Performance Analysis of TCP Variants

Li Xie
Northeastern University

Kaichen Zhang
Northeastern University

***Abstract*—In this paper, we analyze the performance of TCP variants, such as Tahoe, Reno, New Reno, Vegas during congestion by having a gradually increasing a CBR flow, fairness by comparing with a different TCP variant, and queueing algorithms by experimenting with both DropTail and RED.**

Keywords—TCP, Tahoe, Reno, New Reno, Vegas, Sack UDP, DropTail, RED

I. Introduction

TCP stands for Transmission Control Protocol, which is one of the main protocols of the Internet protocol suite. It is a standard that application programs follow to establish and maintain a network conversation so as to exchange data with other application programs, such as World Wide Web and E-mail. The original design of TCP works as expected. However, as the Internet grows, the performance of TCP suffers in large and congested networks. Therefore, several TCP variants with different models and algorithms have been suggested to control and alleviate congestion. In this paper, we will look closely at them.

II. Methodology

In order to perform a complete analysis of the TCP variants, we will use the NS-2 simulator to to perform experiments that simulate TCP on a congested network. We set up the network topology as below in Figure 1.

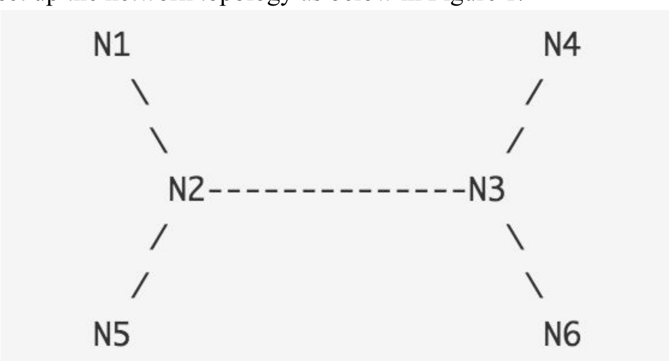

Figure 1, Network Topology

Experiment 1. TCP Performance Under Congestion

Background:

We need to compare the performance of TCP variation (Tahoe, Reno, New Reno, Vegas) under varying load conditions. The Load or Congestion is generated by a CBR source node and a rate varied by 1MB to observe the impact of transmission of TCP packets from N1 to N4. This scenario simulates the real world problem where UDP stream is unfair to TCP stream and consumes most of the bandwidth of a given channel if UDP and TCP transmissions are simultaneous. However, a well-adapted TCP variant will be able to handle congestion and perform efficiently under the influence of random traffic in the network and help deliver data more effectively without losing too much ground to other transmission protocols including other TCP variants.

Methodology:

We are considering to vary the timing and consider performance difference for these 3 scenarios:

1. We start TCP and CBR traffic at the same time for a linearly . varying the rate of CBR flow from 1 Mbps to 10 Mbps.
2. We start TCP flow first let it stabilize and then induce CBR traffic for different rates of CBR traffic.
3. We start CBR flow first let it stabilize and then induce TCP traffic for different rates of CBR traffic.

Expectation:

Of course every case will be different, but we are expecting the general trend that the higher CBR flow rate, the lower throughput. The higher the CBR flow rate, the higher packet drop rate. The higher the CBR flow rate, the higher latency. So, there will be three plots that have CBR flow rate on the x-axis and throughput, packet drop rate and latency on the y-axes respectively with NewReno, Reno, Tahoe and Vegas all on each plot for performance comparison. The best performing TCP variant remains to be seen until we have done the simulation.

Experiment 2. Fairness Between TCP Variants

Background:

We are comparing the fairness of various TCP variants in pairs with an addition of CBR flow. The CBR flow is varying. This experiment helps us analyze the fairness between TCP variants. The result should simulate what is observed in actual network, that different TCP variants are unfair to each other and some can consume more bandwidth than others. An unfair TCP variation would generally try and throttle other connections by consuming as much bandwidth as possible in a given channel shared by different TCP variants during a transmission.

Methodology:

Setup is similar to the first experiment except we are adding another TCP flow in this case to compare the fairness between different TCP variant.

Expectation:

Different case will have different result, but we do expect some TCP variant will outperform others in certain scenarios and therefore, cause unfairness. We will have multiple plots that have CBR flow rate on the x-axis and throughput, packet drop rate and latency on the y-axes respectively with two TCP variant on each plot for performance comparison.

Experiment 3. Influence of Queueing

Background:

We make use of queuing algorithms named Drop Tail and Random Early Drop (RED) to help us understand the influence of queuing discipline on the fair bandwidth and end-to-end latency (overall throughput). We have one TCP flow between N1 and N4 and one CBR flow between N5 and N6.

Methodology:

1. Vary the size of the queue and observe the change in latency.
2. Observe the change of throughput over time to see how that makes a difference.

Expectation:

We will plot time on x-axis vs throughput on y-axis as well size of queue on x-axis vs latency on y-axis. We expect one TCP variant with a specific queuing algorithm will have a better performance. We will have to do the simulation to find out which one and prove our assumption.

III. Analysis

Experiment 1. TCP Performance Under Congestion

Throughput

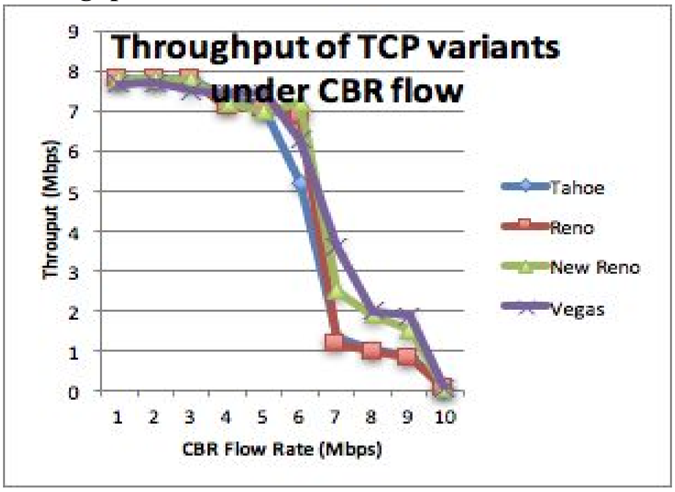

Figure 2, Throughput of TCP Variants in Congestion

As can be seen in Figure 2, New Reno performs best when CBR flow rate is low, while Vegas performs the best when CBR flow rates increases to a larger number.

Since both Tahoe and Reno set their congestion window to 1 when a RTO timeout is detected, Tahoe and Reno will not be as effective as the other TCP variants. New Rene resets the timeout timer, so that it can always have more packets in the window to have a higher throughput. Vegas can dynamically adjust throughput by the round trip delays so it can maintain a high throughput.

T-Test analysis: two sample using unequal variance test for Tahoe, Reno, New Reno and Vegas indicates that the value of t stat < t critical

Latency

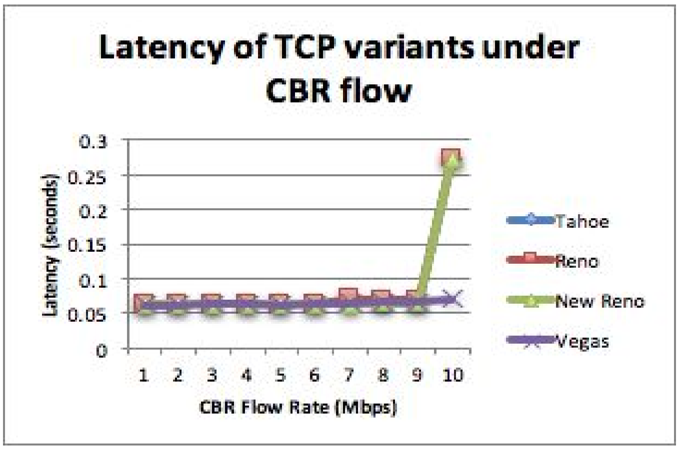

Figure 3, Latency of TCP Variants in Congestion

As can be seen in Figure 3, Vegas performs the best in latency test, especially when the CBR flow rate is high, all other TCP variants start to see big delays.

Vegas outperform other TCP variants because Vegas calculates the RTT to make sure the last packet will reach in time before retransmitting. In this way, Vegas do not waste time sending packets that will eventually be rejected. The reason that Tahoe, Reno and New Reno are not performing well is because Tahoe requires to send the entire pipe of data when retransmission. Reno and New Reno have delays because they are waiting for the 3 ACKs and 1 ACK respectively.

T-Test analysis: two sample using unequal variance test for Tahoe, Reno, New Reno and Vegas indicates that the value of t stat > t critical. Since P value is lesser than initially set alpha value, the null hypothesis cannot be justified, which proves that there is statistically significant difference between the Vegas values compared to others according to the T-Test results, however no statistically significant difference was observed amongst other three variants.

Drop Rate

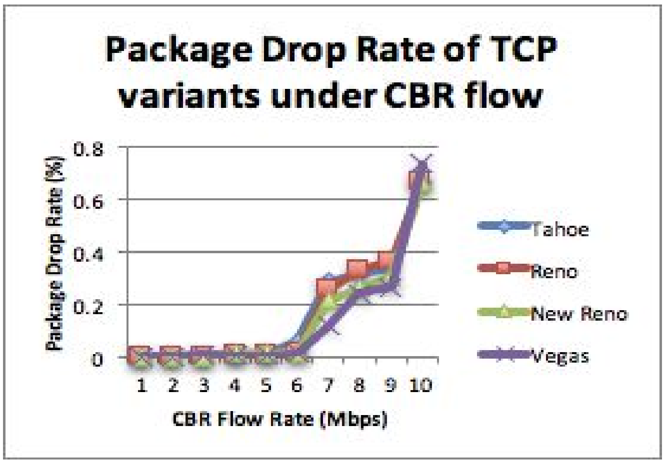

Figure 4, Packet Drop Rate of TCP Variants in Congestions

As can be seen in Figure 4, Vegas is still the winner in the drop rate test. Reno and Tahoe are the worst while New Reno performs in the middle.

Similar to the congestion avoidance, Vegas outperform other TCP variants because Vegas calculates the RTT to make sure if one packet has been lost, it avoid losing more packets by decreasing the packet sending rate. For Tahoe, packet window capacity increases linearly until a packet has lost. By the time of the detection, much of the packet has lost. Reno and New Reno halves the window capacity of congestion encountered, so when Reno and New Reno realize some packets are lost because they didn't receive the 3 and 1 ACKs respectively, they have a lower packet drop rate.

T-Test analysis: two sample using unequal variance test for Tahoe, Reno, New Reno and Vegas indicated that the value of t stat <t critical. And since P value is greater than initially set alpha value, the null hypothesis cannot be rejected, which means that there is no statistically significant difference between the values according to the T-Test results.

Experiment 2. Fairness Between TCP Variants

Reno vs Reno

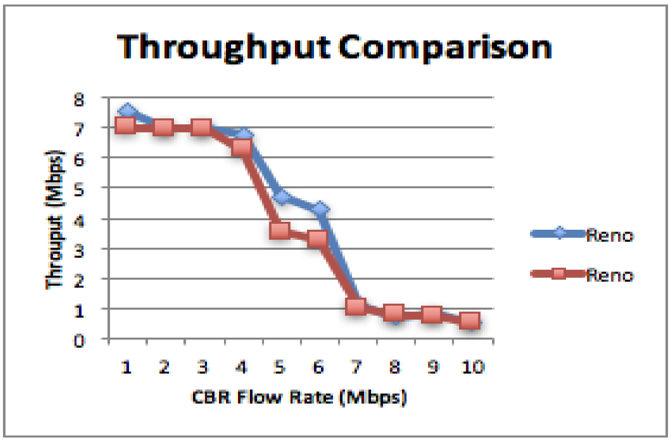

Figure 5, Reno vs Reno

As can be seen in Figure 5, the blue Reno has slight edge when the CBR flow rate is in the middle range, but become very similar when the rate increases. Both Reno are fair to each other.

This can be explained because N5 has more throughput than N1. However, with more packets have been sent, One Reno starts to drop packets and eventually the other Reno catches up on throughput.

T-Test analysis: two sample using unequal variance test between Reno and Reno indicated that the value of t stat <t critical and P value is greater than initially set alpha value that meant that the null hypothesis cannot be rejected. Therefore there is no statistically significant difference between the values according to the T-Test results.

New Reno vs Reno

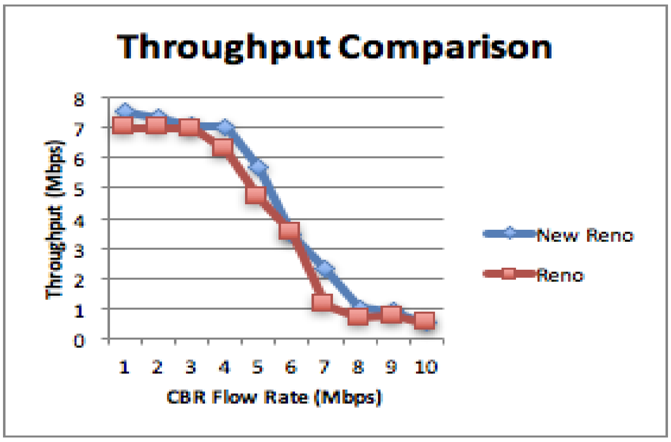

Figure 6, New Reno vs Reno

As can be seen in Figure 6, New Reno has an edge over Reno in throughput. New Reno is not fair for Reno

Similar to the congestion analysis we did for experiment 1, since New Reno only waits for ACK 1 time as opposed 3 times for Reno, New Reno always retransmits faster and has high throughput.

T-Test analysis: since both use unequal variance test, value of t stat <t critical and P value is greater than initially set alpha value, meaning that the null hypothesis cannot be rejected. Therefore, there is no statistically significant difference between the values according to the T-Test results.

Vegas vs Vegas

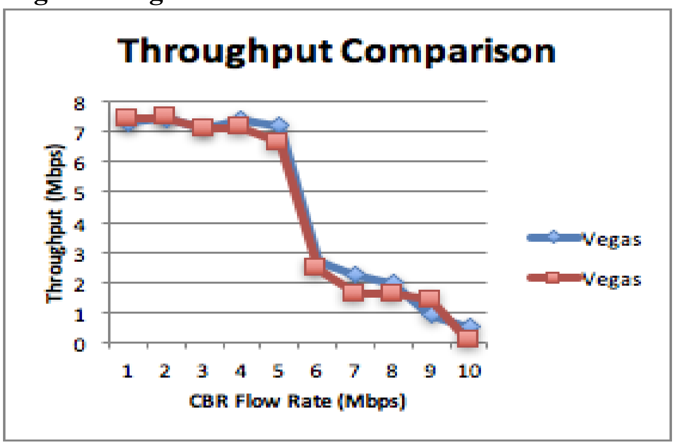

Figure 7, Vegas vs Vegas

As can be seen in Figure 7, both Vegas TCP variant have about the same throughput. Both Vegas are fair to each other.

T-Test analysis: Since two Vegas have unequal variance test, the value of t stat < t critical. Known P value is greater than initially set alpha value, the null hypothesis cannot be rejected, meaning that there is no statistically significant difference between the values according to the T-Test results.

New Reno vs Vegas

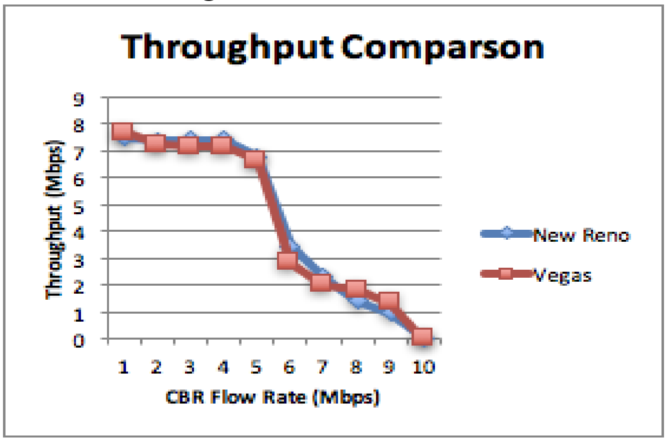

Figure 8, New Reno vs Vegas

As can be seen in Figure 8, New Reno has more throughput than Vegas overall. New Reno is not fair to Vegas.

Since Vegas has a preference in dealing packet latency than packet drop, when congestion happens, Vegas decreases its throughput.

T-Test analysis: Since New Reno and Vegas use unequal variance test, the value of t stat > t critical. With P value lesser than initially set alpha value, the null hypothesis cannot be justified, which illustrates that there is statistically significant difference between the Vegas value as compared to New Reno.

Experiment 3. Influence of Queueing

Latency Comparison

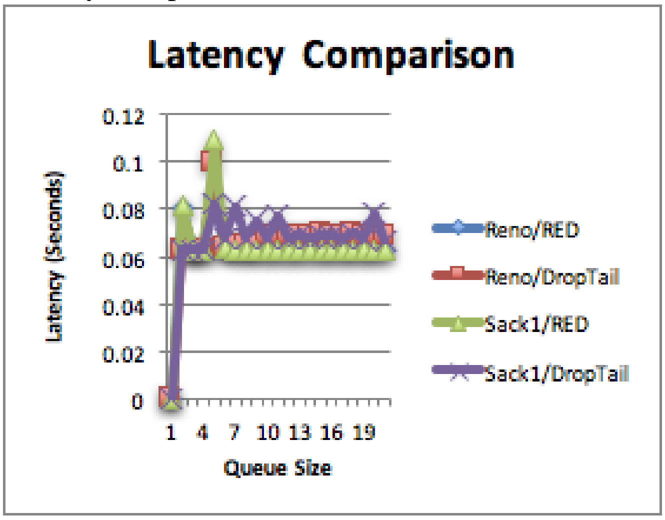

Figure 9, Latency Comparison for Queueing Algorithm

Figure 9 is not very clear, but when zooming in, we can discover that RED algorithm has lower latency when queue size increases. When congestion happens, RED is more stable compared to DropTail.

Throughput Comparison

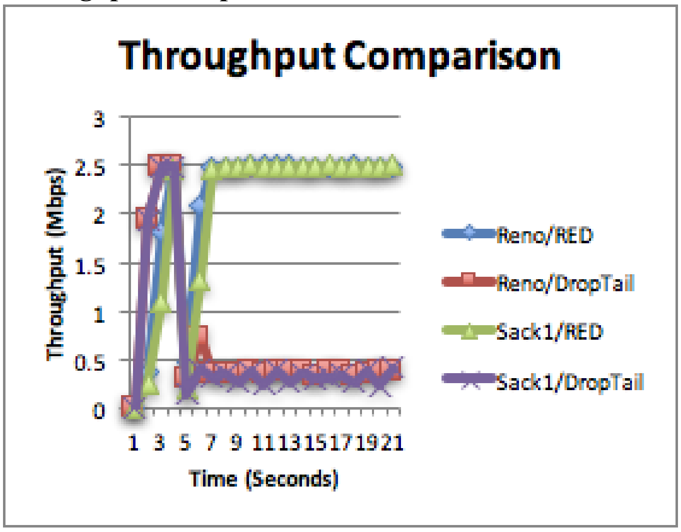

Figure 10, Throughput Comparison for Queueing Algorithm

From Figure 10, we can see that there are two peak throughput for RED queueing algorithms at about 4 and 7 seconds. We can tell that Reno and Sack1 in RED use more bandwidth than the bandwidth used by Reno and Sack1 in DropTail, which we can infer that RED will be given more bandwidth compared to DropTail. This can also be proved by taking average value of throughput for Reno and Sack1 in DropTail and RED over the period when CBR is running.

This happens because RED admits fair shared to TCP streams in its queue and only drop packets based on a probabilistic analysis it does itself. DropTail, however, drops packets when the queue is full and therefore has smaller throughput.

T-Test analysis: Sack1 and Reno use unequal variance test, which results value of t stat <t critical. And since P value is greater than initially set alpha value, we can conclude that null hypothesis cannot be rejected, meaning that there is no statistically significant difference between the values according to the T-Test results.

IV. Conclusion

After all the experiments, we can come to the conclusions that different TCP variants have different advantages and disadvantages. In the first experiment, where we simply measure the performance under congestion, Vegas is the ultimate winner, which has edges over other TCP variants in higher average throughput, lowest average latency and fewest packets drops. In the second experiment, we prove that unfair games exist and when two TCP variant are different, one variant is usually unfair to the other. And in the last experiment, we discover that RED queueing algorithm usually has higher throughput with lower latency.

Overall, Vegas is the better TCP variant and RED is the better queueing algorithms in the three specific experiments we have done here. It will be interesting to have more experiments with more TCP variants and more queueing algorithms to see how they fare in a different game.

V. References

[1] Kevin Fall and Sally Floyd, “Simulation-based Comparisons of Tahoe, Reno, and SACK TCP”