Analysis of TCP Variants and Queuing Methods

Shunlin Liang, Zhenming Shen

College of Computer and Information Science, Northeastern University

I. Introduction

Transmission Control Protocol (TCP) is the commonest transportation protocol in today's network. TCP provides the functions of reliable, ordered, error-checked delivery and so on. During the development of the TCP, scientists have made great efforts to improve the efficiency of TCP. One important aspect they focus on is TCP's congestion control. Scientists have proposed several algorithms for the congestion control, likes slow-start, congestion avoidance, fast retransmit and fast recovery. Congestion avoidance is one of the most important algorithms in TCP and there are different methods to implement it, which are put forward with different TCP variants, for example, TCP Tahoe, TCP Reno, TCP Vegas, TCP New Reno, TCP Cubic, Compound TCP and so on. The main propose of this paper is to analysis the performance of following four TCP variants: TCP Tahoe, TCP Reno, TCP Vegas and TCP New Reno, including the throughput, latency and drop rates of TCP flows in different congestion situations. Besides of analyzing the performance of End-to-end Congestion Control, we conducted an experiment to find out the influence of using different queuing methods to the TCP flow, in specific, Drop Tail and Random Early Detection (RED).

In this paper, three experiments would be conducted to analysis the different performance and fairness between different TCP variants. And different queuing methods would be tested. The methodology of the three experiments would be proposed in Section II. And we will conduct an experiment to analysis different TCP variants' performances in Section III. Another experiment would be performed in Section IV to detect the fairness between different TCP variants. In Section V, we would make the last experiment to analysis how would the different queuing methods influence the TCP's performance. In section VI is a summary of our work.

II. Methodology

This paper is based on three experiments focused on TCP performance of different variant under congestion, fairness between TCP variants and influence of queuing. For the evaluation of performance, fairness and influence of queuing strategy, we choose average throughput, packet loss rate and latency as the value of interest.

These experiments were conducted using NS-2, an event driven network simulator, which consist of event scheduler and network component. We create topology definition and event schedule, and filter useful entries of trace file by adding awk code in tcl script. After various simulations, we use the filtered trace files as the input of a python parse script to calculate the value of interest in our experiments. The papers conclusions are all based on these results.

III. Performance of TCP Variants

Experiment 1 is set up in the topology in Figure 3-1. A constant CBR source is added at N2 and sink at N3. A single TCP stream is added from N1 to N4. And the TCP stream varies from Tahoe, Reno, New Reno and Vegas. In the experiment, the CBR flow rate varies from 1Mbps to 10Mbps, to change the congestion condition in the network. Under each constant CBR flow rate, the throughput, latency and drop rate of each TCP flow would be recorded. By analyzing the experiment result, we could make a comparison of how the four TCP variant react to the presence of congestion.

In Figure 3-2, we could see the following phenomenon easily: with the CBR flow rate increasing, the throughput of TCP flow would decrease. Because the increasing CBR flow rate aggravated the congestion in the network, that's why TCP flow's throughput decreased.

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

Figure 3-1 Simulations on pair of TCP variants

Additionally, we could conclude that the Vegas and New Reno could get about 10% higher average throughput while the CBR flow rate changed. Because the TCP Vegas emphasizes packet delay rather than packet loss, it controls its congestion window by detecting the RTT. So the size of congestion window is more adaptable to the latest network status, that's why TCP Vegas could have higher throughput in the congestion situation. New Reno is modified from Reno and the retransmission is improved during the fast-recovery. With this, New Reno has a better performance when in the situation that lots of packets are dropped. So New Reno could have a better throughput than Tahoe and Reno when in the congestion situation.

By analyzing the results of Figure 3-3, we could find out that the TCP flow of Vegas has smallest latency. The TCP flows of Tahoe, Reno and New Reno have similar latency, which are about 20% higher than the latency of Vegas. Because of the efficient congestion window control algorithm, TCP Vegas could detect and avoid the congestion more easily.

Another obvious phenomenon in the results of Experiment 1 is that, in Figure 3-4, the higher rates of CBR flow, the higher drop rates of TCP flow in all the TCP variants. Tahoe, Reno and New Reno have similar drop rate in this experiment. When the CBR flow rate is less than 8Mbps, they have small drop rates. And while congest increased, their drop rates increase heavily. Because Tahoe, Reno and New Reno are using Additive Increase Multiplicative Decrease algorithm to control the congestion window and they use detecting dropped packets to change their congestion window. On the other hand, TCP Vegas has a better performance on drop rate in Experiment 1, because Vegas use detecting RTT rather than dropped packets.to control its congestion window. It can detect the congestion earlier than Tahoe, Reno and New Reno, and then take action to avoid the congestion. That's why its drop rate is the smallest.

Overall, TCP Vegas has the best performance in the results of Experiment 1: it has the largest throughput and smallest latency and drop rates. But so far we could not conclude that TCP Vegas is the best TCP variant yet, because in Experiment 1, there are just a constant CBR flow and TCP flow in the network, the reality is far more complicated than Experiment 1, for example, there would be more than one TCP flow in the network and problem of fairness should be considered. We will discuss the issue of fairness in next part.

IV. Fairness Between TCP Variants

Experiment 2, using the same topology as Exp.1, like Figure 4-1, conduct experiment by adding two TCP flow, one from N1 to N4, another from N5 to N6. Traffic CBR is set from N2 to N3. Two TCP flow are set to 4 different pairs of TCP agent using the same or different TCP variants.

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

Figure 4-1 Simulations on pair of TCP variants

The experiment adopts average throughput, packet loss rate and latency of TCP flows as evaluation value, plots these values as a function of the bandwidth used by CBR traffic flow. It is conducted on 4 pairs of TCP flows:

(1) Reno/Reno

(2) New Reno/Reno

(3) Vegas/Vegas

(4) New Reno/Vegas

Following discussions about the fairness between TCP variants are based on the result of Experiment 2.

For the first pair of TCP flows, Reno/Reno, three evaluation value of the two TCP flows have similar with the increase of bandwidth used by CBR flow, which shows the fact that TCP Reno flows are fair with each other when they are in homogeneous network.

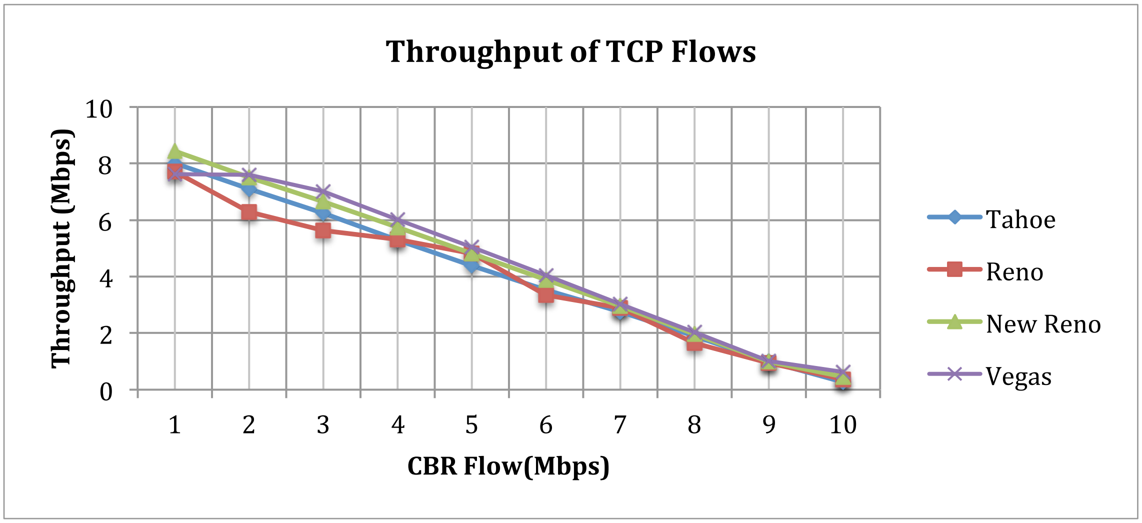

Figure 3-2 Throughputs of TCP Flows

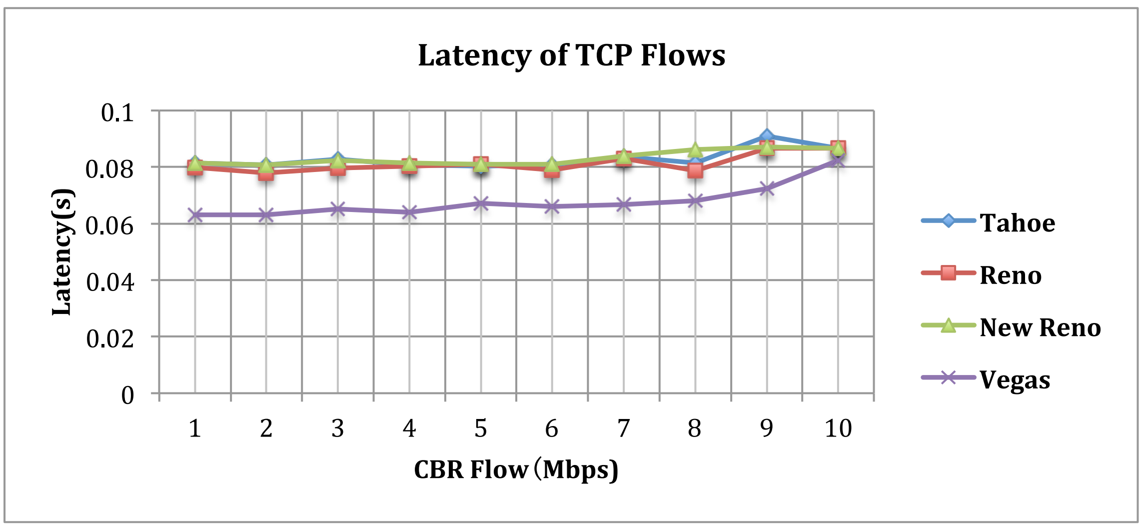

Figure 3-3 Latency of TCP Flows

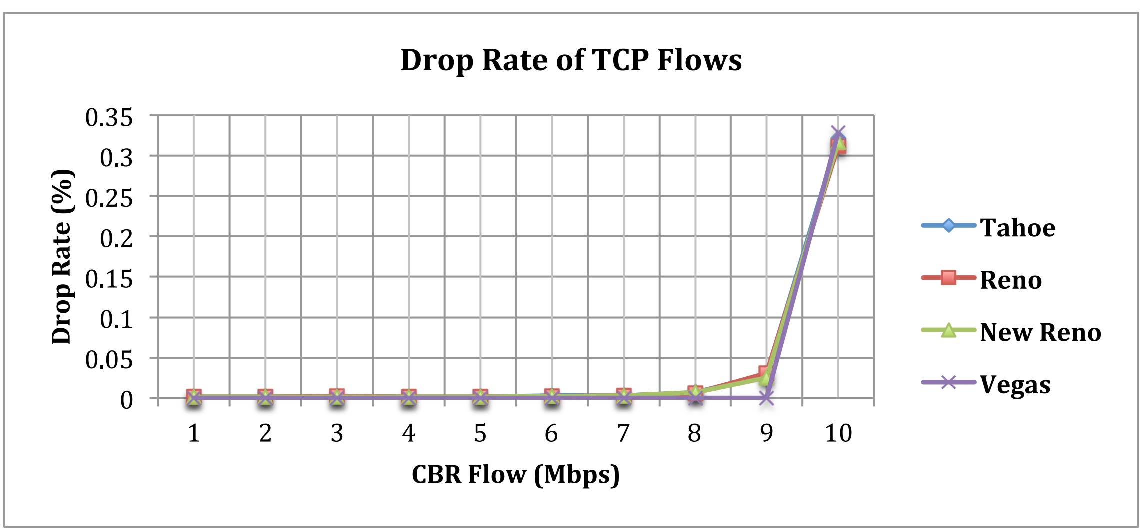

Figure 3-4 Drop Rate of TCP Flows

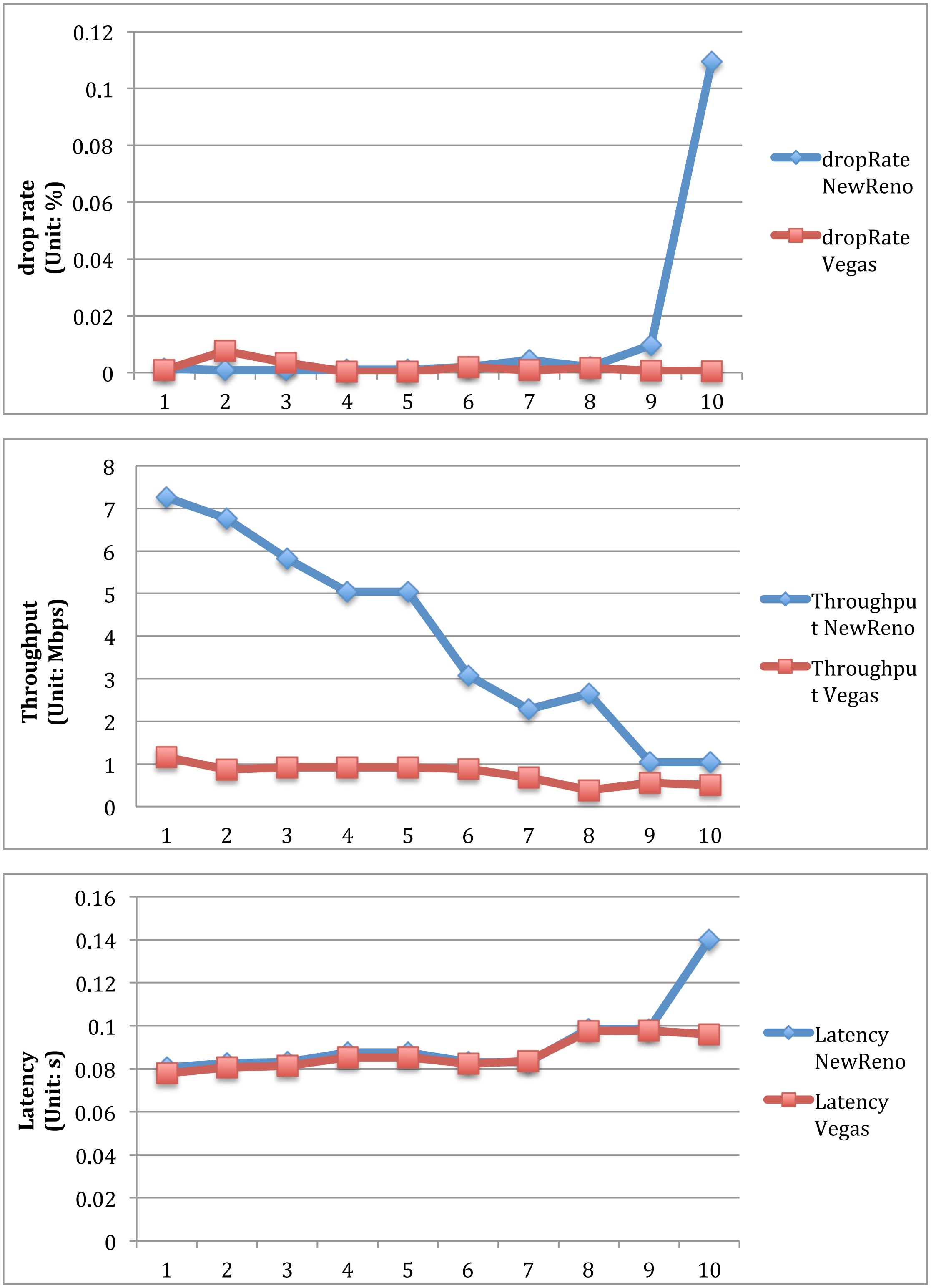

Figure 4-2 Simulations on Reno/Reno

For New Reno/Reno, the conclusion remains tenable, and those three evaluation value of the two TCP flows are similar with each other, not significant difference are shown in the result. So New Reno and Reno flows are fair to each other too.

For Vegas/Vegas, packet loss rate and latency of the two TCP Vegas flows are similar. While the throughput of the two TCP Vegas flows acts differently when the bandwidth used by CBR is very little. But the throughputs of two TCP Vegas flow become similar if the CBR rate is bigger than 3Mbps. We consider this difference as accidental phenomena. It may occur when one of the TCP Vegas flow get slightly earlier than the other one. As TCP Vegas adjust its sending rate according to the latency, the slightly earlier TCP Vegas flow will get higher sending rate than the other at the beginning and due to low traffic, this TCP Vegas flow can maintain its bandwidth advantage for a period of time, which make it achieves better average through -put in an infinite time interval.

For New Reno/Vegas, as shown in Figure 4-2, TCP New Reno has a lower packet loss rate than that of TCP Vegas when the CBR rate is 2, which implies a little unfairness between New Reno and Vegas. Furthermore, New Reno always has a larger throughput than that of Vegas whatever the traffic condition is, which proofs the unfairness between Vegas and New Reno. The phenomenon that New Reno holds more bandwidth than Vegas, results from the different congestion control strategy of these two TCP variants. For New Reno its congestion signal is packet loss, which means that New Reno will continuously enlarge its sending window until it sensed a package loss. On the other hand, Vegas have a more conservative linear increase/decrease mechanism to adjust its sending rate by using latency as its signal. As the latency signal usually comes before than packet loss, so Vegas always lower their sending rate before New Reno, which leads to the result that New Reno occupying more bandwidth than Vegas.

V. Influence of Queuing

Experiment 3 has topology as Figure 5-1, adding TCP flow between N1 and N4, CBR flow from N5 to N6. CBR flow starts after TCP flow stabilized. In this experiment, interest values are the throughput, latency and packet drop rate of TCP and CBR flow, which are plotted over time.

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

Figure 5-1 Simulations on Reno/Reno

It is conducted on four conditions:

(1) Reno + DropTail (2) Reno + RED

(3) SACK + DropTail (4) SACK + RED

In each condition, TCP flow starts at 0s, while CBR flow start at 5s.

Theoretically, DropTail is a queuing strategy that drop packet passively when buffer is full But RED is an initiative packet dropping strategy, which intends for congestion avoidance. It can drop packets by estimating congestion. It used average queue length as the parameter of discard possibility, which determines the possibility of dropping certain packet, and smooths its reaction toward congestion.

But as shown in Figure 5-2, under all four conditions, CBR flow grabs bandwidth from TCP flow and achieves more throughput than that of TCP flow. Which means both queuing strategy cannot provide fair bandwidth to each flow. Even RED queuing does not solve this problem. Because, although it can drop packet initiatively, it treat different flows without difference, so it cannot protect TCP flows from misbehaving flows as the CBR flow in this experiment.

From Figure 5-3, although the end-to-end latency for both TCP flows is shorter when using RED as their queuing strategy than that of DropTail, RED does not provide higher throughput for TCP flows. As shown in Figure 5-2, the combination of SACK and DropTail provides highest throughput than the other combinations. In another word, RED queuing does not show its advantages in this network topology.

VI. Conclusion

After conducting the three experiments and analyzing the results, we could get to the following conclusions:

(1) In Experiment 1, TCP Vegas performed best among the four TCP variants. Under the congestion situation, the TCP flow of Vegas has about 10% higher throughput than Reno and Tahoe, 20% less latency than Tahoe, Reno and New Reno and better performance on the drop rates.

(2) In Experiment 2, the unfairness between New Reno and Vegas is verified. While the performance between Reno/Reno, New Reno/Reno, Vegas/Vegas are proved to be fair, the performance of Vegas is worse than New Reno

when they are in a homogeneous network.

(3) In Experiment 3, we could conclude that the queuing of RED do not perform well in such network situation. In both of the tests of Reno and Sack, the TCP flows with the queuing of Drop Tail have the higher average throughput than that flows with queuing of RED. Because the CBR flow using in the experiment is misbehaving, that is, the packets of CBR flow always be sent in a constant speed, TCP flow do not perform well under the situation using RED queuing.

By analyzing the results stated above, we can come to following conclusion: Although TCP Vegas has a better performance than the other TCP Variants under the congestion situation, the performance of Vegas is unsatisfactory when there are other TCP Variants in the network. That is, in today's network with most of TCP flows are using Reno, Vegas could not have such good performance in experiment 1, even worse than the TCP Reno. If we want to use TCP Vegas to improve our network, we should ask everyone to use TCP Vegas too.

Additionally, although RED could improve our network by the better absorption of packet bursts and avoiding TCP flow synchronization, RED does not have a good protection to deal with the misbehaving flows, such like UDP and CBR. TCP Flows performs worse in RED than Drop Tail with a constant CBR flow. So we should better only use the queuing of RED in the network that most flows are TCP flows. Another solution to this problem is to separate two queues for TCP flows and the other flows in the router, and to use RED for the TCP flows.

The contributions of this paper are as follows. Firstly, we analyzed the performance of different TCP variants. Secondly, we discussed the feasibility of using the variant with the "best" performance in the reality. Thirdly, the performance of different queuing method is analyzed.

Further work includes discussing the performance of TCP variants in a more complicated network situation. And how to improve Vegas and RED to make them more feasible and efficiency in modern network is also interest to us.

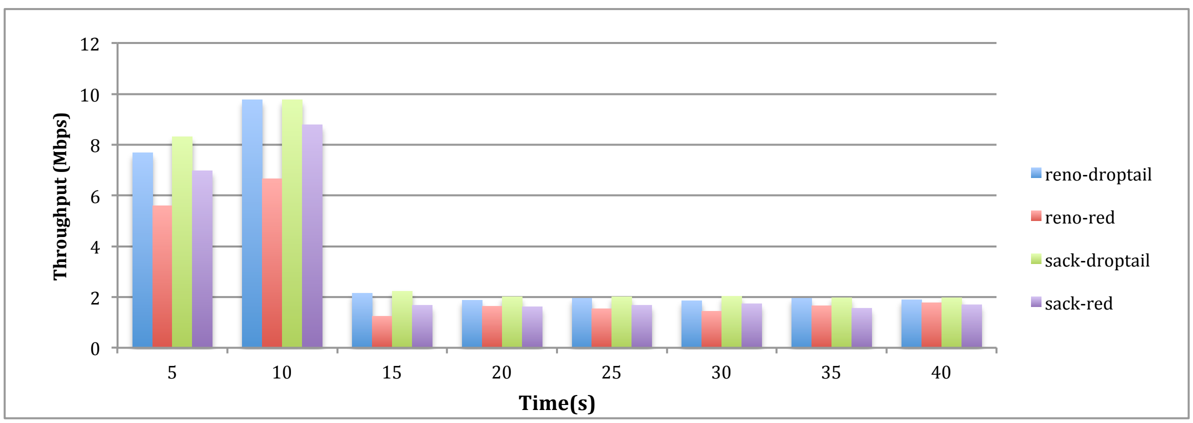

Figure 5-2 Average Throughputs of TCP Flows

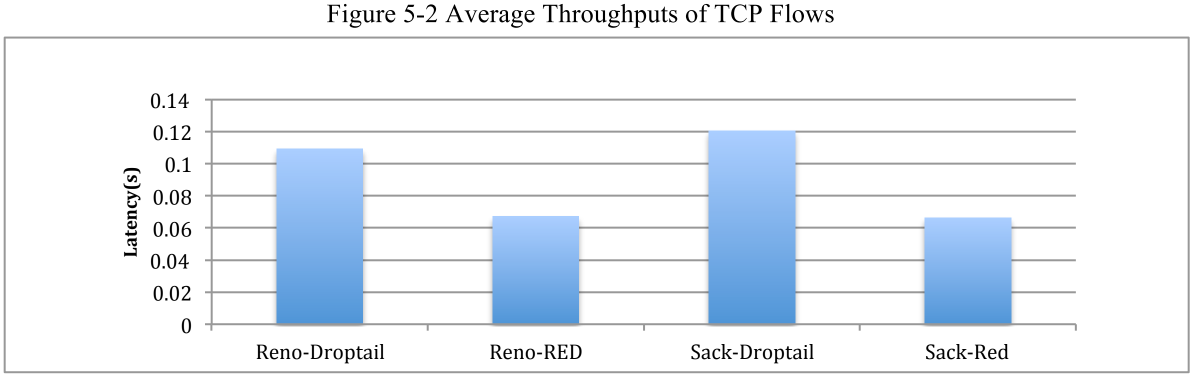

Figure 5-3 Latency of TCP Flows