

Mitchell Caisse

mwcaisse

CS4513

Project 4

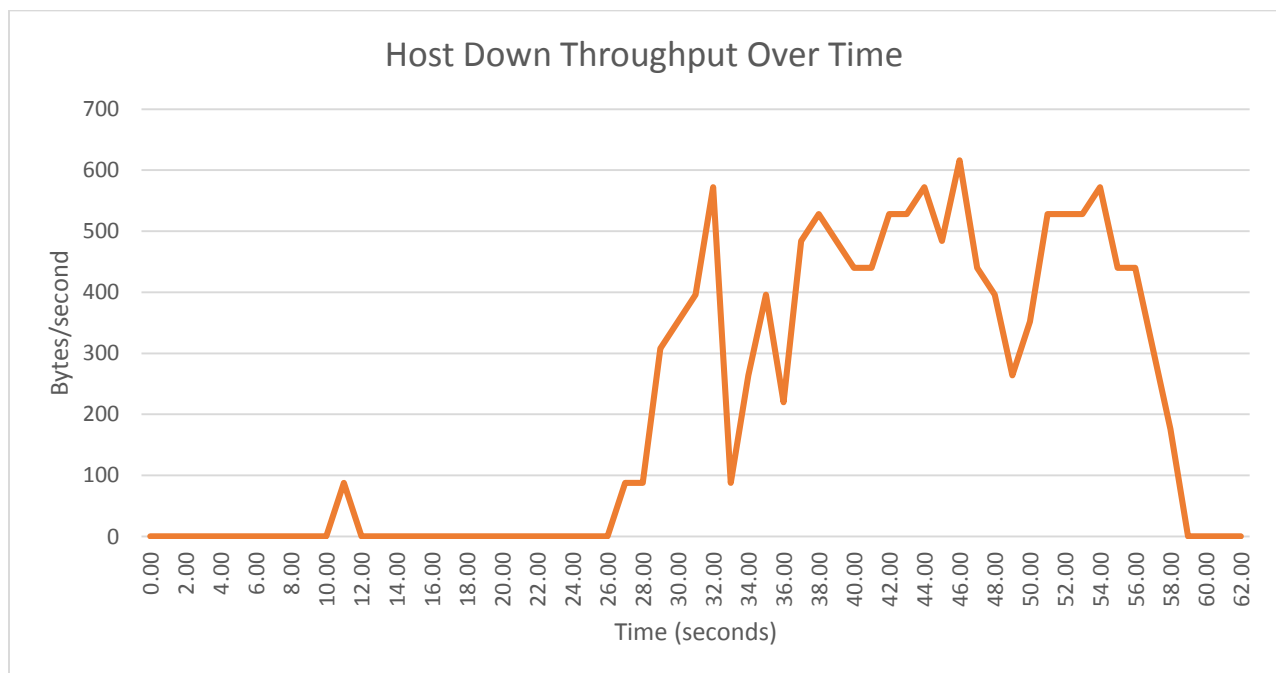
## Design

I designed my experiments by creating a NetworkStats class to record throughput and latency between the client and server. The NetworkStats class subscribes to network events, and each time it receives a message it records how long it was, and adds to the byte counters. Then every second it prints out the bitrate for that second. At the end it prints out the total number of bytes transferred, average throughput, and the average message size.

The NetworkStats also periodically sends messages to the client / host to test the round trip time between them. Every 60 steps or two seconds it sends out a message, and when the other party receives it, it returns it back. The RTT is then calculated by comparing the timestamp in the message to the current time. Each time a RTT message is received it prints out the RTT.

To perform the experiment I used one of the CCC machines and a computer located off-campus. The game was played for a little over a minute, with the network statistics enabled.

## Results



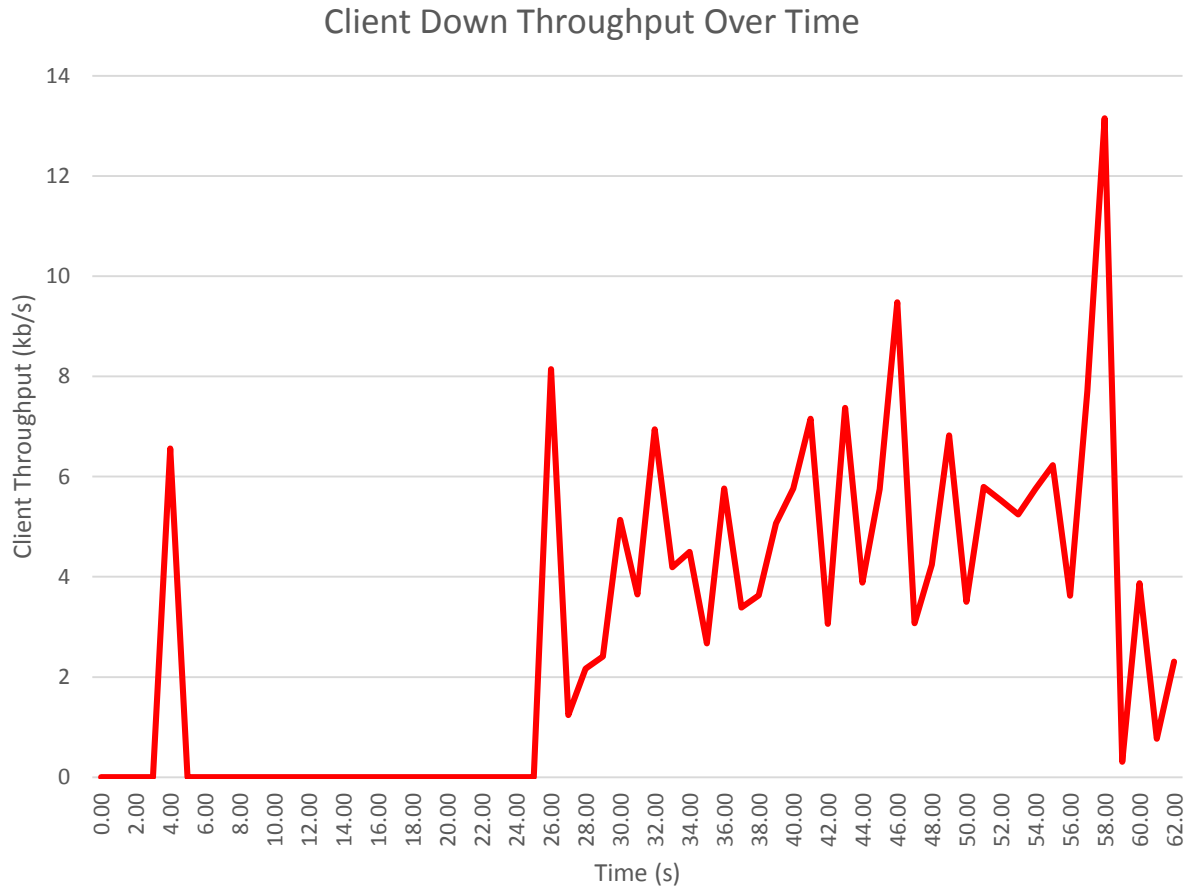


Table 1: Host and Client Throughput

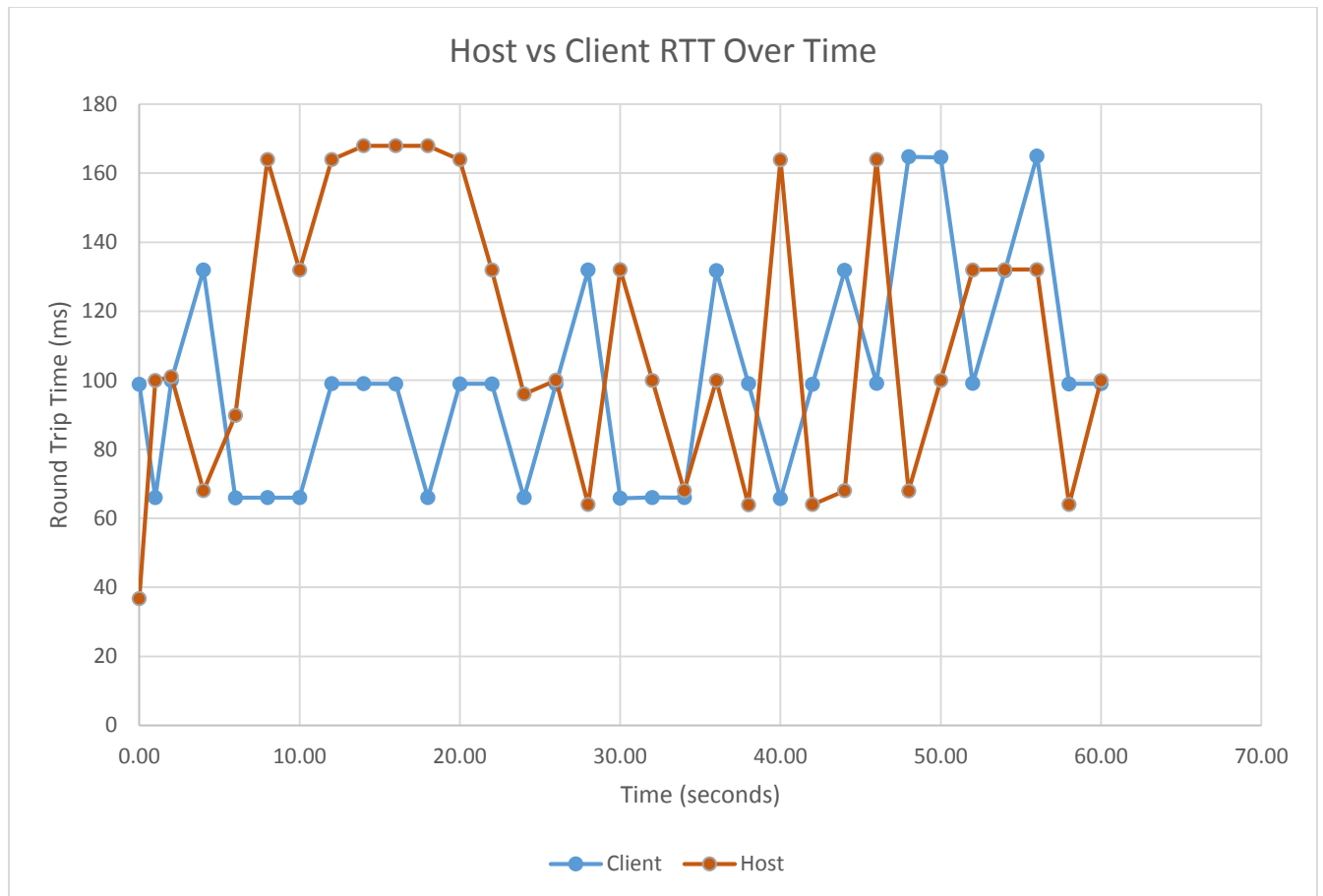
	Average (kb/s)	Max (kb/s)	Min (kb/s)	Stdev (kb/s)
Host	0.205	0.616	0	0.225
Client	2.950556	13.153	0	3.065915

Table 2: Host and Client Message Sizes

	Average (bytes)	Max (bytes)	Min (bytes)	Stdev (bytes)
Host	239.81	461	44	157.93
Client	44	44	44	0

Table 3: Host and Client Round Trip Time

	Min (ms)	Max (ms)	Average (ms)	Stdev (ms)
Host	36.783	167.942	111.4671	39.38215
Client	65.711	164.979	100.0021	30.24144
Total	36.783	167.942	105.7346	35.57542



## Analysis

The results from the host and client throughput show that the bit rate of the game is not that high. On average the amount of data that the client received was 2kb/s, with a max of 13kb/s. While the average is skewed by the long period of no data transfer, which was the pause on the GameStart screen waiting for the game to start, it still shows that the throughput never got too high. According to the graph the throughput during the normal gameplay was around 6kb/s. This is not a great amount, and should not introduce any bandwidth related performance issues on anyone with a modern internet connection.

The results comparing the average message size sent between the host and client, shows that not much data is transferred. Over all the average message size that the host sent was 240 bytes and the maximum was 461. For the client the results were much different, as the client only sends key strokes, each message was only 44 bytes, the size of the message header. As the message size remains relatively small, the game does not consume that much bandwidth while it is running.

The results of the round trip time between the clients was a bit interesting. The round trip times from the client to the host and back were on average smaller than from the host to

the client. The round trip times also stayed around two extremes, around 60ms on the lower end and 160ms on the higher end. I was unable to obtain a base round trip time to compare the results to as both the host and client blocked ping requests. However the variation is likely do to varying network conditions and the difference in the time before the message is pulled from the socket, however this should be minimal.

Overall the results show that the game does not consume that much bandwidth nor does it require a high bit rate.