Mp0 Report

Yuhang Chen 3170111144

Wenbo Ye 3170112243

1. Outline of the codebase and introduction on the design

As mentioned in the mp0.pdf, the main task of mp0 is to make a log file logging the data transmission between the client and server. My code is based on python language and work with TCP in python socket. The node.py file work as the client file to send messages produced by the local generator to the server (where we simply use the local server). The logger.py work to make logs in the local machine by receiving data from the server and doing the data processing and do data processing for drawing the graphs needed.

1. Libraries or package used in the codebase

Sys, socket, time, csv, threading

1. Description on how to measure the delay and bandwidth

The delay can be measured by the difference between the actual time we receive the message on the local server and the actual time that the message was sent from the node. Specifically, the actual time we receive the message can be recorded as we call time.time() in the logger.py file. The actual time we sent the message is recorded and sent by the node.py file and recorded as the log time in the log record.

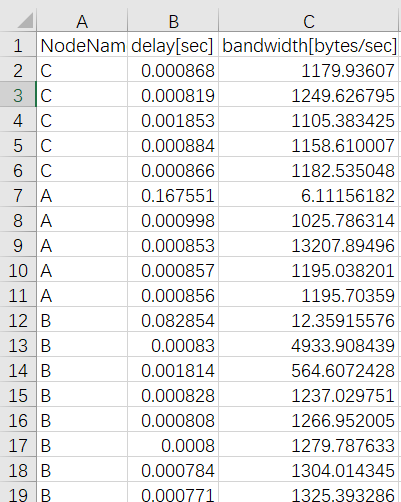
The bandwidth can be measured by the transmitted bytes per seconds from the client to the server, since we use the local machine as both the client and the server, we can simply use the delay as the difference of timestamps between the two node(which is actually one node). which means the transmitted bytes divided by the delay.

Figure 1. The screenshot on a csv file of some trial in logging

1. Testing

Since we need to accurately print out the log file, we can do the testing on the cmd window and check if there exits bugs or something unreasonable by typing

timeout 5 && python3 -u logger.py 8080 | python3 -u generator.py 10 20| python3 -u node.py A 127.0.0.1 8080 &

timeout 5 && python3 -u logger.py 8080 | python3 -u generator.py 10 20| python3 -u node.py B 127.0.0.1 8080 &

timeout 5 && python3 -u logger.py 8080 | python3 -u generator.py 10 20| python3 -u node.py C 127.0.0.1 8080 &

By printing some variables we use in the codes also help a lot to figure out the bugs in the programming procedure (such as the deltat\trans\_bytes)