Tarea 4 Redes: Cliente eco UDP con Go-Back-N para medir performance

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

Preparación

Comando de uso general

- $ ./client_gbn_bw.py size timeout window IN OUT host port

Generar un archivo de 1 GB, uno de 500 KB, y 100 archivos de 10.000 KB y de 5KB (.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 1000 KB**

En base a resultados anteriores, un size de 1.000 suele funcionar bien, por tanto estableceremos ese size fijo e iremos cambiando los timeouts y Windows para la obtención de métricas.

$ time python3 client_gbn_bw.py **1000 timeout window** files_localhost/big_file.bin outputs_localhost/OUT_***size*** 127.0.0.1 1818

- Limpiando cache

Size (bytes)	Timeout	Window	Tiempo real (s)	Sent packets	Retrans packets	% retrans	Total packets	Max window	Rtt est	Ancho de banda (B/s)
1.000	0.1	10	0,053	515	0	0	515	9	0,0003	9.434
1.000	0.1	100	0,060	515	0	0	515	99	0,003	8.333
1.000	0.1	500	0,061	515	0	0	515	499	0,0067	8.197
1.000	0.1	900	0,045	515	0	0	515	514	0,0062	11.111
1.000	0.3	10	0,081	515	0	0	515	9	0,0002	6.173
1.000	0.3	100	0,049	515	0	0	515	99	0,0032	10.204
1.000	0.3	500	0,118	515	0	0	515	499	0,0037	4.237
1.000	0.3	900	0,045	515	0	0	515	514	0,0034	11.111
1.000	0.8	100	0,042	515	0	0	515	99	0,002	11.904

Al comparar los timeouts para una misma window = 100 podemos observar que al aumentar el tiemeout, el bando de ancha aumenta también.

Además al mantener fijo el timeout e incrementar el tamaño de la window, se observan intervalos intermitentes, la razón principal atribuible es que se esta realizando en localhost y el tiempo real varia en torno a los decimales, y por tanto pequeños retrasos en la red producen estas alteraciones al experimentar.

Anakena

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

- **Un archivo de 500 KB**

En base a resultados anteriores, un size de 1.000 suele funcionar bien, por tanto estableceremos ese size fijo e iremos cambiando los timeouts y Windows para la obtención de métricas.

$time python3 client_gbn_bw.py **1000 timeout window** files_anakena/big_file.bin outputs_anakena/OUT_500kb 52.67.31.19 1818

- Limpiando cache

Size (bytes)	Timeout	Window	Tiempo real (s)	Sent packets	Retrans packets	% retrans	Total packets	Max window	Rtt est	Ancho de banda (B/s)
1.000	0.1	10	5,15	515	190	36,89	705	9	0,10	97,08
1.000	0.1	100	2,86	515	2075	402,91	2590	99	0,44	174,82
1.000	0.1	500	0,931	515	45	8,73	560	499	0,40	537,05
1.000	0.1	900	0,133	515	0	0	515	514	0,09	3.759
1.000	0.3	10	5,27	515	10	1,94	525	9	0,10	94,87
1.000	0.3	100	2,26	515	415	80,58	930	99	0,47	221,23
1.000	0.3	500	0,767	515	30	5,82	545	499	0,70	681,89
1.000	0.3	900	0,160	515	0	0	515	514	0,11	3.125
1.000	0.8	100	1,67	515	0	0	515	99	0,44	299,40

Al observar las mediciones para una window = 100, y aumentar el tiemout, podemos comprobar que al ancho de banda también aumentó experimentalmente.

Por otra parte, para timeout fijo, podemos observar que al ir aumentando la window el ancho de banda aumentó y los paquetes retransmitidos tendieron a la baja.

Además, como observación general, siempre se alcanzó una max window menor en solo un valor respecto a la window establecida al ejecutar el comando.

Preguntas

1. Verifique que una ventana de tamaño 1 funcione y demore lo mismo que Stop-and-Wait.

Para Stop and Wait con size 1000 y un archivo de 500KB en 52.67.31.19 con el código del profesor se obtuvo lo siguiente:

Size (bytes)	Tiempo real (s)	Sent packets	Lost packets	Ancho de banda	Tamaño final (KB)
1.000	49,78	514	107	10,04	500

En esta tarea, utilizando una ventana 1, con timeout = 0.1 (como en la T3), y un mismo archivo de 500KB en 52.67.31.19 a size = 1000 se obtuvo:

Size (bytes)	Timeout	Window	Tiempo real (s)	Sent packets	Retrans packets	% retrans	Total packets	Max window	Rtt est	Ancho de banda (B/s)
1.000	0.1	1	50,01	515	107	20,77	622	0	0,089	9,99

Por tanto, los resultados fueron iguales, incluso la cantidad de paquetes pérdidos para Stop-and-Wait coinciden con los retransmitidos para Go-Back-N

2. ¿El protocolo funciona con una ventana de tamaño 1000? ¿Debería fallar?

El protocolo Go-Back-N debería de ser capaz defuncionar con una ventana de tamaño 1000, sin embargo, para esta implementación en particular, si debiese de mostrar incongruencias, por ejemplo obtenemos el siguiente resultado:

Size (bytes)	Timeout	Window	Tiempo real (s)	Sent packets	Retrans packets	% retrans	Total packets	Max window	Rtt est	Ancho de banda (B/s)
1.000	0.1	1000	0,11	515	515	100%	515	514	None	4.545

A partir del cual podemos observar que rtt est dio None, esto se debe a que no llego a recibir un ACK (implícito) que fuera válido. Con una ventana demasiado grande sucede que el receptor recibe el paquete del tamaño n, pero el emisor no puede diferenciar si es un envío actual o uno anterior, es decir, no diferencia entre los ciclos, y debido a esto, el protocolo falla.

3. Una gracia de Go-Back-N es que un ACK para n implica un ACK para todo i <= N en la ventana de envío. En esta tarea, que no hay ACKs, ¿esta propiedad sirve de algo? ¿O simplemente es inútil?

Aunque no hay una implementación de los ACKs de forma implícita, la propiedad si resulta útil, esto se debe por la parte del cliente se utiliza la secuencia del paquete de la respuesta, y con ello avanza en la base, aplicando la propiedad de que un ACK vale para todo i<=N, y por tanto el cliente hará el proceso de avanzar en su ventana sin la necesidad de un ACK individual por paquete (es decir, el procotolo Go-Back-N).

Comparación

Al comparar la T1 con la T3 pudimos observar que el protocolo Stop and Wait suele ser mucho más lento, disminuyendo considerablemente para sizes más grandes debido a la retransmisión de paquetes perdidos. Para el caso de la T4 podemos ver que esto se repite, siendo más lenta que la T1 (donde no se hace nada respecto a retransmitir paquetes), y al compararla con la T3 podemos observar que al aumentar el tamaño de la ventana el ancho de banda supera con creces las mediciones para Stop and Wait.

No hablo específicamente respecto a comparar el ancho de banda con el de la T1 ya que pude observar que los tiempos al medir en anakena y en el otro servidor difieren (en el nuevo me tomaba un tiempo real del doble que en anakena).