Universidad de los Andes
Facultad de Ingeniería

Departamento de Ingeniería de Sistemas y Computación
Infraestructura de Comunicaciones
Profesores: Yezid Donoso / Carlos Lozano
[ydonoso / calozanog] @uninandes.edu.co
Asist. De laboratorio: Arnold Lara Vacca
a.larav@uniandes.edu.co

ABET Engineering Accreditation Commission

LABORATORIO 3: ACTIVIDAD 1 – IMPLEMENTACIÓN DE SERVIDOR Y CLIENTE TCP

1. OBJETIVO (S)

Este laboratorio tiene por objetivo comprender el funcionamiento del protocolo de control de transmisión (TCP). Para este fin, en este laboratorio se realizará la implementación de un servidor TCP y su respectivo cliente en un lenguaje de programación seleccionado por el grupo de trabajo.

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

- Implementar aplicaciones servidor y clientes, para enviar/recibir archivos soportada en sockets TCP.
- Evaluar pruebas de carga y desempeño para la comunicación. Diseñar, implementar y evaluar diferentes tipos de pruebas de desempeño a cada uno de los servidores implementados.
- Completar el desarrollo y el análisis pertinente para uno de los requerimientos del proyecto.

2. LECTURAS PREVIAS

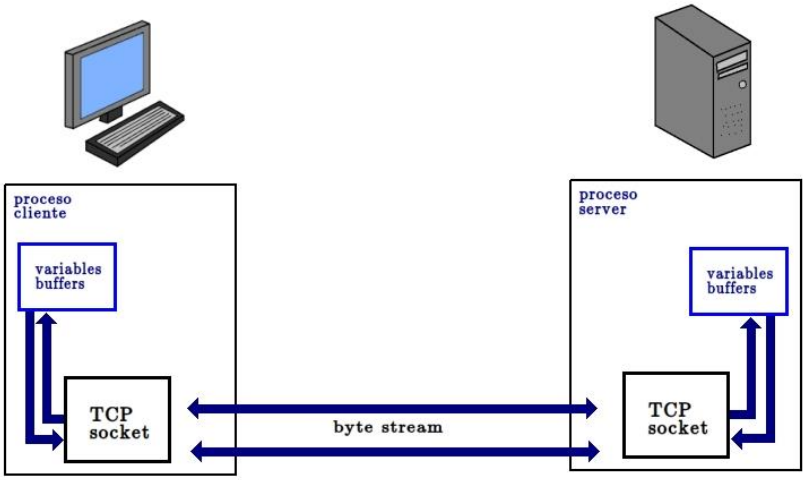
De acuerdo con el lenguaje en el que usted decida implementar la aplicación, aquí encontrara una serie de recursos útiles para el desarrollo de este laboratorio.

- Sección - 3.1 Introduction and Transport-Layer Services. Computer Networking, a top-down approach. James Kurose, Keith Ross. Addison-Wesley, 7th edición.
- Sección - 3.5 Connection-Oriented Transport: TCP. Computer Networking, a top-down approach. James Kurose, Keith Ross. Addison-Wesley, 7th edición.
- The Java Tutorials. Trail: Custom Networking. [3]
- Java Secure Sockets Extension (JSSE). [4]
- Socket – Low-level Networking Interface. [5]
- Manejo de Sockets en PHP. [6]
- Foundations of Python Network Programming. 2nd edition. [8]

3. INFORMACIÓN BÁSICA

- Durante el laboratorio se puede utilizar las máquinas propias de los integrantes del grupo o hacer uso de la infraestructura del laboratorio. En caso de elegir utilizar la infraestructura del laboratorio esta decisión debe ser informada lo antes posible al profesor de laboratorio para que se habilite el acceso a la máquina virtual correspondiente.
- Durante el desarrollo de este laboratorio usted desarrollará un servidor de transferencia de archivos y de un aplicativo cliente usando el protocolo TCP; de tal forma que se logre una comunicación entre estos desarrollos utilizando una arquitectura del tipo cliente-servidor (Ver Figura 1).

Figura. 1. Diagrama de comunicación por socket TCP [1]



- Se recomienda leer la guía completamente antes de iniciar a resolver las actividades propuestas, con el objetivo de tener presente las actividades y los entregables a desarrollar.

4. PROCEDIMIENTO

Tal como se enuncio en el anterior apartado en este laboratorio se debe realizar un desarrollo de un servicio de transferencia de archivo utilizando el protocolo de control de transmisión (TCP por sus siglas en inglés) emulando una arquitectura cliente – servidor.

Finalmente, se espera que se implementen pruebas de carga y desempeño sobre la arquitectura desarrollada, con el fin de evaluar el funcionamiento del servicio mediante métricas de desempeño, como: total de bytes transmitidos por el servidor, tiempo de transferencia promedio medido en servidor, tasa de transferencia promedio medido en servidor, total de bytes recibidos por el cliente, tiempo de transferencia promedio medido en cliente, tasa de transferencia promedio medido en cliente.
Esto permitirá realizar un análisis de rendimiento de las transferencias con el protocolo TCP en diferentes escenarios.

Por último, usted deberá subir sus aplicativos de cliente y servidor en Github. Asegúrese de incluir un archivo **readme** en donde se pueda encontrar instrucciones sobre la instalación y ejecución de cada uno de los programas.

4.1. Configuración del Ambiente de Trabajo

Esta práctica puede ser desarrollada usando los equipos propios de los integrantes del grupo (Computadores personales) o puede utilizar la infraestructura del laboratorio.

4.1.1. Configuración en Equipos Propios

Para poder desarrollar el laboratorio usando los equipos propios de los integrantes del grupo usted debe disponer de un hipervisor para máquina virtuales, específicamente se utilizará el hipervisor desarrollado por VMWare.

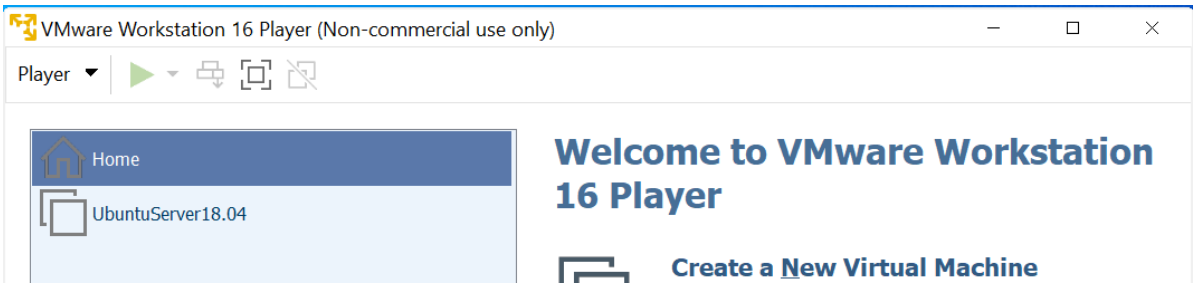
Si su equipo tiene un sistema operativo Windows o Linux, deberá utilizar el software VMWare Workstation Player 16, disponible en el siguiente enlace: https://www.vmware.com/co/products/workstation-player/workstation-player-evaluation.html

En el caso de que su equipo tenga un sistema operativo macOS, deberá utilizar el software VMware Fusion, disponible en el siguiente enlace: https://www.vmware.com/co/products/fusion/fusion-evaluation.html

Posteriormente descargue el archivo de la máquina virtual que utilizará como servidor, esta máquina tiene como sistema operativo base Ubuntu Server 18.04. El enlace de descarga es el siguiente: https://uniandes-my.sharepoint.com/:u:/g/personal/a_larav_uniandes_edu_co/EW03PbEL8B1Eh2-Q2wFFhc8BrmsmsgypP8PndTXZaI1xZw?e=Li0U5a
El archivo a descargar se llama **`UbuntuServer18.04.rar`**.

Descomprima el archivo descargado y entre en la carpeta nombrada UbuntuServer18.04, allí encontrara los archivos de la máquina virtual. Abra el archivo **`UbuntuServer18.04.vmx`**, esto abrirá la máquina virtual en el software Vmware Workstation Player (Ver Figura 2).

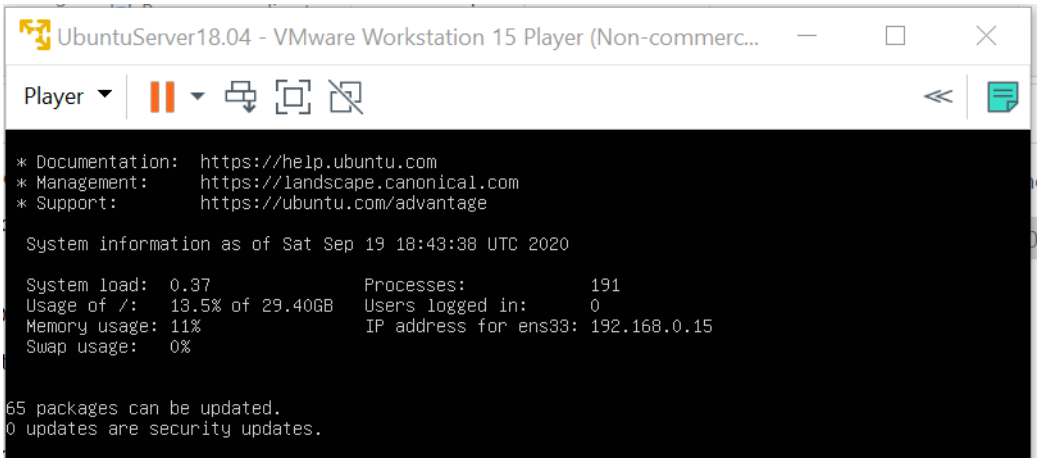
Figura. 2. Ventana inicial VMWare Workstation Player.



Nota: La máquina virtual se encuentra configurada en **`modo Bridge`**; esto significa que se conectara a su red como si fuera un equipo más.

Al iniciar la maquina use las credenciales de acceso **`usuario: infracom`** y **`contraseña: infracom`**. Al iniciar podrá observar la dirección IP que obtiene su máquina virtual; para el caso de prueba se puede observar en la Figura 3 que la maquina ha recibido la dirección **`192.168.0.15`**.

Figura. 3. Funcionamiento Máquina virtual Ubuntu18.04 Server.



Una vez termine la actualización de la máquina, la misma quedará disponible para su trabajo.

En cuanto al equipo cliente, el aplicativo desarrollado puede ser ejecutado en el mismo equipo que ejecuta la máquina virtual.

4.1.2. Infraestructura de Laboratorio

Para este caso se debe informar lo antes posible al profesor de laboratorio para que se habilite el acceso a una máquina virtual Ubuntu Server 18.04 en la que ejecutara el aplicativo de servidor. Para el aplicativo cliente usted podrá utilizar el equipo Windows 10 que le fue asignado a su grupo.

4.2. Requerimientos del Servidor de Transferencia de Archivos

Se deberá implementar un servidor de transferencia de archivos en el lenguaje de programación de su preferencia (Java, Python, entre otros), el cual debe cumplir con los siguientes requerimientos:

1. Debe implementar el protocolo de control de transmisión (TCP por sus siglas en inglés) para transmitir un archivo hacía un cliente
2. Debe correr sobre una máquina con sistema operativo UbuntuServer 18.04
3. La aplicación debe soportar al menos 25 conexiones concurrentes.
4. Debe tener dos archivos disponibles para su envío a los clientes: un archivo de tamaño 100 MB y otro de 250 MB.
5. La aplicación debe permitir seleccionar qué archivo desea transmitir a los clientes conectados y a cuántos clientes en simultáneo; a todos se les envía el mismo archivo durante una transmisión.
6. Debe enviar a cada cliente un valor hash calculado para el archivo transmitido; este valor será usado para validar la integridad del archivo enviado en el aplicativo de cliente.
7. La transferencia de archivos a los clientes definidos en la prueba debe realizarse solo cuando el número de clientes indicados estén conectados y el estado de todos sea listo para recibir.
8. La aplicación debe generar un archivo de **log** en un directorio llamado **Logs**. Se debe generar un archivo de **log** por cada prueba realizada y este debe contener registros para cada conexión recibida en la prueba.

Los requerimientos para la construcción del archivo **log** son las siguientes:

a. El nombre del archivo de Logs debe incluir la fecha exacta de la prueba. Ejemplo: <año-mes-dia-hora-minuto-segundo-log.txt>
b. Incluya el nombre del archivo enviado y su tamaño.
c. Identifique para cada conexión el cliente al que se realiza la transferencia de archivos.
d. Identifique si la entrega del archivo fue exitosa o no.
e. Tome los tiempos de transferencia a cada uno de los clientes, calcule este tiempo desde el momento que se envía el primer paquete archivo y hasta que se recibe la confirmación de recepción del último paquete del archivo.
f. Incluya el número de paquetes enviado, también el valor total en bytes enviado (estos valores están dados por el tamaño del archivo, los paquetes retransmitidos y los encabezados).

Apóyese de comandos como **iptraf**, **ngrep**, **ethstatus**, entre otros para obtener las estadísticas.

4.3. Requerimientos del cliente TCP

El grupo deberá desarrollar y desplegar una aplicación cliente en el lenguaje de programación de su preferencia (Java, Python, entre otros), la cual debe cumplir con los siguientes requerimientos:

1. Debe implementar el protocolo de control de transmisión (TCP por sus siglas en inglés) para recibir un archivo del servidor
2. Debe correr sobre una máquina cliente que pueda conectarse al servidor dispuesto (equipo propio de los integrantes del grupo o cliente de Windows 10 en la infraestructura de la Universidad)
3. La aplicación debe permitir al menos 25 conexiones (Clientes) con el servidor.
4. Se debe mostrar el estado de conexión de cada cliente.
5. Debe permitir que el usuario envié al servidor una confirmación de listo para empezar la recepción del archivo.
6. Las transferencias recibidas deben ser almacenadas en un directorio llamado **ArchivosRecibidos**.
7. El archivo recibido debe conservarse de la siguiente forma: **[Número cliente]-Prueba-[Cantidad de conexiones].txt**

   Ejemplo nombres de archivo para prueba de 5 conexiones:
   - Cliente1-Prueba-5.txt
   - Cliente2-Prueba-5.txt
   - Cliente3-Prueba-5.txt
   - Cliente4-Prueba-5.txt
   - Cliente5-Prueba-5.txt

8. Verificar la integridad del archivo con respeto a la información entregada por el servidor en cada transferencia. Para este fin la aplicación cliente debe calcular el hash del archivo recibido y compararlo con el valor suministrado por el servidor (utilice el mismo algoritmo utilizado para el cálculo en el servidor).
9. La aplicación debe generar un archivo de **log** en un directorio llamado **Logs**. Se debe generar un archivo de **log** por cada prueba realizada y este debe contener registros para cada conexión recibida en la prueba.

Los requerimientos para la construcción del archivo **log** son las siguientes:

a. El nombre del archivo de Logs debe incluir la fecha exacta de la prueba. Ejemplo: <año-mes-dia-hora-minuto-segundo-log.txt>
b. Incluya el nombre del archivo enviado y su tamaño.
c. Identifique para cada conexión el cliente al que se realiza la transferencia de archivos.
d. Identifique si la entrega del archivo fue exitosa o no.
e. Tome los tiempos de transferencia a cada uno de los clientes, calcule este tiempo desde el momento que se recibe el primer paquete del archivo y hasta que se envía la confirmación de recepción del último paquete del archivo.
f. Incluya el número de paquetes recibidos, también el valor total en bytes recibidos (estos valores están dados por el tamaño del archivo, los paquetes retransmitidos y los encabezados).

4.4. Análisis de desempeño del servicio

Una vez finalizado el desarrollo de los aplicativos cliente y servidor y tras realizar pruebas de funcionamiento; se propone la realización de un análisis de desempeño para el servicio de transferencia de archivos desarrollado. Para este fin, se tomará como base los registros de log que se pidió generar para cada prueba que se realiza; y también se utilizará el analizador de protocolos Wireshark para realizar la toma de capturas de tráfico durante las transferencias.

Se propone la ejecución de un conjunto de pruebas que le permitirá obtener la información para generar un análisis del rendimiento alcanzado por el servicio desarrollado.

Trabajo Propuesto

Para este paso se propone la realización de las siguientes actividades:

1. Realice pruebas por cada archivo disponible en el servidor para 1, 5 y 10 clientes. En total usted realizara 6 pruebas siendo estas definidas de acuerdo con la siguiente tabla.

Número de conexiones (Clientes)	Archivo 100MB	Archivo 250MB
1	Prueba 1	Prueba 2
5	Prueba 3	Prueba 4
10	Prueba 5	Prueba 6

Nota: Antes de realizar cada prueba usted debe ejecutar la herramienta Wireshark y efectuar la captura de paquetes de las trasferencias definidas en la prueba.

2. Genere una tabla por cada prueba que contenga al menos la siguiente información de cada transferencia realizada desde el servidor:

a. Transferencia exitosa (Si o No)
b. Valor total de bytes recibidos por cada cliente
c. Tiempo de la transferencia para cada cliente
d. Tasa de transferencia a cada cliente

e. Numero Puerto utilizado para la conexión de cada cliente (Aplicación Cliente)
f. Valor total de bytes transmitidos por el servidor a cada cliente
g. Tiempo de transferencia a cada uno de los clientes medido desde el servidor
h. Numero Puerto utilizado para la conexión con cada cliente (Aplicación Servidor)

Nota: Se pueden adicionar los campos que consideren relevantes para hacer un análisis de la transferencia de tráfico.

3. Analice cada prueba realizada. Considerando como cambian las métricas de acuerdo con el número de clientes conectados, el tamaño del archivo transmitido y la captura de tráfico realizada durante la prueba.
4. Realice un análisis global tomando el resultado del conjunto de pruebas realizado. Apóyese en tablas y graficas que le permitan dar una mejor explicación a su análisis.

5. ENTREGABLES

El entregable para este laboratorio es:

1. Informe en formato **.pdf** con:
   a. Proceso de solución de los requerimientos de la práctica
   b. Enlace a un repositorio en Github de las aplicaciones de Servidor y cliente desarrolladas.
   c. Enlace a un video corto en el cual explican el desarrollo realizado y su funcionamiento.
   d. Enlace de descarga para obtener las capturas de tráfico que obtuvo durante la práctica.
   e. Análisis de las pruebas realizadas para medir el desempeño del servicio de transferencia de archivos desarrollado por el equipo.

Este documento debe ser entregado utilizando el enlace habilitado en Sicua Plus para tal fin.

6. ENTREGABLES

La calificación de este laboratorio será distribuida de la siguiente manera:

Entregable	Valor
Actividad 4.2	25%
Actividad 4.3	25%
Actividad 4.4	30%
Calidad informe, incluyendo el video	20%
Total	100%

7. REFERENCIAS

[1] Arquitectura cliente-servidor en aplicación de transferencia de archivos. https://websarrolladores.com/2019/03/05/socket/

[2] Kurose, James. Ross, Keith. Computer Networking: A Top-Down Approach. 7th edition. Addison-Wesley. Capítulo 3.

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

[4] Java Secure Sockets Extension (JSSE). http://docs.oracle.com/javase/7/docs/technotes/guides/security/jsse/JSSERefGuide.html

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

[6] Manejo de Sockets en PHP. http://php.net/manual/es/book.sockets.php

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

[8] Python - Programming with Networks, https://en.admininfo.info/python-programaci-n-con-redes

HISTORIAL DE REVISIONES

FECHA	AUTOR	OBSERVACIONES
12/02/2021	**Arnold Andres Lara** a.larav@uniandes.edu.co	Actualización de actividades del laboratorio.
11/09/2020	**Arnold Andres Lara** a.larav@uniandes.edu.co	Actualización del laboratorio.
28/02/2020	**Arnold Andres Lara** a.larav@uniandes.edu.co	Ajustes de redacción y actualización del laboratorio
24/07/2019	**Jonatan Legro Pastrana** j.legro@uniandes.edu.co	Actualización de documento.