SENG3500 Laboratory 3 Report – Simulation Experiment III

Introduction

The purpose of the simulations in this report is to examine the TCP congestion control technique, specifically the TCP Reno protocol. Since we are examining traffic load, the primary focus is on the number of clients within our simulations. We will analyse the effects of increasing numbers of clients using a series of data plots, checking congestion window size (*cwnd*), round trip time (RTT), number of duplicate ACKs (DupACKs) and queue length.

Simulation Model

The simulations will run from a basic client-server network model, using a router as an intermediary between the client and server. Depending on the simulation, the network model will contain 1-3 clients, all sharing a single router. Each simulation will run for 200 secs.

We will run 3 simulations in total. Simulation 1 will consist of 1 client, simulation 2 of 2 clients and simulation 3 of 3 clients. From these simulations will come the following plots:

Simulation 1:

- Congestion window size (cwnd) for client #0.
- RTT for client #0.

Simulation 2:

- Congestion window size (cwnd) for client #0.
- RTT for client #0.
- Router queue length for client #0 and client #1.

Simulation 3:

- Congestion window size (cwnd) for client #0 and #2.
- RTT for client #0.
- No. of duplicated ACKs received by client #0.

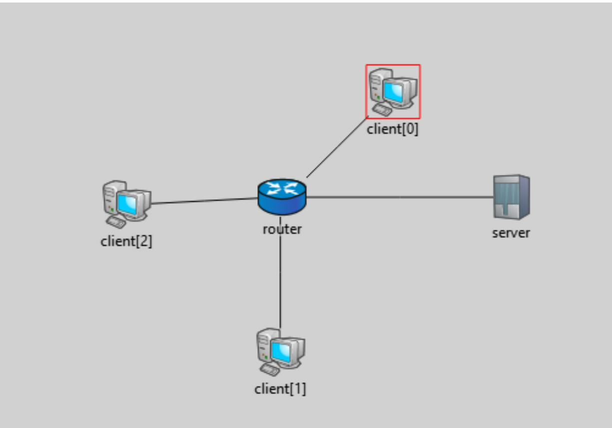

Figure 1. Network with 3 clients, a router and a server.

Results

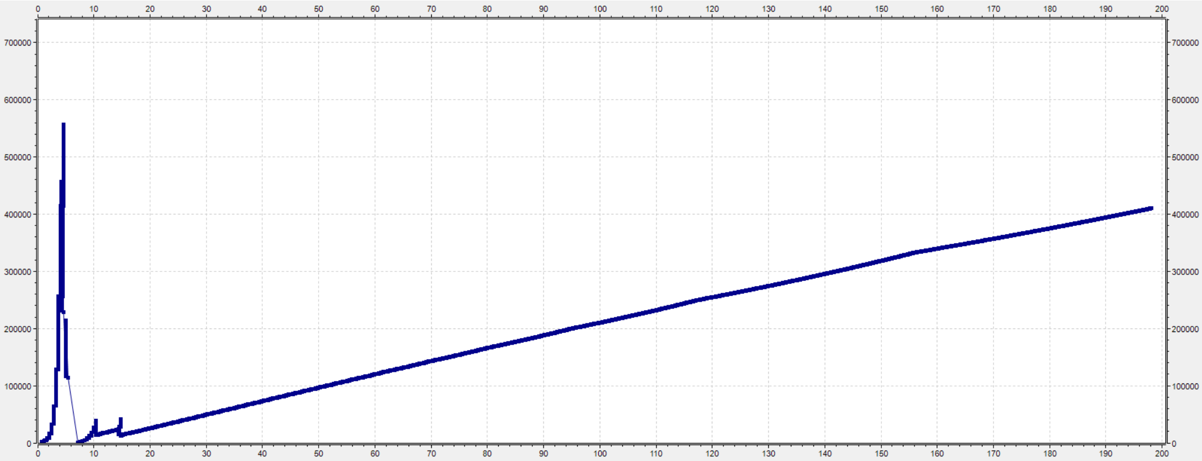

Figure 2. cwnd, client #0, sim. 1

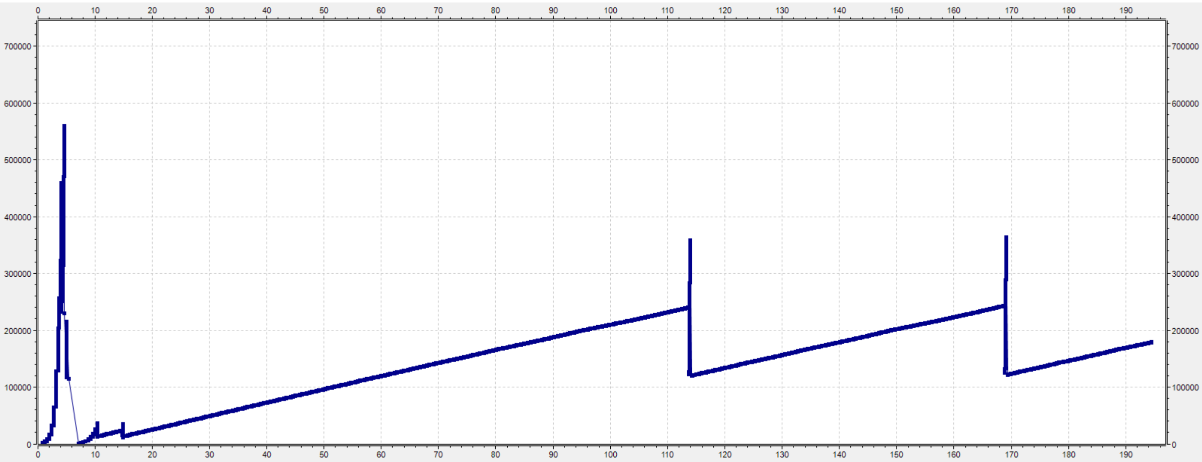

Figure 3. cwnd, client #0, sim. 2

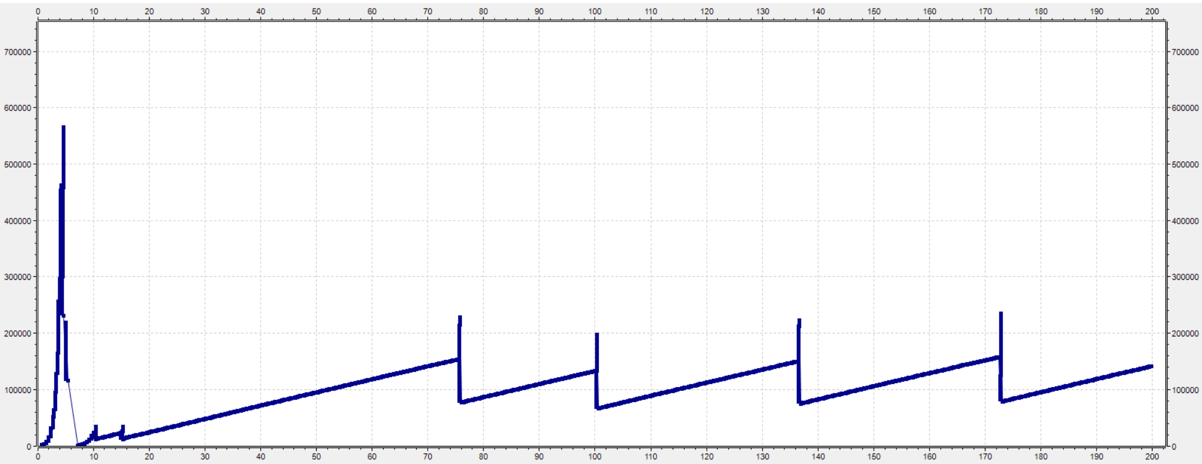

Figure 4. cwnd, client #0, sim. 3

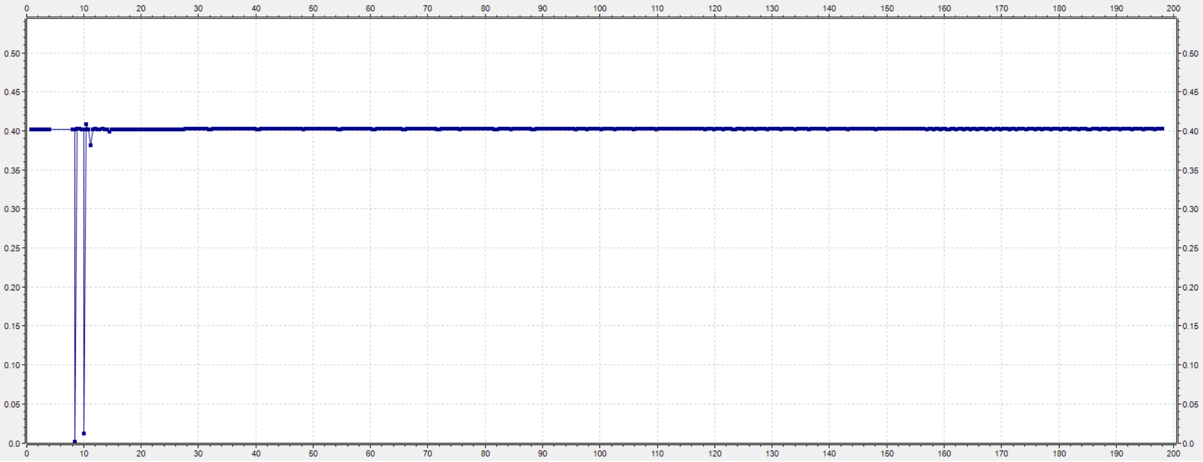

Figure 5. measured RTT, client #0, sim. 1

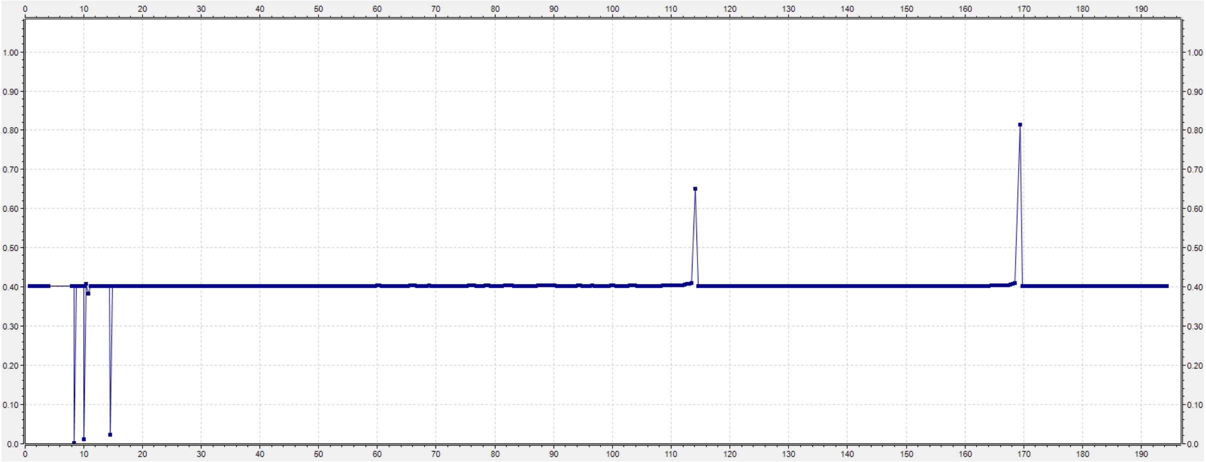

Figure 6. measured RTT, client #0, sim. 2

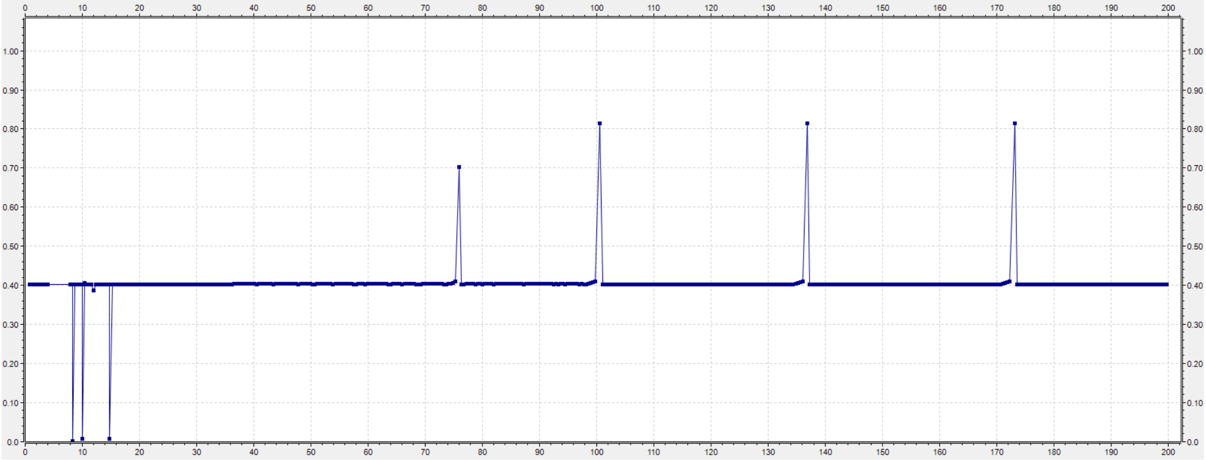

Figure 7. measured RTT, client #0, sim. 3

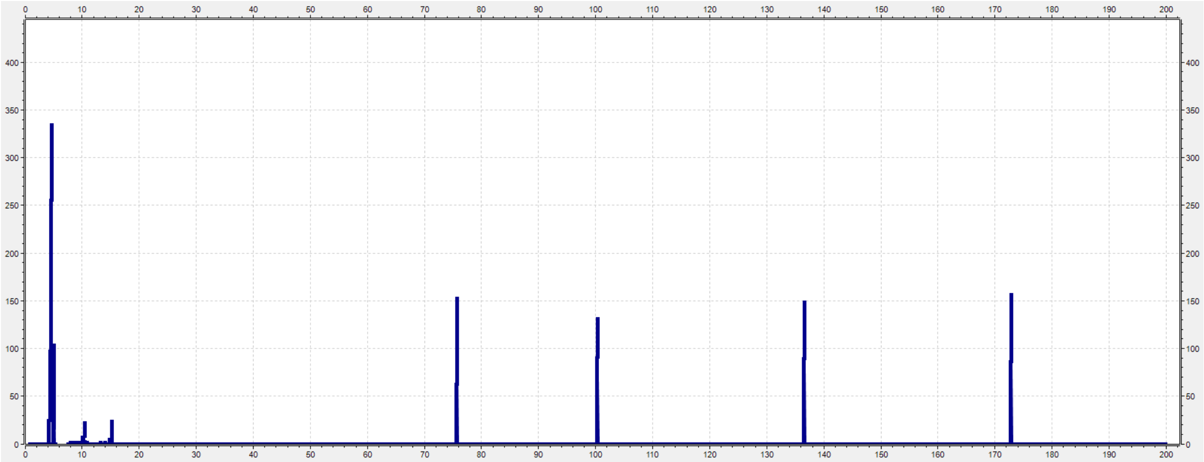

Figure 8. rcvd DupACK, client #0, sim. 3

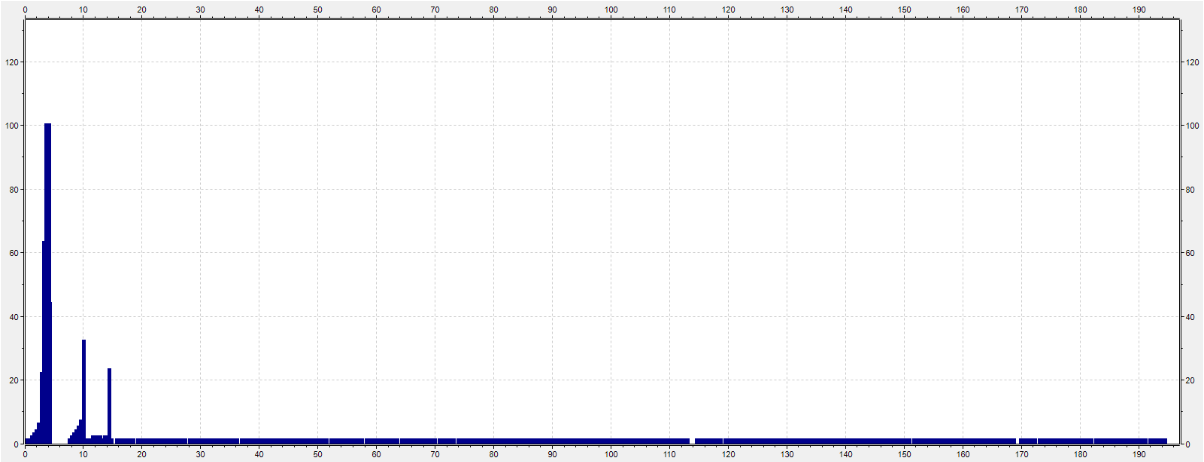

Figure 9. queueLength:vector, client #0, sim. 2

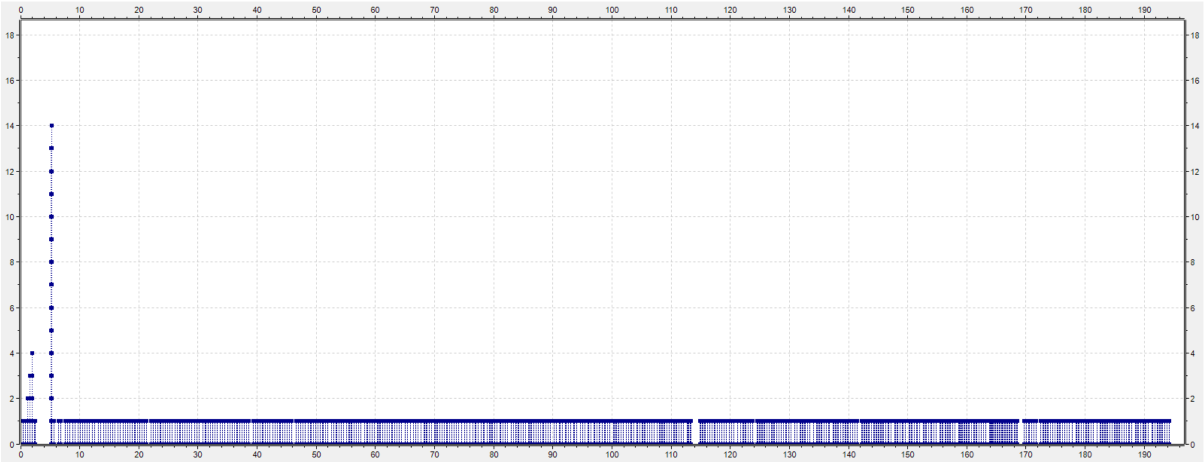

Figure 10. queueLength:vector, client #1, sim. 2

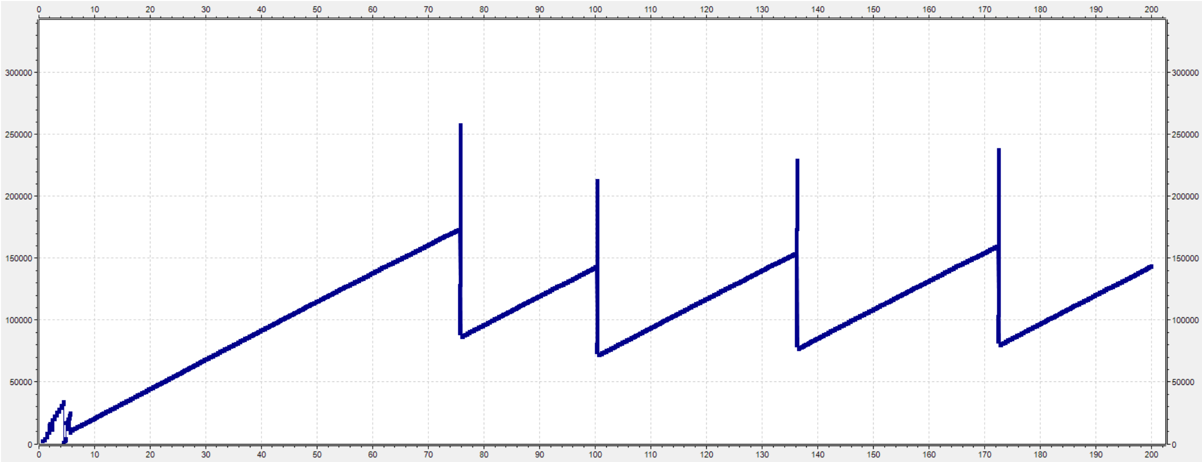

Figure 11. cwnd, client #2, sim. 3

Analysis

Loss events occur in figure 4 (2 clients) at around 240000-250000 *cwnd* and in figure 5 and 12 (3 clients) at around 140000-160000 *cwnd*. Additionally from figure 3 we can see that *ssthresh* is set to around 310000 MSS, meaning that for more than 1 client, congestion has reached the point where loss is indicated by 3 duplicate ACKs – *cwnd* does not reach *ssthresh* before congestion controls are implemented.

Due in part to these more frequent loss events, the total and average *cwnd* sizes decrease with more clients. Under TCP Reno, *cwnd* growth is exponential until either a loss event occurs or *ssthresh* is reached, being linear afterwards. This means a network with more clients will spend less time in the initial growth stage, more time in the linear growth stage and will incur more loss events, halving *cwnd* per client each time.

RTT remains relatively static during periods of *cwnd* growth, keeping to the average of 0.4 regardless of simulation or client. Sharp increases in RTT correlate with *cwnd* loss events and similar increases in dupACKs. The only exception to this is the sharp *decreases* in RTT seen near the beginning of the simulations, correlating with periods of drastic exponential growth followed by loss events.

Knowledge

1. *Explain why the client congestion window (cwnd) size varies differently for different load conditions.*

The *cwnd* size determines how many bytes can be sent at any given time, therefore total/average *cwnd* size will be larger when there is less congestion. Load conditions that increase traffic congestion will result in lower *cwnd*. Increasing the number of clients in the network will add congestion.

2. *Analysing the DupACK plot, do you see any correlation between the congestion window size and the number of DupACKs received by a client? Briefly explain.*

As seen when comparing Figs. 5 and 12 to Fig. 9, the number of DupACKs received by a client increases sharply during a loss event. In all three plots, this can be observed at 76, 100, 136 and 172 seconds. Another, much larger DupACK spike at be observed at 5 seconds, which corresponds to an equally large *cwnd* spike in Fig. 5. Notably, this *cwnd* spike does not occur in Fig. 12.

In TCP congestion control, loss events can occur when the client receives 3 duplicate ACKs for any ACK. Under TCP Reno, this results in *cwnd* being set to (*ssthresh* + 3) MSS. This also means the number of duplicate ACKs received by a client has a direct effect on *cwnd* – the sharp increase in means that eventually at least one ACK will have 3 duplicate ACKs, triggering a loss event and easing congestion by lowering *cwnd*.

3. *Consider the client #0 and client #2 congestion window plots in simulation 3. Are both clients achieving the same average throughput or different? If different, which client has higher average throughput?*

Generally, both clients are achieving approximately the same throughput. However, client #0 experiences a chaotic period of growth and loss that could skew the overall results. This coincides with a period of significantly lower RTT (see Fig. 8).

This discrepancy occurs within the first 10-15 seconds of the simulation. For the first 5 seconds, client #0 shows major exponential growth, reaching a *cwnd* of around 570000 before timing out. They proceed to experience two more loss events before *cwnd* begins to steadily climb. From this point, the *cwnd* of both clients appears to be even for the rest of the simulation.