Performance Analysis of TCP Variants

Ritesh Kurapati, Neelesh Nidadhavolu
Northeastern University

1 Abstract

This paper contains performance of TCP variants like TCP Tahoe, Reno, Vegas and New Reno under different congestion conditions observed in network today. Each TCP variant tries to control and avoid congestion differently. It presents comparison among these variants in terms of throughput, latency, packet drop under various load conditions using different queueing algorithms and tries to identify the best variant in congestive environment.

2 Introduction

Even though there are many differences in implementation of features, TCP and UDP are widely used transport layer protocols for applications in today's world. TCP is connection oriented, provides reliability, flow control and congestion control mechanisms while UDP is not reliable and connection less protocol.

TCP is successful because it establishes three-way handshake for initial connection, does not overwhelm receiver by flow control principle, avoids congestion in network and provides reliable transfer from one end point to other. Because of various network conditions observed today, there are different TCP variants but working principle of TCP does not change.

TCP Tahoe, Vegas, Reno and New Reno are different TCP variants used in this study. TCP Tahoe uses slow start phase, congestion avoidance based on Additive Increase Multiplicative decrease(AIMD) algorithms and starts with congestion window as one MSS segment when it observes time out. Since TCP Tahoe may not perform well during network congestion, new TCP variant Reno came into existence. TCP Reno uses fast Recovery and fast retransmit features where it resends packet based on three duplicate ACK and sets congestion window to ssthresh/2. New Reno uses improved fast recovery feature where it retransmits packet for every duplicate ack. TCP Vegas has delay based window for packet transmission and does not consider window with packet based loss.

Three experiments are presented in this paper. First experiment deals with performance of TCP variants under congestion. TCP Vegas has high average throughput, lowest average latency and fewest drops when compared to other TCP variants in similar conditions of congestion. Second experiment deals with comparison of fairness between two TCP variants. TCP New Reno emerged as winner in most cases whenever it is compared with other TCP variants. Experiment three is about performance of TCP variants under Drop Tail and Random Early Drop (RED) queueing algorithms. Here Random Early Drop (RED) outperforms Drop Tail in providing fair bandwidth for TCP Reno and SACK flows. Details of these experiments would be explained in further sections.

3 Methodology

Experiments were conducted using NS-2 network simulator.NS2 is chosen because it is widely used in academic research, large user base and is frequently updated with packages. It is discrete event scheduler, freely available, flexible and simulates both wired and wireless network. It supports wide range of networking protocols, TCP variants and queueing algorithms. Since it uses TCL as scripting languages, simulations can be easily controlled. NS2 separates data path and control path implementation. Data path objects implemented in C++ makes it fast because it reduces overall event processing time. The above mentioned factors led us to choose NS2.

```
N1                N4
 \               /
  \             /
   N2-------N3
  /             \
 /               \
N5                N6
```

Figure 1 : Network Topology

Figure 1 shows the network toplology which is used in all the experiments.All links of nodes are set to 10Mbps and are full duplex.N1 is TCP source based on FTP application and N4 is TCP sink for flow.N2 is source of CBR over UDP connection to a sink at N3.N5 is also a TCP source based on FTP application and N6 is TCP sink.

In experiment 1 , N1 is TCP source based on FTP application, N4 is TCP sink and N2 is source of CBR over UDP connection that sinks at N3.The CBR is varied from 1Mbps to 10Mbps and perofrmance of TCP variants is comapred under congestion.The following variations were also perfomed.

a. Packet Size variation: Packet sizes of TCP flow and CBR flow are varied.
b. Time variation: Varied TCP and CBR flow start times to observe congestion.

In experiment 2 ,two TCP sources with different variants, one at N1 that sinks at N4 and other at N5 which sinks at N6 are compared.CBR flow from N2 to N3 remains unchanged.Fairness among TCP New Reno , Reno and Vegas are compared.CBR rate is varied from 1Mbps to 10 bps at rate of 1Mbps and start time for both TCP varaints are varied.

In experiment 3, UDP flow is between N5 and N6, TCP Reno/ TCP SACK is between N1 and N4 and queueing algorithm Drop Tail and Random Early Detection(RED) are used to control the treatments of packets in queue.The performance of TCP and CBR is observed over time using Drop Tail and RED algorithm.

For all three experiemnts, NS-2 simulations yield trace files of .tr format in same path of our running configuration.These files are parsed using python scipt and the values are plotted over graph using Excel.T Test analysis is used to check whether differences between values produced are statistically signifcant based on probability. Error bars are also plotted on graphs based on Standard deviation.

4 Result Analysis:

Experiment 1: TCP perfromance under congestion

The main objective of this experiment is to analyze performance of various TCP variants by linearly increasing CBR rate. Each link bandwidth between nodes is set to 10Mbps.For each run, CBR rate is gradually increased at rate of 1 Mbps on all TCP variants.Average Throughput, packet drop rate and end to end latency parameters are measured.

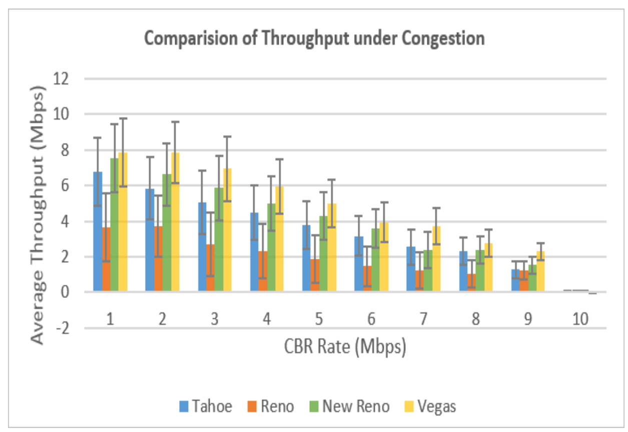

Figure 2 : Comparision of Throughput for TCP Tahoe, Reno,New Reno and Vegas under congestion

In above Fig 2, TCP Vegas has highest average throughput over TCP Tahoe,New Reno and Reno because Vegas acts based on proactive measures like variation in RTT that are much more efficient than reactive ones like packet loss.It detects congestion before packet loss occurs and loss of packet can be detected by major variation in delay. New Reno has almost similar thorughput as Vegas when congestion is less(low CBR) because of its features improved fast retrasmission and fast recovery.Reno has lowest throughput when compared to all others during congestion because it waits long for for retransmission due to triple duplicate ACK even though it has faster recovery mechanisms.So it swings arounds the same value and does not have good throughput even for lower CBR's.TCP Tahoe has high throughput for lower CBR's because it perfroms well when there is minimal congestion and low thorughput in high CBR is due to start of congestion window with 1MSS and slow start phase.

After performing T test with un equal variance and tail value of 2 for TCP Tahoe, Reno with Vegas throughputs, we observered that probability value is less than 0.05 which means values differ statistically while TCP New Reno and Vegas throughputs do not differ signficantly.

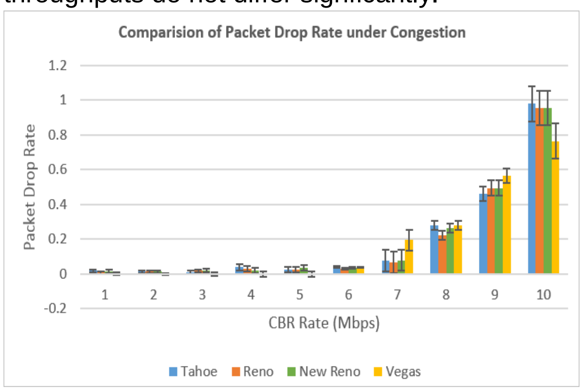

Figure 3 : Packet Drop Rate vs CBR (Mbps)

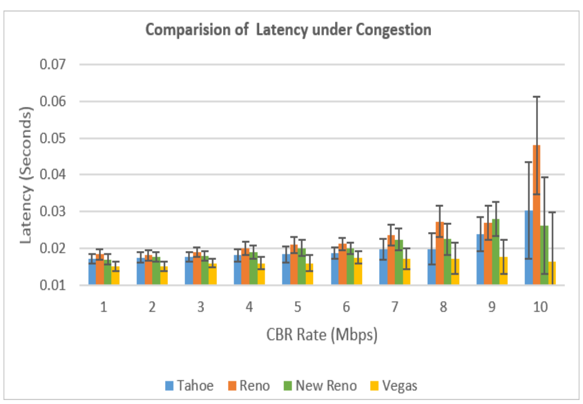

Figure 4 : Latency(secs) vs CBR(Mbps)

As shown in Figure 3, TCP vegas has lowest packet drop rate when compared with other variants and drops are zero for CBR values less than 6 and it increases from 6 Mbps and becomes almost equal with others when congestion is high.Packet drops for other variants except Vegas increase significantly with increase in CBR due to congestion because TCP vegas reacts before the packet loss by observring the variations in RTT due to congestion while others do not take proactive measures based on RTT.Two tailed T tests with unequal varaince on TCP Vegas with TCP Reno/New Reno/Tahoe, we observed that probability value is greater than 0.05 which means values do not differ signficantly.

As shown in figure 4,TCP vegas has lowest latency and is clearly winner over other variants.Vegas has the lowest latency when congestion is high.This is because TCP vegas has delay based window, so it clearly montiors the variations in RTT and reduce the sending rate whenever estimated RTT is greater than base RTT .Other variants have similar latency for all CBR's because they act only based on packet loss and don't have delay based window.Results for T tests with unequal variance on TCP Vegas with TCP Reno, New Reno, Tahoe showed probability is less than 0.05 which means values differ significantly.

For this experiment, TCP Vegas is overall best variant since it has outperformed other TCP varinats in average Throughput , Packet drop rate and latency parameters.

Experiment 2: Fairness Between TCP varaints

The main objective of this experiement is to measure the fairness when TCP variants exist together in same network.The TCP fairness is measured based on Throughput,Latency and Packet drop parameters between each TCP variant pair New Reno/Reno , New Reno / Vegas , Reno / Reno and Vegas/Vegas.

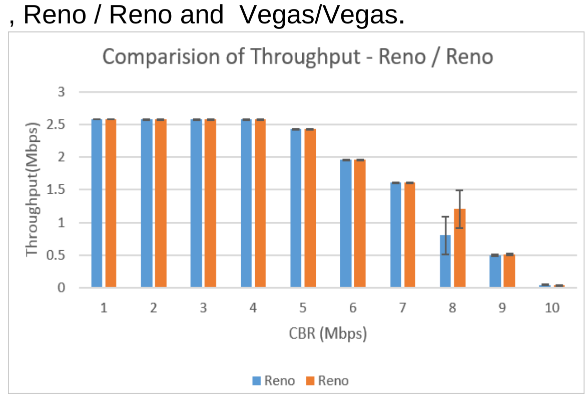

Figure 5 : Reno – Reno Throughput vs CBR

As shown in figure 5, throughput values are mostly same for both TCP flows Reno and Reno when they exist in same network, share the bandwidth equally and are fair to each other. This is because both TCP flows behave based on fast retransmit and fast recovery algorithm under congestion as CBR increases. Packet drop rate and latency has been the same for both flows and there is no significant difference.

The results of T–Test with un equal variance and two- tail value on both TCP flows also confirm the probability is greater than 0.05 which means there is no statistcal difference between the both.

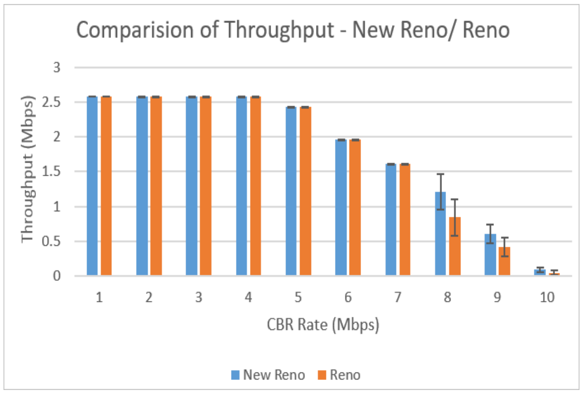

Figure 6 : New Reno – Reno Throughput vs CBR

Figure 6 shows the throughput values are almost same for both New Reno and Reno for less congestion (low CBR's) and there is significant difference for high CBR values because New Reno is an extension of Reno where New Reno uses improved fast retransmit algorithm while fast recovery remains the same for both TCP flows.When congestion is high, there are more retransmissions because of high packet drops.So TCP NewReno retransmits for every duplicate ack while Reno waits longer for retransmission.As a result,there is high throughput difference because window slides faster for TCP New Reno than Reno. TCP varaints NewReno and Reno are fair to each other when there is less congestion and unfair to each other when congestion is high.

T-Test results with unequal varaiance and tail value of two for packet drop, throughput and latency parameters show the probability is greater than 0.05 which means there is no significant statistical difference overall.

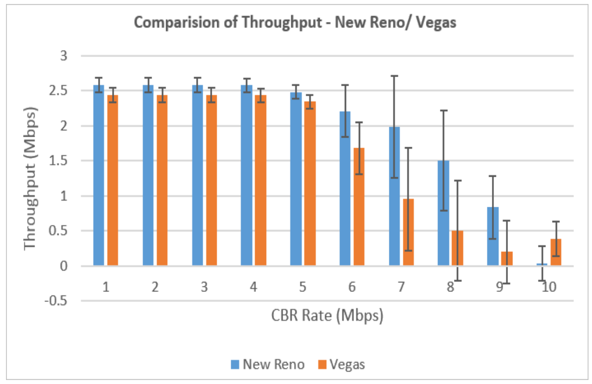

Figure 7: New Reno-Vegas Throughput vs CBR

In Figure 7, we compared the New Reno /Vegas throughput over CBR.We observed NewReno always has higher throughput over Vegas and are unfair to each other.This is because NewReno has improved fast retrasmit algorithm and takes decision on packet loss while Vegas takes decision on delay.So as the New Reno traffic increases in network , Vegas see it as congestion and reduces the sending rate.For CBR value of 10, when congestion is really high Vegas has better throughput than NewReno because Vegas considers RTT variation to determine rate of sending packets while NewReno considers only packet loss.So TCP variants New Reno , Vegas are unfair to each other.

T-Test with groups having unequal varaiance and tail value as two for packet drop, thoughput and latency parameters also confirm that probability is less than 0.05 which means there is significant statistical difference between the two flows.

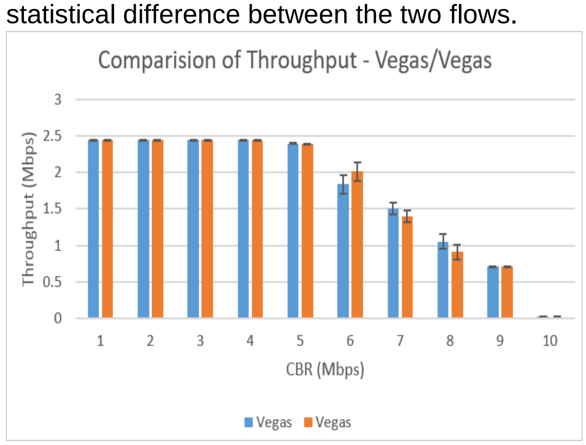

Figure 8 : Vegas – Vegas Throughput vs CBR

From figure 8, TCP variant combination Vegas/Vegas are fair to each other. They have similar throughput even when congestion increases because both TCP flows uses the same variant and have same congestion control algorithm.Both flows increase sending rate when congestion is less because of less delay variation and both reduce their sending rate when the congestion is high.

Two tailed T-test results with unequal variance also confirm that probability is greater than 0.05 which means there is no significant difference in values.

Experiment 3 : Influence of Queueing

The main objective of experiment is to study the behaviour of queueing algorithms DropTail and RED on TCP variants Reno and SACK.This expereiment uses the same topology as shown in Fig. 1.Here, TCP flow is started on N1 node to sink at N4 and once TCP flow is steady, CBR flow is started on N5 that sinks at N6 at constant rate of 7Mbps.Throughput is recorded for TCP and CBR flows under Drop Tail and RED queueing by varying the time.

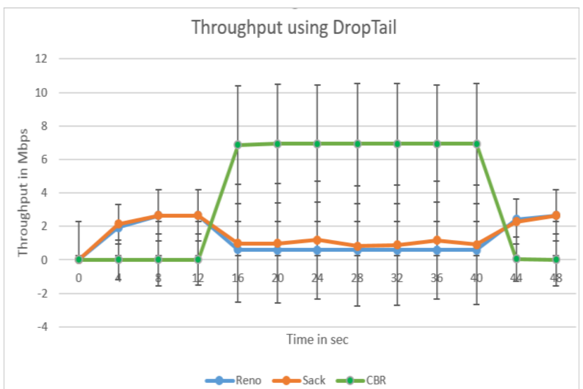

Figure 9: Throughput (Mbps) vs Time (Seconds)

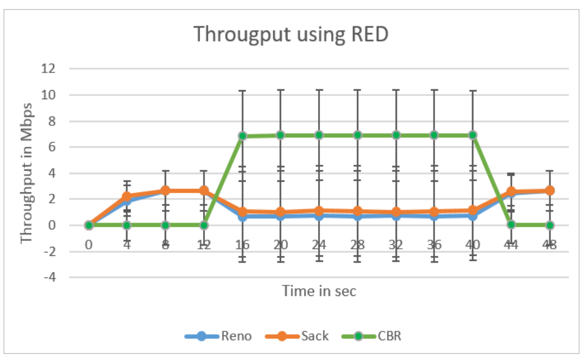

Figure10:Throughput(Mbps) vs Time(Seconds)

From Figure 9 and 10, initially when there is no CBR flow till 12 seconds, both TCP flows RENO and SACK uses the whole bandwidth and has maximum throughput. When CBR flow is started, throughput of both TCP flows RENO /SACK is dropped and fluctuates around the same value. At the end once CBR flow is stopped after 40 seconds, both TCP flows RENO and SACK retain the same maximum throughput which they initially had.

Drop tail does not provide fair bandwidth to each flow because it drops the packets as soon as queue is full and does not distinguish traffic between different flows. From figure 10, we observe RED provides fair bandwidth to each flow because the difference of throughput between SACK and RENO is less. In RED, the probability of dropped packet for flow depends on data it already has in the queue. So each flow has fair amount of bandwidth in queue while using RED.

SACK selectively tells sender about missing and out of order sequence numbers when there is multiple packet drops due to congestion. RED analyzes the congestion by reactively dropping the packets based on probability before the queue is full and provides fair bandwidth to different flows. So RED is good idea with SACK because it provides high throughput under congestion.

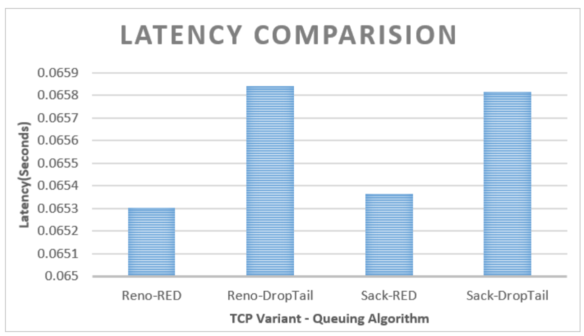

Figure11: Latency vs TCP Variant – Queueing Algorithm

From Figure 11, we observe that end-to-end latency for RED is less when compared to DropTail for TCP variants SACK and Reno. This is because Red defines limit on burst traffic allowed for flow in queue.

5 Conclusion

The main objective of the paper is to study the performance of TCP variants Tahoe, Reno, New Reno, Vegas under congestion and effect of queueing algorithms DropTail and RED on TCP variants. In first experiment, we observed that TCP Vegas performed better under congestion when compared to TCP Reno, New Reno and Tahoe for average throughput, latency and packet drop parameters. In second experiment, we compare two TCP variants like in real world and observed different variants are unfair to each other. TCP New Reno performed well with other variants TCP Reno and Vegas because of fast recovery and improved fast retransmit algorithm. Similar TCP variants are fair to each other because they used the same algorithm. In third experiment, we compared queueing algorithms RED and DropTail on TCP variants Reno, SACK and observed that RED provides fair bandwidth to flows unlike DropTail and RED has less latency compared to DropTail under congestion.

TCP Vegas performed well under congestion when compared to other variants and RED queuing algorithm is better than DropTail. Comparison of more TCP variants under congestion and different queueing algorithms in future would help us in deciding the best TCP variant.

6 References

[1] S.Floyd, T.Henderson "The NewReno Modification to TCP's Fast Recovery Algorithm" RFC 2582, Apr 1999.

[2] O. Ait-Hellal, E.Altman "Analysis of TCP Reno and TCP Vegas".

[3] K.Fall, S.Floyd "Simulation Based Comparison of Tahoe, Reno and SACK TCP" .

[4] S. Floyd. "SACK TCP: The sender's congestion control algorithms for the implementation "sack1" in LBNL's "ns" simulator (viewgraphs).,". Technical report, Mar. 1996. Presentation to the TCP Large Windows Working Group of the IETF, March 7, 1996. URL ftp://ftp.ee.lbl.gov/talks/sacks.ps.