidor se queda escuchando la respuesta de los clientes esperando a que envíen el reporte. Este reporte se queda en los logs:



* Los Logs se guardan en la carpeta logs.
* El repositorio del servidor se encuentra en el siguiente enlace:

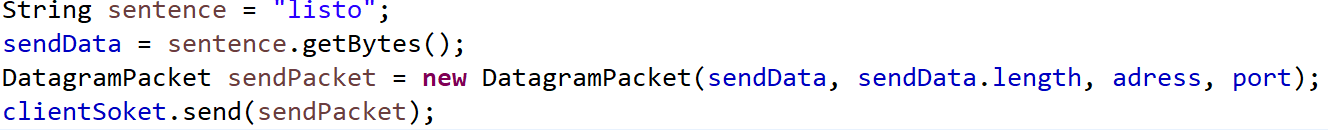
<https://github.com/sicard6/lab4-redes>

**NOTA:** Debido a que el cliente se ahogaba ya que el servidor enviaba los paquetes muy rápido, se puso a dormir el hilo por 5 milésimas de segundo para que le pudieran llegar un numero aceptable de paquetes. Sin embargo, en ningún caso llegaron todos los paquetes completos. En el caso de que se llevaran pocos paquetes (entre 10 y 500) la integridad se mantenía, pero de lo contrario siempre había un paquete o 2 que se perdía.

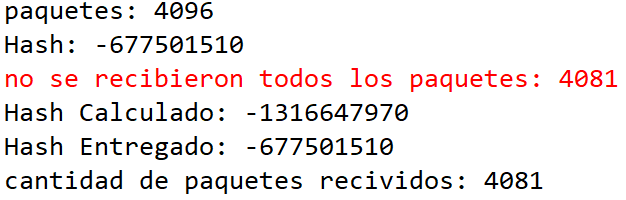
### Cliente

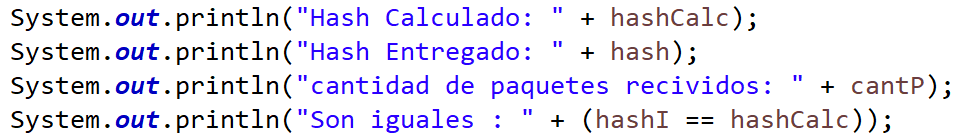
Cuando se envía la conexión de que el cliente esta listo para recibir datos, se le informa a el usuario que está hecha la conexión.

Como se ve en el diagrama de secuencia de la clase anterior se informa que se envia una notificación diciendo que el cliente esta listo para recibir mensajes:



Después de recibir los archivos, se verifica que el archivo enviado este correcto. En caso de que lo este, se notifica en un log, de lo contrario se envía un mensaje de error.





Los datos obtenidos como tiempo, bytes recibidos y si llego correcto se envía a el servidor para generar el reporte.

El repositorio del servidor se encuentra en el siguiente enlace:

<https://github.com/sicard6/lab4-redes>

## Tablas y análisis de resultados

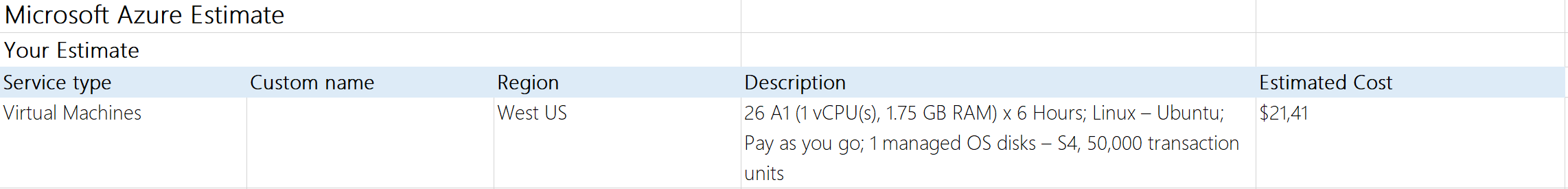
En la siguiente tabla se encuentra la relación de cliente vs tiempo. Como se puede observar, la transmisión de datos es prácticamente constante (cambio de 1 milisegundo entre un cliente y 25). Esto se debe a que se utilizaron hilos para enviar el archivo a cada cliente. Además, como se colocó un sleep en el hilo después de que se enviara un paquete (para que el cliente pudiera recibir el mensaje y no se ahogara), se obtiene un tiempo relativamente alto. Los tiempos son constantes debido a que UDP no tiene control de trafico y tampoco tiene control de integridad, causando que los tiempos no varíen por tiempos con variaciones en la red.

Los bytes enviados y los bytes recibidos son incrementalmente lineales. Es decir, a mayor cantidad de clientes, mayor cantidad de datos es enviados. Esto causa que el servidor tenga que trabajar mas para enviar mas datos a diferentes clientes. Los bytes recibidos son menores a los que se envían debido a que se tienen menos datos por las propiedades de UDP, y eso incluye los paquetes de los headers:

Como se puede observar en la siguiente tabla, hay una degradación del servicio a medida que hay mayor cantidad de clientes, debido a que hay congestion en la red, causando que hallan mas paquetes perdidos:

## Cotización en servidores Cloud (Microsoft Azure)

Con la calculadora de Microsoft Azure, encontramos que las maquinas virtuales para hacer las pruebas (dado que nos demoramos 6 horas y que utilizamos aproximadamente 20000 transacciones) fue:



Una transacción de Microsoft azure es un llamado a la consola de Ubuntu, entonces como es una consola, no se hacen muchos comandos. Si fuera una interfaz, habrían muchas transacciones.