Analysis of Vegas Using Network Simulator

Jitender Sharma[1], **Dr Amit Kumar Garg**[2]

ECE Department
MM University, Maulana, Haryana (INDIA)

Abstract- Information sharing has become the basic need of time, and internet supports us to share the information in one of the fastest possible ways. Internet has emerged as the basic need of the time. Internet has influenced every part of our life. Shopping, communication, entertainment, business, information, education all aspects of one's life are available on internet There has been a tremendous increase, almost an exponential rise, in the number of internet users in the recent times, which resulted in the form of congestion problem over the wide area network (WAN). Window size is an important parameter to avoid congestion. The basic idea of this work is to simulate TCP variant Vegas using NS2 at different delay times and window size, to find which is best suited window size for this variant, depending on the parameters like bandwidth and delay time.

Index Terms- RTT, AIMD, TCIP/IP, FAST TCP, TCP RENO, TCP TAHOE, TCP VEGAS, cwnd.

I. INTRODUCTION

Transmission Control Protocol (TCP) is one of the core protocols of the TCP/IP Protocol Suite. TCP is used to provide reliable data between two nodes and works at the transport layer of the TCP/IP model. TCP operates at a higher level, concerned only with the two end systems, for example, a Web browser and a Web server. In particular, TCP provides reliable, ordered delivery of a stream of bytes from a program on one computer to another program on another computer. Besides the Web, other common applications of TCP include e-mail and file transfer. Among its other management tasks, TCP controls message size, the rate at which messages are exchanged, and network traffic congestion [1]. Different variants of TCP use different algorithms to control congestion over a network so as to provide communication of data on a wide area network like internet.

As the global Internet traffic increases, many popular sites are often unable to serve their TCP/IP workload, particularly during peak periods of activity. For example, Web servers for sports events are often swamped by requests during and after games. To address this problem, many sites allocate multiple server hosts to concurrently handle the incoming requests. To support workload sharing, they need a method to distribute the requests among the servers. Since network traffic is self-similar, with waves of heavy traffic at peak times, this requires dynamic feedback control.

Commonly used TCP variant is TCP Reno and uses basic AIMD mechanism only to adjust their congestion window size. TCP Reno was the modified version of TCP Tahoe. These protocols are not scalable as the delay-bandwidth product of the network becomes larger [2] because additive increase is too slow and multiple decreases is too fast. Basic TCP uses packet loss only to adjust the congestion window size.

So, TCP Vegas and FAST TCP were proposed to cope up the same problem. FAST TCP uses packet loss as well as queuing delay as the congestion control parameter and to adjust window after every RTT (Round Trip Time) [3], [4], [5-6].

II. CLASSIFICATION OF TCP PROTOCOLS

TCP protocols are differentiated from each others on the basis of their congestion control strategy and are classified as shown in Figure 1.

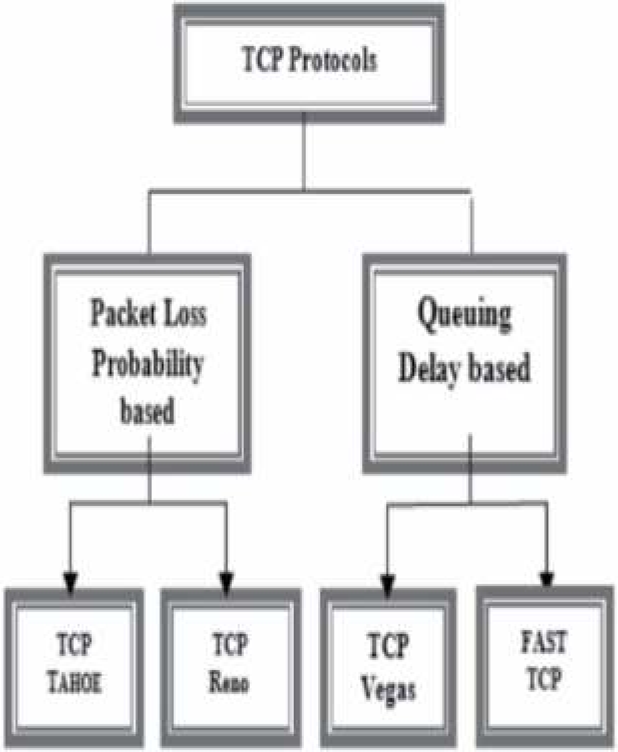

Figure 1: Classification of TCP Protocols

TCP is one of the core protocol used in the communication world. TCP uses basic AIMD (Additive Increase Multiple Decrease) algorithm for the congestion control over a network. TCP Tahoe, TCP Reno, TCP Vegas, FAST TCP are some TCP variants which uses different algorithms to control congestion [2], [7], [8].

A. Loss-based TCP protocols variants

These are the protocols which uses packet drop probability as the main factor for adjusting the window size. These variants of TCP use congestion control algorithms. There were developed initially and are still used. Loss based TCP protocols are more aggressive than the delay based TCP protocols [9]. These are classified as TCP Tahoe and TCP Reno.

B. Delay-based TCP Protocols

Delay-based algorithms were developed so as to provide stable throughput at the receiver end. These TCP variants use congestion avoidance algorithms to avoid the packet loss and are less aggressive than packet loss based TCP protocols. Delay-based algorithms can maintain a constant window size, avoiding the oscillations inherent in loss-based algorithms [6]. However, they also detect congestion earlier than loss-based algorithms, since delay corresponds to partially filled buffers, while loss results from totally filled buffers. This can be either strength or a weakness. If the only protocol used in a network is delay-based, then the inefficiency of loss can be avoided; however, if loss-based and delay-based protocols share the network, then delay-based algorithms tend to be less aggressive. These are the protocols which uses queuing delay as the main factor for adjusting the window size. These variants were developed so as to provide stable throughput at the receiver end. These TCP variants use congestion avoidance algorithms to avoid the packet loss and are less aggressive than packet loss based TCP protocols. These are classified as TCP Vegas and Fast TCP.

C. TCP VEGAS

TCP Vegas is a congestion control or network congestion avoidance algorithm that emphasizes packet delay, rather than packet loss, to determine the rate at which to send packets. TCP Vegas detects congestion during every stage based on increasing Round Trip Time (RTT) values of the packets in the connection unlike Reno, Tahoe etc. which detect congestion only after it has actually happened via packet drops. The algorithm depends heavily on accurate calculation of the Base RTT value. TCP Vegas provides high throughput as compare to TCP Reno and TCP Tahoe.

TCP Vegas was proposed in 1994 with the claim that it can offer higher throughput than Reno and being able to achieve throughput improvement ranging from 37% to 71% compared to Reno [10]. It was developed at the University of Arizona by Lawrence Brakmo and Larry L Peterson. However, the performance of Vegas connection degrades significantly when they coexist with other concurrent Reno connections. Despite its limitations, an innovative idea in Vegas is that it uses a very simple mechanism to measure the network conditions.

- TCP Vegas uses packet delay to adjust the source rate rather than packet loss.
- TCP Vegas adjusts the source rate before actually packet is dropped.
- Queuing delay is the difference between baseRTT and avgRTT.
- TCP Vegas decreases the source rate in case of increase in queuing delay value and increases in case of decrease in queuing delay.

TCP Vegas increases or decreases the window size with fixed number of packets and it is less aggressive as compare to TCP Tahoe and TCP Reno. So, FAST TCP was proposed [11].

The difference between TCP Vegas and FAST TCP lies in the way in which the rate is adjusted when the number of packets stored is too small or large. TCP Vegas makes fixed size adjustments to the rate, independent of how far the current rate is from the target rate. FAST TCP makes larger steps when the system is further from equilibrium and smaller steps near equilibrium [12]. This improves the speed of convergence and the stability [4], [13].

III. PROPOSED WORK

For the comparison of TCP Vegas at different window sizes, simulation is done for the dumbbell topology as shown in Figure 2, in which there are three source nodes (i.e. S1, S2 and S3) which are sending data to sink nodes (i.e. D1, D2 and D3) through a bottleneck link between nodes S0 and D0. Node S0 and node D0 acts as router which forward data to the sink nodes over the network. The delay for all the side links is kept constant, at 1ms as shown in Figure 2. Simulation can be done for different values of link capacities (C) but the results shown are only for C = 100 Mbps. Delay on the bottleneck link (i.e. X) is varied on bottleneck link and simulation is done for four values of X i.e. for X=8 ms, 18 ms, 48 ms and 98 ms, so as to make total delay from source node to the sink node equals to 10ms, 20ms, 50ms and 100ms respectively. Simulation is done for 100 seconds in every case and window size is varied as 200, 300, 400, 500, 600, and 700 and so on, so that the comparison can be made on the basis of the window size.

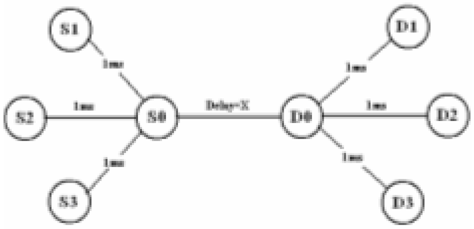

Figure 2: Dumbbell Topology

Here 3 source nodes are taken so as to generate congestion over the bottleneck link. All source nodes use FTP protocol (used on the Application layer of the TCP/IP layer model) to generate bulk amount of data. The source rate is controlled by the different congestion control algorithms used by different TCP variants. There are three active flows used during the simulation for the above mentioned topology: Flow_1 takes place between nodes S1 and D1 from 0 to 100 seconds. Flow_2 takes place between node S2 and D2 from 20 to 80 seconds and Flow_3 takes place between node S3 and D3 from 40 to 60 seconds.

The major responsibility is to develop the code in TCL, which can be simulated in ns2 and then to simulate TCP Vegas in ns2 and to generate a comparison on the basis of Bandwidth-delay product value.

Software used is ns or the network simulator (also called ns-2) is a discrete event network simulator.

- ns is popularly used in the simulation of various protocols. ns supports simulation for wired as well as wireless networks.
- Linux operating system (e.g.. Fedora 9.0.x) or Ubuntu (GUI for Linux).
- Topology Used: Bottleneck or Dumbbell topology.

IV. RESULTS AND DISCUSSIONS

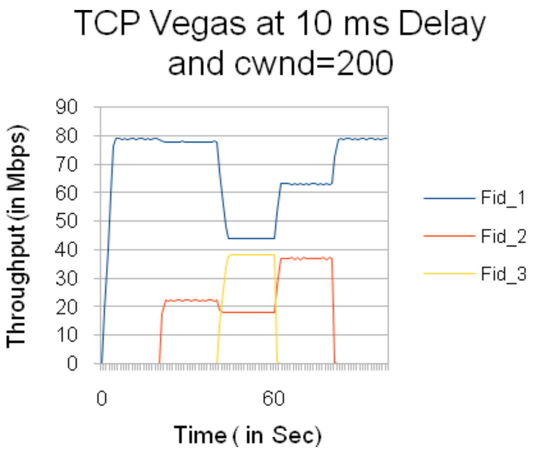

Figure 3(a)

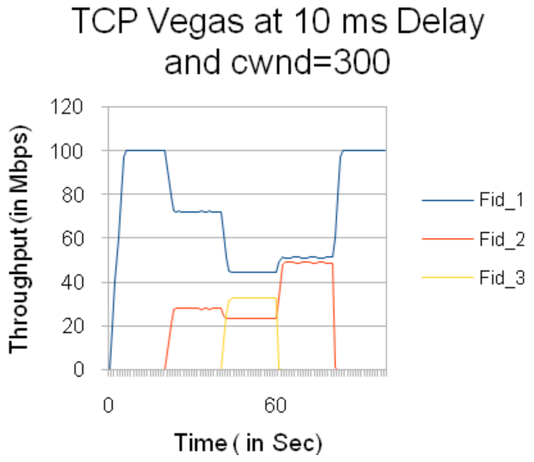

Figure 3(b)

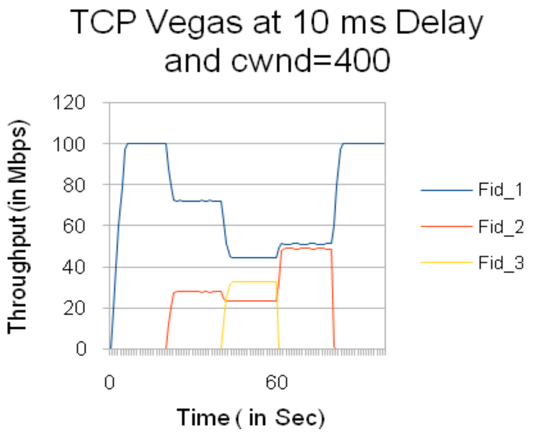

Figure 3(c)

Figure 3: TCP Vegas at delay of 10ms and window size (a) 200 (b) 300 (c) 400

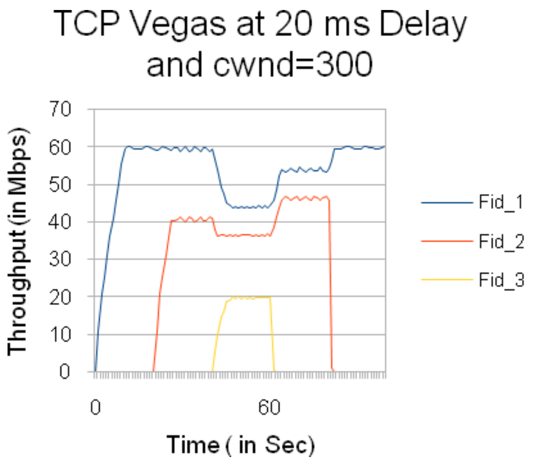

Figure 4(a)

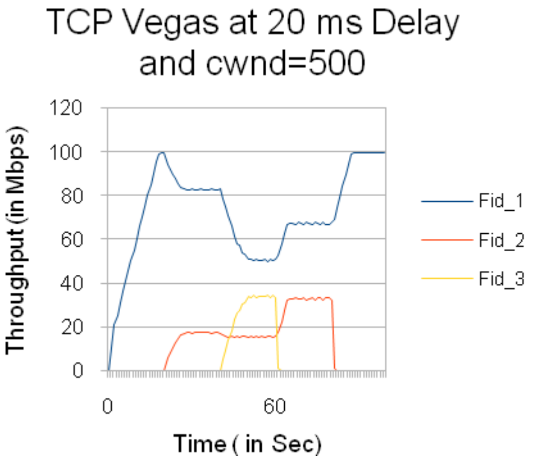

Figure 4(b)

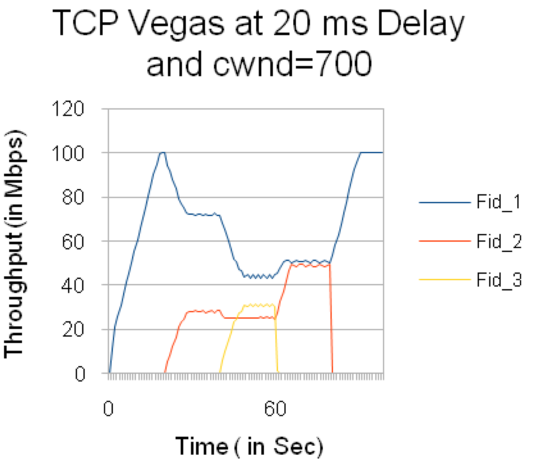

Figure 4(c)

Figure 4: TCP Vegas at delay of 20ms and window size (a) 300 (b) 500 (c) 700

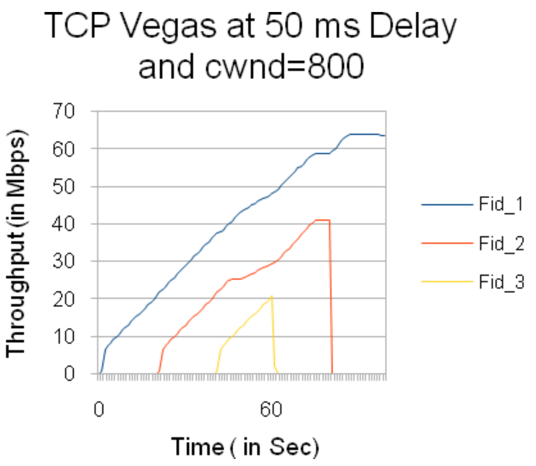

Figure 5(a)

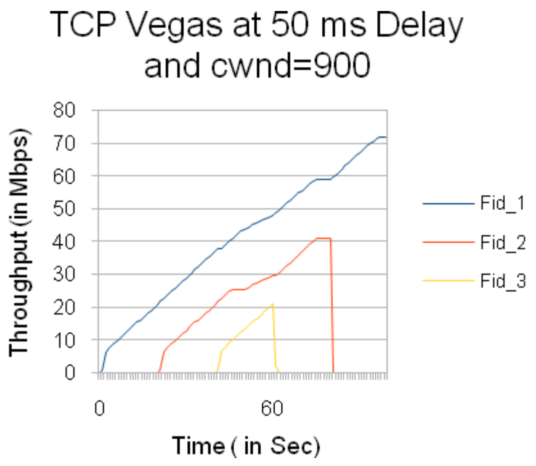

Figure 5(b)

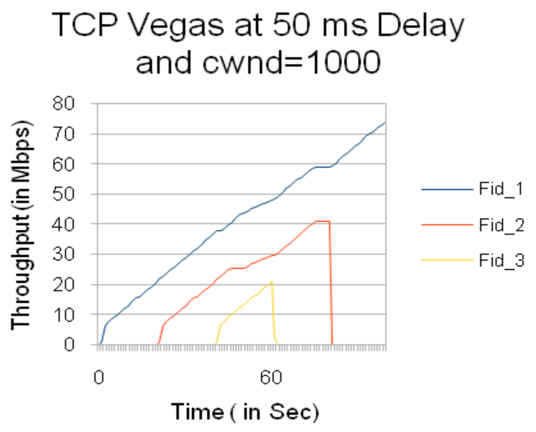

Figure 5(c)

Figure 5: TCP Vegas at delay of 50ms and window size (a) 800 (b) 900 (c) 1000

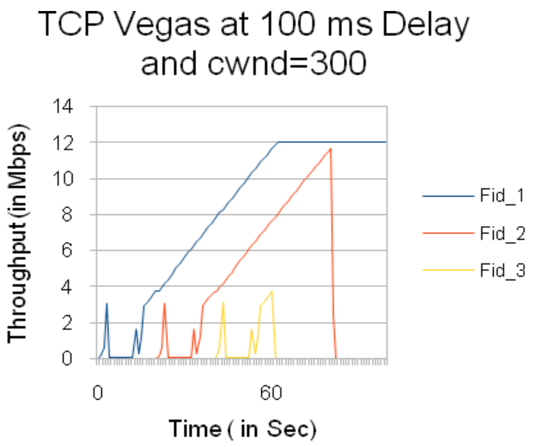

Figure 6(a)

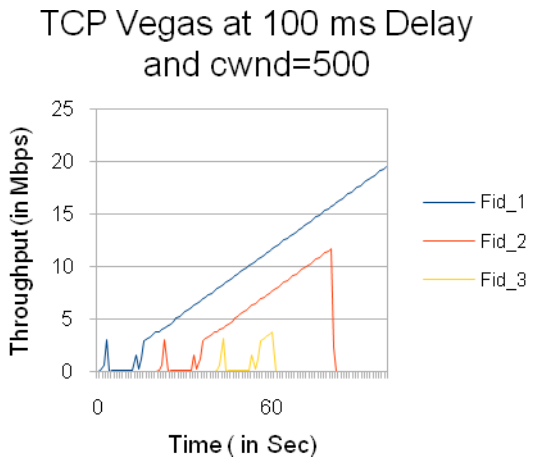

Figure 6(b)

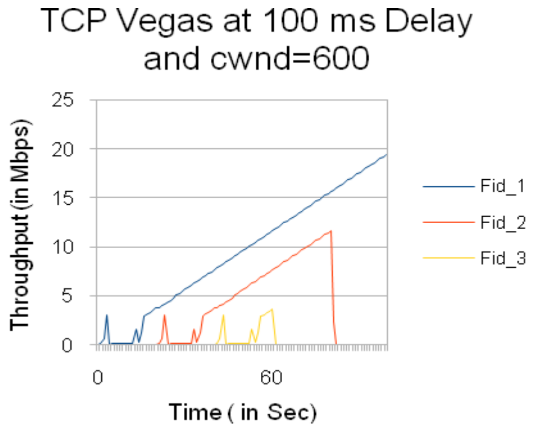

Figure 6(c)

Figure 6: TCP Vegas at delay of 100ms and window size (a) 300 (b) 500 (c) 600

Table 1: Throughput of TCP Vegas at different window size and delay

Congestion window	Delay (ms)	TCP Vegas's Throughput (Mbps)
200	10ms	49.99
	20ms	36.71
	50ms	14.7
	100ms	5.2
300	10ms	53.8
	20ms	43.58
	50ms	20.27
	100ms	6.5
400	10ms	53.8
	20ms	47.81
	50ms	24.84
	100ms	7.12
500	10ms	53.8
	20ms	50.16
	50ms	28.51
	100ms	7.31
600	10ms	53.8
	20ms	49.37
	50ms	30.12
	100ms	7.31
700	10ms	53.8
	20ms	49.37
	50ms	31.62
	100ms	7.31
800	10ms	53.8
	20ms	49.37
	50ms	32.48
	100ms	7.31
900	10ms	53.8
	20ms	49.37
	50ms	32.85
	100ms	7.31
1000	10ms	53.8
	20ms	49.37
	50ms	32.87
	100ms	7.31

V. CONCLUSION AND FUTURE SCOPE

From the graphs obtained by simulation we obtain the value of throughput at different congestion window size and delay and the tabular representation of the data so obtained is shown in Table 1

Therefore from the graphs and the table, we conclude that

A. At 10 ms delay

This gives us almost constant throughput for each flow. But the problem here is that all sources does not get equal access to the available bandwidth.The congestion window size should be greater than or equals to 300 because the difference between the throughput values for the different flows is smaller.

B. At 20 ms delay

The congestion window size should be greater than or equals to 600.

C. At 50 ms delay

The congestion window size should be greater than or equals to 1000 as the throughput is maximum.

D. At 100 ms delay

The congestion window size should be greater than 500 so as to achieve maximum throughput value.

- TCP Vegas provides the constant throughput for the different flows.
- When any source starts sending data, then it decreases the window size by constant size.
- Problem for TCP Vegas is the bandwidth is not shared equally by all the flows. Some sources get access to the maximum available bandwidth whereas others uses the available bandwidth.

For multiple flows over an internet connection, TCP Vegas can lead to oscillations in the throughput values.

For TCP Vegas, if two or more users wants to send the same amount of data, then this may require different amount of data depending upon the amount of bandwidth shared by the individual user or source.

REFERENCES

[1] http://en.wikipedia.org/wiki/Transmission_Control_Protocol

[2] Cheng Peng Fu, Bin Zhou, Jian Ling Zhang, "Modeling TCP Veno Throughput over Wired/Wireless Networks," IEEE COMMUNICATIONS LETTERS, VOL. 11 NO. 9, SEPTEMBER 2007.

[3] C. Jin et al., "FAST TCP: From Theory to Experiments," IEEE Network, vol. 19, no. 1, pp. 4– 11, Jan./Feb. 2005.

[4] J. Wang, D. X. Wei, and S. H. Low, "Modelling and Stability of FAST TCP," in Proc. IEEE INFOCOM 2005, Miami, FL, Mar. 2005.

[5] C. Jin, D. Wei, and S. H. Low, "FAST TCP for high-speed long-distance networks," Internet draft draft-jwl-tcp-fast-01.txt. [Online]. http://netlab.caltech.edu/pub/papers/draft-jwl-tcp-fast-01.txt.

[6] David X., Wei Cheng Jin, Steven H. Low Sanjay Hegde, "FAST TCP: Motivation, Architecture, Algorithms, Performance," IEEE/ACM Transactions on Networking, 14(6):1246-1259, Dec 2006.

[7] Lori A. Dalton, Ciji Isen, "A Study on High Speed TCP Protocols," IEEE Communications Societ, Globecom 2004.

[8] Liansheng Tan, Cao Yuan, Moshe Zukerman, "FAST TCP: Fairness and Queuing Issues," IEEE COMMUNICATIONS LETTERS, VOL. 9, NO. 8, AUGUST 2005.

[9] Tomoya Hatano, Hiroshi Shigeno and Ken-ichi Okada, "TCP-friendly Congestion Control for High Speed Network," Proceedings of the 2007 International Symposium on Applications and the Internet (SAINT'07), 0-7695-2756-6/07 $20.00 © 2007 IEEE.

[10] Tony Cui, Lachlan L. H. Andrew, Moshe Zukerman, Liansheng Tan, "Improving the Fairness of FAST TCP to new Flows," IEEE COMMUNICATIONS LETTERS, VOL. 10, NO. 5, MAY 2006..

[11] Sandor Molnar, Sandor Kardos, and Tuan Anh Trinh, "Parameter Setting Games in TCP Vegas and FAST TCP," High Speed Networks Laboratory, Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics, H-1117, Budapest, Hungary.

[12] Joon-Young Choi, Kyungmo Koo, Jin S. Lee, Steven H. Low, "Global Stability of FAST TCP in Single-Link Single-Source Network," in Proceedings of the 44th Conference on Decision and Control, and the European Control Conference 2005, Seville, Spain, December 12-15, 2005.

[13] T. V. Lakshman, Member, IEEE, and Upamanyu Madhow, Senior Member, IEEE, "The Performance of TCP/IP for Networks with High Bandwidth-

Delay Products and Random Loss," IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 5, NO. 3, JUNE 1997.

First Author – Jitender Sharma, M Tech (ECE) student, Maharishi Markandeshwar University, Maulana, Haryana(India)
email :Er_jitender2007@yahoo.co.in,

Second Author – Dr Amit Kumar Garg, Professor and Head, ECE Department, Maharishi Markandeshwar University, Maulana, Haryana(India), email: garg_amit03@yahoo.co.in