CS5229 Project

AY 2020/2021 Sem 1

**Zhang Hao A0210996X**

**Link to experiment code and results:**

[**https://github.com/iamzhanghao/CS5229-Advanced-Computer-Networks**](https://github.com/iamzhanghao/CS5229-Advanced-Computer-Networks)

**Task 1 (6pt)**

The objective for task 1 is to observe how different TCP variants are affected by packet loss. You need to modify topology\_parkinglot.py to perform the following experiments.

1. (8pt) Measure the throughput for a single TCP flow (H1 -> H8 using *iperf3 -C option*). Also, use ping to measure the RTT between H3 to H7 and record the maximum RTT observed for the following cases
   1. (1pt) A single TCP Cubic flow (with myLossPercentage = 0) from H1 to H8.

H1->H8: 47.3Mbps

H3 <-> H7 max RTT: 1302.06ms

* 1. (1pt) A single TCP Cubic flow (with myLossPercentage = 1) from H1 to H8.

H1->H8: 1.66Mbps

H3 <-> H7 max RTT: 44.086ms

* 1. (1pt) A single BBR flow (with myLossPercentage = 0) from H1 to H8.

H1->H8: 46.3Mbps

H3 <-> H7 max RTT: 131.33ms

* 1. (1pt) A single BBR flow (with myLossPercentage = 1) from H1 to H8.

H1->H8: 46.6Mbps

H3 <-> H7 max RTT: 54.862ms

* 1. (2pt) Explain your observations in the above experiments.

From the experiment results in (a) and (c), we can notice both BBR and TCP flows can achieve high throughput that utilise the link bandwidth under condition that no random packet loss, however, With the flow of TCP Cubic running, the max RTT become significantly large (1302.06ms) comparing to the flow with TCP BBR running (131.33ms). This indicates TCP Cubic flow will take up huge amount of buffer size in switches that caused queuing latency.

From (b) and (d) , with the introduction of random packet loss 1%, running both BBR flow and Cubic flow, other connections can still achieve relatively low RTT. However, TCP Cubic flow under this condition showed a dramatic drop in throughput(from 47.4Mbps to 1.66Mbps) while TCP BBR can still maintain the same level of throughput.

The conclusion is that both BBR and Cubic can achieve high throughput with no random packet dropping. Also, BBR cause less link congestion comparing to Cubic. Under condition of random packet loss, BBR still achieve high throughput and take up very few buffer in switches while Cubic behave badly in throughput but acceptable in RTT. In this experiment, BBR can get the best throughput and RTT in either condition.

**Task 2 (10pt)**

The objective for task 2 is to observe how buffer size affects the performance of different TCP variants. You need to modify topology\_parkinglot.py to perform the following experiments.

1. Set **myLossPercentage** to 0. Vary **myQueueSize** and measure the total throughput over all TCP flows. Also, use ping to measure the RTT between H3 to H7. This RTT measurement provides an estimate for the buffer occupancy for the following 2 cases:
   1. 1 TCP Cubic flows (H1->H8 using *iperf3*).
   2. 1 BBR flows (H1->H8 using *iperf3*).
   3. (2pt) For each of the 2 cases, plot how **total throughput** varies with different buffer sizes (myQueueSize). Chart

      Description automatically generatedChart, line chart

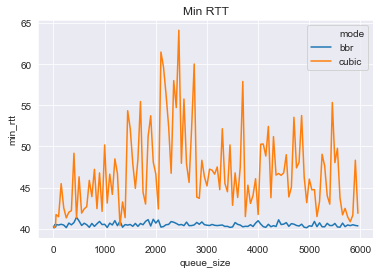
      Description automatically generated
   4. (2pt) For each of the 2 cases, plot how **RTT** varies with different buffer sizes (*myQueueSize*).

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generatedChart, histogram

Description automatically generatedChart, line chart

Description automatically generatedChart, line chart

Description automatically generatedChart, histogram

Description automatically generatedChart, line chart

Description automatically generated

* 1. (2pt) Explain if use of large buffer is always a good option.

No using large buffer is not always a good option.

* Large buffer usually cost more in hardware because of larger cache size, it may not be cost effective
* In the case of TCP Cubic flow, a large buffer will cause significant queuing delay if the number of packets in the buffer is large
* When the buffer size in range of small (0-40), increase the buffer size will improve the network throughput and RTT, however, as buffer continue increasing, network throughput become stagnant couldn’t improve further and RTT become larger in the case of Cubic flows
  1. (2pt) Explain how you could choose the “right” buffer size for **TCP Cubic** flows for this experimental configuration.

For this configuration, queue size of 20 seems to be the optimal value. At this configuration, maximum throughput around 47Mbps can be achieved. Also, average RTT, max RTT, min RTT and mdev RTT all achieved the lowest value.

* 1. (2pt) Explain how you could choose the “right” buffer size for **BBR** flows for this experimental configuration.

The optimum values achieved after queue size of 40. When queue size is larger than 40, there isn’t any significate improvements for throughput and RTT, therefore, the “right” buffer size for BBR flows in this configuration is 40, if we consider using less hardware resources.

**Task 3 (8pt)**

The objective for task 3 is to observe how differences in RTT affects the performance of different TCP variants. You need to modify topology\_parkinglot.py to perform the following experiments.

Consider 2 flows are served by the same server but have different end-to-end RTTs. Ideally, to provide a fair service, the hosting server should provide similar throughput to both flows.

To testify this, set the experiment as follows; **Flow 1** goes from H8->H1 and **Flow 2** goes from H8 to H4. Let Flow 1 start 10 sec before Flow 2. Set **myLossPercentage** to 0, **myQueueSize** to 1000, and fix the delay of one bottleneck link <S2, S3> to 10ms. You can vary the delay of the other bottleneck link <S1, S2> to emulate different delay between S1 and S2. Use enough monitoring time (e.g., > 300 seconds).

Timeline

Description automatically generated

1. (2pt) Use only TCP Cubic, plot how the final flow throughputs of H8->H1 and H8->H4 vary with different end-to-end delays by varying the delay between S1 and S2.

Chart, line chart

Description automatically generatedChart, line chart, histogram

Description automatically generated

1. (2pt) Use only BBR, plot how the flow throughputs vary with different end-to-end delays by varying the delay between S1 and S2.

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

1. (4pt) Explain your observations in the above experiments.

In the case of BBR, when delay value is low, the flow started first(flow1) will take advantage. As delay increases, the throughput is equal around 800ms delay. Thereafter, the flow started later(flow2) will take advantage. Since BBR is delay based congestion control, as delay between S1 and S2 increase, the delay of flow 1 will increase while flow 2 has less delay. Therefore, flow 1 throughput will become slower as delay between S1 and S2 increase. Flow 2 still has the same delay therefore it will use up the remaining link bandwidth.

As for Cubic case, flow 1 and flow 2 has similar throughput for the range of delay value 0-300ms, however, flow 1 has slightly higher throughput on average. After throughput is greater than 300ms, the throughput of flow1 continuously decreases while flow2 increases. Since Cubic is loss-based congestion control, as delay between S1 and S2 increases while the delay is low(<300ms), very few packets is dropped at S2. As delay continue increasing, buffer in S2 filled up quickly and more packets are dropped, then the throughput of flow 1 is impacted. Flow2 takes up the remaining bandwidth and its throughput increases.

**Task 4 (16pt)**

In the near term, the internet would likely be dominated by TCP Cubic and BBR. Discuss the potential problems or issues. Design your own experiments to support your arguments/discussions.

You can use the parking lot topology or create your own topology to perform your experiments. However, you must clearly specify the topology (or topologies) you use.

You can also vary the number of TCP flows (number of hosts), TCP variants, **myBandwidth, myDelay, myQueueSize, and, myLossPercentage.**

As Internet continues to evolve and develop, it will become more busy with various type of traffic. Especially in the near future TCP BBR will become more famous because of its overall better performance. However Internet is huge and contains billions of devices, it is not possible to let all application switch to the same protocol at the same time. Thus, there will be a prolonged period of mixing TCP variants in the Internet. In the following experiments, we only focus on one TCP Cubic flow and one TCP BBR flow. Testing is using Parking Lot Topology with different number of switches in between hosts, the testing parameters are following:

* **myBandwidth=50Mbp**
* **myDelay=10ms**
* **myQueueSize= variable**
* **myLossPercentage = variable**

In all the testing scenarios:

1. Start iperf3 server on H1 and H2
2. Start iperf3 TCP Cubic flow on H3 -----> H1
3. Wait for 50 seconds
4. Start iperf3 TCP BBR flow on H4 -----> H2
5. Wait for 200 seconds
6. Stop TCP BBR flow
7. Wait for 50 seconds
8. Stop TCP Cubic flow
9. ***BBR and Cubic, 2-Hop Parking Lot Topology, Variable Queue Size, No Loss***

In this experiment, we will use the following topology, this is a simplified parking lot topology.

A close up of a sign

Description automatically generated

Running experiments with queue size with queue size in range 0 – 3000, we get following results

Chart, line chart, histogram

Description automatically generatedChart, line chart

Description automatically generated

From the plots on the left, when queue size is less than 100, throughput of BBR is better than Cubic. When queue size is larger than 100, throughput of Cubic is better than BBR. To take a detailed look of what exactly happen when BBR flow started transmission, the detailed TCP transfer speed vs times is plottedChart

Description automatically generated

When queue size is 72, it is very clear that TCP BBR took advantage at T=50s when it started transmission, and after BBR flow finishes, TCP Cubic flow restored to the same level of throughput.

Chart

Description automatically generated

When queue size is 850, TCP BBR slowly increased its speed to roughly same as TCP Cubic flow, then they kept at equilibrium states. After transmission finishes, BBR flow gained full bandwidth of the link.

Chart, histogram

Description automatically generated

When queue size is 2900, TCP BBR struggled in crease its speed. The overall speed is far lower than TCP BBR flow

1. ***BBR and Cubic, n-Hop Parking Lot Topology, Variable Queue Size, No Loss***

In this experiment, we try to increase the number of hops from host to host to simulate real world internet connection, which usually involves several hops

1. 2-Hop Topology (same as section I)

A close up of a sign

Description automatically generated

Chart, line chart, histogram

Description automatically generatedChart, line chart

Description automatically generated

1. 4-Hop Topology

Chart, waterfall chart

Description automatically generated

Chart, line chart

Description automatically generated

1. 8-Hop

Chart, waterfall chart

Description automatically generated

Chart, line chart

Description automatically generated

From the results in 1) 2) 3) , we can observe there is always a change in average throughtput as we increase the queue size. And the point of banlance become at larger queue size as the number of hops increases.

1. ***BBR and Cubic, 8-Hop Parking Lot Topology, Fixed Queue Size, Variable Loss***

Chart, waterfall chart

Description automatically generated

Using the same 8-hop topology with queue size = 1000, and try loss percentage from 0% -10% on link S4-S5 and S5-S6. Selecting these two links and central nodes to simulate a very bad Internet congestion.

Chart, line chart

Description automatically generated

From the results, we can see the BBR flow can handle random packet loss well, even at 10% loss the performance is still decent given that 10% is very harsh network situation. TCP Cubic almost loss all the speed, effective speed is less than 1Mbps

1. ***Conclusion***

Internet is a complex network, It is not sufficient to simulate Internet using the above topology but we can still have some findings:

* BBR performs well in bad network condition
* BBR will compete with other TCP flows in some condition. In this experiment, we can conclude that BBR will have more advantage than Cubic given max\_queue\_size is small, however, when the queue size is large enough BBR can only get very little throughput.
* Tuning the queue size in switches might be helpful to optimize network traffic and achieve better fairness.
* Combining multiple variants of TCP in the network can cause fairness issue.