International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-1 Issue-2, December 2011

Analysis of Tahoe: A TCP Variant

Jitender Sharma, Amit Kumar Garg

Abstract- 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 Tahoe 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 [10]. 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 [5] because additive increase is too slow and multiple decrease is too fast. Basic TCP uses packet loss only to adjust the congestion window size.

So, TCP Vegas and FAST TCP are 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) [1], [2], [3-4].

Manuscript published on 30 December 2011.
* Correspondence Author (s)
Jitender Sharma*, M Tech (ECE) Student, MM University Mulana, Ambala, India, 09780811656, email: er_jitender2007@yahoo.co.in.
Dr.Amit Kumar Garg, Professor and Head ECE Department, MM University Mulana, Ambala, India, email: garg_amit03@yahoo.co.in

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.

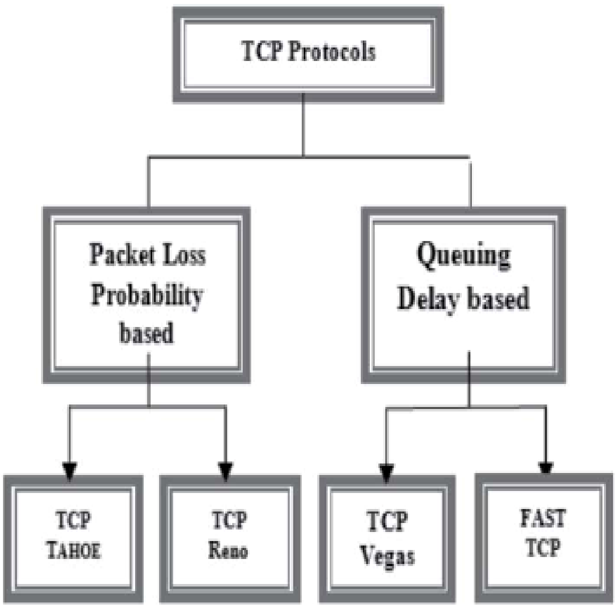


Figure 1: Classification of TCP Protocols

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 [5], [7], [8].

Loss-based TCP protocols variants

These are the protocols which uses packet drop probability as the main factor

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

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 [6]. These are classified as TCP Tahoe and TCP Reno.

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 [4]. 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 a 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.

II. TCP TAHOE

TCP Tahoe is the TCP variant developed by Jacobson in 1988 [9]. It uses Additive Increase Multiplicative Decrease (AIMD) algorithm to adjust window size. It means that increases the congestion window by one for successful packet delivery and reduces the window to half of its actual size in case of data loss or any delay only when it receives the first negative acknowledge. In case of timeout event, it reduces congestion window to 1 MSS [11].

- TCP Tahoe uses packet loss probability to adjust the congestion window size.
- During Slow Start stage, TCP Tahoe increases window size exponentially i.e. for every acknowledgement received, it sends two packets.
- During Congestion Avoidance, it increases the window size by one packet per Round Trip Time (RTT) so as to avoid congestion.
- In case of packet loss, it reduces the window size to one and enters in Slow Start stage.

III. PROPOSED WORK

For the comparison of TCP Tahoe 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.

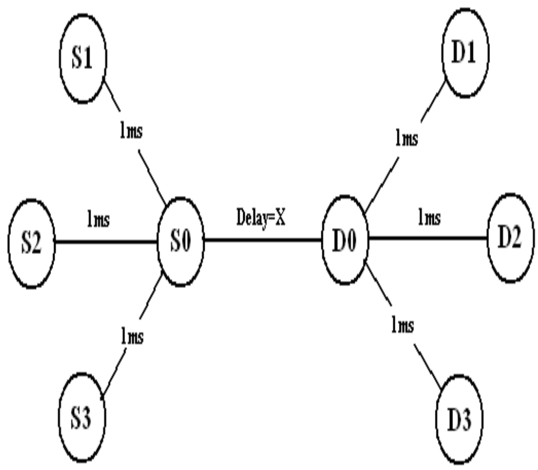


Figure 2: Dumbbell Topology

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 Tahoe 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: Bottelneck or Dumbell topology.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

OPEN ACCESS
International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-1 Issue-2, December 2011

IV. RESULTS AND DISCUSSIONS:

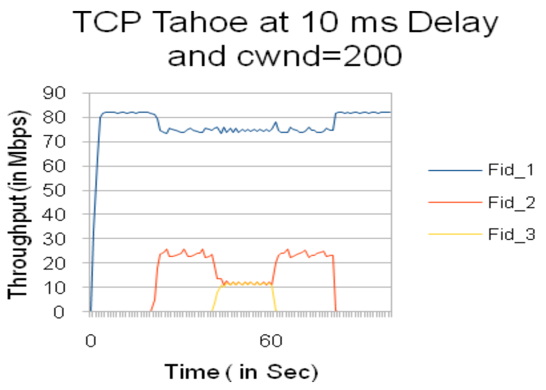


Figure 3 (a)

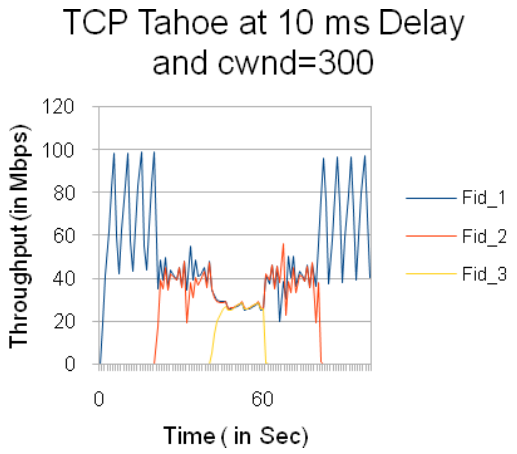


Figure 3(b)

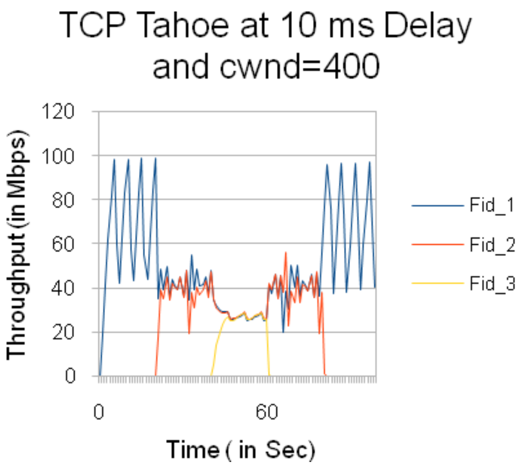


Figure 3(c)

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

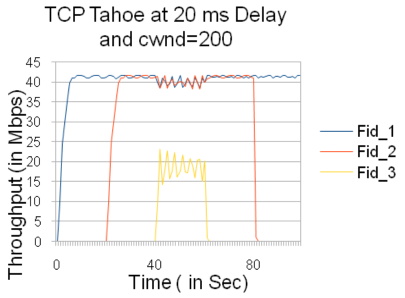


Figure 4(a)

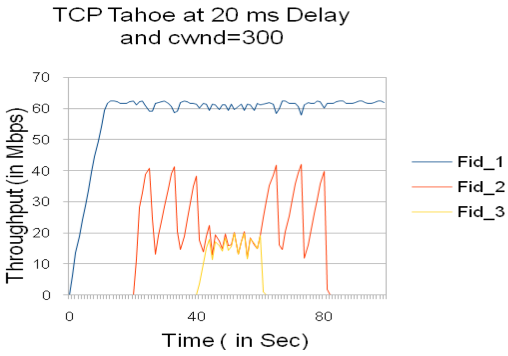


Figure 4(b)

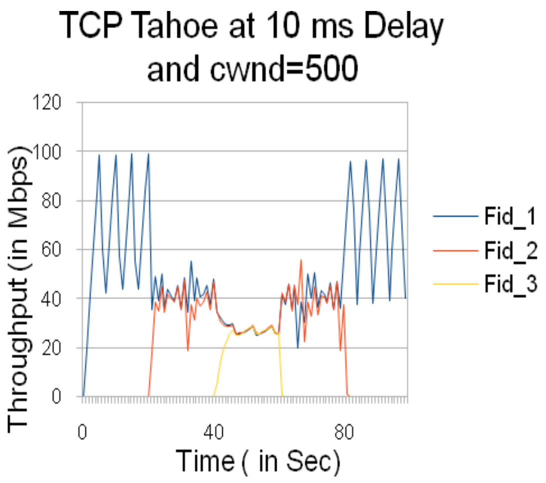
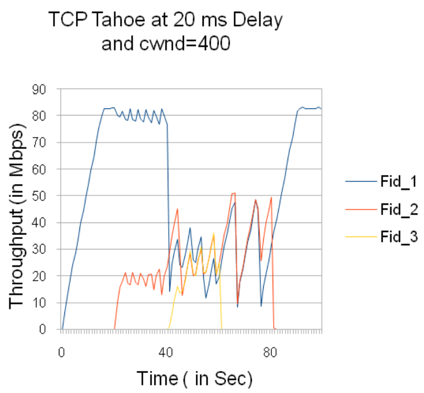


Figure 4(c)



Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

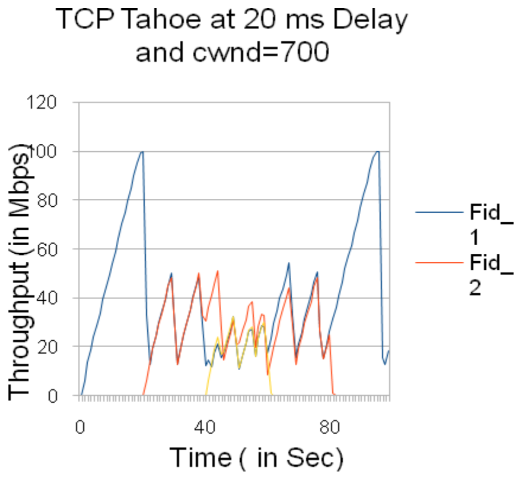


Figure 4(d)

Figure 4: TCP Tahoe at delay of 20ms and window size (a) 200 (b) 300 (c) 400 (d) 700

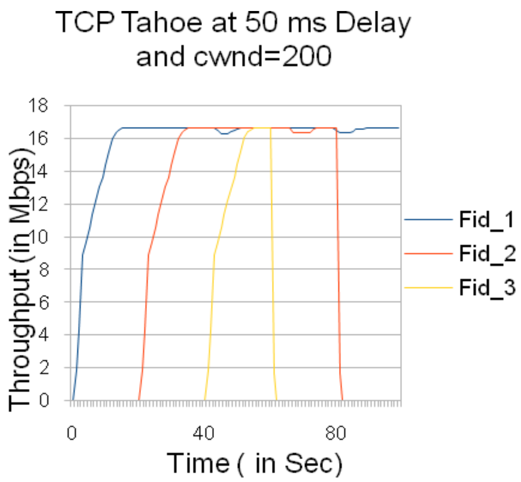


Figure 5(a)

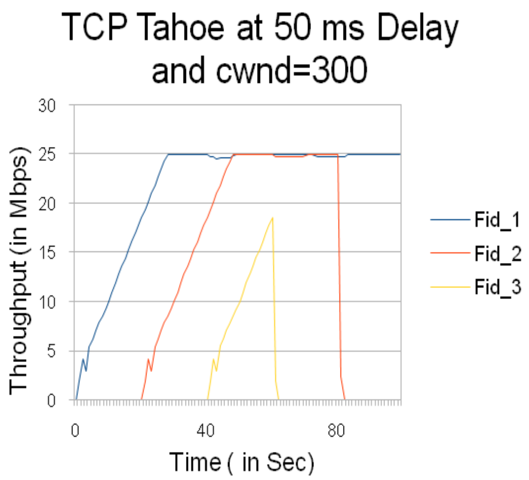


Figure 5(b)

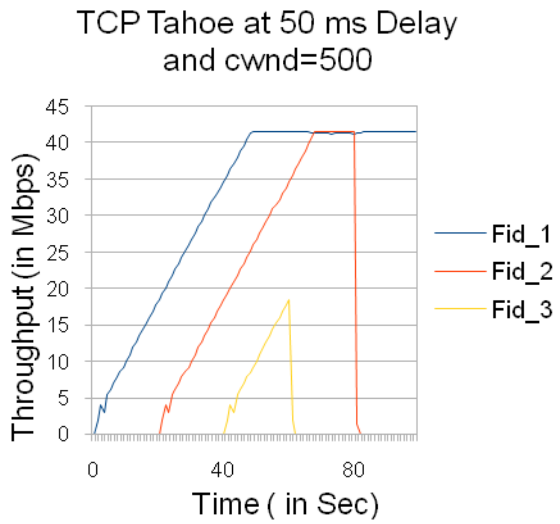


Figure 5(c)

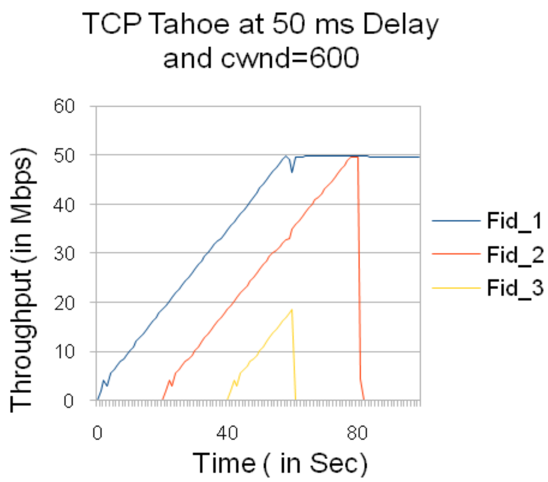


Figure 5(d)

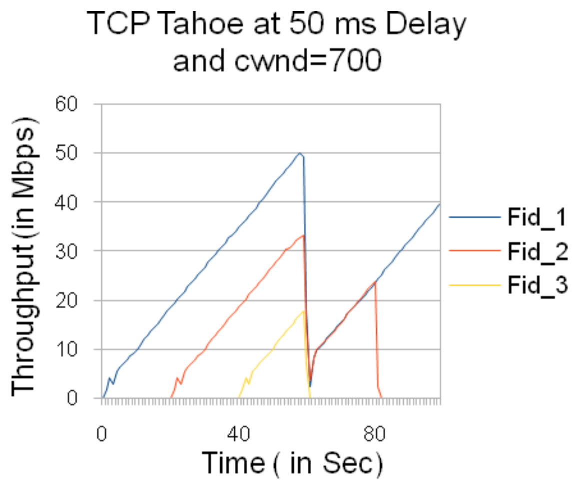


Figure 5(e)

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Figure 5: TCP Tahoe at delay of 50ms and window size (a) 200 (b) 300 (c) 500 (d) 600 (e) 700

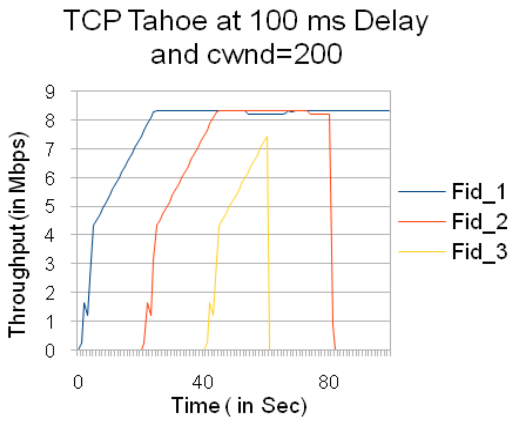


Figure 6(a)

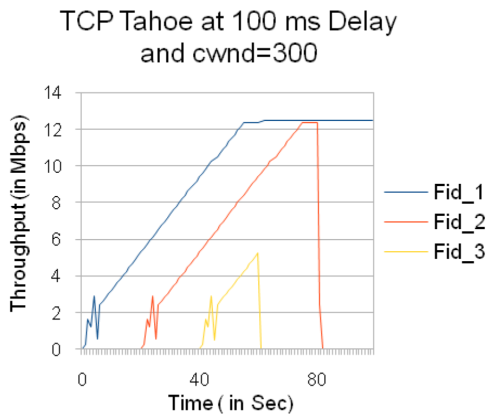


Figure 6(b)

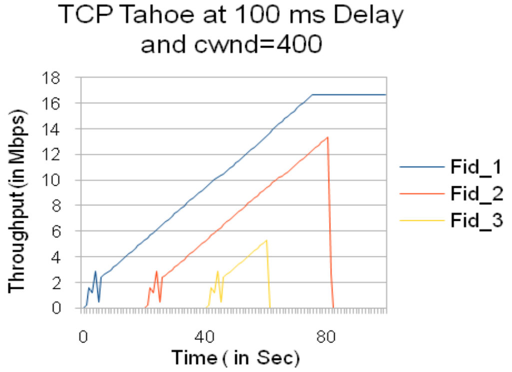


Figure 6(c)

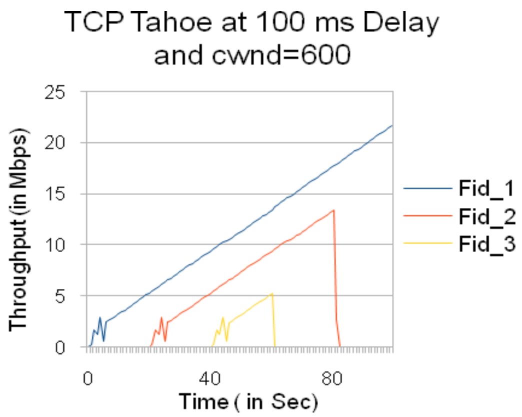


Figure 6 (d)

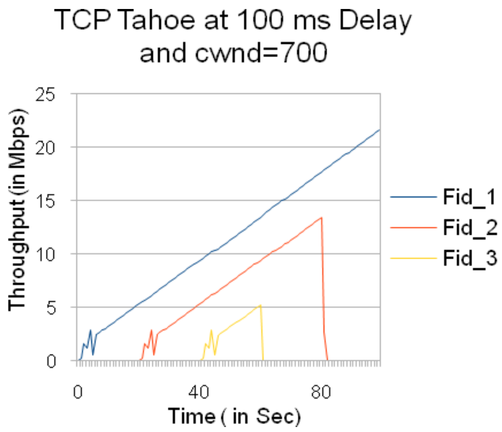


Figure 6(e)

Figure 6: TCP Tahoe at delay of 100ms and window size (a) 200 (b) 300 (c) 400 (d) 600 (e) 700

Congestion window	Delay (ms)	TCP Tahoe's Throughput (Mbps)
200	10ms	50.58
	20ms	37.67
	50ms	15.46
	100ms	7.18
300	10ms	41.8
	20ms	42.26
	50ms	19.86
	100ms	7.99
400	10ms	41.8
	20ms	40.12
	50ms	24.24
	100ms	8.81
500	10ms	41.8
	20ms	50.67
	50ms	27.7
	100ms	9.15
600	10ms	41.8
	20ms	34.59
	50ms	30.18
	100ms	9.16
700	10ms	41.8
	20ms	34.59
	50ms	21.01
	100ms	9.16

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

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

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

(i) At delay=10ms: the congestion window size should be greater than or equal to 400.

For congestion window=200, TCP Tahoe provides almost constant throughput but only flow 1 using the maximum available bandwidth. With increase in congestion window size, flow 1 uses the maximum bandwidth between time 1 to 100 seconds. When other flow starts, it reduces its source rate so that all the flows get equal access to the network.

(ii) At delay=20ms: congestion window size should be greater than or equals to 600.

(iii) At delay=50ms: When congestion window is 200, all three flows share the bandwidth equally between time 40 to 60 seconds. But as the overall throughput is less, therefore higher value of window size is preferred. Congestion window size should be greater than or equals to 700.

(iv) At delay=100ms: congestion window size should be greater than 600.

V. CONCLUSION

We can say that as the distance between the source and destination increases the delay time also increases and hence the total throughput starts reducing so to improve the throughput we increase the window size. Therefore on the WAN when the distance increases between the source and destination the throughput starts to reduce and hence to increase and support the large number of users we increase the window size.

REFERENCES

1. C. Jin et al., "FAST TCP: From Theory to Experiments," *IEEE Network*, vol. 19, no. 1, pp. 4– 11, Jan./Feb. 2005.
2. J. Wang, D. X. Wei, and S. H. Low, "Modelling and Stability of FAST TCP," in *Proc. IEEE INFOCOM 2005*, Miami, FL, Mar. 2005.
3. 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.
4. 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.
5. 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.
6. 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.
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. 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.
10. http://en.wikipedia.org/wiki/Transmission_Control_Protocol
11. By Steven H. Low, Fernando Paganini, and John C. Doyle, "Internet Congestion Control," *IEEE Control Systems Magazine,* 0272-1708/02/$17.00©2002 IEEE.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)