Networks Programming Assignment 1

William Frazier

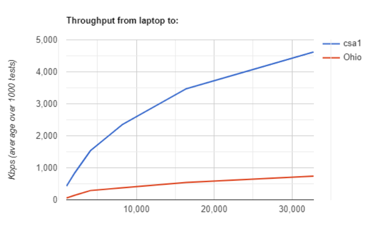
October 2020

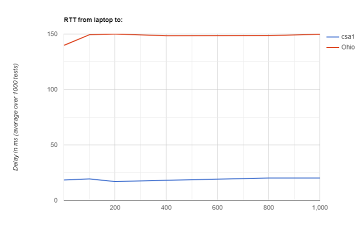
1 Experiments

Following the guidelines in the assignment, I calculated RTT and throughput of various servers. In all occasions, the client was run from my Windows laptop in my apartment in Boston. Tests were occasionally run on different days but collections of tests were always run as complete blocks (e.g., I tested every message size for RTT in a single sitting). For every test I used 1,000 messages and took the average. For calculating RTT and throughput I used the python method time.time_ns() which provides a high resolution timer. I ran my server on csa1.bu.edu (CentOS) and used the provided server 140.254.14.107 in Ohio. I ran a speedtest of the WiFi network I used and got 42.7 Mbps download and 46.0 Mbps upload.

2 Results

2.1 Server Delay 0

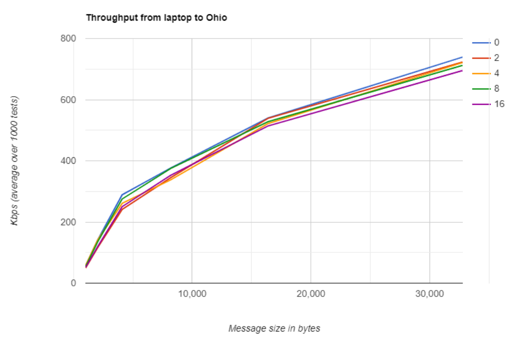


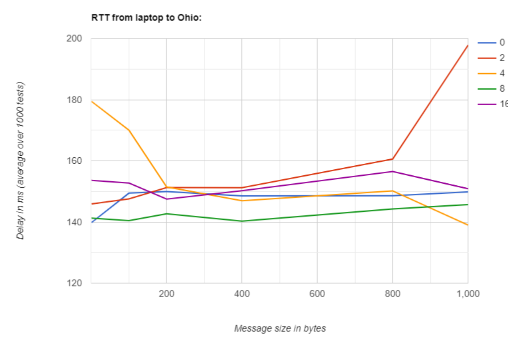


The results of my throughput testing showed that throughput continued to increase as the size of my messages increased. That being said, the rate of increased appears to be somewhat leveling off at the higher end of my graph. I tryed a small experiment of sending huge messages to the server in Ohio and confirmed that messages significantly larger have higher throughput but not massively higher. Specifically, messages of size 1Mb (roughly 30 times larger than the largest message in the graph) had a throughput of 1,192Kbps (only 1.6 times higher than the largest message in the graph). Compare this to the fact that moving from the smallest to the second smallest message size shown in the graph (a two-fold increase) results in throughput nearly three times higher.

The result was a bit stranger for RTT. There was not a persistent increase in delay as message size increased. Overall it trended this way but it was not consistent. I ran the test multiple times after seeing that result and found that this was likely just a fluke as that did not continue to be the case. Nonetheless, each point on my graph is built from an average of 1,000 tests and I don't want to not include my original data simply because it's unexpected. My intuition is that RTT is very sensitive to even small changes so those random fluctuations may have a stronger effect than message size. To demonstrate this, I also sent huge messages (around 100,000 bytes in size) and saw that RTT was over 800ms. This gives some evidence that large messages do have larger RTTs, even if the graph doesn't show this.

2.2 Varying Server Delay





Like the above experiments, throughput showed a consistent increase as the message size increased. Although there is some fluctuation, we can see that the lower the server delay, generally the higher the throughput. Server delay does not change the overall shape of the graph, only how large the throughput is.

For RTT, I once again got wildly inconsistent results. Looking only at the graph, it's not at all clear that message size or server delay have a consistent effect on the delay. Perhaps the network is just too random for these results to be clearly seen with server delay values this low but nonetheless I expected that averaging over 1,000 tests would introduce some smoothness.

If I were to run these tests again, I'd like to try with much larger server delay values. To be honest, I expected a server delay of 16 to affect the results much more than it did. I still found this result interesting as it shows just how non-uniform RTTs can be.