

LABORATORIO 4 – ANÁLISIS DE COMUNICACIÓN POR SOCKET TCP

1. OBJETIVO (S)

El objetivo del presente laboratorio se centra en implementar sockets TCP para establecer comunicación entre un servidor y un cliente, implementado en un lenguaje de programación seleccionado por el equipo de trabajo. Adicionalmente, se busca realizar una transferencia de archivos por medio de los canales implementados, evaluando la calidad del servicio cuando se realizan cambios en los parámetros del mismo.

Al finalizar la práctica, el estudiante estará en capacidad de:

- Implementar un cliente para recibir archivos de un servidor haciendo uso de sockets TCP.
- Implementar un servidor que permita establecer una comunicación a través de sockets TCP y realice transferencia de archivos por medio del mismo protocolo.
- Implementar y evaluar pruebas de carga y desempeño para la comunicación.

2. LECTURAS PREVIAS

Los temas a tratar en esta práctica son los siguientes:

- Generalidades de los protocolos TCP [1]
- Implementación de sockets en Java [2]
- Implementación de sockets en Python [3, 4]
- Ejecución de pruebas de carga y desempeño con Apache JMeter [5, 6]

3. INFORMACIÓN BÁSICA

En esta práctica se desarrollará un servidor que implemente servicios TCP, para comunicarse respectivamente con un cliente. De tal forma, se supone el diseño e implementación de una arquitectura cliente-servidor, donde la comunicación se establezca a través de sockets. En la Figura 1 se presenta la arquitectura esperada.

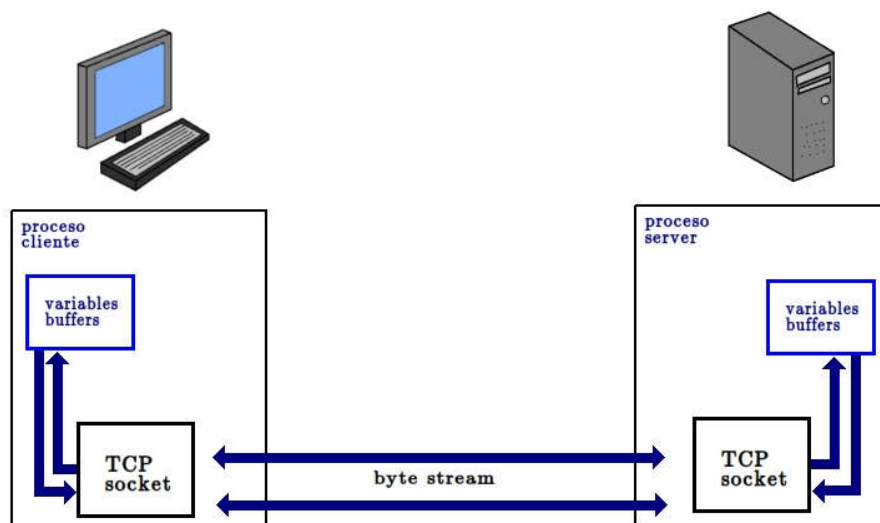


Figura 1 – Diagrama de componentes
(http://wiki.inf.utfsm.cl/index.php?title=Socket_programming_with_TCP)

Adicionalmente, se espera que se implementen pruebas de carga y desempeño sobre la arquitectura desarrollada, con el fin de evaluar las características ofrecidas por el protocolo de comunicación. Esto permitirá evaluar otras aplicaciones del mercado, analizar su desempeño y sugerir los protocolos de comunicación empleados en las mismas.

Las primeras dos partes del laboratorio se centran en la implementación del cliente y del servidor, junto con los mecanismos necesarios para garantizar una comunicación sobre TCP. La tercera parte se enfoca en el diseño e implementación de pruebas de carga y desempeño sobre el servidor. La práctica finaliza con una evaluación del protocolo de comunicación implementado, y análisis de aplicaciones del mercado.

4. PROCEDIMIENTO

Para el presente laboratorio deberán desarrollar los siguientes requerimientos para un cliente y un servidor que se encuentran en dos máquinas diferentes; deberán subir el servidor a Amazon Web Services, Google Cloud, Microsoft Azure, u otra nube:

4.1. Implementación de cliente TCP

El grupo deberá desarrollar y desplegar un cliente en el lenguaje de programación de su preferencia (Java, Python, etc.). Este deberá cumplir con los siguientes requerimientos:

1. Conectarse al servidor y mostrar que se ha realizado dicha conexión. Mostrar el estado de la conexión.
2. Recibir un archivo del servidor por medio de una comunicación a través de sockets TCP.
3. Permitir al usuario seleccionar el archivo que desea recibir.
4. Permitir al usuario iniciar y detener la captura de datos. Recordar que iniciar y detener la captura se debe efectuar después de elegir el archivo, dado que el envío se debe realizar en tiempo real.
5. Mostrar cada paquete recibido (puede ser por consola) y adicionalmente mostrar el archivo recibido.

6. Desarrollar una interfaz usable donde se reflejen las funcionalidades solicitadas (La conexión establecida y estado de la conexión, los archivos disponibles para su descarga, botones para iniciar y detener la captura y el archivo recibido (no los paquetes), debe permitir visualizar el archivo).
7. Tener un repositorio de los archivos recibidos.

4.2. Implementación de servidor TCP

El equipo deberá implementar un servidor en el lenguaje de programación de su preferencia (Java, Python, etc.). Este deberá cumplir con los siguientes requerimientos:

1. Recibir conexiones TCP. El equipo de trabajo deberá determinar el número óptimo (adecuado, apropiado) de máximas conexiones disponibles simultáneamente. Este número deberá determinarlo de acuerdo a su diseño.
- 1. Describa el proceso para determinar el número máximo de conexiones.**
2. Tener 3 tipos de archivos disponibles para su descarga: De tamaño pequeño 3 MB aproximadamente, de tamaño mediano 15 MB aproximadamente y de tamaño grande 45 MB aproximadamente.
3. Realizar la transferencia de archivos que los usuarios hayan seleccionado, por medio del protocolo de comunicación TCP.
4. Definir el tamaño del buffer apropiado para su diseño.
- 2. Cambie el tamaño del buffer a un 10% de su tamaño inicial y realice la transferencia de un archivo grande, vuelva a modificar el tamaño del buffer a un 500% del tamaño inicial y realice la transferencia de un archivo grande. Describa los cambios observados, haga el análisis comparativo de las dos situaciones y concluya determinando cuál es el valor adecuado para su diseño.**
5. Definir el tamaño de los mensajes en que se van a fragmentar los archivos.
- 3. Modifique el tamaño de los mensajes a un 10% del valor inicial y realice la transferencia de un archivo mediano, luego a un 200% del valor inicial y realice la transferencia de un archivo mediano, describa los cambios observados, realice el análisis comparativo de las dos situaciones y concluya determinando cuál es el valor adecuado para su diseño.**
6. Definir un tiempo en el cual el servidor bloquee la comunicación con un cliente si en ese tiempo determinado no ha hecho ninguna petición.
- 4. Explique por qué es importante definir ese tiempo en donde el servidor bloquee la comunicación con un cliente.**
7. Establecer las preferencias de rendimiento en términos de tiempo de conexión, baja latencia y alto ancho de banda.
- 5. Cambie los valores de las preferencias iniciales, describa los cambios percibidos, realice el análisis comparativo de las diferentes situaciones y concluya determinando cuál es el valor adecuado para su diseño. Para esta prueba use la transferencia del archivo de tamaño grande.**

Para la primera y la segunda parte del laboratorio, se debe grabar un video de máximo 50 MB donde se demuestre el desarrollo de cada uno de los requerimientos con una breve explicación. Si estos no se evidencian en el mismo, se asumirá que no fueron desarrollados.

4.3. Pruebas de carga y desempeño sobre la arquitectura

Se espera que para este laboratorio el grupo efectúe pruebas de carga y desempeño sobre el servidor TCP implementado en la segunda parte de la práctica. Para ello, haga un grupo de pruebas haciendo uso de JMeter.

Grupo de pruebas en JMeter:

- Se recomienda hacer uso del Aggregate Report de Apache JMeter.
- Se deben efectuar pruebas de carga y desempeño para el servicio TCP de manera independiente. Para ambos casos se debe determinar el tiempo promedio de atención de los clientes, porcentaje de error frente a peticiones de usuario y throughput.
- Efectuar pruebas sobre escenarios de concurrencia. Por ejemplo, 10, 40, 90, 210, y 300 usuarios. La selección de estos escenarios dependerá directamente del comportamiento de su desarrollo, por lo que será necesario hacer pruebas exploratorias que le permitan determinar el número de solicitudes a evaluar¹. Manejar un mismo período de ramp-up² para 3 escenarios seleccionados, y efectuar 2 iteraciones para cada uno de estos escenarios.

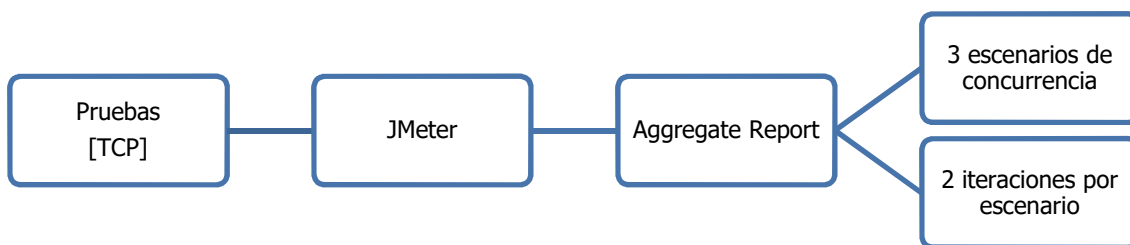


Figura 2 – Vista general plan de pruebas

- 6. Para esta parte de la práctica, se deberá hacer una descripción del diseño e implementación de cada conjunto de pruebas. Así mismo, se solicita la muestra de resultados, análisis y conclusiones para el grupo de pruebas. Apoyarse en las gráficas necesarias para sustentar lo anterior.**
- 7. Consulte 3 herramientas adicionales para realizar pruebas de carga y explique la operación de cada una de ellas. (Sólo explicar cómo funciona, no deben realizar pruebas adicionales de carga).**

¹ No exceder los 300 usuarios concurrentes, ya que para este escenario sería necesario crear un clúster de JMeter.

² Para definir adecuadamente el ramp-up, consultar la siguiente fuente: http://jmeter.apache.org/usermanual/test_plan.html.

4.4. Evaluación y análisis de aplicaciones del mercado

Para la última parte del laboratorio deberá dar respuesta al siguiente requerimiento:

- 8. Analizar las aplicaciones Waze³, Netflix⁴, y Skype⁵. Guarde una captura de wireshark mientras usa Netflix y otra captura de wireshark mientras realiza una videollamada en Skype, analice las diferencias entre ambas aplicaciones. Finalmente, con relación a los análisis efectuados en la práctica y a referencias adicionales, argumentar cuál es el tipo de comunicación empleada por cada aplicación.**

6. ENTREGABLES

- Informe digital que contenga el proceso de solución de los requerimientos efectuados en cada una de las secciones de la práctica. Si estos no se evidencian en el mismo, se asumirá que no fueron desarrollados.
- El video con la demostración.
- Carpeta con el desarrollo efectuado para el cliente y el servidor.
- Capturas de wireshark solicitadas.

Nota: Deben hacer una sola entrega, en una carpeta comprimida con todos archivos. La entrega, como es costumbre, la deben realizar por Sicua Plus.

7. REFERENCIAS

[1] Kurose, James. Ross, Keith. Computer Networking: A Top-Down Approach. 5th edition. Addison-Wesley. Capítulos 2 y 3.

[2] The Java Tutorials. Trail: Custom Networking.
<http://docs.oracle.com/javase/tutorial/networking/index.html>

[3] Rhodes, Brandon. Goerzen, John. Foundations of Python Network Programming. 2nd edition. 2010.

[4] Documentación oficial Python: Socket – Low-level Networking Interface.
<https://docs.python.org/2/library/socket.html>

[5] Página oficial de Apache JMeter. <http://jmeter.apache.org/>

[6] Ejecución de las pruebas con JMeter. <http://carloszuluaga.wikidot.com/pruebascarga:ejecucion-pruebas-jmeter>

³ Página oficial: <https://www.waze.com/es/>

⁴ Página oficial: <https://www.netflix.com/co/>

⁵ Página oficial: <http://www.skype.com/en/>

HISTORIAL DE REVISIONES

FECHA	AUTOR	OBSERVACIONES
09/03/2017	Laura María Ruiz Gómez lm.ruizg@uniandes.edu.co	Versión inicial del documento.
18/01/2018	Jonatan Legro Pastrana j.legro@uniandes.edu.co	Modificaciones de contenido.