Communications on Applied Electronics (CAE) – ISSN : 2394-4714
Foundation of Computer Science FCS, New York, USA
Volume 2 – No.1, June 2015 – www.caeaccess.org

Simulation Study of Variants of Transmission Control Protocol

Meenakshi Moza, Suresh kumar
Faculty of Engineering and Technology,
Manav Rachna International University, Faridabad

ABSTRACT

Transmission Control Protocol (TCP) provides reliable, connection oriented, end to end communication. TCP is used mostly by wired links. Source of delay and data corruption, due to peripheral parameters, is minimal in wired links. Thus packet loss due to congestion is the main concern. Therefore quality of service parameters, namely throughput, delay and packet loss need to be analyzed. At present many TCP variants are available and therefore their performance can be compared based on these parameters. In this paper various TCP variants are simulated and their performance is compared in terms of throughput, delay, packet delivery ratio and packet loss for a particular network topology. NS2, an event driven simulator, is used for simulation. The result shows that TCP Vegas is better than other TCP variants for sending data and information.

Keywords

Transmission control protocol, Congestion Window, Congestion Control, Congestion Avoidance, Round Trip Time, Acknowledgement, Quality of service..

1. INTRODUCTION

The transmission control protocol has window based flow control mechanism. The flow control of TCP allows the sender to send new packets after receiving the acknowledgement for the previous packet. The congestion control algorithm prevents overrunning the resources of network. The transmission control protocol makes use of slow start, congestion avoidance, fast retransmit and fast recovery congestion control algorithms[1].

Slow Start: The TCP connection starts with slow start for congestion control. Network congestion can be prevented if data is sent after calculating the maximum available bandwidth. The size of the window is increased exponentially. The window size is increased as the number of segments acknowledged increases. Receiving an acknowledgment or a predetermined threshold level not being arrived at, are the requisites for this procedure to take place. TCP is of the assumption that network congestion gives rise to packet loss. Under such circumstances reduction in network load is the solution. Once threshold is attained, congestion avoidance is the resultant phase. An increment in the window size, by one segment takes place for each round trip time[2]. Only a loss event can put a stop to this procedure.

Congestion Avoidance: As long as receiver window is large the slow start mechanism will start discarding packets. Conduction of slow start or congestion avoidance by TCP isdetermined by the slow start threshold (SSTreshhold) as shown in Figure 1.

Fast Retransmit:TCP generates a duplicate acknowledgment, whenever segments are received out of order and sends this acknowledgement to the sender. The cause of a duplicate acknowledgement could be a lost segment or simply because of reordering. Segment is retransmitted, when the sender receives three duplicate acknowledgments. If instead of three, this limit were lowered, then chances of reordered segments would have increased, causing creation and transmission of duplicates needlessly. So TCP need not wait, for the retransmission timer to expire and indication of a lost segment could be given by three duplicate acknowledgments. The lost packet, is retransmitted by TCP sender, going into the fast retransmit mode.

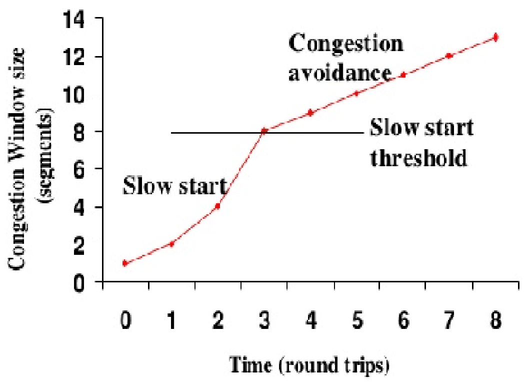

Figure. 1. Congestion window trace

SSThreshold is made equal to half of the current congestion window by the sender. The new congestion window, is equalized to the sum of new SSThreshold and the number of received duplicate acknowledgements[3]. Entry into the Fast Recovery phase starts here.

Fast Recovery:TCP New Reno, during fast recovery makes a distinction between a partial ACK and a full ACK. All segments, which are yet to be acknowledged, at the start of fast recovery are acknowledged by a full ACK and when only some of this outstanding data is to be acknowledged, partial ACK is used. When congestion occurs, higher throughput is generated by fast recovery algorithm. The behavior of various variants of TCP under different scenarios have been analysed. The results will help in deciding which of the TCP variant is suitable for a particular application area.

2. TCP VARIANTS

2.1 TCP Tahoe

In 1988, TCP Tahoe was released with three congestion control algorithms: slow start, congestion avoidance and fast retransmit. Network capacity is reflected by congestion window. Initially if the traffic is more, it may overload the network and the connection for communication may not start at all. This is the reason, why a TCP connection has to go through slow start, as suggested by Tahoe. Slow start means that the sender initializes the congestion window (CWD) to one. The congestion window is then increased by one, whenever an acknowledgement is received. This means that one packet is sent in the first round trip time (RTT), two packets are sent in the second and four packets are sent in the third. Traffic increases exponentially, until a packet is lost which is an indication of congestion. Reduction in the sending rate is the way to counter the congestion. The congestion window is reduced to one and the procedure is started again. The sender is notified about congestion of packet loss which is detected by timeouts.

2.2 TCP Reno

TCP Reno came up in 1990. Fast recovery was the algorithm provided by it. Tahoe's basic principle was retained by TCP Reno. Reno requires, immediate acknowledgement, whenever segment is obtained. When the next expected sequence is delayed, duplicate acknowledgment is the result. This gives rise to packet loss or segments being out of order. Segment needs to be retransmitted, whenever three duplicate ACK's are received, without waiting for timeout. Congestion window is not reduced to one, after a packet loss in TCP Reno. Fast retransmit is the resultant algorithm. Small packet losses are dealt with perfectly in TCP Reno whereas multiple packet losses can pose serious threat.

2.3 TCP New Reno

TCP New Reno is a modification of TCP Reno. When multiple duplicate packets are received Fast retransmit mode is experienced. When acknowledgement for the data, that was out standing at the time, it entered fast recovery, is received, Fast recovery mode is exited[4]. The fast-recovery phase proceeds as in Reno except for the case when a fresh acknowlegement is received. Under such circumstances, two cases mentioned below are experienced:

a. Full ACK, makes it to exit fast recovery. Congestion window size, is set to threshold value and congestion avoidance is proceeded with.

b. Partial ACK, helps to deduce, that the next segment in line was lost. That particular segment is retransmitted. Number of duplicate acknowledgements received, is set to zero[5].

In New Reno, one RTT is required to detect each packet loss. To identify the lost segment, the time when the acknowledgement for the first retransmitted segment was received needs to be known.

2.4 TCP Sack

TCP with selective acknowledgement detects, multiple lost packets, and retransmission of more than one lost packet, is carried out in a round trip time[6]. TCP Sack requires, selective segment acknowledgement and not the cumulative segment acknowledgement. When there are no segments outstanding, then it sends a new packet**.** Thus more than one lost segment, can be sent in one RTT[7].

2.5 TCP Vegas

TCP Vegas is an extension of Reno. TCP Vegas being proactive in nature, does not allow packets to be dropped in the network and therefore has a better packet delivery ratio and almost zero packet loss[8]. Here window size is dependent on round trip time of packet, whereas protocols like TCP Tahoe, Reno and Sack1 vary their window size in ascending order till packet loss is experienced. The packet loss is detected by a presence of large number of duplicate acknowledgements.Congestion of network is prevented by modification of slow start algorithm[9].

TCP Vegas includes the three modifications listed below

2.5.1. New Re-Transmission Mechanism:

RTT is calculated, by keeping track of two things. The first is the time when each segment is sent, and the time the acknowledgement for the same is received.Therefore, it is clear that Vegas modifies the retransmission mechanism of Reno[10] .

2.5.2. Congestion avoidance:

Behaviour of TCP Vegas during congestion avoidance is different from all the other implementations. The loss of a segment is not an indicator of congestion. Congestion is determined by decrease in the sending rate as compared to the expected rate, because of large queues in the routers.

2.5.3. Modified Slow-start:

TCP Vegas implements the slowstart phase in a different way. At the start of a connection , if bandwidth is overshot due to exponential increase in the traffic, congestion will be induced. Exponential increase, is experienced in Vegas only for every other RTT. While doing so, calculation of the actual sending throughput to the expected throughput is carried out. If the difference, reaches a certain threshold, slow start is exited and congestion avoidance phase is experienced.[11].

3. LITERATURE REVIEW

Brakmo[12] put forward a new method of controlling,detecting and avoiding congestion called TCPVegas**.** It is responsible for modifying the earlier TCP congestion control algorithm which detected congestion on the basis of packet delay instead of packet loss.

Vendictis et.al. [13]have evaluatedthe behavior of TCP Vegas and TCP Reno in a wired but heterogeneous network.Thehave concluded that it is not possible to achieve fairness in TCP Vegas and TCP Reno.

Dunaytsev et.al.[14] evaluated the performance of TCP in wired and wireless networks.Evaluation of parameters like Bit error rate and throughput was taken up.The models they came up with were capable of optimizing performance, minimizing losses and capable of operating in wide range of conditions.

4. METHODOLOGY

Ns2 simulator is used to analyze the behavior and performance of different TCP variants. Performance of any network, is evaluated on the basis of the parameters namely Throughput, Packet delivery ratio, Packet loss, Average delay. All these performance metrics, can be calculated using the equations[15] listed below:

Throughput = (∑ received packet size) / (stop time – start time)

Packet delivery ratio = (Number of packets received / Number of packets generated)*100

Packet loss = (Number of packets received - Number of packets generated)

Average delay = (∑ packet receive time - ∑ packet send time)

5. SIMULATION ENVIRONMENT

Simulation and its implementation is used to examine the performance, of various variants of TCP. NS2 is an event driven simulator, used to analyze communication in networks. The topology shown in Figure. 2 has been used to study TCP variant communication for varied packet sizes. The topology consists of 5 senders and receivers with 2 routers in between connected by a 5Mbps channel. The senders and receivers are connected to routers through a 1Mbps channel. Data flows from n0 to n7, n1 to n8, n2 to n9, n3 to n10 and n4 to n11. The requirement for TCP communication is a TCP agent, FTP traffic generator and TCP sink. In NS2, sender agent is represented by Agent/TCP and by default it represents TCP Tahoe. To the base name are added variant names to obtain other TCP agents.

6. RESULT ANALYSIS

The main focus in this paper, is on the effect of packet sizes on throughput, packet delivery ratio, packet loss and delay for various variants of TCP communication, for a particular topology created in NS2. From the simulation results it is observed that TCP Vegas shows a slightly different behavior, than other variants of TCP. TCP Vegas induces three major changes. The first is New Retransmission mechanism, by means of which it detects multiple packet loss.The second is during congestion avoidance it does not utilize loss of segment for saying that congestion has occurred but a reduced sending rate tells us that congestion has occurred. The third is that during modified slow start, Vegas gives an exponential increase for every other RTT and only after the difference between actual sending throughput and expected throughput goes beyond a threshold value, it exits slow start.The performance of TCP Vegas is different, as far as throughput, packet delivery ratio, packet loss and average delay is concerned. TCP Vegas being proactive in nature, does not allow packets to be dropped in the network and therefore has a better packet delivery ratio and almost zero packet loss. The reason for this is, that in TCP Vegas, the window size is made dependent on round trip time (RTT) of packet whereas protocols like TCP Tahoe, Reno and SACK1 .vary their window size in ascending order till packet loss is detected. The tables with actual values obtained for all the parameters namely Throughput, Packet delivery ratio, Packet loss and delay for different TCP variants are drawn on the next page.

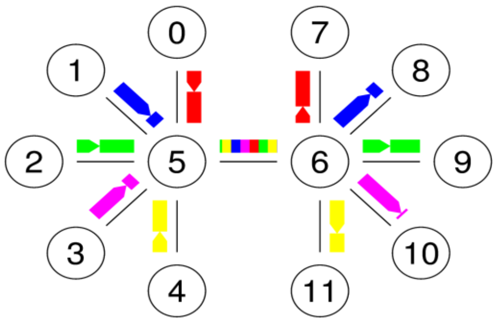

Figure. 2. Snap shot of the above analysis

Table 1 Actual values of various parameters for Tahoe

PKT SIZE	THROUGHPUT	PACKETS SENT	PACKETS RECEIVED	PDR %	PACKET LOSS	AVERAGE DELAY	TOTAL DELAY
100	10648	6655	6655	100%	0	0.0111	74
200	10328	6455	6455	100%	0	0.0119	77
400	9494	5934	5934	100%	0	0.0141	84
500	7792	4870	4870	100%	0	0.0152	74
800	5086	3180	3179	99.9%	1	0.0168	54
1000	4140	2592	2588	99.8%	4	0.0197	51
1500	2845	1790	1778	99.3%	12	0.0229	41

Table 2 Actual values of various parameters for Reno

PKT SIZE	THROUGHPUT	PACKETS SENT	PACKETS RECEIVED	PDR %	PACKET LOSS	AVERAGE DELAY	TOTAL DELAY
100	10648	6655	6655	100%	0	0.0111	74
200	10328	6455	6455	100%	0	0.0119	77
400	9494	5934	5934	100%	0	0.0141	84
500	7792	4870	4870	100%	0	0.015	74
800	5086	3180	3179	99.9%	1	0.016	54
1000	4140	2592	2588	99.8%	4	0.0197	51
1500	2845	1790	1778	99.3%	12	0.02	41

Table 3 Actual values of various parameters for Vegas

PKT SIZE	THROUGHPUT	PACKETS SENT	PACKETS RECEIVED	PDR %	PACKET LOSS	AVERAGE DELAY	TOTAL DELAY
100	10432	6520	6520	100%	0	0.010	70.4
200	10112	6320	6320	100%	0	0.011	73.3
400	9289	5806	5806	100%	0	0.013	76.68
500	7792	4870	4870	100%	0	0.014	69.36
800	5082	3176	3176	100%	0	0.016	52.76
1000	4112	2570	2570	100%	0	0.018	47.8
1500	2760	1725	1725	100%	0	0.023	39.67

Table 4 Actual values of various parameters for Sack

PKT SIZE	THROUGHPUT	PACKETS SENT	PACKETS RECEIVED	PDR %	PACKET LOSS	AVERAGE DELAY	TOTAL DELAY
100	10648	6655	6655	100%	0	0.011	74
200	10328	6455	6455	100%	0	0.0119	77
400	9494	5934	5934	100%	0	0.014	84
500	7792	4870	4870	100%	0	0.015	74.39
800	5086	3180	3179	99.9%	1	0.016	53.58
1000	4140	2592	2588	99.8%	4	0.019	51
1500	2845	1790	1778	99.32%	12	0.022	41

The above tabulated values are shown below in the form of graphs for each of the parameters and for the four variants of TCP.

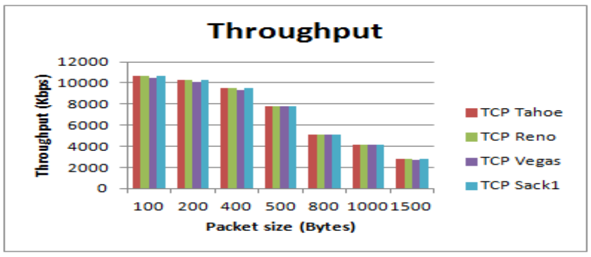

Figure. 3. Packet size vs Throughput for TCP variants

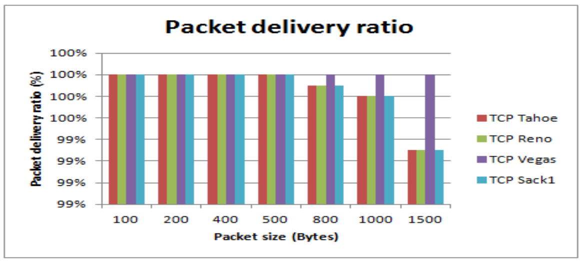

Figure. 4. Packet size vs Packet Delivery Ratio for TCP variants

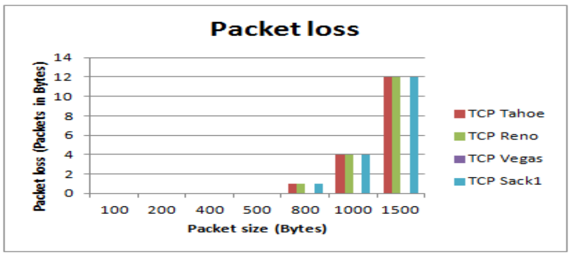

Figure. 5. Packet size vs Packet loss for TCP variants

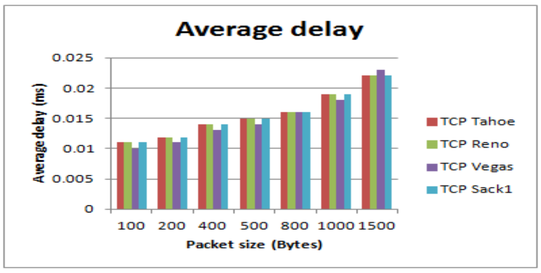

Figure. 6. Packet size vs Average delay for TCP variants

7. CONCLUSION AND FUTURE SCOPE

A detailed analysis of performance of five TCP variants is presented here. From the figures it is observed that TCP Vegas shows a slightly different behaviour than other variants of TCP. The performance of TCP Vegas is different as far as throughput, packet delivery ratio, packet loss and average delay is concerned. TCP Vegas being proactive in nature does not allow packets to be dropped in the network and therefore has a better packet delivery ratio and almost zero packet loss . The future work can be done in the following areas:

a. Analysis can be extended to newer TCP's like TCP Westwood, TCP Fack.

b. b. Parameters like overheads involved in routing, product of bandwidth with delay incurred and the sum total requests for route sent can also be measured and analysed.

8. REFERENCES

[1] Habbal, A. M. M., & Hassan, 2010 September, Second IEEE International Conference on Network Applications Protocols and Services pp. 48-54, Loss detection and recovery techniques for tcp in mobile ad hoc network.

[2] Al Hanbali, A., Altman, E., &Nain, 2005 IEEE Communications Surveys and Tutorials, 7(1-4), 22-36, A survey of TCP over ad hoc networks.

[3] Qureshi, B., Othman, M., & Hamid, 2009, February,2nd IEEE International Conference on Computer, Control and Communication, pp. 1-6, Progress in various TCP variants.

[4] Bisen, D., & Sharma, 2011, International Journal of Computer Science and Communication, 1(2), 395-399, Improve performance of tcp new reno over mobile ad-hoc network . . International Journal of Computer Science and Communication, 1(2), 383-386.

[5] Dobhal, D. C., & Sharma, 2010,International Journal of Computer Science and Communication, 1(2), 383-386, Simulation base analysis of TCP Reno and TCP Westwood over IEEE 802.11 wireless Adhoc Networks.

[6] Sharma, J., & Garg, 2010,International Journal of Engineering and Advanced Technology ISSN, 2249-8958, Analysis of Tahoe: A TCP Variant.

[7] Fall, K., & Floyd, 1996, ACM SIGCOMM Computer Communication Review, 26(3), 5-21, Simulation-based comparisons of Tahoe, Reno and SACK TCP.

[8] Tayade, M., & Sharma, 2011,International Journal of Engineering Science & Technology, 3(3), Review of different tcp variants in ad-hoc networks.

[9] Greis, M, 1998http://www.isi.edu/nsnam/ns/tutorial/Tutorial for the UCB/LBNL/VINT Network Simulator “ns”.

[10] Meher, P. K., & Kulkarni, 2011 IEEE International Conference on Communication Systems and Network Technologies, pp. 67-70 , Analysis and Comparison of Performance of TCP-Vegas in MANET.

[11] Khelage, P. B., & Kolekar, D. U, 2014 International Journal of Scientific & Engineering Research, 5(1), Survey and Simulation based Performance Analysis of TCP-Variants in terms of Throughput, Delay and drop Packets over MANETs.

[12] Brakmo et.al.,1999 In the proceedings of IEEE Journal on selected areas in Communications Vol 13,no.8,pp. 1465-1480,TCP Vegas End to End Congestion Avoidance on a Global Internet.

[13] Vendicitis et.al.,2003In the proceedings of 10th International Conference on Computer Communication and networks,Analysis and enhancement of TCP congestion control in a mixed TCP Vegas and TCP Reno network scenario.

[14] Dunaytsev,2010Phd Thesis University of Technology,TCP Performance Evaluation over wired networks.

[15] Terry Slatter, 2011,Computer Communication review, . TCP performance and Mathis equation.