Performance Analysis of TCP Variants

Zhengda Wu
CCIS, Northeastern University
wu.zhen@husky.neu.edu

Xiangshi Sun
CCIS, Northeastern University
sun.xian@husky.neu.edu

1.Introduction

TCP, a well-known protocol in our network, which originated in the initial network implementation in which it complemented the Internet protocol(IP). Since early stage implementation of the TCP has many faults which will waste lots of network resource in computer and cause congestion in our network. Nowadays we have four types of implementation of TCP and we could utilize these ways to solve congestion problems we meet.

In this paper, we will analyze these four different variants which are Reno, NewReno, Tahoe and Vegas. So, we will use N2 to simulate a network environment to test the performances of dour different variants of TCP implementation to show pros and cons of every type of TCP implementation.

2. Methodology

2.1 Network Topology

In this experiment, we simulate a network with six nodes as Figure 1 shows. Each of node has connected with a link of bandwidth of 10MB and default delay 10ms.

In experiment 1, we set up a TCP flow at the source node n1 and sink at node 4. We also set up a CBR flow at the source node n2 and sink at the node n3.

In experiment 2, we set up two TCP flows in the network. We set the first one at source node n1 and sink at n4, the second one at source node n5 and sink at n6. Finally, we set up CBR flow at source node.

In experiment 3, the TCP flow is set at source node n1 and sink at n4, CBR flow is set at source node n5 and sink at n6.

3. TCP Performance Under Congestion

3.1 Throughput

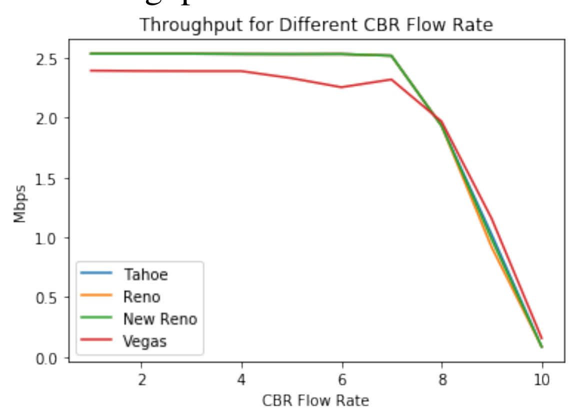


Figure 3.1.1

From the Figure 3.1.1 we could see that when CBR flow is lower than 7Mbps, NewReno's throughput is more than other variants. However, since we increase CBR flow over 7Mbps we could observe that NewReno's throughput decreases sharply while throughput of Vegas decreases slower than other variants. The reason of this phenomenon is that since Vegas uses different strategy to detect congestion with other variants. Vegas use congestion avoidance strategy and others use ACK strategy. In this environment, Vegas usually perform better than other variants. On the other hand, Reno showed lowest average throughput when CBR flow increased over

7Mbps.

3.2 Packet drop rate

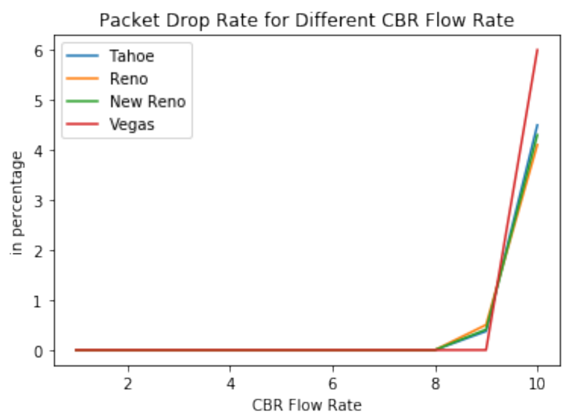


Figure 3.2.1

Figure3.2.1 above shows the average packet drop rate of four TCP variants when against CRB flow from 1m to 10m.

We observed that when CBR flow is lower than 8M, four variants have almost same packet drop rate. From 8M to 9M, all variants except Vegas increased their packet drop rate. From 9M to 10M, packet drop rate for all variants grows rapidly.

3.3 latency of the TCP stream

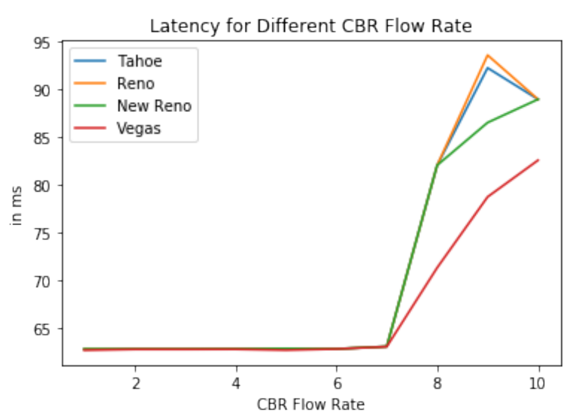


Figure3.3.1

As shown in figure above, we could observe that when CBR flow rate before 7Mbps, all variants almost have same latency between 60ms to 65ms.

However, when CBR flow rate is increased over 7Mbps, latency of all variants increased sharply especially for Tahoe, Reno and New Reno while Latency of Vegas increased comparably slower than other three types of variant. Which means Vegas performance better than other variants in these environments.

3.4 Summary

In sum, for throughput part, when there is little congestion or no congestion, NewReno usually has more throughput and Vegas has most throughput while there is lots of congestions.

For latency part, we could make a conclusion that when CBR flow is lower than 7Mbps, latency for all variants are almost same, but when CBR flow is over 7Mbps, Vegas is better than other variants which has lowest latency.

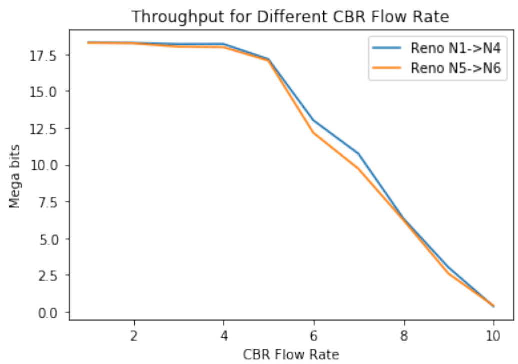
For packet drop rate, Vegas performance better which has fewest drops.

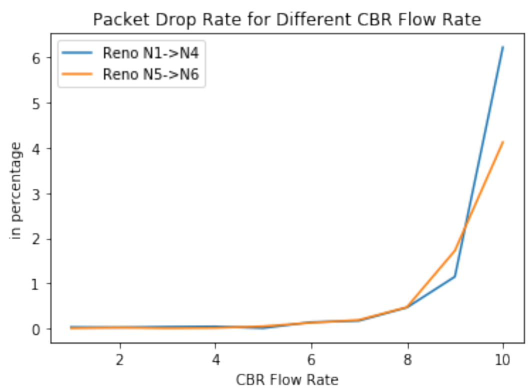
From all data above, there is no absolute “best” variant in network, we need to choose the best one depends on current circumstance such as CRB flow or some other cases.

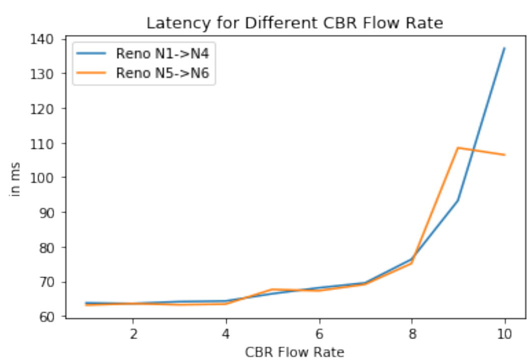
4.Fairness Between TCP Variants

For this experiment, we compared many pairs of variant to test their performance in different circumstances and to test their features.

4.1 Reno vs Reno

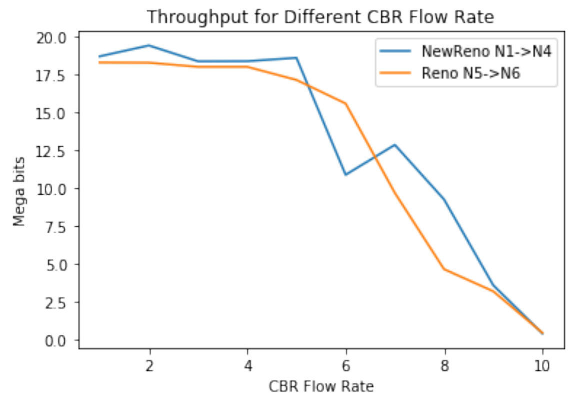


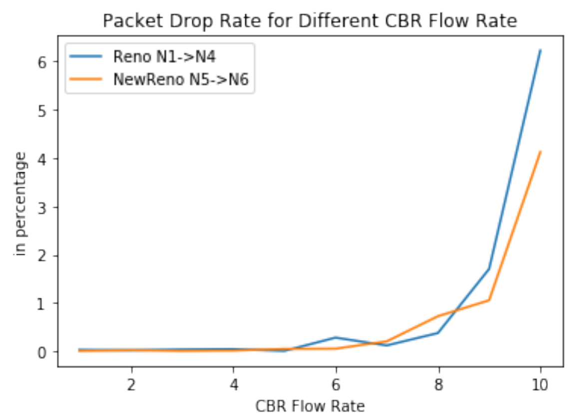


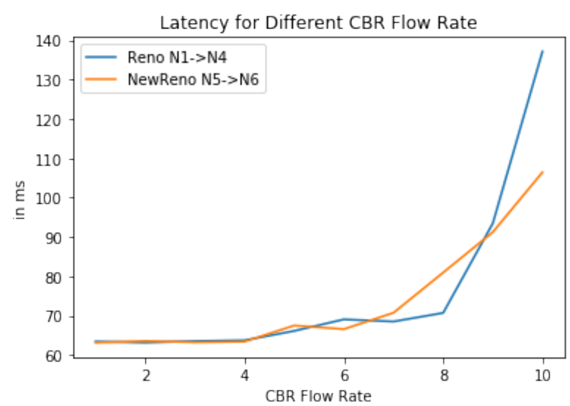


Although there is a little oscillation for throughput, two Reno lines change with similar tracks and close with the increase of the CBR flow, the oscillation is acceptable because bandwidth is constant. And the oscillation means the two TCP agents alternately occupy the more bandwidth, so it is fair for two Reno TCP running in the same network.

4.2 NewReno vs Reno

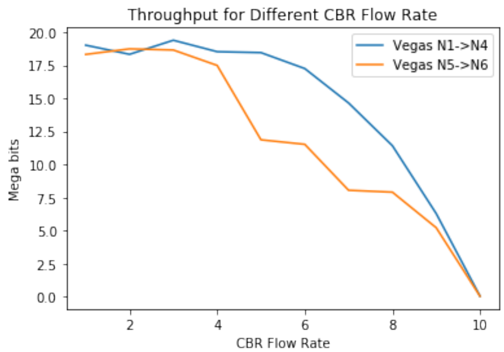


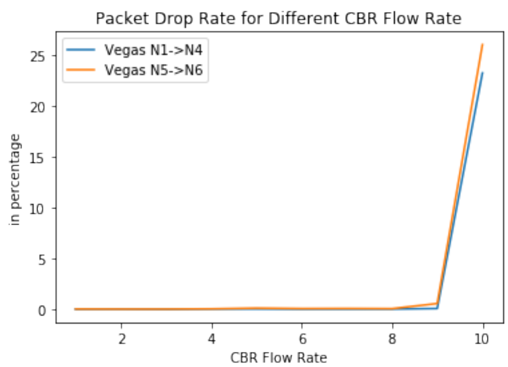


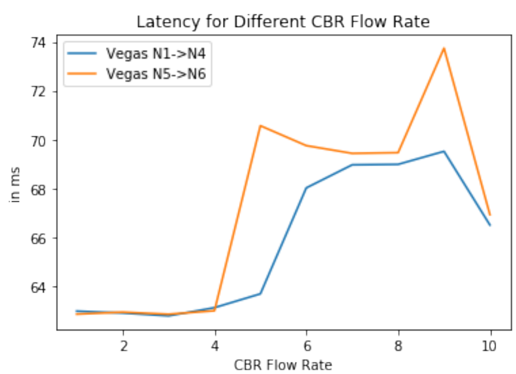


When CBR flow is lower than 5Mbps, Reno and NewReno have almost same throughput, packet drop rate and latency. However, when CBR flow is higher than 5Mbps, NewReno began to suppress Reno, NewReno occupied more bandwidth than Reno. This is due to its improved retransmission during the fast recovery phase as opposed to Reno's waiting time for a triple duplicate ACK.

4.3 Vegas vs Vegas

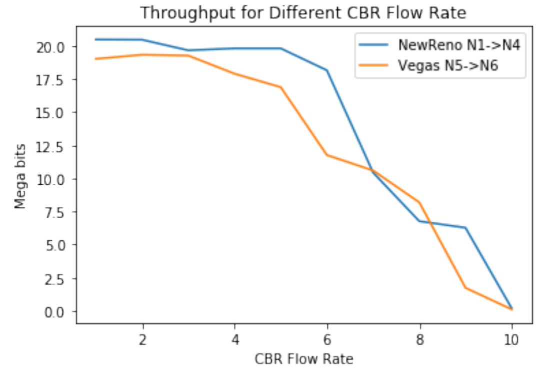


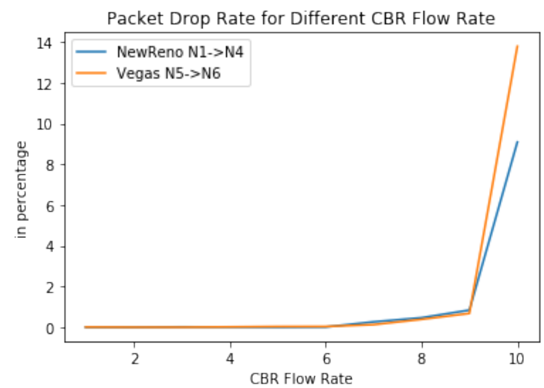


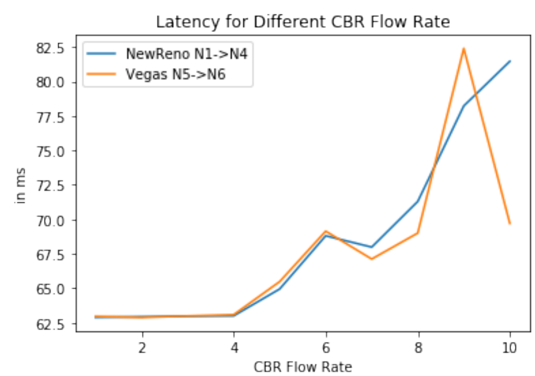


From figures above we could that the relationship between two Vegas is fairness, which is similar to the conclusion between two Renos.

4.4 NewReno vs Vegas







From figures above we could see that NewReno and Vegas are not fair between each other and NewReno performance slightly better than Vegas. This is because Vegas's strategy will detect congestions prior to NewReno, which means some of bandwidth will be given to NewReno. We could draw a conclusion that there is an unfairness among these two TCP variations.

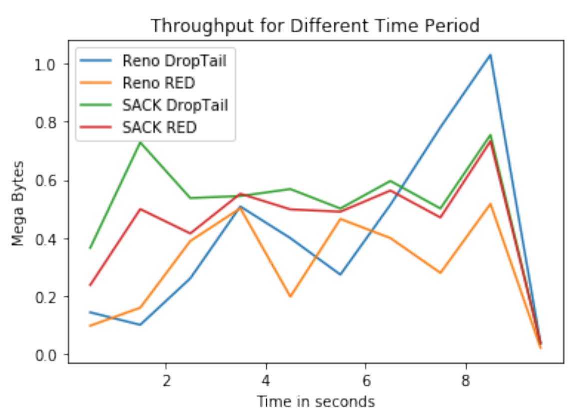
4.5 Summary

From all data above, it is safe to say that when two TCP variants are same, usually they will be fair to each other since they have same strategy to deal with congestion. However, while different variants cannot be completely fair to the co-existing variants. Because different variants have different implementation of dealing with data, so if two variants work together, one variant flow usually will suppress the other one. The visual maps show us that early time Vegas will be suppressed by NewReno, since NewReno became the dominant in the network, Vegas will assume there is a congestion and never come back to be the dominant again.

5.Influence of Queueing

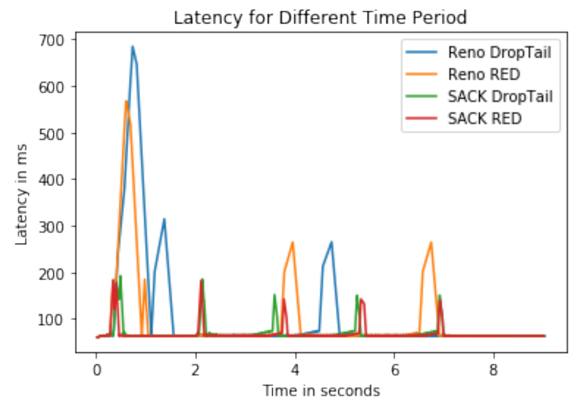
At experiment3 tested Reno and SACK TCP flows with different queue algorithms, we assume that different queue will influence fairness of TCP variants including throughput and latency. In this experiment, we used two types of queue which are DropTail and RED.

5.1 Average throughput



From figure above we could draw a conclusion that SACK variant could get more throughput than Reno variant, Also, compare to RED, DrapTail could always produce slightly more throughput. Besides, Reno with RED queue type has the worst performance in bandwidth, because RED will drop some packets even before the queue is full. Once a packet is dropped, Reno enters fast recover and changes its congestion window size to the half of the current threshold size.

5.2 Average latency



The figure above shows us that SACK variant with RED data structure works well than other combinations, which means that SACK works well with RED. Also, the latency of SACK is obviously lower than Reno. because RED uses different strategy to deal with latency which avoids TCP Global Synchronization compare to DropTail.

5.3 Summary

From data above we could see that queuing discipline doesn't provide fair bandwidth to each flow. For Reno variant, DropTail provides more throughput than RED. For SACK variant, utilizing RED or DropTail gets quite similar amount of throughput.

From the figure above we could draw a conclusion that compare to DropTail, RED could provide slightly lower latency than Droptail.

TCP flows gradually reduced when CBR flows begin to be created. When CBR flows became higher, on the other hand, the TCP flows will be lower according to data above.

According to the data above we could draw a conclusion that RED works well with SACK. Because throughput of SACK with RED is higher than SACK with DropTail and latency is lower than DropTail.

6. Conclusion

From all experiments above, we tested four different TCP variants and obtain their throughput, packet drop rate and latency of the TCP streams based on amount of congestion within the network.

For experiment1, we could make a summary that usually Vegas could performance better than other TCP variants in terms of throughput, packet drop and latency.

For experiment2, we could draw a conclusion that usually same variants could performance fair in the same network. However, it will be unfair for different variants in the same network even one variant maybe suppresses the co-exist one.

For experiment3, we could conclude that RED is a good idea when deal with SACK and it is better than DropTail since RED could avoids TCP Global Synchronization phenomenon.

7. Reference

[1] Wikipedia, *TCP Vegas*, http://en.wikipedia.org/wiki/TCP Vegas

[2] Wikipedia, *TCP, https://en.wikipedia.org/wiki/Transmission_Control_Protocol*

[3] Kevin Fall and Sally Floyd, *Simulation-based Comparisons of Tahoe, Reno, and SACK TCP*.