**UNIVERSIDAD DE LOS ANDES**

**FACULTAD DE INGENIERÍA**

**DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN**

**INFRAESTRUCTURA DE COMUNICACIONES**

**Profesor: Harold Enrique Castro Barrera**

[**hcastro@uniandes.edu.co**](mailto:hcastro@uniandes.edu.co)

**Asist. De laboratorio: Laura María Ruiz Gómez**

[**lm.ruizg@uniandes.edu.co**](mailto:lm.ruizg@uniandes.edu.co)

|  |  |  |
| --- | --- | --- |
| |  | | --- | | **INFORME DE LABORATORIO 5 – ANÁLISIS DE COMUNICACIÓN POR SOCKETS TCP** | |  | |

**Nombres: - José Daniel Fandiño Códigos: -201423487**

* **Juan Sebastián Cardona -20142**

1. **Describa el proceso para determinar el número máximo de conexiones.**

Mediante pruebas de carga se verificó hasta cuantas conexiones podía soportar el servidor y se estableció el número máximo de conexiones en el valor máximo en donde las pruebas no tenían error. En nuestro caso eso fue alrededor de 500.

1. **Cambie el tamaño del buffer a un 10% de su tamaño inicial y realice la transferencia de un archivo, vuelva a modificar el tamaño del buffer a un 500% del tamaño inicial y** **realice la transferencia de un archivo. Describa los cambios observados, haga el análisis comparativo de las dos situaciones y concluya determinando cual es el valor adecuado para su diseño.**

Inicialmente al buffer le pusimos 8KB pues es el valor por defecto\*. Así, al probar con el 10% de dicho tamaño (es decir, 800Bytes) obtuvimos un tiempo de descarga del archivo grande de 268 segundos. Al aumentarlo al 500% (40KB) obtuvimos un tiempo de descarga del mismo archivo de 261 segundos, 7 segundos menos. También observamos que en consola para el caso del 10% los paquetes llegaban a una velocidad constante mientras que en el de 500% al inicio llegaban muy rápido pero luego se volvían más lentos. Así, es bueno tener más buffer para mayor velocidad en un inicio, pero luego puede generar latencia. En nuestro caso, como vimos que se mejoraba, decidimos subirlo a 40KB.

\*Fuente: http://www.linuxquestions.org/questions/linux-general-1/default-value-for-tcp-send-buffer-size-699906/

1. **Modifique el tamaño de los mensajes a un 10% del valor inicial y realice la transferencia de un archivo, luego a un 200% del valor inicial y realice la transferencia de un archivo, describa los cambios observados, realice el análisis comparativo de las dos situaciones y concluya determinando cual es el valor adecuado para su diseño.**

En un inicio elegimos partir los archivos en paquetes de 8KB. Al transferir un archivo con el 10% de dicho tamaño (es decir, 800Bytes), obtuvimos un tiempo de descarga de 66 segundos. Al hacerlo con el 200% (16KB), obtuvimos un tiempo de descarga de 107segundos. Al ser más pequeños los paquetes se mandan más rápido pues el TCP por debajo debe manejar pocos bytes y así hay menor probabilidad de error, y aún si llegase a haber errores reenviar el paquete no le cuesta mucho trabajo. Por el contrario, con un paquete muy grande TCP por debajo igual lo debe partir en paquetes más pequeños y asegurarse de que todos llegaron bien antes de seguir con el siguiente paquete, aumentando la latencia. Es mejor entonces quedarse con algo pequeño, pero no tanto porque igual puede ser contraproducente (pues cada paquete debe llevar su dirección y entre más paquetes entonces se transmite más datos). Para saber si valía la pena cambiarlo, comparamos también el tiempo de paquetes de 8KB y obtuvimos que se demora 89 segundos. Así, decidimos cambiar el tamaño de los paquetes a 2KB.

1. **Cambie los valores de las preferencias iniciales, describa los cambios percibidos, realice el análisis comparativo de las diferentes situaciones y concluya determinando cual es el valor adecuado para su diseño. Para esta prueba use la transferencia del archivo de tamaño grande.**

Se realizarán algunos casos para ver la diferencia. Todos se compararán con el caso de igual importancia (tiempo=latencia=ancho de banda, caso base). Casos:

* **Caso base:** Observamos que los paquetes se descargan a distintas velocidades y que la descarga se demora 210.
* **tiempo>(latencia=ancho de banda):** Aquí observamos que la descarga se realiza igual al caso base. Lo que cambia es que el tiempo de conexión disminuye, es decir, si el usuario dura inactivo se desconectará más rápidamente.
* **latencia>(tiempo=ancho de banda):** Aquí observamos que durante la descarga los paquetes se bajaban con una velocidad consistente. Es decir, los paquetes se imprimían en consola a la misma velocidad, no como el caso base donde a veces iba más rápido y otras más lento. Además, se demoró 179 segundos.
* **ancho de banda>(tiempo=latencia):** Aquí observamos que durante la descarga los paquetes van un poco más lento respecto al caso base, lo cual se ve reflejado en el tiempo de descarga de 205 segundos.

Por los anteriores resultados decidimos darle más importancia sólo a la latencia.

1. **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.**
2. **Analizar las aplicaciones Waze[[1]](#footnote-1), Netflix[[2]](#footnote-2), y Skype[[3]](#footnote-3). De acuerdo a los análisis efectuados en la práctica y a referencias adicionales, argumentar cuál es el tipo de comunicación empleada por la aplicación.**

Las pruebas de los puntos 2, 3 y 4 se realizaron en un PC con las siguientes características:

RAM: 3,9 GB

CPU: 2,4 GHz, 2 núcleos.

Procesadores lógicos: 4

1. Página oficial: <https://www.waze.com/es/> [↑](#footnote-ref-1)
2. Página oficial: <https://www.netflix.com/co/> [↑](#footnote-ref-2)
3. Página oficial: <http://www.skype.com/en/> [↑](#footnote-ref-3)