CSC 573 - INTERNET PROTOCOLS PROJECT 2

Experiment 1

Experiment 1 is completed with solely S1 sending traffic to D1 using TCP CUBIC. This gives us our baseline average throughput and average flow completion time using TCP CUBIC over a 1 Gbps link for a single sender. The average throughput was 922.351 Mbps with a standard deviation of 6.104e-05 Mbps. The average flow completion time was 0.478 seconds with a standard deviation of 2.980e-08 seconds. This throughput is very close to the ideal throughput of 1000 Mbps, with a very small standard deviation. This makes sense because we are utilizing a small delay, which would be common in a data center environment.

Experiment 2

Experiment 2 was configured so that S1 and S2 are both sending traffic over the shared link to their respective destinations, D1 and D2, using TCP CUBIC. Here the average throughput for S1 was 466.614 Mbps with a standard deviation of 0.890 Mbps. Its average flow completion time was 0.941 seconds with a standard deviation of 0.002 seconds. For S2 its average throughput was 530.349 Mbps with a standard deviation of 5.690 Mbps. Its average flow completion time was 0.828 seconds with a standard deviation of 0.009 seconds. Due to both senders competing over the shared link, the average throughput and average rate of flow completion time decreased from experiment 1 (per link), while the throughput was divided quite evenly between the two links. It's also interesting that both of the standard deviations for S2 increased from our previously measured standard deviations.

Experiment 3

Experiment 3 was configured so that solely S1 was sending traffic to D1, this time using DCTCP. The average throughput was 922.351 Mbps with a standard deviation of 6.104e-05 Mbps. The average flow completion time was 0.478 seconds with a standard deviation of 2.980e-08 seconds. Compared to experiment 1, which is identical to this experiment besides the variant of TCP which is used, we notice identical results. This makes sense because there is no congestion or competition between two senders, so the TCP variants operate similarly.

Experiment 4

Experiment 4 was conducted so that S1 and S2 send traffic to their respective destinations, D1 and D2, over the shared link, this time using DCTCP (as opposed to experiment 2 with TCP CUBIC). The average throughput for S1 was 464.387 Mbps with a standard deviation of 0.549 Mbps. Its average flow completion time was 0.945 seconds with a standard deviation of 0.001 seconds. For S2 its average throughput was 529.172 Mbps with a standard deviation of 12.5322 Mbps. Its average flow completion time was 0.830 seconds with a standard deviation of 0.020 seconds. Compared to experiment 3, we can conclude the average throughput and average rate of flow completion time have decreased (per link), due to the competition of the shared resource of the link. However, comparing this to experiment 2, which is identical to this experiment except for the variant of TCP which is used, we can similarly note that throughput is divided nearly evenly between the two links. We can also conclude that in these two experiments, TCP CUBIC and TCP DCTCP have very similar average throughput and average rate of flow completion time. It makes sense that this is similar, as in both experiments (2 and 4) we have two competing variants of TCP, and TCP is designed to achieve fairness between flows of the same variant.

Experiment 5

In this experiment, S1 is sending traffic to D1 using TCP CUBIC, while S2 is simultaneously sending traffic to D2 using DCTCP. The average throughput for S1 was 465.937 Mbps with a standard deviation of 0.672 Mbps. Its average flow completion time was 0.943 seconds with a standard deviation of 0.001

seconds. For S2 its average throughput was 540.409 Mbps with a standard deviation of 15.330 Mbps. Its average flow completion time was 0.813 seconds with a standard deviation of 0.023 seconds. Comparing these values, we can conclude that in this experiment, TCP DCTCP outperforms TCP CUBIC when contending over the resource of a shared link, in terms of throughput (although throughput is generally split evenly between the two links) as well as in average flow completion time. Comparing these results to experiments 2 and 4 (which use the same TCP variant with 2 senders over a shared link), we conclude that differing variants of TCP being used has a significant impact on which variant performs better. Experiments 2 and 4 have the first link performing better than the second, while DCTCP is the second link in this experiment, which performs better. This makes sense that DCTCP has a slightly higher throughput, because we have a low point-to-point delay, which simulates a data center environment (which DCTCP is designed for). Additionally, we configured DCTCP to adapt quickly and aggressively to congestion, which is another reason why it came out slightly on top for throughput. More information on our specific configuration for DCTCP can be found in the readme_hcench_adlapp file.

Conclusion

TCP CUBIC and TCP DCTCP have very similar performance with one sender over a single link, as well as when there are two senders of the same variant contending over the shared link. TCP DCTCP outperforms TCP CUBIC in a small network with a 1 Gbps shared link, with 2 competing senders. This is due to the way we configured TCP DCTCP to react quickly to congestion, and our data center based setup of our network environment (due to our small point-to-point delay). It’s also important to note that by decreasing the point-to-point delay, our average throughput significantly increased while our average flow completion time decreased (for experiments with a single sender). This makes sense, since we are sending a small amount of data over a comparably large link, with no competition.