An experimental study of the performance and effectiveness of TCP Variants in IP and MPLS Networks

Madiha Kazmi[1], Muhammad Younas Javed[1], Muhammad Khalil Afzal[2]
[1]Dept of Computer Engineering
CE & ME, NUST
Pakistan
[2]COMSATS Institute of Information Technolgy
Wah Cantt. Pakistan
myjaved@ceme.nust.edu.pk, {madihakazmi, khalil_78_pk}@yahoo.com

ABSTRACT: *TCP variants are significant as they are crucial in IP supporting networks. Hence, research is directed towards the management of TCP variants in network management and functioing. TCP provides a trustworthy end to end data transfer under changeable wired networks. The unreliability of IP network is a major problem where TCP is deployed to address the issue. Network management now heavily relies on MPLS over Internet to transfer data, to have edge over the traditional transferring strategies. While studying the protocol networks, it is found that the TCP show varying behavior. The current study is an extension of the work we did for TCP variants under IP and MPLS networks by focusing Tahoe, Reno, New Reno, Sack and Vegas under File Transfer Protocol (FTP).*

Keywords: MPLS Networks, IP supporting networks, TCP variants, Transport Layer Protocols

Received: 1 June 2011, Revised 14 July 2011, Accepted 19 July 2011

1. Introduction

The development of transport layer protocol leads to many implications out of which the TCP emerges as the most popular protocol. TCP provides in sequence deliverance of data and an unfailing data transmission among communicating nodes. One of the strengths of TCP is its high responsiveness towards network congestion. TCP is also a defensive protocol as it detects incident congestion and in result to that it tries to lessen the impacts of this congestion, which will prevent collapse of communication.

Currently, the net based communication is carried to a large extent using TCP, and as a result many researchers are concentrating on modeling and understanding it on different parameters, i.e. time to transmit a file and network consumption. TCP is the fastest growing protocol even in future and we are presenting a comparative study of different TCP variants in MPLS and IP domain. For analytical results of the proposed solution, demonstration of the IP and MPLS network is simulated over a limited number of nodes. The flow of descriptors to maintain network topology determines average delay, throughput, variance of delay, packets sent, packets received, packet dropped. Conclusions are drawn on the basis simulation results, while comparisons between them have been elaborated.

Organization of rest of the paper is as follows: The Section II briefly describes the TCP variants. A summary of the work already done in the field of TCP variants, IP and MPLS is presented in the third section. The Section IV introduces MPLS network. In section V, we present the simulation result followed by their interpretation. Finally we present the analysis on simulation result.

2. TCP Varaints

A brief description of TCP variants is given below. More details of TCP Variants can be found in [13]and [14].

2.1 TCP Tahoe

TCP Tahoe has method to pay compensation for the efficiency plunge caused by congestion after packets are dropped. Tahoe is the very first variant of TCP that uses three mechanisms to organize the flow and handle congestion that is congestion avoidance, slow start, and fast retransmit[13].

2.2 TCP Reno

In 1988 by V. Jacobson proposed a variant of TCP that is typical implementation of TCP protocol, it includes the congestion control algorithm. TCP Reno uses four mechanisms to control the flow and deal with congestion three are those used by Tahoe and fourth algorithm is termed additive increase multiplicative decrease (AIMD) [13].

2.3 TCP New Reno

TCP Reno algorithms are efficient in dealing with single packet lost in a congestion window.

But in case of multiple packets dropped, it will retransmit the packet whose duplicate acknowledgment was received leading Fast Recovery phase to finish. TCP Reno will re-enter the Fast Recovery phase when it comes to know that more packets are dropped. Effectiveness of protocol is affected by again and again entering the Fast Recovery as TCP New Reno will stay in this phase until all lost packets are retransmitted. New Reno works on the mechanism of partial ACK [13].

2.4 TCP Sack

TCP Tahoe, Reno, and New Reno all acknowledge cumulative packets therefore are unable to detect multiple lost packets per round trip time. TCP SACK's selective acknowledgements algorithm deals effectively with multiple packets lost [13].

2.5 TCP Vegas

TCP VEGAS detects congestion before it really occurs and also follows the AIMD paradigm. TCP Vegas switches to congestion avoidance phase as soon as it senses an early congestion by keep on calculating the difference of current and expected throughput [14]

3. Related Work

One of the performance comparison research is conducted in [1], that focuses on performance evaluation of certain variants of TCP protocol over IP and MPLS network .

Another research was conducted to examine various variants of TCP on two types of traffic, i.e. FTP and Telnet. In Universal Mobile Telecommunications System (UMTS) network, there is a significant impact of different type of traffic on as a whole performance of TCP. [2]

Zhong Ren et. al. in [3] integrated mobile IP and MPLS networks. Techniques for controlling and signaling this integration are argued in detail, it also points out some scalability issues of Mobile IP. A similar sort of study is performed by M. Asante in [4] by analyzing the mobile IP and MPLS union architecture. This paper highlighted benefits of this union.

Wierman et al [5] gave a framework for analyzing TCP variations, i.e. Vegas, Sack and Reno. He analyzed that the induced slow start algorithm of Vegas do not help to reduce packet loss but this algorithm wastes a lot of time in slow start phase.

Mazleena Salleh et.al. [6] compared TCP Tahoe, NewReno, Vegas, and Sack over self-similar traffic. They found that NewReno did better than other TCP variants with respect to efficiency and throughput. TCP Vegas showed better throughput then Reno.[7]

Jeonghoon, et .al. [8]results emphasize on former discussed research results.

Go Hasegawa et.al. [9] compared performance of Reno and Vegas sharing bottleneck link on internet found out Reno to be a better performer. Similar results were concluded by Cheng P. Fu et.al. [10]where they compared performance of Reno and Vegas on asymmetric networks having bottleneck.

Thomas Bonald in [11]compared Reno and Vegas keeping RTT measurement as testing template. They focused on long-term performance criterion, i.e. average throughput and average buffer taken up.

Yi-Cheng Chan in [12]has reported a few problematic sides of TCP Vegas while congestion avoidance, which makes its less successful. To trim down impact of Vegas problems authors have also presented congestion avoidance scheme based on router.

4. MPLS Network

The unstable growth of the Internet and the introduction of complicated services require an epoch-making change. MPLS was proposed as an alternative. MPLS is a protocol specified by Internet Engineering Task Force (IETF). MPLS provides many services through networks i.e., routing, effective /efficient designation, forwarding of packets and traffic flows switching. The most salient functions of MLPS is to supervise the traffic flows among heterogeneous applications, hardware and machines. MPLS is not reliant on Layer-2 protocols and Layer-3 protocols [15].

5. Simulation and Analysis

Simulation has 13 nodes in total divided into IP and MPLS domains. The 7 nodes in MPLS domain are named as LSR1 to LSR7, while there are 6 nodes in IP domain labeled as node0 to node5. The IP domain consists of a sender and a receiver network. Both sender and receiver network having three nodes each. The bandwidth between nodes of IP and MPLS network is set 1 MB with a 5ms link processing delay. All MPLS enable nodes provides mechanism for label distribution as they are Label Distribution Protocol (LDP) enabled.

The traffic in FTP having packet size of 1500 KB has varying time interval. The simulation runs for 100 seconds. Both source and destination networks are IP based. Some common networks parameters are revealed in Table 1.

Network parameters	**Values**
Number IP nodes	6
Number of MPLS nodes	7
Number of hops	7
Link processing Delay	5ms
Packet size	1500
Bandwidth	1MB

Table 1. Network parameter

The different TCP variants in MPLS/IP network are analyzed using different scenarios. The numbers of flows of the link were varied to check the effect of different flows on the delay and throughputs. These traffics are run in FTP.

(a) Single Traffic

(b) Multiple Traffic

- a. Two flows Traffic
- b. Four flows Traffic
- c. Eight flows Traffic

FTP traffic has been run on single and multiple flows and results are recorded thereby. These results are then presented in structure of tables and figures for TCP New Reno, TCP Reno, TCP Sack, TCP Tahoe and TCP Vegas.

An FTP connection is set up between node0 and node1 and this simulation is executed for New Reno, Reno, Sack, Tahoe and Vegas. Table 2 shows the values of these simulation average delays in milliseconds. The percentage throughput of different variants on single flow and it can be examined that all variants are giving 100% throughput.

On FTP flow there is no packet loss. Average delay of all the variants is more or less alike and TCP Vegas shows 25% lesser average delay than other variants. The accomplishments of TCP variants can also be observed by recording the delay of packets, .i.e. average delay of simulation traced until 50, 100, 150 and so on, as in Figure 1.

Protocol	Delay (ms)
TCP New Reno	202.551
TCP Reno	202.551
TCP Sack	202.551
TCP Taho	202.551
TCP Vegas	149.270

Table 2. Delay of TCP Variants on Single Flow FTP

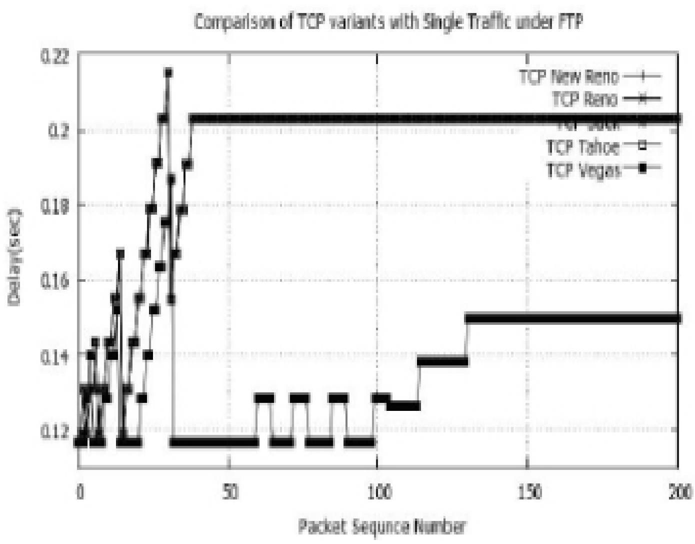

Figure 1. Comparison of TCP Variants with Single Trafcs under FTP

The Table 3 is about average delay in milliseconds on same topology. In this model of FTP flow, there is no packet loss. Average delay of all the variants is roughly similar but TCP Vegas shows 40% lesser average delay than other variants. In other words, TCP Vegas performs superior on two flows than other variants keeping average delay under discussion.

Protocol	FlowID	Delay(millisec)
TCP New Reno	Flow1	440.777
	Flow2	441.031
TCP Reno	Flow1	440.777
	Flow2	441.031
TCP Sack	Flow1	440.777
	Flow2	441.031
TCP Tahoe	Flow1	440.777
	Flow2	441.031
TCP Vegas	Flow1	173.062
	Flow2	173.152

Table 3. Delay of TCP Variants on Two Flows FTP

Figure 2 shows the behavior of TCP variants observed by recording the delay of the packets plotted with their sequence numbers, i.e. average delay after packet number 50, 100, 150 and so on, sent by every variant, is traced.

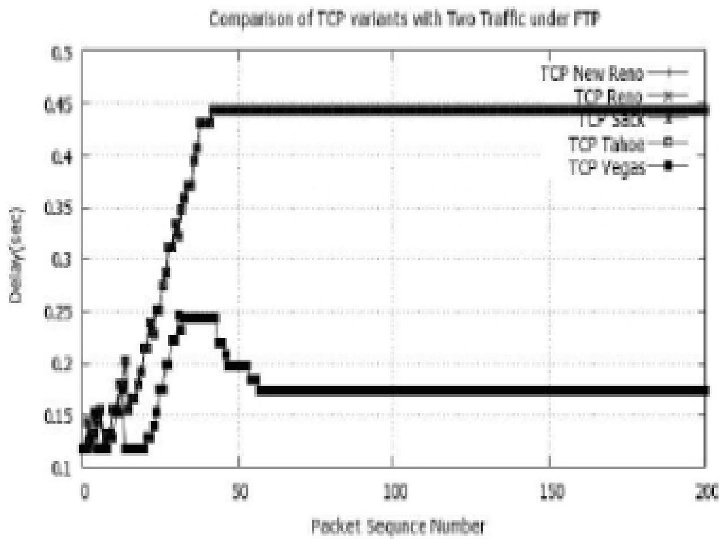

Figure 2. Comparison of TCP Variants with Two Trafcs under FTP

Figure 3 illustrates the behavior of TCP variants observed by recording the delay of packets on basis of their sequence numbers, i.e. average delay of packets after transmission of 50,100,150 and so on packets for every variant under four flows of FTP. It gives a glimpse of behaviors of TCP variants by plotting average delay of packets in seconds across y-axis. Each sharp edge shows the abrupt change in delay of that particular variant. Vegas shows a little jitter in the beginning but later on illustrates a smooth performance, which shows that the delay of the packets remains the same throughout simulation time. There is no packet loss in TCP Vegas. TCP Reno has highest delay reaching 1.8 second, until the end of simulation Reno is giving highest peaks of delay.

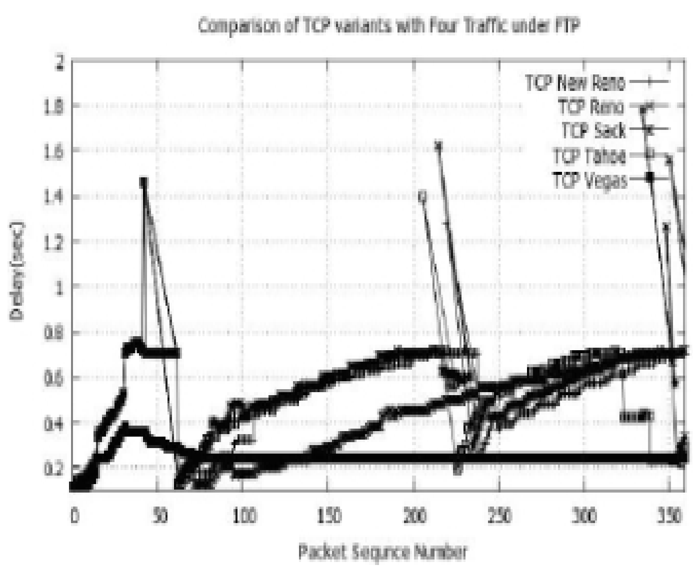

Figure 3. Comparison of TCP Variants with Four Trafcs under FTP

5.1 Error rate induction in FTP Single Flow

TCP provides reliable end to end transport layer protocol. Because of this feature this a protocol used by 90% traffic on internet approximately. Congestion avoidance in TCP allows the application to increase by one packet whenever an acknowledgement is received; allowing full utilization of available bandwidth.

The most important feature of TCP is its Congestion Control strategy. In wired network, whenever there is a packet loss, it indicates that network is congested. Inducing error rate explicitly shows the behavior of different TCP variants, as some variants reduce congestion window unnecessarily. Performance of TCP is affected by various factors like link capacity, RTT, random losses, short flows etc.

As IP is an unreliable network, adding TCP to it will provide reliability via sliding window scheme, ACK, sequence number and control flow to avoid overflowing of receiver buffer [16].

TCP Tahoe is the scheme of TCP that deploys slow start mechanism to prevent the problem of congestion. It is a reactive mechanism. New Reno is an active variant of TCP which is used for multiple packet losses. It provides the solution for oscillating congestion window to resolve problem faced by TCP Reno. For the solution to the problems of TCP's inability to tell about the multiple packetdropping, TCP Sack was proposed. TCP Vegas is the proactive variant of TCP that anticipates the intended congestion on the basis of round trip times of the data packets.

Figure 5 gives an idea about percentage throughput achieved by TCP variants. As the probability of packet loss is increased 1 to 5% the throughput of variants deteriorates. This is because every time a retransmitted packet is lost, the frequency of dropped packet increases and all variants of TCP suffer from numerous drops and timeouts. When random packet loss was introduced, throughput of all variants remained the same i.e. 98, 97, 96, 95 and 94 percent for 1, 2, 3, 4 and 5 percent error rate respectively. Vegas gave lowest end-to-end delay till 3% error rate but dramatically TCP Sack gave delay even lower than Vegas for 4% and 5 % error rate.

Figure 4 gives an idea about percentage throughput achieved by TCP variants. There is significant deteriorates in throughput of variants with increase in the probability of packet loss (1 to 5%).Reason behind this decline is that with every lost packet frequency of dropped packet increases and results in numerous drop and time outs. When random packet loss was introduced, throughput of all variants remained the same i.e. 98, 97, 96, 95 and 94 percent for 1, 2, 3, 4 and 5 percent error rate respectively. Vegas gave lowest end-to-end delay till 3% error rate but dramatically TCP Sack gave delay even lower than Vegas for 4 and 5 % error rate.

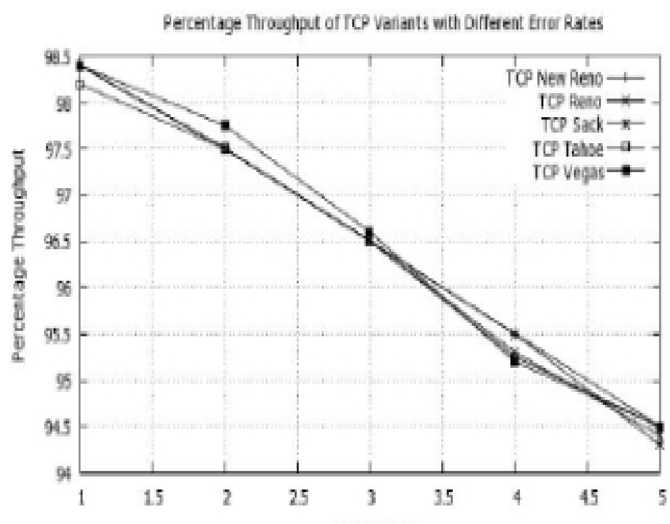

Figure 4. Percentage throughput of TCP variants with diferent Error Rates

The congestion window of all the variants rapidly goes down to the smaller value. New Reno had largest value of average congestion window in 1% error rate case. But, Vegas had largest value and Tahoe had smallest value of average congestion window throughout the experiment scenarios, which shows Tahoe's inferiority to Vegas.

6. Conclusion

TCP is debatably the most significant protocol in the internet today. Congestion control algorithm is special feature of TCP.

TCP tries to achieve the best bandwidth rate vigorously on any network. It keeps on pushing high transfer rate continuously. It also reduces the transfer rate on detecting errors from time to time. Observing the behavior of TCP is quite a revealing experience about behaviors of different variants of TCP on IP and MPLS network.

It is evident that as loss rate increases, throughput decreases. So, it can be concluded that there exists an inverse relationship between loss rates and throughput. As error rate increases throughput decreases gradually. Similarly congestion window size also.

References

[1] Muhammad Saeed Akbar, Zubair Ahmed, S., Muhammad Abdul Qadir, (2006). Quantitative Analytical Performance of TCP Variants in IP and MPLS Networks, IEEE Multitopic Conference, 2006. INMIC'06.

[2] Dubois, X. (2005). Performance of Different TCP Versions over Common/Dedicated UMTS Channels. Master Thesis, University of Namur.

[3] Zhong Ren Chen-Khong Tham Chun-Choong Foo Chi-Chung Ko, (2001). Integration of Mobile IP and MPLS, IEEE ICC 2001, V 7, p 2123- 2127, Helsinki, Finland, June.

[4] Asante, M., Sherratt, R.S. (2004). Mobile IP Convergence in MPLS-based Switching. *In*: Proceeding of Wireless and Optical Communication MultiConference, July.

[5] Adam Wierman, Takayuki Osogami, Jorgen Olsen, (2003). A Unified Framework for Modeling TCP-Vegas, TCP-SACK, and TCP-Reno. Proceedings of MASCOTS.

[6] Mazleena Salleh Ahmad Zaki Abu Bakar (2005). Comparison of TCP Variants over Self-Similar Traffic, *In*: Proceedings of Computers, Communications, & Signal Processing with Special Track on Biomedical Engineering, p.85 - 90.

[7] Ols'en, J. (2003). Stochastic Modeling and Simulation of the TCP Protocol. PhD thesis, Department of Mathematics, Uppsala University, Sweden, Oct.

[8] Mo, J., La, R. J., Anantharam, V., Walrand, J. (1999). Analysis and comparison of TCP Reno and Vegas, *In:* Proc. INFOCOM '99, V. 3, New York, Mar.

[9] Go Hasegawa, Kenji Kurata, Masayuki Murata, (2000). Analysis and Improvement of Fairness between TCP Reno and Vegas for Deployment of TCP Vegas to the Internet, *In:* Proceedings of the 2000 International Conference on Network Protocols, p. 177.

[10] Cheng P. Fu, Soung C. Liew, (2003). A Remedy for Performance Degradation of TCP Vegas in Asymmetric Networks, *Communications Letters*, IEEE 7 (1) 42 - 44, Jan.

[11] Thomas Bonald, (1998). Comparison of TCP Reno and TCP Vegas via Fluid Approximation, Available as INRIA Research Report, November.

[12] Yi-Cheng Chan, Chia-Tai Chan, Yaw-Chung Chen and Cheng- Yuan Ho, Performance Improvement of Congestion Avoidance Mechanism for TCP Vegas, *In:* Proceedings of the 10th International Conference on Parallel and Distributed Systems (ICPADS'04) 1521-9097/04 2004 IEEE

[13] Kai Xu, Ye Tian, Nirwan Ansari, (2005). Improving TCP performance in integrated wireless communications networks, *Computer Networks*, 47, p. 219-237, Elsevier.

[14] Dongkyun Kim Juan-Carlos Cano and Manzoni, P. (2006). A comparison of the performance of TCP-Reno and TCP-Vegas over MANETs, C-K. Toh, IEEE

[15] Rosen, E., Viswanathan, A., Callon,et al., R. (1999). Multiprotocol Label Switching Architecture, IETF Internet Draft, Aug.

[16] Floyd, S. (1996). Issues of TCP with SACK, Tech. Report, http://www.icir.org/floyd/sacks.html. (Cited on page 6 and 58.), January.