Performance Analysis of TCP Variants

Project Report of FCN cource

Shengdong Zhu
CCIS
Northeastern University, Boston, MA
360 Huntington Ave., Boston, Massachusetts 02115
zhu.shen@husky.neu.edu

Weijia Ben
CCIS
Northeastern University, Boston, MA
360 Huntington Ave., Boston, Massachusetts 02115

***Abstract*—Transmission Control Protocol(TCP) was the fourth layer in the OSI network model. It is designed to provide reliable, in-order, bi-directional byte stream in the network environment. To improve the performance of TCP, several new types of TCP was developed to solve the problems such as congestion control, flow control. In this paper, it mainly shows that how TCP is performed under congestion and the fairness between different types of TCP. It will also show how queuing influence the TCP performance.**

Keywords—TCP; fairness; congestion; queuing;

I. Introduction

The transmission control protocol(TCP) is one of the most popular and popular protocols of network protocol. It enables two hosts to establish a connection and exchange streams of data. TCP guarantees delivery of data and also guarantees that packets will be delivered in the same order in which they were sent. So it is important to avoid congestion by implementing different algorithms. Understanding how these algorithms vary from each other and how they would preform under different conditions can help us gain a better view of TCP. In this paper, we evaluate the performance of TCP in three part. In the first part, we use simulator to simulate the TCP behavior under the congestion condition and observe the three main parameters: throughput, drop rate and latency. In the second part, we set two TCP flow in a same network environment and to see whether they are fair to share bandwidth. In the third part, we try to figure out how queuing algorithm may influence the performance of TCP.

II. Methodology

A. Summary

In this section, we will describe how our experiment will be conducted and introduce what tools we will use. We will give the key definition and formula that will be used in the experiments.

B. Experiment Conduction

In this paper, we will conduct three experiments.

- The first one is to show the TCP performance under congestion. We build the network topology as designed. Bind a CBR sender in node2 and bind a sink in node3. Bind a TCP sender in node1 and a sink in n4. Change the bandwidth of CBR flow, which indicates the network congestion and see how it will influence the performance of the TCP flow. We will focus on the throughput, packet drop rate and latency to evaluate the performance of TCP.
- The second experiment is to show the influence between the same and different TCP variants. We set a CBR flow from node2 to node3, and set a TCP from node1 to node 4 and another from node5 to node6. Change the CBR bandwidth and use different pairs of TCP flows to see the performance between them. We also use throughput, packet drop rate and latency as the parameters.
- The third experiment is to show the influence of different queue types that may impact the TCP performance. We start TCP flow 3 second before CBR flow start to wait it to be steady. Then use the different queue and TCP variants to run the simulation. Calculate the throughput and latency of different combinations and generate the graphs to see the results.

C. Tool

The experiments are composed of four parts. First part is to simulate the TCP under congestion network environment and record the result. Then use tools to collect the experiment data. After data collection and calculation, use graphs to show the result. At last, analyze the results and give the conclusion. For each part, we use different tools to implement it.

Network Simulator 2 (NS-2) is a discrete event simulator targeted at networking research. Ns provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks. [1] We use NS-2 to simulate the TCP because it is the most popular network simulator nowadays.

We use Python to collect the data generated by NS-2, which is .tr file and calculate the performance parameter with the formulas shown in Part D. Python is an efficient coding programming language and is great to handle the data process. Most importantly, we are familiar with python and can focus more on the network problems solving.

At last we use Excel to generate graphs. Excel is the software developed by Microsoft and is excellent to process data. We use it to generate graphs because the quality of graphs generated by

Excel is great and there is a huge number of template in Excel that we can use for our graphs.

Fig. 1. General Network Topology

D. Network Topology

We conduct our experiments in the network topology shown in figure1. In each single experiments, we will implement the TCP source and CBR source to the node and see how the TCP performs.

E. Defination and Formula

- Throughput: Throughput is a measure of how much actual data can be sent per unit of time across a network, channel or interface. [2] The formula of calculate throughput of TCP:

$$\text{Throughput} = \frac{Packets\ received\ number}{Transfer\ Time} \quad (1)$$

- Packet drop rate: Packet drop rate is the radio that the packets drop compares to the total packets. The formula of packet drop rate:

$$\text{Drop rate} = \frac{Dropped\ Packets}{Total\ Packets} \quad (2)$$

- Latency: Latency refers to the timing of data transfers on a communications channel or network. [3] The formula of latency:

$$\text{Latency} = \frac{Sum(Packet\ Received\ Time - Packet\ Send\ Time)}{Total\ Number\ of\ Packets} \quad (3)$$

III. TCP PERFORMANCE UNDER CONGESTION

A. Summary

In this section, we conducted the first experiments that show how TCP performed under congestion network. We set set six nodes and added a CBR source at N2 and a sink at N3. We set a TCP stream from N1 to a sink at N4. The bandwidth is set to be 1 Mbps at the beginning. During this experiment, the varieties are CBR rate and TCP variants, we calculate the throughput, drop rate and latency varies, which reflects how the TCP variants performs differently under congestion.

B. Experiment Result and Analysis

The figure2 shows the throughput performance of each TCP variants. We can see that all the TCP variants work steadily at first. Tahoe, Reno and New Reno use the same slow start algorithm so they have the same throughput before the congestion occurs. Vegas is delayed-base congestion control, the throughput of it smaller than other. As the CBR bandwidth increase up to 7Mbps, congestion occurs and the throughput of TCP variants performs differently with each other. Throughput of all four TCP variants decreases and we can see that Tahoe decreases more than other three TCP. So we can make the conclusion that when the network environment is good and there is few congestion, the Vegas performs worse. But when there occurs congestion, the advantage of Vegas is it can gain a higher throughput. Among the other three TCP protocol, Tahoe works worst. The performance of Reno and New Reno is quite close. The throughput of New Reno is a little higher than Reno. Overall, New Reno and Vegas may have the highest throughput and which is better depends on the network environment.

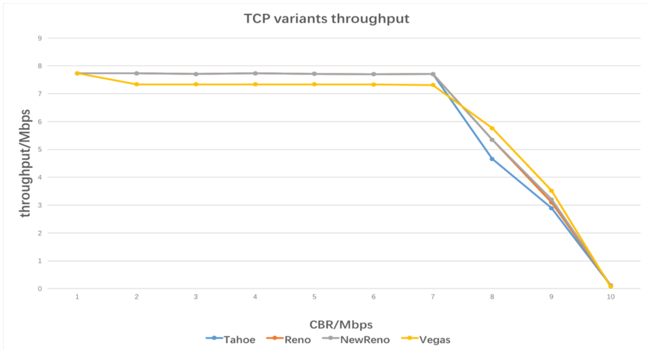

Fig. 2. Throughput of various TCP variants

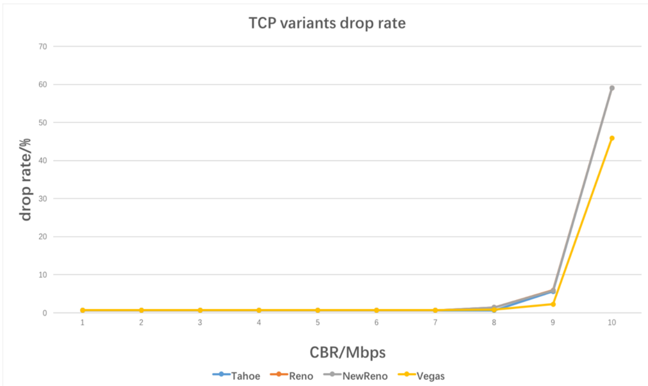

Fig. 3. Drop rate of various TCP variants

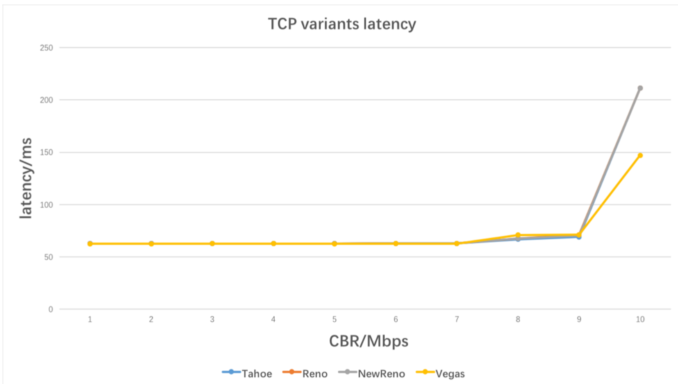

Fig. 4. Lantency of various TCP variants

Figure 3 shows the drop rate of the TCP variants. We can see that after CBR increases to 7Mbps, the packet drops rate increases significantly, which means there is congestion in the network. The drop rate of New Reno is much high than other. That is because New Reno is a aggressive protocol. It will retransmission once receive a duplicate ACK packet. So the drop rate will be higher than other. Vegas has the lowest drop rate.

Figure 4 shows the latency of various TCP variants and we can see that Vegas has the lowest latency.

C. Conclusion

From the experiment conducted above, we can see that Vegas performs great when there is a congestion environment. However, when the network condition is good, the throughput of Vegas is quite low and it can use the bandwidth efficiently. We believe that there is no 'best' variant and the performance of variant depends on the circumstances. If the bandwidth is large enough, we can implement Reno or New Reno to use the bandwidth efficiently. While if the bandwidth is small, we suggest to use Vegas.

IV. Fairness between TCP variants

A. Summary

In this section, we conducted the second experiment. we need to compare two different TCP flow and to see whether two different TCP flows share the bandwidth fairly and discuss what may cause this unfairness. Most part of the experiment is the same as experiment 1. We set a CBR flow from node2 to node3, and set a TCP from node1 to node 4 and another from node5 to node6. Change the CBR bandwidth from 1 Mbps to 10 Mbps and use different pairs of TCP flow: Reno with Reno, New Reno with Reno, Vegas with Vegas and New Reno with Vegas). We calculate and analyze the throughput, drop rate and latency of the TCP flow pair as the parameter of fairness between them. The result and analysis is shown in next section.

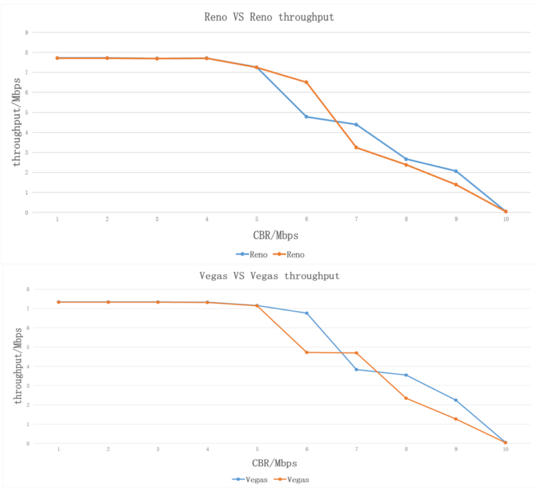

Fig. 5. Throughput comparasion of same TCP Variants

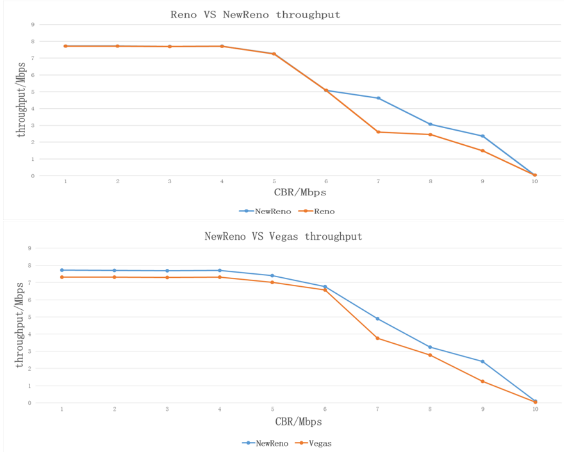

Fig. 6. Throughput comparasion of different TCP variants.

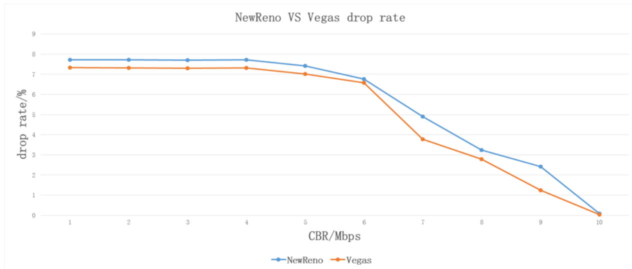

Fig. 7. Drop rate comparison of New Reno with Vegas

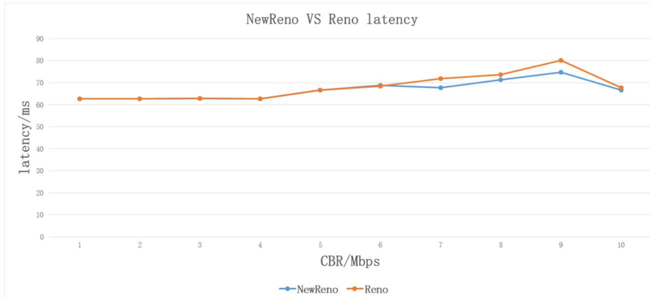

Fig. 8. Latency comparason of New Reno with Reno

B. Experiment Result and analysis

Figure5 shows the throughput of two same TCP variants and figure6 shows the throughput of two different TCP variants. According to the results and figures of the experiments, we can empirically make the conclusion that two same TCP variants shares better fairness than two different TCP variants.

As the CBR flow grows the throughput of both TCP variants decrease because of the packet drops. As the packets loss happen, TCP start function of retransmission. When two TCP flow is the same, they have the same functionalities. So they can share the network bandwidth equally. For example, the two Reno will both start slow start and fast recovery, they will both try retransmission after receiving three duplicate ACK packets. We can see the result of drop rate and latency comparison are also fair between two same TCP variants.

However, when it occurs to two different TCP variants, situation may change. As shown in figure 6, which is comparison of the throughput of Reno with New Reno and New Reno with Vegas, we can see that the throughput of new Reno is always larger than than that of Reno. It seems an unfairness between them. In the comparison of New Reno and Vegas, the throughput of New Reno is larger than Vegas through the whole transmission. So we can empirically make the conclusion that the combinations of New Reno with Vegas and New Reno with Reno are unfair. After learning how these TCP variants work, we can tell why the unfairness happens.

For New Reno and Reno, when packet loss occurs, they both use slow start and fast recovery. However, New Reno will start retransmission when receiving a duplicate ACK packets but Reno starts retrains-mission after receiving three duplicate ACK packets. So New Reno is a more aggressive protocol. It may cost more bandwidth waste because only a duplicate ACK packets may caused by not receiving the ACK packets in order. It may not really lose.

For New Reno and Vegas, New Reno earns a better throughput than Vegas. It is because Vegas implements a delay-based congestion avoidance. It will detect the congestion by RTT. When the network condition is good, New Reno will slow start, which is actually not slow at all. It will gain a huge throughput quickly until the packet loss occur. So the combination of New Reno and Vegas is unfair as well.

Figure 7 and 8 shows the some of the drop rate and latency of the comparison of the combination of TCP variants. We can see the unfairness between the different TCP variants and fairness between the same variant.

C. Conclusion

In this section, we try to set two TCP flow in one network environment and to see the performance of TCP flow influence each other. From the result of the experiments we can make the conclusion that two same TCP variants can share a good fairness when working in the same network. It is because they have the same function to handle the retransmission, packet drop, packets sending rate, etc. However, different TCP variants may not have the fairness. Some of TCP variant may be designed as more aggressive so it may gain more throughput and performance. However, this kind of aggressiveness may cause the congestion in the network more severe.

V. Influence of Queuing

A. Summary

In this section, we conducted the third experiment to show how the two most popular queuing types, DropTail and Random Early Drop(RED) may influence the TCP performance. We use the same topology as experiment 1. Set TCP flow from node1 to node4 and CBR from node 5 to node 6. Start TCP flow 3 second before CBR. Use different queues (DropTail and RED) and use Reno/SACK. We shut down TCP flow at 30 second and shutdown CBR flow at 35 second.

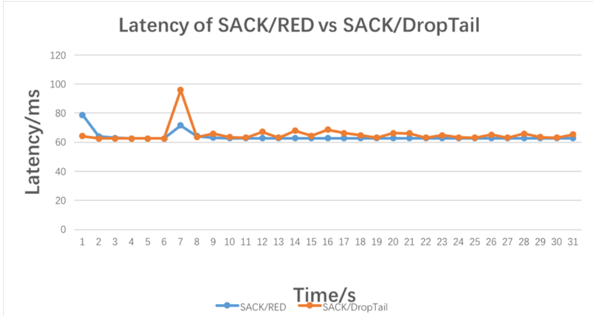

Fig. 9. Latency comparison of diffierent combination of variants and queue

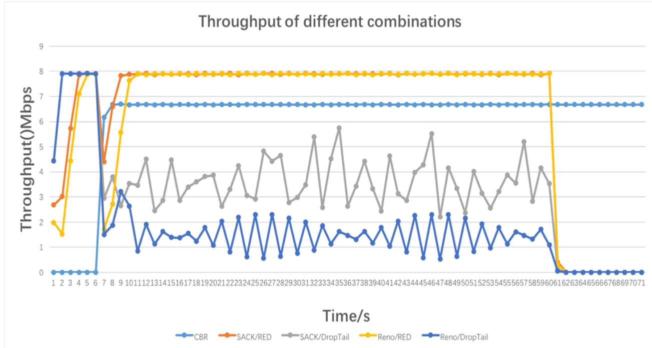

Fig. 10. Throughput comparison of diffierent combination of variants and queue

B. Experiment Result and analysis

From the result shown in the figure 10 above, we can figure out that each queuing discipline provides unfair bandwidth to each flow. The combination of Reno/RED and SACK/RED gains much more bandwidth than SACK/DropTail and Reno/DropTail.

As shown in the figure 10, when CBR flow is created, all the throughput of TCP flows decreased significantly. That is because CBR flow deliveries data in a constant rate and it does not consider congestion. So it will take the bandwidth of the TCP flows. Besides, we can observe that flows that use RED queue recover back to its normal throughput but the DropTail does not.

The latency of flows that use RED is more stable than those use DropTail and the average latency is a little shorter. So we may make the conclusion that RED actually performs better than DropTail.

Figure 11 shows the details of the combination of SACK/RED. The latency of RED/SACK is the shortest among all the combinations while the throughput of SACK/RED is the largest one. So according to the experiment result we make, RED is a good idea while dealing with SACK.

C. Conclusion

In this section, we make the experiment of the combination of TCP variants and queuing types. It shows that the different combination actually differs a lot. Reno/DropTail performs worst among all the combinations, which has the lowest throughput and highest latency. Furthermore, the latency of it is really not stable. Instead, the combination of SACK/RED perform best. Averagely, RED performs better than DropTail.

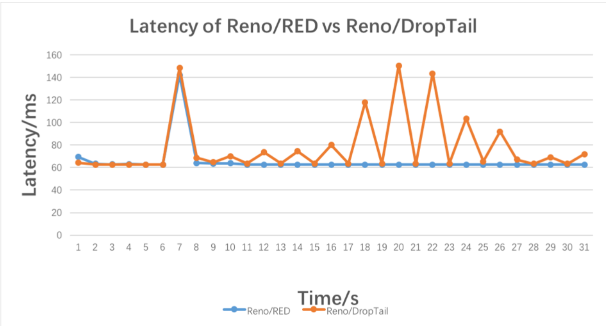

Fig. 11. Detail of latency of Reno/Red and RropTail

VI. Conclusion

In this paper, we make three experiments to analyze the performance of TCP variants.

To evaluate the performance of TCP variants, there are three key parameters: throughput, latency and packet drop rate. We do the experiment one to show the performances of various TCP variants under congestion. We find out that there is actually no 'best' TCP variant. The performance of TCP variant depends on the network environment. When the network condition is good, Reno and New Reno should be implemented. When the bandwidth is small and the congestion may occur frequently, Vegas is a smarter choice.

Then we compare the different TCP variants in the same network environment to see whether it is fair between TCP variants. It shows that between the same TCP variant it is fair between each other. While between different TCP variants, some of TCP variants are more aggressive and may gain more throughput. So when implement the TCP in the real world scenario, we should implement the TCP variants carefully and try to avoid use different TCP variants in the same network.

In the last experiment, we try several combination of TCP variants and queuing types. It shows that averagely, RED has a better performance than DropTail and the combination of SACK/RED is a good choice when implement the TCP protocol.

We make study on Tahoe, Reno, New Reno, Vegas and SACK. While nowadays, there is several new TCP variants that aim to solve the fairness between different variants and to gain more throughput to use the bandwidth of the network more efficiently. In our next step study, we may take more other protocol into consideration.

References

[1] http://nsnam.sourceforge.net/wiki/index.php/User_Information#The_Network_Simulator_-_ns-2

[2] http://www.tcpipguide.com/free/t_PerformanceMeasurementsSpeedBandwidthThroughputand-2.htm