

Gabriel Gorospe

SID: 1696580

Lab04

Due: March 2, 2023

Test Cases:

1) Control

- a) 1 client, 3 servers, unique ip's and ports for each server
- b) droppc: 5
- c) window size: 5
- d) mtu: 25000
- e) File size (Discord.dmg)

Results: All files replicated successfully

2) Increase droppc to 10

- a) 1 client, 3 servers, unique ip's and ports for each server
- b) droppc: 10
- c) window size: 5
- d) mtu: 25000
- e) File size (Discord.dmg)

Results: All files replicated successfully

3) Increase window size to 10

- a) 1 client, 3 servers, unique ip's and ports for each server
- b) droppc: 5
- c) window size: 10
- d) mtu: 25000
- e) File size (Discord.dmg)

Results: failed 2 times after 3 trials

4) Increase total clients to 3

- a) 3 client, 3 servers, unique ip's and ports for each server
- b) droppc: 5
- c) window size: 5
- d) mtu: 25000
- e) File size (Discord.dmg)

Results: All files replicated successfully

5) Decrease mtu to 8000

- a) 3 client, 3 servers, unique ip's and ports for each server
- b) droppc: 5
- c) window size: 5
- d) mtu: 25000
- e) File size (Discord.dmg)

Results: All files replicated successfully

The documentation should describe how to use your application and the internal design, as well as any shortcomings it might have.

Usage:

`./myserver <port> <droppc> <root folder path>`

`./myclient <servn> <servaddr.conf> <mtu> <winsz> <in file path> <out file path>`

Design:

- Threading in client to handle multiple servers (array of threads of size *servn*)
- Argument struct which is used to pass variables to the threaded function. Holds a pointer to a `char**` array representing all the packets to be sent to the server, total number of packets to be sent, the port, ip, and window size.
- In main, an array of the argument structs is allocated to store the arguments of each thread

Shortcomings:

- Difficult to record DATA and ACK information to make graph, need to implement code for this rather than doing it by hand.

