

Tarea 1

Cliente Echo UDP

Alumno: Ignacio Pinto
Profesor: José M. Piquer
Auxiliares: Diego S. Wistuba La Torre
Manuel Olguín Muñoz
Ayudantes: Gabriel Montañana
Roberto Araya Day
Sebastián Contreras P.
Fecha de entrega: 1 de Octubre de 2021
Santiago, Chile

README

- Para correr el programa se debe ejecutar desde la línea de comandos con la siguiente sintaxis:

```
py client_echo3.py archivo.txt tamaño_maximo
```

Un ejemplo de uso sería:

```
py client_echo3.py bee_movie.txt 1500
```

- Para generar los archivos de textos se hizo un script que genera múltiples archivos de texto con diferentes tamaños, para ejecutarlo basta con correrlo con un intérprete de python:

```
py generador.py
```

Esto generará 6 archivos de texto con tamaños de paquetes fijos, si se desea hacer otros habría que cambiar la lista sobre la que se itera.

Pregunta 1

Se efectuaron múltiples mediciones con distintos tamaños de archivo y una serie de tamaños máximos de paquete.

Los resultados obtenidos se muestran en las tablas adjuntas a continuación:

Archivo de 1000000 bytes

max_packet_size [bytes]	loss [bytes]	time [s]
10	423650	1.3693899
100	394400	0.1807877
200	378400	0.1035257
300	322300	0.0772472
400	287600	0.0747387
500	325000	0.0542369
600	492000	0.0431685
700	568100	0.0404828
800	419200	0.0466465
900	379800	0.052891
1000	99000	0.0518755
5000	935000	0.0563237
10000	890000	0.0590569
15000	910000	0.042697
18000	910000	0.0513474
21000	874000	0.0509554
25000	875000	0.0460368
28000	580000	0.0580534
31000	876000	0.0418407
35000	685000	0.057051
39000	766000	0.0627028
41500	792500	0.0544628
45000	910000	0.0529319
47600	999600	0.0402576
50000	900000	0.0419269

Archivo de 10000 bytes

max_packet_size [bytes]	loss [bytes]	time [s]
10	3220	0.0396884
100	0	0.0244143
200	0	0.0276266
300	0	0.020748
400	0	0.0255191
500	0	0.0253308
600	0	0.0343302
700	0	0.0182676
800	0	0.0238889
900	0	0.0235217
1000	0	0.0331096
5000	0	0.0344445
10000	0	0.0322835
15000	0	0.030084
18000	0	0.0292224
21000	0	0.0235173
25000	0	0.0303148
28000	0	0.024719
31000	0	0.0335413
35000	0	0.0235208
39000	0	0.0336806
41500	0	0.0239085
45000	0	0.0292765
47600	0	0.0222641
50000	0	0.0249134

Archivo de 1000 bytes

max_packet_size [bytes]	loss [bytes]	time [s]
10	0	0.0418125
100	0	0.0270658
200	0	0.0253011
300	0	0.0283166
400	0	0.0241758
500	0	0.0267121
600	0	0.0341221
700	0	0.0299882
800	0	0.0239978
900	0	0.0296491
1000	0	0.0249838
5000	0	0.0314633
10000	0	0.0204265
15000	0	0.0206798
18000	0	0.0298287
21000	0	0.0339857
25000	0	0.0334902
28000	0	0.0276273
31000	0	0.0267118
35000	0	0.0794851
39000	0	0.0318569
41500	0	0.0280581
45000	0	0.0300239
47600	0	0.0174995
50000	0	0.0230877

Con la información de la primera tabla se confeccionan los siguientes gráficos:

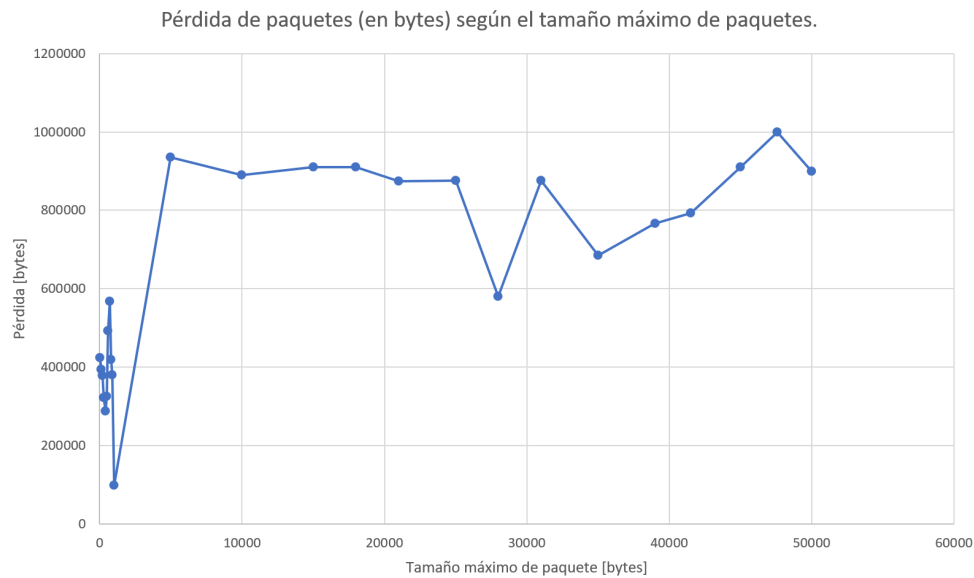


Figura 1: Pérdida vs. Tamaño Máximo de paquetes para archivo de 1000000[bytes]

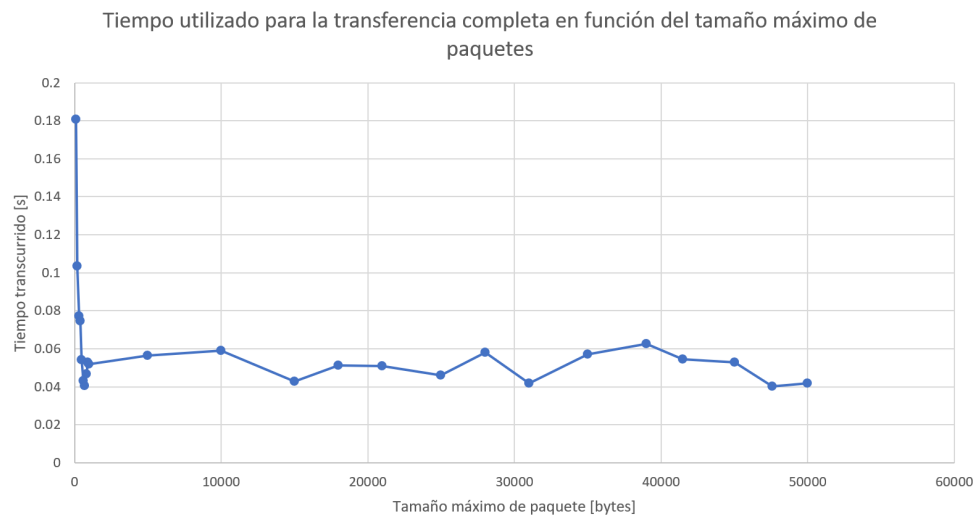


Figura 2: Tiempo transcurrido vs. Tamaño Máximo de paquetes para archivo de 1000000[bytes]

A partir del análisis del gráfico y de las tablas obtenidas se puede reconocer que si hay alguna influencia entre el tamaño máximo de paquetes y la cantidad de bytes perdidos ya que como se puede observar en el gráfico de la primera tabla hay un valle que optimiza la mínima pérdida de paquetes.

Con el tiempo, sin embargo, no ocurre lo mismo, si bien hay oscilaciones entre los 0.04 y 0.06 no parece haber una relación por lo que esta información no es suficiente para concluir alguna relación para esta variable.

Pregunta 2

Con los datos obtenidos de la pregunta anterior la recomendación sería utilizar paquetes de máximo 1000[bytes] para minimizar la pérdida de paquetes.

Pregunta 3

El tamaño máximo del paquete UDP del socket del sistema operativo si puede ser cambiado, sin embargo, existe un límite físico dado por las capas de más abajo, esto significa que podemos movernos en el rango mientras no sobrepasemos ese límite, por ejemplo, en la capa 2 para enlaces ethernet hay un límite estándar de 1500 bytes donde si los paquetes superan este tamaño se fragmentan en varios paquetes más pequeños que no sobrepasan los 1500 bytes.

En Linux se pueden cambiar los parámetros `SO_SNDBUF` y `SO_RCVBUF` para cada socket.