Tarea 3 Redes: Cliente eco UDP con Stop-and-Wait para medir performance

Nombre: Patricio Espinoza A. **Sección**: 1

Preparación

Comando de uso general

- $ ./client_bw.py size timeout IN OUT host port

Generar un archivo de 1 GB, uno de 500 KB (.bin)

- $ python3 files_generator.py

Ejecución del servidor local

- $ python3 server_echo_udp3.py 1818

Limpiar cache

- $ sudo sync && echo 3 | sudo tee /proc/sys/vm/drop_caches

Ejecuciones y Resultados

Localhost

- **Un archivo de 1000000 KB (1GB) → 1000000000 bytes**

$ time python3 client_bw.py ***size*** 0.01 files_localhost/big_file.bin outputs_localhost/OUT_***size*** 127.0.0.1 1818

- Limpiando cache

Size (bytes)	Tiempo real (s)	Sent packets	Lost packets	Ancho de banda (B/s)	Tamaño final (GB)
100					
500					
1.000	4912,84	1.085.785	61.784	203.548	1 GB
10.000	123,30	103.323	928	8.110.300	1 GB
100.000	0	0	0	Msg too long	0 GB
1.000.000	0	0	0	Msg too long	0 GB
10.000.000	0	0	0	Msg too long	0 GB

Obs: Para sizes 100 y 500 el tiempo real fue demasiado por lo que no se tomaron las medidas y se llego solo hasta el size = 1.000 que tomo 81 minutos.

Anakena

sudo sync && echo 3 | sudo tee /proc/sys/vm/drop_caches

- **Un archivo de 500 KB**

$ time python3 client_bw.py ***size*** 0.1 files_anakena/big_file.bin outputs_anakena/OUT_***size*** anakena.dcc.uchile.cl 1818

- Limpiando cache

Size (bytes)	Tiempo real (s)	Sent packets	Lost packets	Ancho de banda	Tamaño final (KB)
10	2.401,72	56.487	5.286	0,20	500
100	200,55	5.319	198	2,49	500
500	43,20	1.034	9	11,57	500
1.000	27,30	524	11	18,31	500
10.000	22,28	228	175	22,44	500
100.000	3	0	0	Msg too long	0
1.000.000	3	0	0	Msg too long	0
10.000.000	3	0	0	Msg too long	0

Preguntas

1. Si el archivo de salida es más pequeño que el de entrada, ¿puede ocurrir aunque mi protocolo esté bien implementado?

   Con un protocolo de Stop-and-Wait bien implementado no debería ocurrir que el archivo de salida sea más pequeño que el de entrada (pérdida de datos). Esto ya que, si bien se pueden perder paquetes o señales ACK, este protocolo tiene un sistema de retransmitir paquetes perdidos, asegurando que todos los datos sean enviados. Además, utiliza un orden para enviar cada paquete, en donde se debe confirmar que fue recibido correctamente (señal ACK) para recién continuar con el siguiente.

2. Los valores de ancho de banda medidos en esta tarea son muy distintos a los de la T1 y T2. ¿Cuál es la causa principal de estas diferencias?

   La causa principal de esta diferencia es el protocolo Stop-and-Wait implementando, si bien en la T1 el protocolo TCP garantizaba un mayor respaldo para que los datos sean enviados y recibidos correctamente a comparación de la T2, el TCP sigue permitiendo que la red fluya más rápido ya que no se debe esperar la confirmación del recibimiento de un paquete para continuar con el siguiente como ocurre con Stop-and-Wait. Esto último genera un aumento considerable en el tiempo que toma enviar todos los datos, más aún cuando los sizes son pequeños y por ende la cantidad de paquetes aumenta notoriamente.

3. Dijimos que un protocolo bien implementado no debiera nunca morir por el timeout de 3 segundos puesto en el socket de recepción. ¿Por qué es esto así?

   Esto es así ya que si el protocolo esta bien implementado, entonces el receptor debería estar recibiendo constantemente paquetes siempre y cuando el transmisor los envíe. En un funcionamiento correcto esto se mantendría hasta que el último paquete sea enviado, finalizando la conexión con todos los datos recibidos correctamente.

   La función del timeout de 3 segundos es una medida ante la interrupción del envió de datos por parte del emisor, ya sea por desconexión, desincronización, un error, etc.

Comparación

Dentro de las diferencias principales entre la tarea 1 con TCP, y la tarea 3 con Stop and Wait, es que esta última resulta ser muchísimo más lenta, en donde para sizes muy pequeños el ancho de banda disminuye considerablemente. Esto sucede ya que una de las ventajas del protocolo implementado es que toma medidas respecto a la pérdida de paquetes, retransmitiendo los perdidos y creando un sistema de señales de confirmación a costo de un tiempo extra para el proceso.