MP0: Event Logging Report

Zhicong Fan(zhicong2), Kehang Chang(kehang2)
02/08/2020
Cluster Number: 01 (Under mp0_group folder) (All other VMs are clients)
Revision Number: 85c376fd709546f5e7016a283ad7ea412f549747
Website: https://gitlab.engr.illinois.edu/zhicong2/cs425mps_group35/tree/master

Introduction:

In this mp0, we completed a logging system by creating a server and up to 9 clients. In the logger, events, time delay and bandwidth are recorded in order so as to plot the graphs representing the relationship among these clients. In the server terminal, events are printed as required as the time they are received.

Code Implementation:

Server:

In the server, we firstly introduced a function to create logger filename, such that once a server is started, we check if the corresponding logger file is created, if not, we build one. Then, we wrote a function to support us write into the logger. The third function we implemented is used to determine what to do when a message is received by the server. We tried to poll from the clients' channels, if the channel is a new channel, we use the channel as the key in the map matching with the client. Based on this, when we are polling from the channels, once we find that a channel has no response, we know that the client is disconnected when we will print out the information it disconnected. If not, we calculate the delay and the bandwidth, writing them into the logger.

Client:

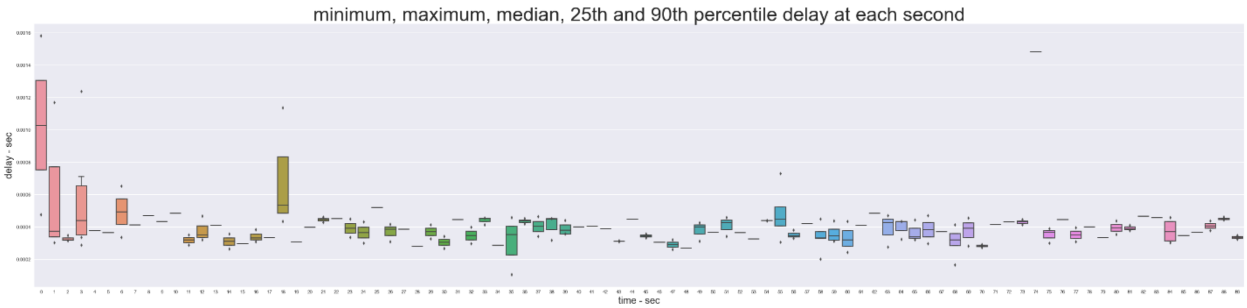
In the server, we only have the main function where we request the host port and dialing to the server. By receiving messages generated by the generator.py, the client constantly sending events to the server.

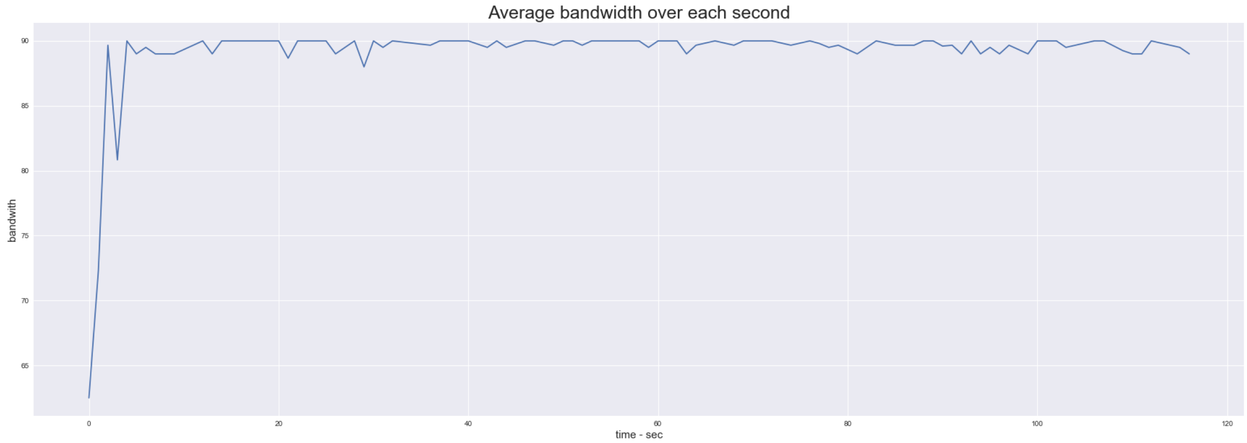
Graph:

One single delay is measured by calculating the time difference between the time the event is generated on the client server and the time the event is being received on the host server. The maximum, minimum, median, 25th, 90th percentile are calculated by the dataset over that entire second range. One single bandwith is calculated by calculating the length of each message received by the server. The average bandwidth over each second is found by calculating the mean of the dataset over each second.

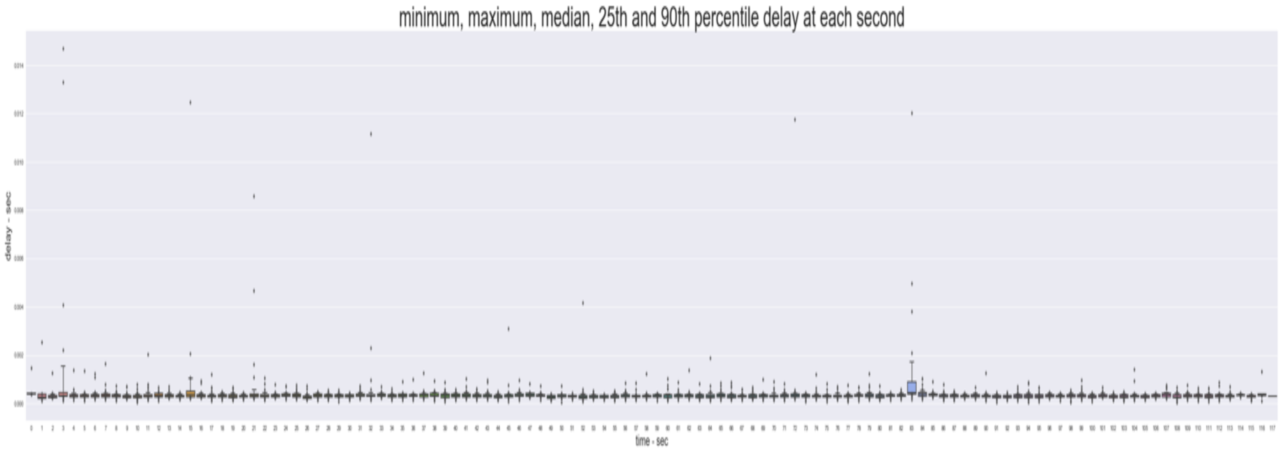
Language used: "Go" for server and client, "Python" for plotting the graph

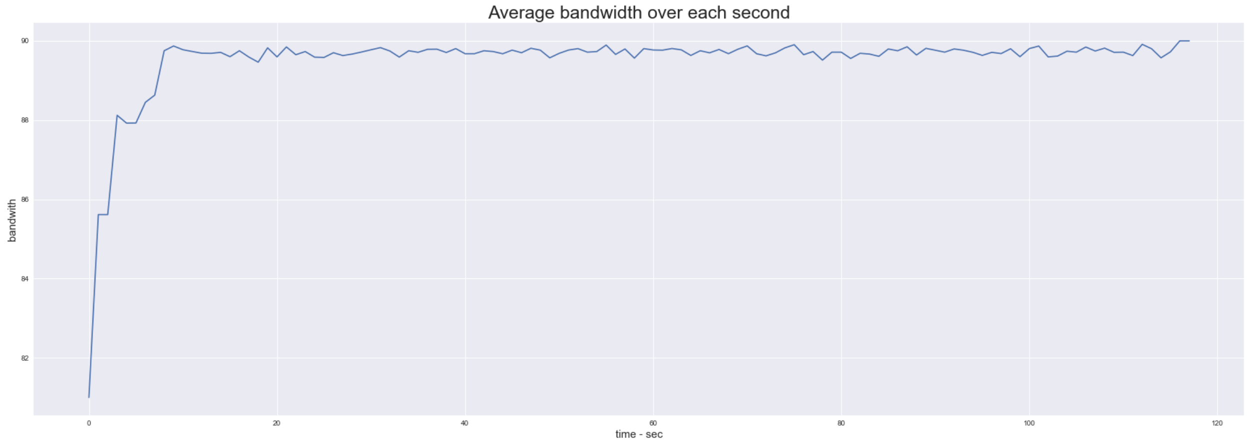
3 nodes, 0.5 Hz each, running for 100 seconds:





8 nodes, 5 Hz each, running for 100 seconds:





To run the code:

1. go build mp0.go
2. go build client.go
3. ./mp0 1234
4. python3 -u generator.py [frequency you want] | ./client node(number you want) 172.22.158.115:1234