

# Performance Comparison between TCP Sack and TCP Vegas using NS-2 Simulator

Heena Dave

Department of E&C Engineering  
Technocrats Institute Of Technology  
Bhopal,MP.

Vikas Gupta

Department of E&C Engineering  
Technocrats Institute Of Technology  
Bhopal,MP.

Parul Dihulia

Department of E&C Engineering  
Technocrats Institute Of Technology  
Bhopal,MP.

## ABSTRACT

In this paper, two TCP variants are compared in different scenario. Transmission Control Protocol (TCP) is a reliable, end-to-end transport protocol which is most widely used for data services and is very efficient for wired networks. It also performs well in wireless networks. It is the backbone protocol of most of the internet based applications. Our analysis is on performance of TCP variants. In this paper we carry out performance study of TCP Sack and TCP Vegas to be able to classify which variant of TCP performs better in various possible scenarios. This paper describes an NS-2 based simulation analysis. By varying different parameters and congestion control mechanism we will check its effect on Throughput.

**Keywords:** TCP variants, Error rate, Delay, Gateways, NS-2, Throughput.

## 1. INTRODUCTION

Now-a-days the world is becoming smaller and various ways of communication are being used to avail the facilities for people. Internet is one of the widely used techniques to serve different purposes like data transfer, for entertainment, for education purpose, for paying bills online, for shopping, to be aware of the new researches and innovations, etc. The Internet is expanding rapidly, major contribution in this expansion is of the global acceptance of the TCP/IP protocol [1] stack and use of wireless links, particularly in case of remote areas. TCP has three control mechanisms: Flow control, Error control, Congestion control. Flow control defines the amount of data source can send before receiving an acknowledgment from the destination. Flow control mechanisms accomplish by sliding window protocol. For error control TCP uses three simple tools: checksum, acknowledgment, and time-out.

Transmission capacity is governed by congestion window; CWND is reduced to 1 from its current value, which also reduces the flow of packets. Thus it controls congestion in network. Here we will discuss about congestion control [3] mechanisms. There are four algorithms used in TCP: Slow start, Congestion Avoidance, Fast retransmit and Fast Recovery. Slow Start, a requirement for TCP software implementations is a mechanism used by the sender to control

the transmission rate. Thus the rate of acknowledgments (ACKs) returned by the receiver determine the rate at which the sender can transmit data. During the initial data transfer phase of a TCP connection the Slow Start algorithm is used. However, there may be a point during Slow Start that the network is forced to drop one or more packets due to overload or congestion. If this happens, Congestion avoidance [10] is used to slow the transmission rate. When

three or more duplicate ACKs are received, the sender does not even wait for a Retransmission timer to expire before retransmitting the segment (as indicated by the position of the duplicate ACK in the byte stream). This process is called the Fast Retransmit algorithm. After retransmitting the segment, Sender knows that still data is flowing to the receiver. The reason is because duplicate ACKs can only be generated when a segment is received. So instead of reducing the flow of data abruptly by going all the way into Slow Start, the sender only enters Congestion Avoidance mode. The fast recovery algorithm then governs the transmission of new data until non-duplicate ACKs arrive.

## 2. TCP Variants

### 2.1 TCP Sack

TCP with 'Selective Acknowledgments'[2] is an extension of TCP Reno and it works around the problems face by TCP RENO and TCP New-Reno, namely detection of multiple lost packets, and re-transmission of more than one lost packet per RTT.

SACK retains the slow-start and fast retransmits parts of RENO. It also has the coarse grained timeout of Tahoe to fall back on, in case a packet loss is not detected by the modified algorithm. SACK TCP [5] requires that segments not be acknowledged cumulatively but should be acknowledged selectively. Thus each ACK has a block which describes which segments are being acknowledged. Thus the sender has a picture of which segments have been acknowledged and which are still outstanding. Whenever the sender enters fast recovery, it initializes a variable pipe which is an estimate of how much data is outstanding in the network, and it also set CWND to half the current size. Every time it receives an ACK it reduces the pipe by 1 and every time it retransmits a segment it increments it by 1. Whenever the pipe goes smaller than the CWND it checks which segments are un received and send them. If there are no such segments outstanding then it sends a new packet. Thus more than one lost segment can be sent in one RTT.

### Disadvantage of TCP Sack

The biggest problem [4] with SACK is that currently selective acknowledgments are not provided by the receiver to implement SACK we will need to implement selective acknowledgment which is not a very easy task.

### 2.2 TCP Vegas

TCP Vegas [12] is a TCP implementation which is a modification of RENO. It builds on the fact that proactive measure to encounter congestion is much more efficient than reactive ones. It tried to get around the problem of coarse

grain timeouts by suggesting algorithm which checks for timeouts at a very efficient schedule. Also it overcomes the problem of requiring enough duplicate acknowledgments to detect a packet loss, and it also suggests a modified slow start algorithm which prevents it from congesting the network. The three major changes induced by Vegas are:

**New Re-Transmission Mechanism:** Vegas extend on the re- transmission mechanism of RENO. It keeps track of when each segment was sent and it also calculates an estimate of the RTT by keeping track of how long it takes for the acknowledgment to get back.

**Congestion avoidance:** TCP Vegas is different from all the other implementation in its behavior during congestion avoidance. It does not use the loss of segment to signal that there is congestion. It determines congestion by a decrease in sending rate as compared to the expected rate, as result of large queues building up in the routers. It uses a variation of Wang and crow croft's Tri-S scheme.

**Modified Slow-start:** TCP Vegas differs from the other network asymmetry, also the behavior of the underlying routing algorithms during its slow-start phase. The reason for this modification is that when a connection first starts it has no idea of the available bandwidth and it is possible that during exponential increase it over shoots the bandwidth by a big amount and thus induces congestion. To this end Vegas increases exponentially only every other RTT, between that it calculates the actual sending through put to the expected and when the difference goes above a certain threshold it exits slow start and enters the congestion avoidance phase.

### 3. PERFORMANCE ANALYSIS

Here the performance evaluation of TCP Sack and TCP Vegas using NS-2 simulator [3] is shown. One can simulated the performance of these variants for different parameter. Here by one can consider the Error Rate, Delay, Packet Size, Window Size and different Gateways. The Topology is created by NS-2 simulator as shown in figure 1. There are three nodes Sender (node-0), Router (node-1) and Receiver (node-2) which represents the flow of data transmission.

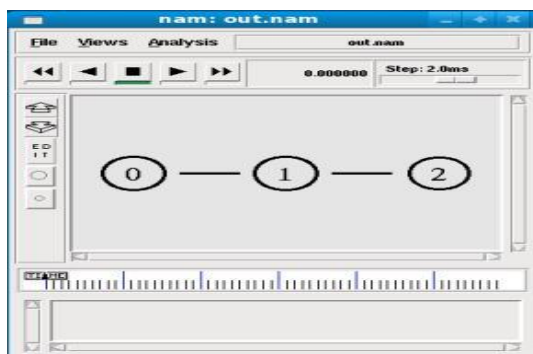


Fig 1: Simulation topology

Numbers of Nodes	3
Simulation Time	16 sec
Data Rate	1Mbps
Delay	1ms,10ms
Queue Limit	100
Queue Gateway Types	Drop Tail, RED,DRR,SFQ,FQ

Table 1: Simulation Parameters

### 3.1 Analysis of Graph of Throughput Vs. Error rate

In this case analysis is done on both TCP variants by changing Error Rate.

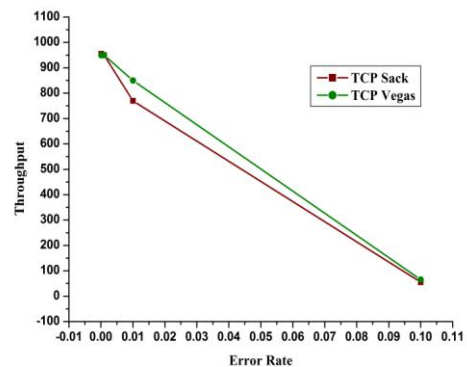


Figure 2: Throughput Vs. Error rate

It is evident from Figure 2 that the performance of TCP Vegas is better than TCP Sack at higher loss rate. Also one can see that the performance of TCP Vegas is similar to TCP Sack at 0.001 error rate. From figure 2 it can be observed that for low error rates (0.0001), TCP Vegas is slightly less efficient than others, but as the error rate increases (0.1) TCP Vegas emerges as the better option [9].

### 3.2 Analysis of Graph of Throughput Vs. Delay

Here Delay is transmitting time between two packets. In this simulation the range of Delay is taken from 1ms to 25ms, after 25ms. Throughput is not changing due to small scale projection in our experiment. From figure 3, one can see that when Delay increases, throughput is decreases. Due to large delay speed will be affected.

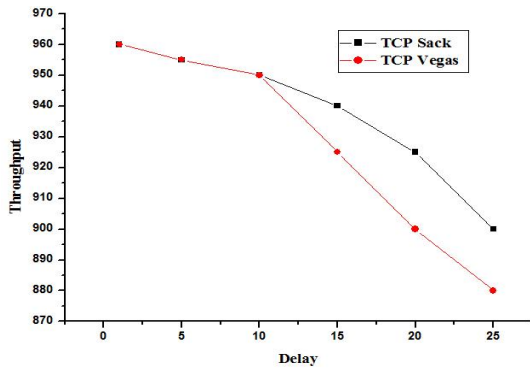


Figure 3: Throughput Vs. Delay

It will reduce the speed of transmission so ultimately throughput will also reduce. At 15ms throughput of TCP Vegas is somewhat less than TCP Sack but by taking graph from NAM editor from NS-2 simulator one can find that response (smooth graph) of Vegas is very good compared to Sack. So TCP Vegas emerges as the better option.

### 3.4 Analysis of Graph of Throughput Vs. Packet Size

The effect of Packet size on Throughput is as shown in Figure 4. Here packet size is kept from 50 to 1500. By increasing size the Throughput is also increase. At lower

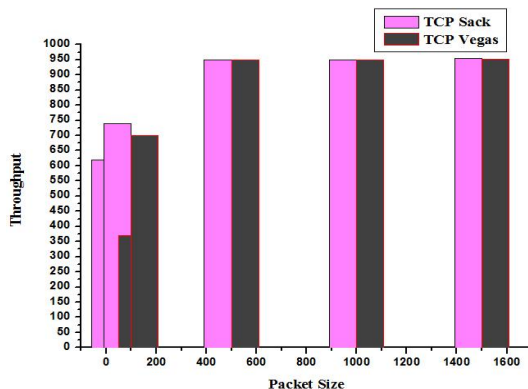


Figure 4: Throughput Vs. Packet size

Packet size TCP Vegas is not efficient but if we increase the size of packet one can find the difference in throughput. However higher packet size is acceptable because it indicates that efficiency becomes good. At large packet size throughput of TCP Sack and TCP Vegas is almost same. So it is better to use TCP Vegas instead of TCP Sack.

### 3.5 Analysis of Graph of Throughput Vs. Window Size

In this case, simulation results are taken by varying window size. From figure 5 it can be seen that at very small window size throughput of both TCP variants is very poor. By increasing the size of window throughput is increases up to fixed value than it becomes constant. Generally the size of window is kept 20 as standard.

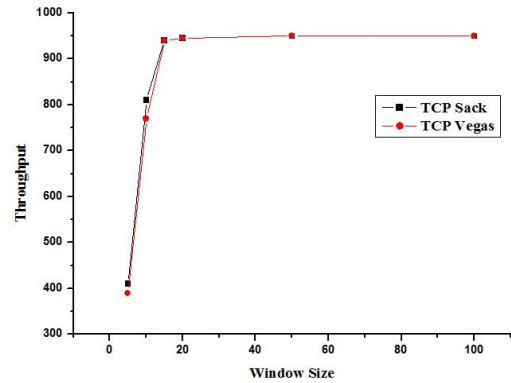


Figure 5: Throughput Vs. Window size

One can see from above figure that throughput of TCP Sack and TCP Vegas are same after window size 15. So it proves that TCP Vegas is similar efficiency giving variant.

### 3.6 Comparison using different Queue Gateways

There are many Queue Gateways in TCP. Each has different policy to manage the congestion in network. Here five gateways named Drop Tail, RED, DRR, SFQ and FQ are taken for simulation.

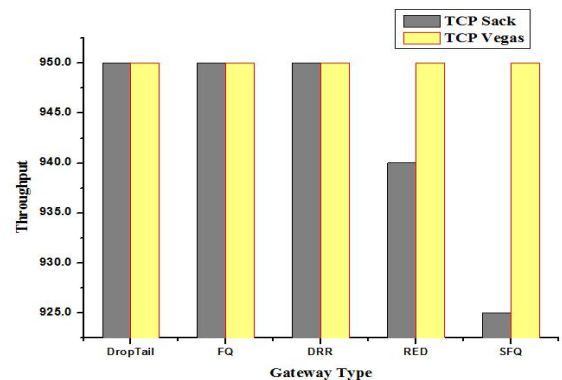


Figure 6: Throughput Vs. Types of Gateways

By changing gateway Drop Tail to RED and SFQ, throughput of TCP Sack decreases but throughput of TCP Vegas remains stable. From figure 6 one can say that TCP Vegas is good compared to TCP Sack in different gateways.

## 4. CONCLUSION

In this paper, analysis is done on two TCP variants named TCP Sack and TCP Vegas. After analyzing the performance from simulated data and graphs obtained, results are found that TCP Vegas is almost similar to TCP Sack. But in some cases one found that TCP Vegas emerges as the better option. The behavior of TCP with SACK is unlikely to cause undesirable network effects. We know that TCP Sack is required same implementation at receiver side and which is not a very easy task. So it is better to use TCP Vegas.

## 5. ACKNOWLEDGMENT

This research has been conducted under the guidance of Mr. Rahul Mehta. Authors are thankful to him.

## **6. REFERENCES**

- [1] RFC 793- Transmission Control Protocol
- [2] RFC 2018 TCP Selective Acknowledgment
- [3] RFC 5681 TCP Congestion Control
- [4] Sally Floyd “Issues of TCP with SACK” March 9, 1996
- [5] Yuvaraju B N, Dr. Niranjana N Chiplunkar “Scenario Based Performance Analysis of Variants of TCP Using NS2- Simulator”,IJCA August 2010.
- [6] Network Simulator - 2 (NS-2) <http://mohit.ueuo.com/NS-2.html> , The ns Manual
- [7] Kevin Fall and Sally Floyd, “Simulation-based Comparisons of Tahoe, Reno, and SACK TCP”
- [8] Shagufta Henna “A Throughput Analysis of TCP Variants in Mobile Wireless Networks”DOI10.1109/NGMAST.2009.71
- [9] Suhas Waghmare, Aditya Parab, Pankaj Nikose, Prof. S. J. Bhosale.”Comparative Analysis of different TCP variants in a wireless environment ”978-1-4244 -8679-3/11/\$26.00 ©2011 IEEE
- [10] JACOBSON, V. Congestion avoidance and control. In Proceedings of SIGCOMM '88 (Stanford, CA, Aug. 1988), ACM.
- [11] A Comparative Analysis of TCP Tahoe, Reno, New-Reno, SACK and Vegas -From Google.
- [12] Lawrence S. Brakmo, Sean W. O'Malley, Larry L. Peterson.”TCP Vegas: New Techniques for Congestion Detection and Avoidance”