MP0 Report

Xiao Shuhong 3180111551
Pan Bo 3180111515

1. Introduction

Mp0 mainly composes of two part: the first part is a basic communication network contain a given generator, a node used to initialize the TCP connection and hand out events and a logger which is responsible for receiving and recording the events through network; the second part is a experiment checking the delay and bandwidth situation with relatively different event generating rate, the csv file named with Raxx.csv will contain all the generated data and a plot file is used for the results.

2. Library used

socket, sys, threading, time, csv, matplotlib, pandas, numpy

3. Test code

For part 1, test code below can be used directly, the generating rate, event number and port number can be changed.

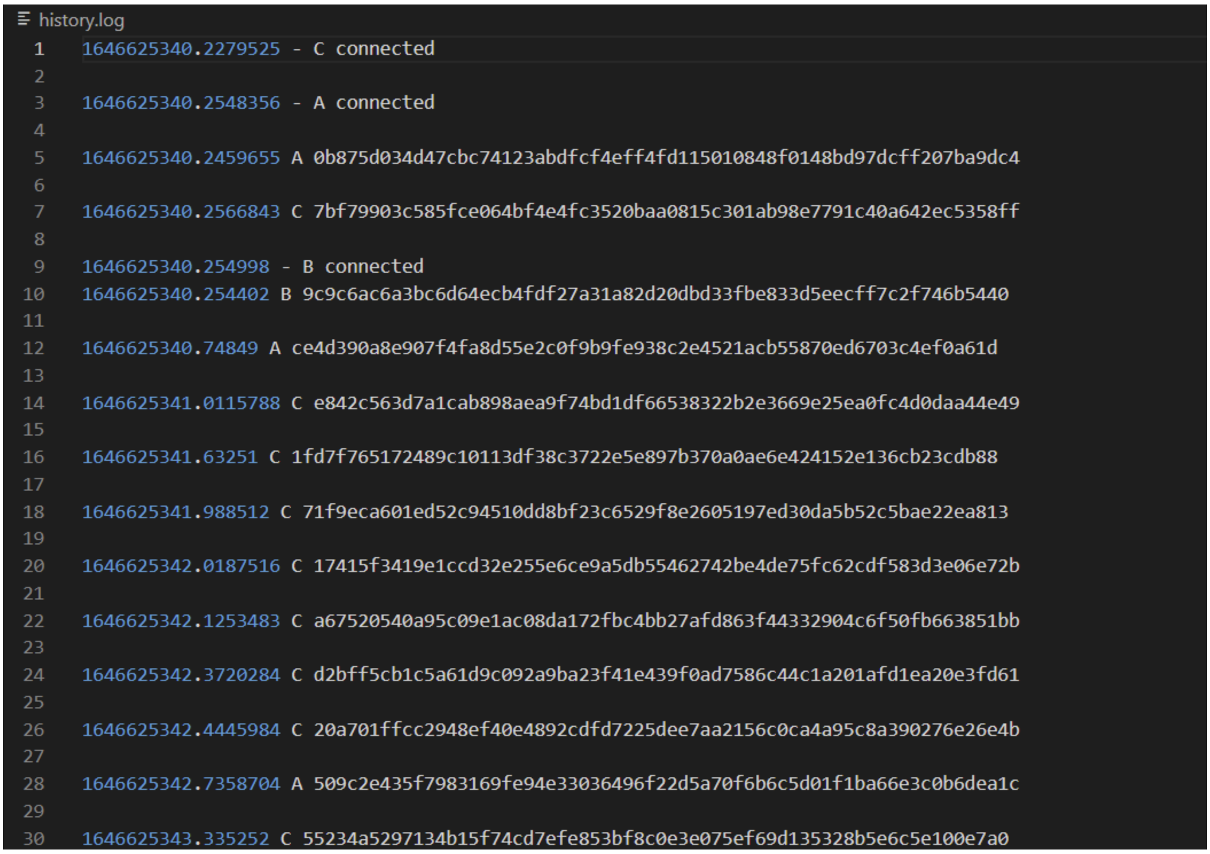
```
sleep 10 && python3 -u generator.py 5 10 | python3 node.py A 127.0.0.1 8080 &

sleep 10 && python3 -u generator.py 5 10 | python3 node.py B 127.0.0.1 8080 &

sleep 10 && python3 -u generator.py 5 10 | python3 node.py C 127.0.0.1 8080 &

python3 -u logger.py 8080 > history.log
```

The event received from logger will be written into history.log like the following.



```
1646625340.2279525 - C connected

1646625340.2548356 - A connected

1646625340.2459655 A 0b875d034d47cbc74123abdfcf4eff4fd115010848f0148bd97dcff207ba9dc4

1646625340.2566843 C 7bf79903c585fce064bf4e4fc3520baa0815c301ab98e7791c40a642ec5358ff

1646625340.254998 - B connected
1646625340.254402 B 9c9c6ac6a3bc6d64ecb4fdf27a31a82d20dbd33fbe833d5eecff7c2f746b5440

1646625340.74849 A ce4d390a8e907f4fa8d55e2c0f9b9fe938c2e4521acb55870ed6703c4ef0a61d

1646625341.0115788 C e842c563d7a1cab898aea9f74bd1df66538322b2e3669e25ea0fc4d0daa44e49

1646625341.63251 C 1fd7f765172489c10113df38c3722e5e897b370a0ae6e424152e136cb23cdb88

1646625341.988512 C 71f9eca601ed52c94510dd8bf23c6529f8e2605197ed30da5b52c5bae22ea813

1646625342.0187516 C 17415f3419e1ccd32e255e6ce9a5db55462742be4de75fc62cdf583d3e06e72b

1646625342.1253483 C a67520540a95c09e1ac08da172fbc4bb27afd863f44332904c6f50fb663851bb

1646625342.3720284 C d2bff5cb1c5a61d9c092a9ba23f41e439f0ad7586c44c1a201afd1ea20e3fd61

1646625342.4445984 C 20a701ffcc2948ef40e4892cdfd7225dee7aa2156c0ca4a95c8a390276e26e4b

1646625342.7358704 A 509c2e435f7983169fe94e33036496f22d5a70f6b6c5d01f1ba66e3c0b6dea1c

1646625343.335252 C 55234a5297134b15f74cd7efe853bf8c0e3e075ef69d135328b5e6c5e100e7a0
```

For part 2, we will first generate 10 csv file contain node, delay and bandwidth corresponding to each Ra. For each Ra, you should run like below and remember to change the generating rate for node A. In every run, there will be a data.csv contain the information we want, you should manually change its name to Raxx.csv with xx related to the generating rate from 0.1 to 1.0.

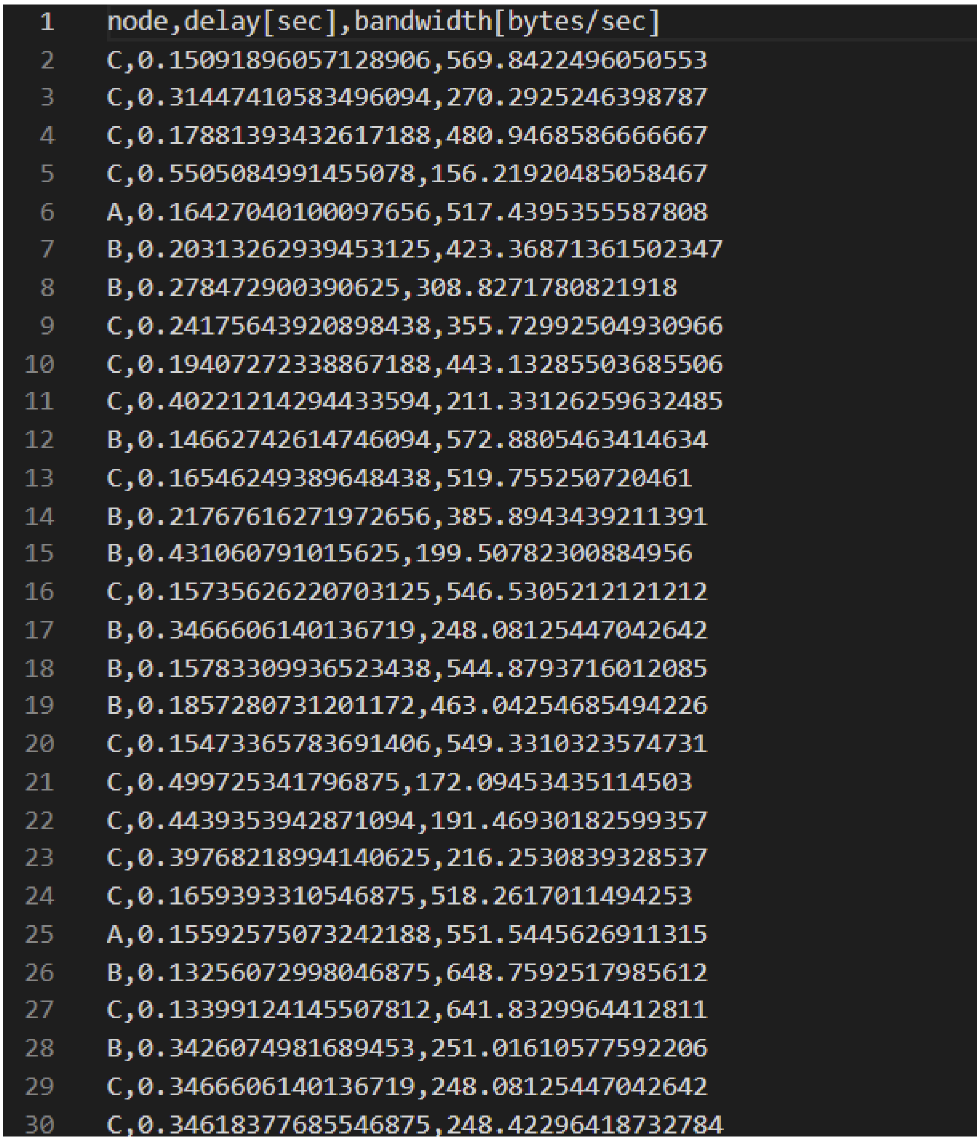
```
sleep 10 && python3 -u generator.py 1 100 | python3 node.py A 127.0.0.1 8080 &

sleep 10 && python3 -u generator.py 1 100 | python3 node.py B 127.0.0.1 8080 &

sleep 10 && python3 -u generator.py 2 100 | python3 node.py C 127.0.0.1 8080 &

python3 -u logger.py 8080 > history.log
```

The csv file generated will be like the following.



```
node,delay[sec],bandwidth[bytes/sec]
C,0.15091896057128906,569.8422496050553
C,0.31447410583496094,270.2925246398787
C,0.17881393432617188,480.9468586666667
C,0.5505084991455078,156.21920485058467
A,0.16427040100097656,517.4395355587808
B,0.20313262939453125,423.36871361502347
B,0.278472900390625,308.8271780821918
C,0.24175643920898438,355.72992504930966
C,0.19407272338867188,443.13285503685506
C,0.40221214294433594,211.33126259632485
B,0.14662742614746094,572.8805463414634
C,0.16546249389648438,519.755250720461
B,0.21767616271972656,385.8943439211391
B,0.431060791015625,199.50782300884956
C,0.15735626220703125,546.5305212121212
B,0.3466606140136719,248.08125447042642
B,0.15783309936523438,544.8793716012085
B,0.1857280731201172,463.04254685494226
C,0.15473365783691406,549.3310323574731
C,0.499725341796875,172.09453435114503
C,0.44393539428710094,191.46930182599357
C,0.39768218994140625,216.2530839328537
C,0.1659393310546875,518.2617011494253
A,0.15592575073242188,551.5445626911315
B,0.13256072998046875,648.7592517985612
C,0.13399124145507812,641.8329964412811
B,0.3426074981689453,251.01610577592206
C,0.3466606140136719,248.08125447042642
C,0.34618377685546875,248.42296418732784
```

After all 10 files are generator, you can simply run python3 -u myplot.py to generate the figure for delay and bandwidth, there will be six figures names delay.jpg and bandwidth.jpg each contains the maximum, minimum and average.

4. Data generated for task2

There are 3 types of data generated in task2: node, delay and bandwidth. The node can be directly get for the message receive from socket; to calculate delay, we first get the event generating time also from message through socket, then generate a local received time and subtract them to get the delay; for bandwidth, we use the bytes send each time divide the delay.

5. Experiment result

First, for the delay part, both max delay and min delay change a lot, as there is also event generated early but arrive late as well as those arrive very quickly. While from the average part, we could say that as the Ra become larger, the average delay will become relatively smaller.

For the bandwidth part, also both the max bandwidth and min bandwidth bounce a lot, while from the average, as the Pa increase, which made the difference between generating rate of A, B and C become smaller, the bandwidth used increase.