Improving the Performance of TCP Using Window Adjustment Procedure and Bandwidth Estimation

R.Navaneethakrishnan
Assistant Professor (SG)
Bharathiyar College of Engineering and Technology, Karaikal, India.

ABSTRACT— *The Internet plays a significant role nowadays in our lives. Two main protocols are being implemented in the transport layer of the internet, namely UDP (User Datagram Protocol) and TCP (Transmission Control Protocol). They are distinguished by their connection type. UDP is a connectionless protocol that suits for multimedia transmissions. TCP is a connection - oriented protocol used to provide a reliable data transfer between two systems. Internet traffic is TCP-based, which causes Congestion. To avoid the Congestion existing method used Slow Start, Congestion Avoidance, and Fast Retransmit. To overcome the drawbacks Of the existing system, we propose two methods, Window Adjustment Procedure and Bandwidth Estimation. In window adjustment procedure the congestion window size is reduced in terms of 10%. Hence, it provides high utilization, minimum delay, maximum throughput, fairness. The bandwidth is estimated by the rate of returning acknowledgement.*

Keywords— Slow Start, Congestion Avoidance, Fast Retransmit, Window Adjustment Procedure, Bandwidth Estimation, Congestion Control

1, INTRODUCTION

The Transmission Control Protocol (TCP) is one of the core protocols of the Internet Protocol Suite. TCP is one of the two original components of the suite, complementing the Internet Protocol (IP), and therefore the entire suite is commonly referred to as TCP/IP. TCP provides the service of exchanging data directly between two network hosts, whereas IP handles addressing and routing message across one or more networks. In particular, TCP provides reliable, ordered delivery of a stream of bytes from a program on one computer to another program on another computer. TCP is the protocol that major Internet applications rely on, applications such as the World Wide Web, e-mail, and file transfer. Other applications, which do not require reliable data stream service, may use the User Datagram Protocol (UDP) which provides a datagram service that emphasizes reduced latency over reliability. [1] in which some problems of TCP in high speed network environments are analyzed. Such problems are related to the standard algorithms Slow-Start and Congestion Avoidance, which were designed when networks provided much lower bandwidth that the current ones do. Two solutions have been described: Limited Slow-Start and High Speed TCP. Limited Slow-Start allows the TCP sender to better estimate the available bandwidth and to achieve the transfer with a higher value of congestion window than for the case of standard slow-Start. The High Speed TCP modifies the standard TCP's response function in order to make the sender more aggressive in congestion avoidance phase for large sizes of the congestion window. Experiments show that both for improving TCP in high speed networks. The former because it reduces the effect of Slow-Start overshoot phenomenon and helps to better probe the available link bandwidth during the initial Slow-Start phase. The latter because it uses a more suitable TCP's response function for TCP in high speed networks, which makes the sender better reacting to congestion-related events. Limited Slow-Start and High Speed TCP have been implemented in Network Simulator [NS]. The two algorithms have been explored in simulations and with experiments in real environments. However, further research is needed to better tune the algorithms parameters and to study the fairness of TCP connections using the Limited Slow-Start and the High-Speed TCP variant against standard TCP versions.

2, EXISTING SYSTEM

2.1 Slow Start:

(Multiplicative Increase) It operates by observing that the rate at which new packets should be injected into the network is the rate at which the acknowledgments are returned by the other end. Slow start adds another window to the sender's TCP: the congestion window, "cwnd". When a new connection is established with a host on another network, the congestion window is initialized to one segment (i.e., the segment size announced by the other end, or the default, typically 536 or 512). Each time an ACK is received, the congestion window is increased by one segment. The sender can transmit up to the minimum of the congestion window and the advertised window.The congestion window is flow control imposed by the sender, while the advertised window is flow control imposed by the receiver. The former is based on the sender's assessment of perceived network congestion; the latter is related to the amount of available buffer space at the receiver for this connection. The sender starts by transmitting one segment and waiting for its ACK. When that ACK is received, the congestion window is incremented from one to two, and two segments can be sent.

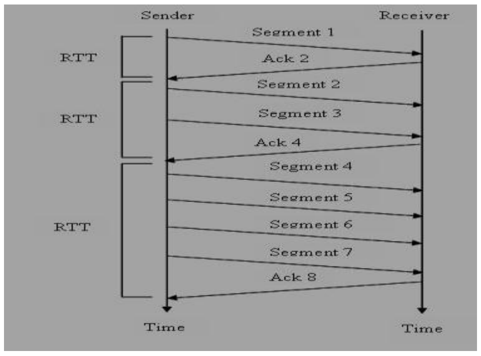

Figure:1. Slow Start, Exponential

Increase

When each of those two segments is acknowledged, the congestion window is increased to four. This provides an exponential growth, although it is not exactly exponential because the receiver may delay its ACKs, typically sending one ACK for every two segments that it receives.

2.2 Congestion **Avoidance : (Additive Increase):**

Congestion can occur when data arrives on a big pipe (a fast LAN) and gets sent out a smaller pipe (a slower WAN). Congestion can also occur when multiple input streams arrive at a router whose output capacity is less than the sum of the inputs. Congestion avoidance is a way to deal with lost packets. Congestion avoidance and slow start require that two variables be maintained for each connection: a congestion window, cwnd, and a slow start threshold size, ssthresh. The combined algorithm operates as follows:

1. Initialization for a given connection sets cwnd to one segment and ssthresh to 65535 bytes. The TCP output routine never sends more than the minimum of cwnd and the receiver's advertised window.

2. When congestion occurs (indicated by a timeout or the reception of duplicate ACKs), one-half of the current window size (the minimum of cwnd and the receiver's advertised window, but at least two segments) is saved in ssthresh. Additionally, if the congestion is indicated by a timeout, cwnd is set to one segment (i.e., slow start).

3. When new data is acknowledged by the other end, increase cwnd, but the way it increases depends on whether TCP is performing slow start or congestion avoidance.[3].

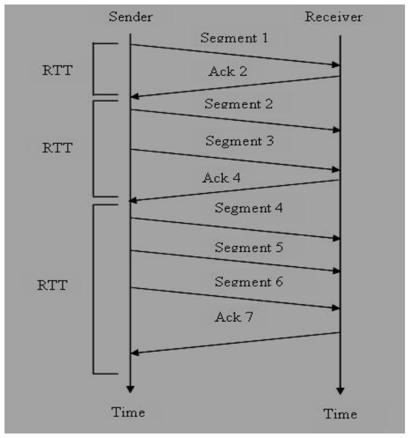

Figure: 2.Congestion Avoidance, Multiplicative Increase

2.3 Fast Retransmit:

TCP may generate an immediate acknowledgment (a duplicate ACK) when an out-of-order segment is received. This duplicate ACK should not be delayed.

The purpose of this duplicate ACK is to let the other end know that a segment was received out of order, and to tell it what sequence number is expected. Since TCP does not know whether a duplicate ACK is caused by a lost segment or just a reordering of segments, it waits for a small number of duplicate ACKs to be received.

It is assumed that if there is just a reordering of the segments, there will be only one or two duplicate ACKs before the reordered segment is processed, which will then generate a new ACK. If three or more duplicate ACKs are received in a row, it is a strong indication that a segment has been lost. TCP then performs a retransmission of what appears to be the missing segment, without waiting for a retransmission timer to expire.

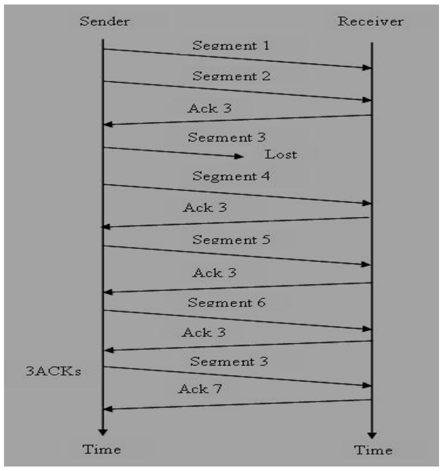

Figure: 3 Fast Retransmit

3, PROPOSED SYSTEM

3.1 Window Adjustment Procedure:

In the existing system, the window adjustment procedure reduces the window size into half, if congestion occurs during packet transmission. But in the proposed system, the window adjustment procedure will reduce the window size in terms of 10%. The procedure generally increases the throughput and minimizes the delay in packet transfer. Initially get the number of users and the packet size for each user in bytes as an input. Then calculate the total packet size for each user.

Window Adjustment Procedure: If congestion occurs the window adjustment procedure reduces the congestion window size is in terms of 10%. This advantage over this procedure is its high utilization. It also increases the throughput but minimizes the time delay. Initially set the bandwidth as 8 Mbps and the window size as 72000 bytes. Convert the bandwidth value in to bytes and compare it with the total value. The message will be sent if the total is less than the bandwidth value. After sending the message, calculate the Round Trip Time (RTT) value for each acknowledgement (ACK). Already we have the bandwidth as,

$$\mathbf{W.S = RTT * B.W}$$

Where Window Size (W.S) is the product of Round Trip Time (RTT) and Bandwidth

(B.W). From that we have,

RTT = W.S / B.W

If the total value is greater than the bandwidth, apply the window adjustment procedure. This procedure reduces the window size in terms of 10%. It continues sending until the whole is sent. So that we have,

10/100 * Total = Total.

3.2 System Architecture:

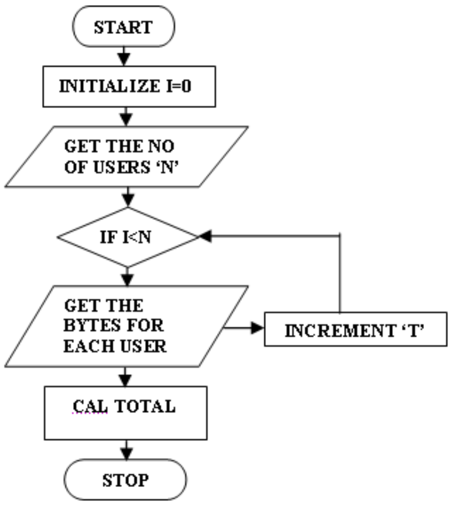

Figure : 4 Input Module

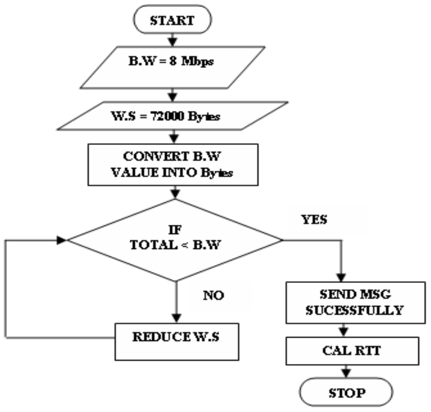

Fig:5 Window Adjustment Module

3.3 Bandwidth Estimation:

The bandwidth is estimated by the rate of returning acknowledgements. Measuring maximum data throughput rate of communication and network access. The bandwidth is used to compute congestion window and slow start threshold. Sender uses ACK reception time and the amount of data recently delivered to the destination. [4]. TCP bandwidth estimation formula

B.W (in Mbps) = RCV buffer size(in bytes) / RTT(in ms)

Ex:(64Kbyte*8bit)/0.17=3011764bps=3Mbps.

Here,

(RTT =170ms)

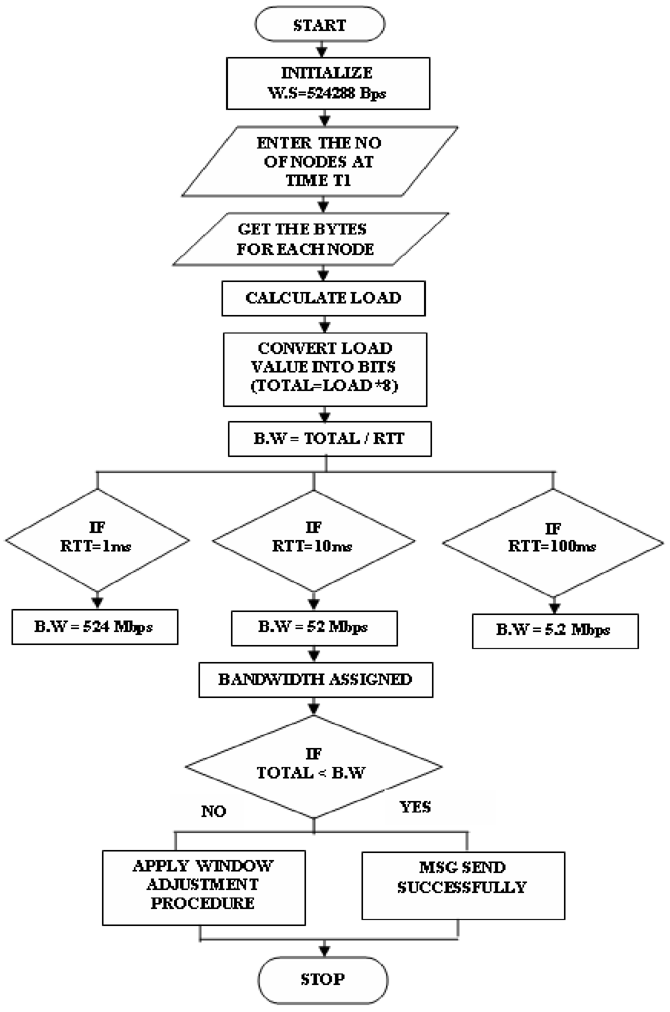

Figure :6 Bandwidth Estimation

TABLE 1

CALCULATION OF RTT

CAPACITY IN Mbps	**CONV E RT Mbps**	**WINDOW SIZE * 10^3**	**ROUND TRIP TIME IN millisecond**
10	10240	15	0.0018
15	15360	20	0.0024
20	20480	25	0.0031
25	25600	30	0.0036
30	30720	35	0.0043
35	35840	40	0.0048
40	40960	45	0.0054
45	46080	50	0.0061
50	51200	55	0.0067

4, STIMULATION RESULTS

The performance in TCP is improved using Window Adjustment procedure and Bandwidth Estimation. The Simulation results shows maximum throughput, minimum delay, high utilization. The simulation results are run in NS2.

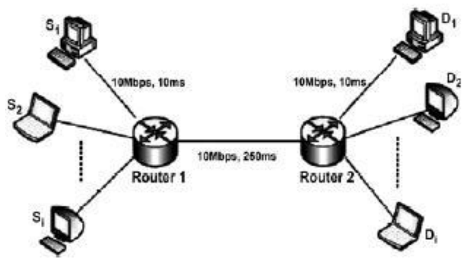

Figure :7 A Simple Network Configuration for NS-2 Simulation.

5, CONCLUSION

In this paper, window adjustment procedure and bandwidth estimation in TCP congestion control is proposed. In Window adjustment procedure, window size gets reduced to 10% in every round trip time, when the sender's total packet exceeds its receiver side bandwidth. The bandwidth is estimated by the rate of returning acknowledgements. Through the simulations, the proposed scheme is shown to have better performance than the existing scheme.

REFERENCES

[1]. Davide Astuti "Improving TCP Performance in High Speed Networks" Research seminar on Internet protocols.

[2] Brakmo, L. S., and Peterson, L. L., 1995, "TCP Vegas: End to End Congestion Avoidance on a Global Internet," IEEEJournal on Selected Areas in Communications, Vol. 13, No. 8, pp. 1465-1480.

[3]. Floyd, S., and Jacobson, V., 1993, "Random Early Detection Gateways for Congestion avoidance", IEEE /ACM Transactions on Networking, Vol. 1, No. 4, pp. 397-413.

[4]. Gerla, M., Sanadidi, M. Y., Wang, Ren, Zanella, A., Casetti, C., and Mascolo, S., 2001, "TCP Westwood: congestion window control using bandwidth estimation," IEEE GLOBECOM '01, Vol.3, pp. 1698-1702.

[5]. Jin, Guojun "Packet Drop Avoidance for High-speed Network Transmission Protocol", Distributed Systems Department Lawrence Berkeley National Laboratory Berkeley, CA 94720.

[6]. Jacobson, V., 1988, "Congestion Avoidance and Control," in Proceedings of ACM SIGCOMM, pp.314-329, Stanford, CA, USA.

[7]. K. Satyanarayan Reddy and Lokanatha C. Reddy, "A Survey on Congestion Control Mechanisms in High Speed Networks", IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.1, January 2008.

BIOGRAPHY

R.Navaneethakrishnan Assistant Professor (SG) from Bharathiyar College of Engineering and Technology, Karaikal, India