Advances in Computational Sciences and Technology
ISSN 0973-6107 Volume 10, Number 6 (2017) pp. 1715-1723

http://www.ripublication.com

TCP Congestion Control and Its Variants

Harjinder Kaur and Dr. Gurpreet Singh*

CSE, GZSCCET, MRSPTU, Dabwali Road, Bathinda, Punjab, India,
**Corresponding author*

Abstract

Transmission Control Protocol provides reliability and end to end delivery of packets in the network. TCP was designed to handle the congestion collapse problem of the network. TCP has various improved versions, which were created time to time as per necessities. In this paper we worked on different TCP variants for instance TCP Reno, TCP New Reno, TCP SACK, TCP FACK and TCP Vegas. We present a comparative analysis of TCP variants on the basis of different parameters.

Keywords: Reno; NewReno; SACK; FACK; Vegas.

1. INTRODUCTION

TCP (Transmission Control Protocol) is mostly used Internet Protocol. The main feature of TCP is that it can handle the congestion. When the packet sending rate is increased than the receiving rate then congestion arises. TCP provides the reliable and connection oriented network. To deals with congestion, TCP provides different variants for example:- TCP Tahoe, TCP Reno, TCP New Reno, TCP Vegas, TCP SACK, TCP FACK, TCP Asym, TCP RBP, Full TCP, CUBIC etc. In this paper we work on a few of them using the NS-2. Network Simulator (NS-2) is used for research and teaching that is evaluated in 1989. These variants basically used three algorithms, slow start, congestion avoidance and Fast Retransmit. TCP Reno uses an additional mechanism, fast recovery. It directly enters in congestion avoidance and skips the slow start, because in fast recovery threshold value and the new congestion window (CWND) is set as CWND= 1/ Current CWND.

TCP New Reno is an extension of the TCP Reno. It has the same features like Reno but it also has an advantage over Reno, it does not leave the fast recovery until the acknowledgment (ACK) of all packets not received. TCP SACK provides selective acknowledgment. SACK can detect multiple packet loss. TCP Vegas is mainly focused on congestion avoidance rather than congestion detection. It introduced three new mechanisms: - New Retransmit mechanism, congestion avoidance and modified slow start. A very important variant known as TCP FACK is working on the top of SACK options. FACK is modified version of SACK that comes with various new features. Actually, all TCP variants work same, but they are used different mechanism in congestion handling. We can say that the performances of TCP variants are affected by its congestion control mechanism.

2. LITERATURE SURVEY

Md. Shohidul Islam et al: - In this research paper author discuss the "TCP Variants and Network Parameters: A Comprehensive Performance Analysis". This research shows how TCP versions respond to various network parameters like propagation delay, bandwidth, time to live, round trip time, rate of packets sending and so on. In this authors select the best version of TCP in each parameter. Such analysis is helpful in selecting the suitable TCP for certain criteria [10].

Yuvaraju B N and Dr. Niranjan N Chiplunkar: - This paper is on "Scenario Based Performance Analysis of Variants of TCP Using NS2 – Simulator". It evaluates the performance of TCP Tahoe, TCP Reno, TCP New Reno, TCP SACK, TCP FACK and TCP Vegas under different scenarios. After analyzing the performance this research found that TCP Vegas is better than other TCP variants for sending data and information [9].

Neha Bathla et al: -"Estimating Performance of TCP Alternatives in Wireless Environment" this paper compares TCP variants on the basis of different parameters using AODV routing protocol on NS-2. After obtaining results, it shown that TCP Vegas have higher efficiency and better performance than other TCP variants [3].

Subramanya P et al: - This paper is on "Performance Evaluation of High Speed TCP Variants in Dumbbell Network" and it uses dumbbells network for calculating the performance of TCP variants. According to this paper the performance of TCP variants depends on the congestion in the network and TCP New Reno performs very well for congestion control [6] [18].

Abhishek Sawarkar and Himanshu Saraswat: - In this paper authors discuss "Performance Analysis of TCP variants". It contains the three experiments in terms of throughput, latency and packet dropped ratio. In the first experiment TCP Vegas performs best whenever apply T-test, variance, mean and simulation analysis. According to second experiment when two flows use the same TCP variant both are fair to each other. But when in different flows they, are not fair to each other in terms of bandwidth utilization. In experiment three, uses the Droptail and RED to get better both throughput and latency because throughput increases whenever using Droptail queuing algorithm and latency when the RED queuing algorithm uses [1].

3. TCP VARIANTS

3.1. TCP Reno

It was introduced in 1990 by Van Jacobson. It has the same features like TCP Tahoe. We can also represent it as follows:-

TCP Tahoe + Fast Recovery = TCP Reno [5]

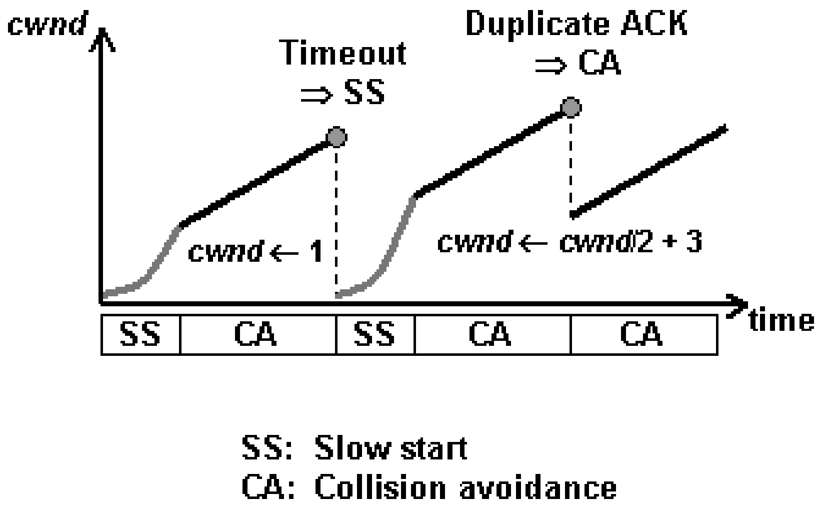

Fig.2. Behavior of TCP Reno

In TCP Reno when three duplicate packets are received, then it is the sign of congestion. If congestion occurs, then TCP Reno retransmits the packets and enters a new mechanism that is fast recovery. The following shows the algorithm for TCP Reno:-

if (cwnd<ssthresh)

cwnd = cwnd + 1 # slow start

else if (cwnd>= ssthresh)

cwnd = cwnd + 1/cwnd# congestion avoidance

if (duplicate ACK)

If (duplicate ACK = = (1 || 2))

cwnd = ssthresh #packet delayed/ out-of-packet received

ssthresh = cwnd/2

else (duplicate ACK > 2)

cwnd = cwnd + Number (ACK) # packet loss due to congestion

ssthresh = cwnd/2

The algorithm shows if cwnd (congestion window) is less than the threshold value (that is represented using variable ssthresh) then congestion window is increment by one otherwise it enters the slow start. As in algorithm shows if one or two acknowledgments are received, then threshold value is set half of the congestion window, but if more than two acknowledgments are received then it indicates the congestion. For each duplicate acknowledgment received increase congestion window by 1. TCP Reno has a limitation that, it can detect only single packet loss [2] [7].

3.2. TCP New Reno

TCP New Reno is the extension of TCP Reno. It has some advantages over TCP Reno that can detect the multiple packet loss and it does not leave the fast recovery until it receives acknowledgment of all packets, present in the window [15]. The fast recovery phase proceeds as in TCP Reno, when a fresh acknowledgment is received then there are two cases:-

(i) If it acknowledges all the packets which are outstanding when entered fast recovery, then it exits fast recovery and set cwnd to ssthresh and still continues congestion avoidance.

(ii) If the acknowledgment is an incomplete acknowledgement, then it deduces that the next packet in line was lost and it retransmits that packet and sets the number of duplicate acknowledgment received on 0 [14].

We have some advantages of TCP New Reno these are given below:-

- It can detect multiple packet loss.
- Its congestion avoidance mechanism is very efficient and utilizes network resources much more efficiently.
- TCP New Reno has few retransmits because of its modified congestion avoidance and slow start.

3.3. TCP SACK

TCP SACK or selective acknowledgement requires that packets should acknowledge selectively. It is an option enabling a receiver to tell the sender the range of non-contiguous packets received. Without SACK, the receiver can only tell the sender about sequentially received packets. The sender uses this information to retransmit selectively only the lost packets. It uses a variable pipe to store outstanding data in the network, which is absent in TCP Reno and New Reno [13] [16]. When pipe < cwnd, then it sends data and Set pipe = pipe + 1. However, when the sender receives an acknowledgement from receiver then set pipe = pipe – 1. In this manner, the sender transmits all of the outstanding data in the network. When all the outstanding packets are acknowledged, then SACK exits fast recovery and enters next phase that is congestion avoidance.

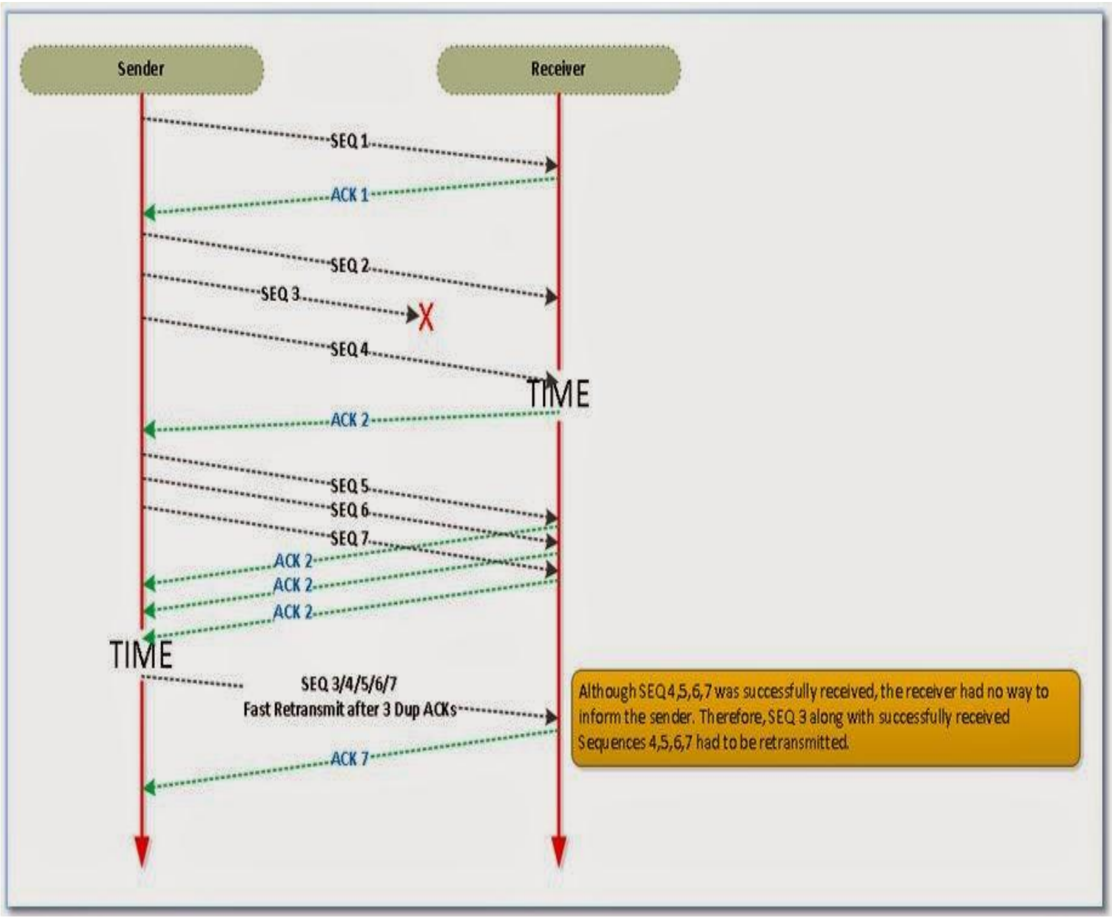

Fig.3. Selectively acknowledge packets

3.4. TCP FACK

The development of TCP SACK with Forward Acknowledgement is called TCP FACK. It has some extra features over TCP SACK, to estimate the amount of data transmit it uses SACK options very efficiently and it also introduces a better way to ½ the window when congestion is detected. The goal of FACK is to perform precise congestion control and improve connection throughput during the recovery phase [8] [12].

3.5. TCP Vegas

According to above mentioned research papers TCP Vegas is better than the other TCP variants. Like Reno, Vegas uses triple duplicate acknowledgments always result in packet retransmission. TCP Vegas introduced a new retransmission mechanism for lost packets. It uses fine-grained round trip time measurements for a compute timeout period of each packet [14]. If the timeout period of the oldest unacknowledgement packets has expired then, packet is retransmitted. Fine-grain timeouts enables faster detection of packet losses and recovery from multiple drops without restarting the slow start phase if the retransmission timer does not expire before. Thus, it solves the problem of multiple packet drops in the same window that we faced in TCP Reno. TCP Vegas also provide modified slow start and congestion avoidance. The original slow start and congestion avoidance need packet losses to realize the congestion, but the modified slow start finds the correct window size without incurring a loss. It happens when Vegas exponentially increasing its window at every other round trip time. It leaves the slow start and enters in the congestion avoidance phase when the actual throughput is lower than the expected throughput [11] [17].

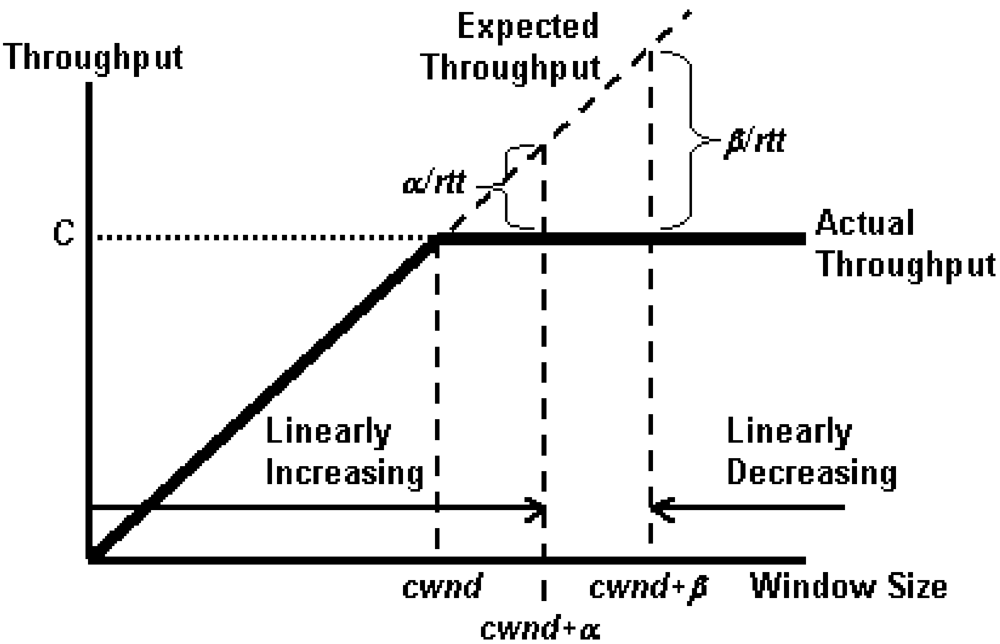

Fig.4. Congestion avoidance in TCP Vegas

4. TCP VARIANT COMPARISON STUDY

TCP variants will be effective based on the parameters that are to be taken into the consideration. The following table shows the comparison of Reno, NewReno, SACK, FACK and Vegas:-

Table 1. Comparative study of different TCP Variant

TCP variants→ Parameters ↓	Reno	NewReno	SACK	FACK	Vegas
Slow Start	Yes	Yes	Yes	Yes	Modified version
Congestion Avoidance	Yes	Yes	Yes	Yes	Modified version
Fast Retransmit	Yes	Yes	Yes	Yes	Yes
Fast Recovery	Yes	Modified version	Modified version	Modified version	Yes
Retransmit Mechanism	Simple	Simple	Simple	Simple	A new mechanism
Congestion Control Mechanism	Simple	Simple	Simple	Simple	A new mechanism
Selective Acknowledgement mechanism	No	No	Yes	No	No
Forward Acknowledgement mechanism	No	No	No	Yes	No
Type of ACK	Immediate	Immediate	Immediate	Immediate	Immediate
Detection of packet loss	A single packet loss	Multiple packet loss	Multiple packet loss	Multiple packet loss	Multiple packet loss
Problem	Can't handle multiple packet loss	Take one RTT for detecting packet loss	It has the problem that implements the selective acknowledgement is a very difficult task.	May not avoid unnecessary inflation of congestion window through delay sensing technique	Do not have any mechanism for re-routing

5. CONCLUSION

In this paper, we discuss the five basic TCP variants; these are TCP Reno, TCP NewReno, TCP FACK, TCP SACK and TCP Vegas. Every variant has a different mechanism to handle the congestion. There is no single variant that can completely overcome the problem to congested and unreliable nature of the network. All variants have their own advantages and disadvantages to solve the network problem of TCP protocol.

REFERENCES

[1] Sawarkar, A. and Saraswat, H., 2016, "Performance Analysis of TCP Variants", International Journal of Computer Science and Network Security, vol. 16, pp. 102-106.

[2] Kamboj, R and Singh, G., 2015, "Various TCP Options for Congestion Evasion", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET). Vol. 4(4), pp. 1534-1539.

[3] Bathla, N., Kaur, A. and Singh, G., 2014,"Estimating Performance of TCP Alternatives in Wireless Environment", National Conference on Cloud Computing & Big Data, vol. 4, pp. 1-12.

[4] Chhabra, A., Dhiman, A. and Joshi, M., 2014, "Performance Evaluation of Variants of TCP Based on Buffer Management over WiMAX", International Journal of Computer Science & Communication Networks, vol. 4, pp. 67-75.

[5] Miyani, H. V., Kukadiya, V. B., Raviya, K. S. and Sheth, D., 2014, "Performance Based Comparison Of Tcp Variants 'Tahoe, Reno, Newreno, Sack' In Ns2 Using Linux Platform", Journal Of Information, Knowledge And Research In Electronics And Communication, vol. 3, pp. 998-1000.

[6] Bathla, N., Kaur, A. and Singh, G., 2014, "Relative Inspection of TCP Variants Reno, New Reno, Sack, Vegas in AODV", International Journal of Research in Engineering and Applied Sciences, Vol. 4, Iss 5, pp. 1-12.

[7] Zafar, F., Mahmood, Z., Ayoub, O. M. and Zhao, Z., 2012, "Throughput Analysis of TCP SACK in comparison to TCP Tahoe, Reno, and New Reno against Constant Rate Assignment (CRA) of 2500 and 4500 bps", Journal of Computer Science & Computational Mathematics, vol. 2, pp. 35- 41.

[8] Bhatia, K. and Valanjoo, A., 2011, "Progress of Different TCP Variant",Journal of Engineering Sciences & Research Technology, vol.11, pp. 3843- 3848.

[9] Yuvaraju, B. N. and Dr. Chiplunkar,N. N., 2010, "Scenario Based Performance Analysis of Variants of TCP Using NS2 – Simulator", International Journal of Computer Applications, vol. 4, pp.20-24.

[10] Islam, M. S., Kashem, M.A., Sadid, W. H., Rahman, M. A., Islam, M. N. and Anam, S., 2009, "TCP Variants and Network Parameters: A Comprehensive Performance Analysis", Proceedings of the International MultiConference of Engineers and Computer Scientists, vol. 1, pp. 978-988.

[11] Singh, G., Kumar, D. and Kaur, A., 2009, "Simulation based comparison of performance metrics for various TCP extensions using NS-2", IEEE Sponsored

International Conference on Innovative Technologies, vol. 9, pp. 106-110.

[12] Ho, C. Y., Chen, Y. - C., Chan, Y. - C. and Ho, C. - Y., 2008, “Fast retransmit and fast recovery schemes of transport protocols: A survey and taxonomy”, Science direct, vol. 52, pp. 1308–1327.

[13] Jamal, H. and Sultan, K., 2008, “Performance Analysis of TCP Congestion Control Algorithms”, International Journal of Computers and Communications, vol. 2, pp. 30-38.

[14] Low, S. H., Peterson, L. and Wang, L., 2002, “Understanding TCP Vegas: A Duality Model”, Journal of the ACM, vol. 49(2), pp. 207-235.

[15] Fahmy, S.; Karwa, T.P. (2001): TCP congestion control: Overview and survey of ongoing research. Computer Science Technical Reports, vol. 11, pp. 1-12.

[16] Kaur, A., Singh, G. and Chauhan D., 2011, "Radical Analysis of TCP Alternatives: Tahoe, New Reno, Sack, Vegas on the basis of imitation in NS-2", Yamuna Journal of Technology and Management, Vol 1, Issue 1, pp. 41-46.

[17] Sharma S., Singh, G. and Kaur, A., 2014, "Study Based Comparison of Multi-Path Routing Protocols in MANETs", International Conference on Futuristic Trends in Engineering & Management, pp. 191-196.

[18] Subramanya, P., Vinayaka K. S., Gururaj, H. L. and Ramesh B., 2014, “Performance Evaluation of High Speed TCP Variants in Dumbbell Network”, Journal of Computer Engineering, vol. 16, pp. 49-53.