

Redes de Computadores

Laboratorio 2

Implementación de Protocolos, Programación en TCP y UDP

Profesor: Oscar Encina
Ayudante: Daniel Tapia

Requerimientos

Lenguaje de Programación: Java 1.7/1.8 Preferido o C

Sistemas Operativos: Windows / Linux / MacOSX

IDE: IntelliJ 14 (Opcional)

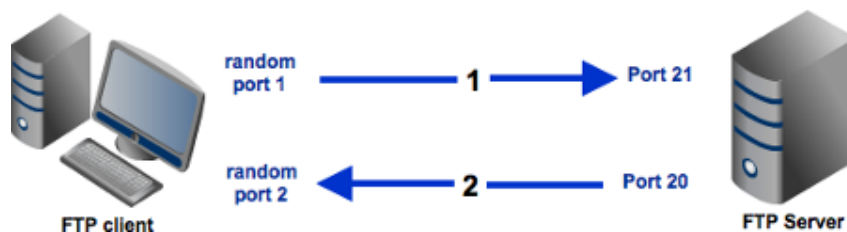
Introducción

Cloud Productions necesita una solución de servicios FTP (File Transfer Protocol), el cliente ha pedido dos servidores FTP para poder comprar el rendimiento de cada una y así tomar una mejor decisión . Para esto se deberá implementar dos servidores FTP, uno con el protocolo TCP y el segundo UDP, como también el cliente FTP.

Una vez implementado se deberá subir el contenido de un directorio llamado “web_application” que contiene los archivos:

- index.html
- application.conf
- logo.png
- video.mp4

Para esto se deberán implementar los dos canales que forman parte del protocolo. El cliente deberá conectarse por el puerto 21 hacia el servidor FTP y el servidor FTP deberá responder hacia el cliente en el puerto 20. Es decir, implementar el canal de control y el de datos.



Parte 1: Cliente y Servidor Canal de Comandos

La primera parte del laboratorio consiste en programar el cliente y servidor FTP para permitir la comunicación entre ellos. Para esto se deberá hacer una implementación TCP / UDP del cliente y servidor que use Sockets y las clases DataInputStreams/DataOutputStreams del lenguaje de programación Java. Los siguientes comandos deberán ser implementados:

- *OPEN* (comando de inicio de conexión)
- *CD* (Change Directory) Comando de cambio de directorio para navegar dentro del sistema de archivos
- *LS* (Directory Listing) Comando para ver los contenidos actuales del directorio dentro del sistema de archivos.
- *GET* Comando para extraer un archivo del servidor ej: GET nombreArchivo.pdf
- *PUT* Comando para subir un archivo al directorio actual de la conexión.
- *QUIT* Comando para terminar la sesión

Se deberá implementar un sistema de login para hacer que el comando OPEN funcione, como se muestra en Consideraciones.

Parte 2: Cliente y Servidor Canal de Datos

El canal de datos es la vía por la cual el servidor se conecta devuelta al cliente para iniciar la transferencia de datos, es decir, el servidor inicia una conexión desde el puerto 20 hacia el cliente en busca del archivo, y se transmite por una conexión TCP/UDP. Se deben usar Sockets y Data Input/Output Streams.

Se deberán implementar sockets que acepten la transmisión de archivos en formato de byte stream (Ver BufferedInputStream) y guardar estos archivos en el servidor o cliente dependiendo si utilizó el comando PUT o GET.

Consideraciones

El servidor deberá responder con un código de estado al request del cliente y escribirlo en el response.

Una vez implementada la solución, la siguiente interacción deberá ser posible:

open 192.168.0.100 (IP hacia el servidor FTP)

Conectado a 192.168.0.100

220 Hola, Soy el servidor FTP. (Respuesta del servidor con su Status Code)

Name (192.168.0.100:root): **admin** (Usuario digita usuario)

331 Password requerido para ingresar a cuenta admin.

Password: **passwordSecreto** (Usuario digita password)

230 Log in OK

cd web_aplication

250 "/home/admin/ Aplicación Web " es el nuevo directorio de trabajo

PUT index.html (348348 bytes)

226 index.html subido correctamente

PUT application.conf (319502 bytes)

226 application.conf subido correctamente

PUT logo.png (85840 bytes)

226 logo.png subido correctamente

LS

150 Mostrando listado del directorio Aplicación Web

-rw-r--r-- 1 admin admin 14886 Apr 3 15:22 index.html

-rw-r--r-- 1 admin admin 317000 Apr 3 17:40 application.conf

-rw-r--r-- 1 admin admin 317000 Apr 3 17:40 logo.png

226 Listado Completado.

GET index.html

150 Datos de conexión para index.html (282323 bytes)

226 Transferencia Completada

QUIT

221 Sesión terminada por usuario.

Comparación

Una vez implementado ambos servidores se deberá usar la herramienta WireShark para ver el tráfico entre ambas implementaciones y de la cadena de comandos de ejemplo.

Se deberán guardar los archivos .pcap de la transmisión del servidor UDP y TCP.

Comparar ambos y responder las siguientes preguntas:

- 1) ¿Cuál Transmite más rápido y por que?
- 2) Para un archivo cualquiera de los que se encuentran dentro del directorio web_applicaiton, monitorear la secuencia que involucra el inicio, transmisión, y termino con WireShark. Observar el orden en el cual los paquetes son enviados entre TCP y UDP, existe diferencia?
- 3) ¿Existe pérdida de paquetes al enviar archivos por los servidores? Si es que existen, ¿en que caso, detalle?

Como se estará implementando un protocolo ustedes podrán conectarse con el cliente hacia el servidor de sus compañeros. Hagan la prueba para estar seguros que implementaron bien el protocolo.

Restricciones

- No se puede usar ninguna implementación de java que generen servidores FTP, deben programarla ustedes.
EJ: org.apache.commons.net.ftp.*
- Cualquier librería que tengan dudas que puedan usar que no sean de Java.IO, Java.net deben ser consultadas al ayudante antes de ser usadas.

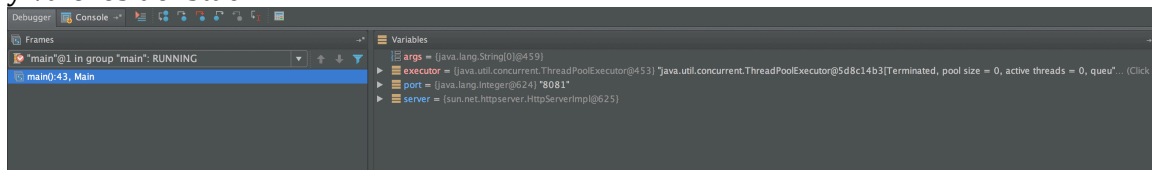
Consideraciones

- Se debe trabajar en grupos de a lo más dos personas.
- La entrega es por Moodle.
- Las dudas o aclaraciones pueden preguntarlas por Moodle.
- La entrega debe realizarse en un ZIP y debe llevar el nombre Tarea2RDC-RolIntegrante-1-RolIntegrante-2.
- Debe ir dentro del ZIP un archivo README.txt que debe contener nombre y rol de los integrantes del grupo y cualquier supuesto que se haya tomado durante la realización de la tarea, y un instructivo de cómo compilar el código adjunto.
- La entrega será usando un método que será informado a la brevedad y el plazo máximo de entrega es hasta el 22 de Abril de 2015 a las 23:59.
- Por cada día de atraso se descontarán 20 puntos.
- Las copias serán evaluadas con nota 0.

Herramientas

IntelliJ14 (IDE)

En IntelliJ14 existe la opción de compilar en modo debug, esto hace que la aplicación sea capaz de parar la ejecución del programa para poder ver los valores de variables y valores del stack.



Wireshark (Aplicación)

Aplicación que permite monitorear el tráfico de un enlace.

<https://www.wireshark.org/download.html>