Performance Evaluation of TCP Congestion Control for Reliable Data Transmission in TCP

[1]Sowjanya H S, [2]Mrs. Pushpalatha S
[1]PG Student, [2]Assistant Professor
[1]Department of Digital Electronics and Communication system
[1]VTU PG Center, Mysuru, Karnataka, India

ABSTRACT
The transmission control protocol TCP is then protocol that is responsible for the reliable data transmission in the TCP/IP standard. Therefore, it has to recover the packets lost via data propagation and retransmit these packets after three duplicate acknowledge. Wait for three duplicate acknowledge could increase the total delay in the network. This work aims to investigate shorter retransmission time. It was proposed that two duplicate acknowledge and the NS2 program was used to simulate the transmission process in the network, taking into account all the factors that affect the transmission. From this project, it has been found that retransmitted lost packets by least time happen when just two duplicate acknowledges were used, and this time rises up when three duplicate acknowledges were used

Keywords: TCP, Congestion Avoidance, E_New Reno.

I. INTRODUCTION

TCP is a vital component of the Transport layer of the Internet protocol suite. It is intended to provide connection oriented reliable service over an underlying unreliable network. In the past, TCP detected the wrong things inside the network, such as packet loss, network congestion, etc., by using only the "timeout" mechanism. After sending data packet, TCP sets up its own timer particularly for the sent packet. The timer is usually set to the retransmission timeout period (RTO) which is determined by some other algorithms. If TCP correctly receives an ACK corresponding to the data packet before the timer is expired, TCP assumes that everything inside the network is fine. TCP, then, automatically resets the timer of just received ACK packet and continuously waits

The Transmission Control Protocol (TCP) is the most dominant protocol used to deliver data reliably over the Internet. The TCP is often described as a byte stream, connection-oriented and reliable delivery transport layer protocol. It guarantees the delivery of data supplied from an application to the other end reliably, although the underlying layers (IP) offer only unreliable data delivery. This reliability is achieved by assigning a sequence number for each transmitted packet and expecting a positive acknowledge (ACK) to come back from the receiver. This acknowledges maintains the next data expected to be received by the receiver. The sequence number is used to reorder the packets which may arrive at the receiver either out of order, duplicated or corrupted.. The role of TCP is then to provide End-to-End mechanisms to ensure that data is reliably delivered.

II. RELATED WORK

New Reno algorithm, in this algorithm the congestion window was set to half of its value based on state network not packet loss but reordered in this algorithm performed by more than three duplicate acknowledges as expressed in Fig. 1 Additional modification performed on New Reno to improve performance of TCP unfortunately this inefficient in utilization of link capacity and unfair in throughput. To minimize delay in TCP by adjusting congestion window based on the status of network, they were using Round trip time to define the status of network and so, based on this specify size of congestion window.

Since transmission control protocol "TCP" ensures reliable data transmission but experience significant delays. When packet loss must be retransmitted after three duplicate acknowledges. This waiting may be caused delay in the network. The proposed mechanism effort to minimize this problem by modifying E_New Reno algorithm. The basic idea is to minimize number of duplicate acknowledges that were required to perform retransmission process for lost packets in TCP.

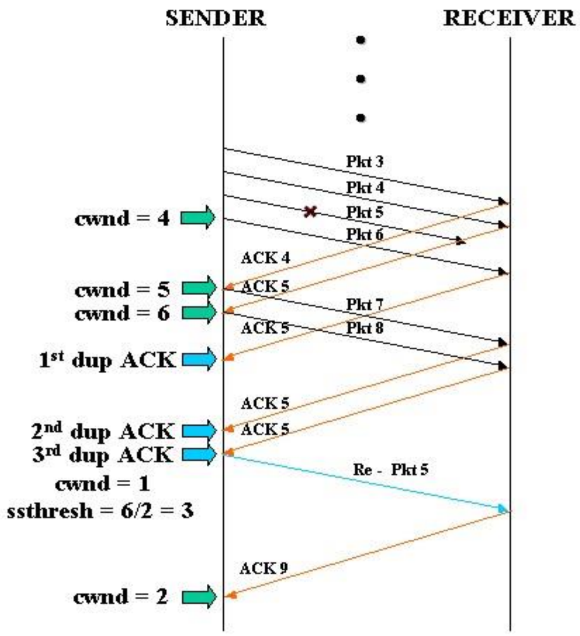
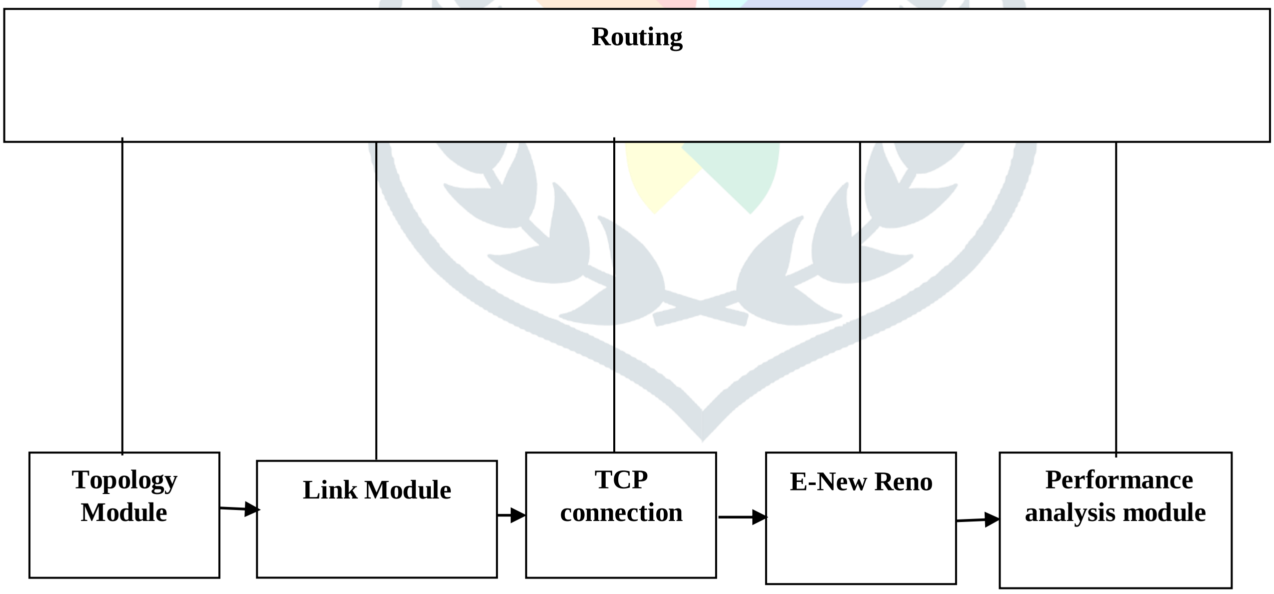


Fig. 1: Retransmission in New Reno algorithm

III. PROPOSED SYSTEM

The transmission control protocol "TCP" ensures reliable data transmission but experience significant delays. When packet loss must be retransmitted after three duplicate acknowledges. This waiting may be caused delay in the network. The proposed mechanism effort to minimize this problem by modifying E_New Reno algorithm. The basic idea is to minimize number of duplicate acknowledges that were required to perform retransmission process for lost packets in TCP and using congestion avoidance algorithm. The key idea of the enhanced congestion avoidance algorithm is to adjust the congestion window at the TCP sender dynamically according to the available connection capacity at any time.The project is divided into five main modules as shown below.



A. Basic idea

Using two nodes of network on ns2 program one as sender, another as receiver, then connection between them was established for sending some packet with sequence number, when the receiver receives packet and sends acknowledge for that sender who sends this packet. This acknowledge indicates to request for sending next packet, from this request the sender understands that packet was received(normal situation).Once some packets that are expected not arrived then the receiver concludes something loss, and sends (message) acknowledge, to replay from sender sending these packets(lost packets) at that time other side ’’sender’’ continuous sending next packets.. This message will be sent two times forming two duplicate acknowledges ,after receiving second duplicate acknowledge TCP at the sending side retransmit the missing packet without waiting for the timer to be expired. This sending process is based on applying E_New Reno algorithm by adjusting the TCP parameters.

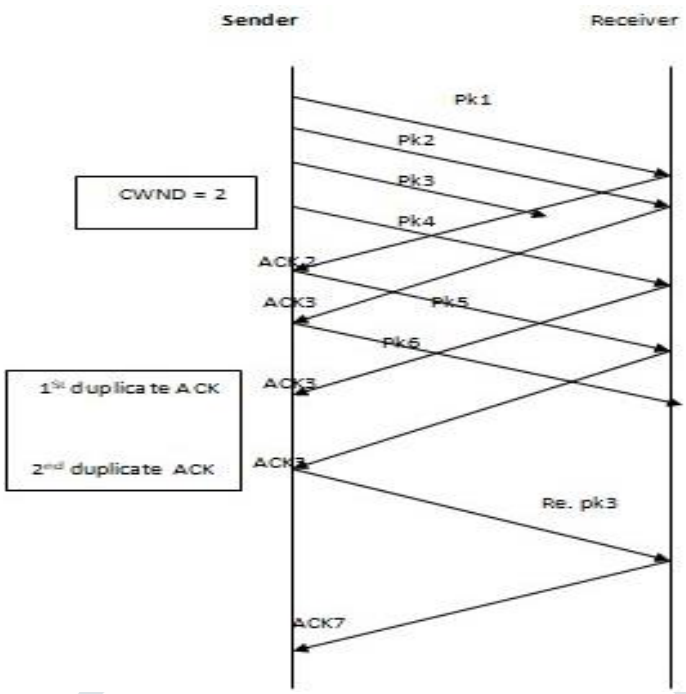


Fig.2: Retransmission in E-New Reno algorithm

In this mechanism network was configured on ns2, it consisted of eight nodes, there were three nodes act like senders and reminder of the nodes act receivers. Node zero and two formed main sender, five and six formed main receiver in our network, but nodes three and four formed source and sink respectively. Any constant bit rate can be sent between them based on UDP protocol that helps to create congestion in the network followed by lost packets as illustrate in Fig.3.

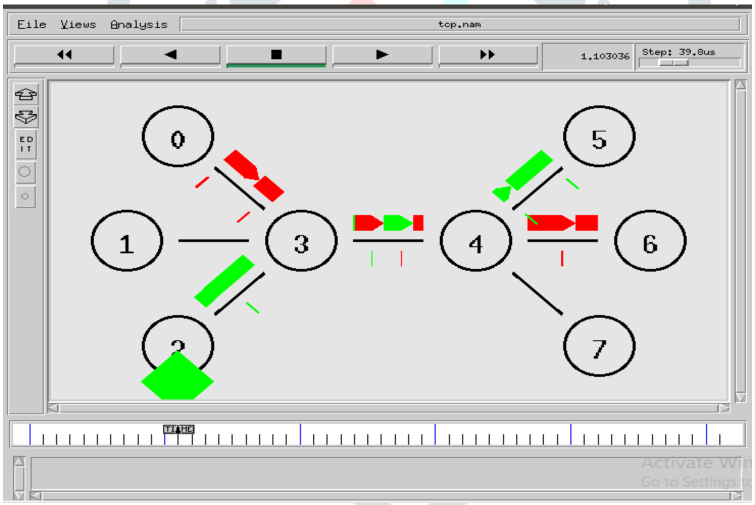


Fig. 3: Simulation Scenario

B. E_New Reno

The principle idea of this mechanism is to adjust congestion window, and hence the transmission rate, at the TCP sender dynamically based on the network status. In other words, the size of the congestion window is dynamically calculated based on the level of congestion in the network instead of updating (increasing/decreasing) the congestion window blindly regardless the current load at the network. The main goal is to maintain the "right" amount of extra data in the network to improve the overall network performance. Obviously, if a connection is sending too much extra data, it will cause congestion; if it's sending too little extra data, it cannot respond rapidly enough to transient increase in the available bandwidth.

C. Congestion Avoidance Algorithm

During congestion avoidance phase, most of the current congestion control mechanisms "blindly" increase the congestion window size linearly by constant value as long as no losses are detected. The key idea of the enhanced congestion avoidance algorithm is to adjust the congestion window at the TCP sender dynamically according to the available connection capacity at any time. In other words, the strategy is to adjust the source's sending rate based on the network load. The increasing rate of the cwnd goes on until an indication for congestion accurance is reached. At this point, the transmission rate should be slow down to resolve the network congestion. Congestion window used in receiver side of the TCP to avoid the duplicate acknowledgements this will reduce retransmission.

IV. RESULTS

Four nodes are used of network on NS2 two as sender and another two as receiver to sending packets with increasing in data rate by 0.5 Mbps, when congestion happen in the network some packets lost, in this case these packets should be retransmitted after two duplicate acknowledges and measure the throughput, packet delivery ratio and number of packets received for this network. From these results, it has been found that throughput for E_New Reno better than that with Reno. With increasing number of packets sent in the network and compared to other methods, E_New Reno has the lowest drop rate

V. EXPERIMENTAL RESULTS

A. **THROGHPUT**: throughput is known as the number of bits that the destination has successfully received. Expressed in kilobits per second (kbps). Throughput measures a routing protocol's efficiency in receiving data packets by destination. This result below shows the throughput for proposed system.

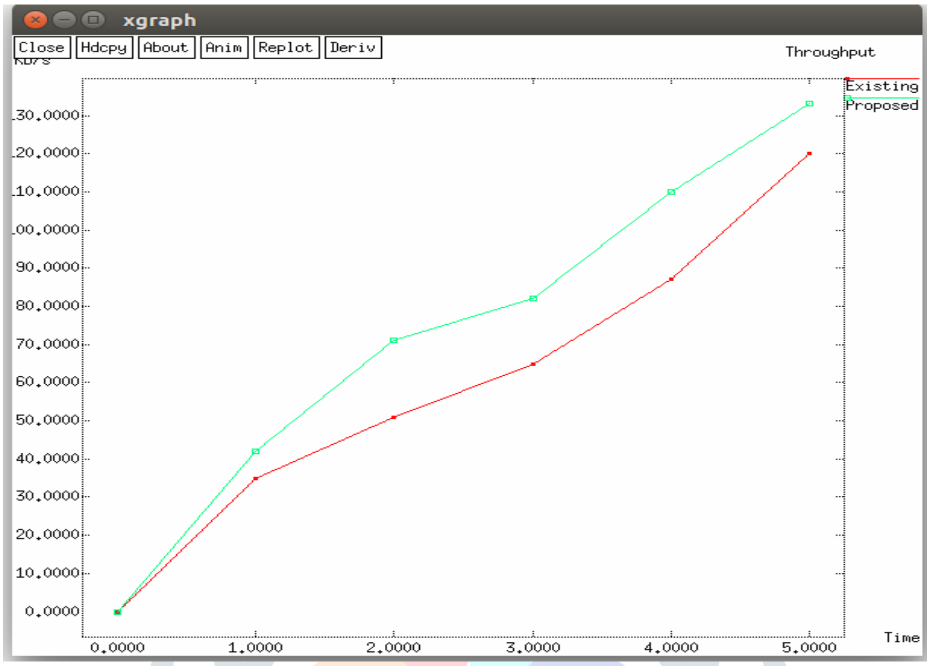


Figure.1: Graph of throughput

B. **PACKET DELIVERY RATIO**: it means the ratio of the data packet that were delivered to the destination node to the data packets that were generated by the source. This metric shows a routing protocol's quality in its delivery of the data from source to destination. This result below shows the packet delivery ratio for proposed system.

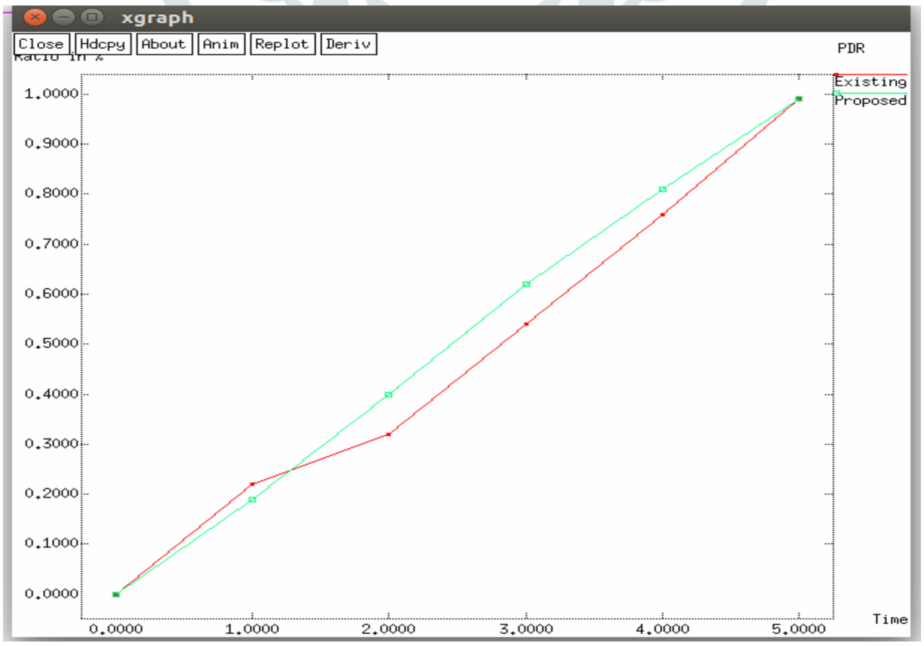


Figure.2: Graph of packet delivery ratio

C. NUMBER OF PACKET RECEIVED: it means that number packets delivered to the receiver. This result below shows the number of packets received for proposed system.

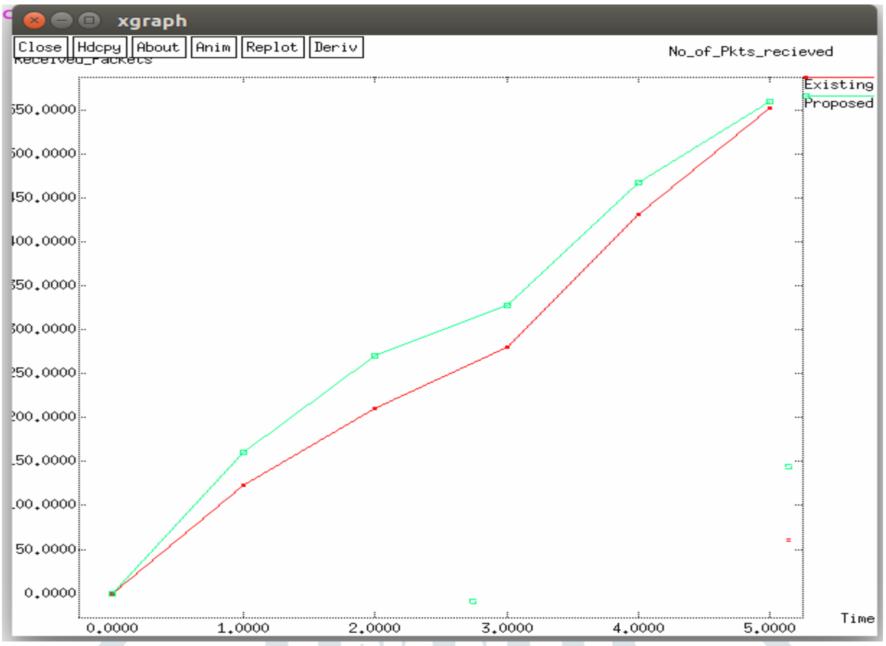


Figure.3: Graph of number of packets received

VI. CONCLUSION

To meet requirement of this project, NS2 is used as a simulator to emulate the normal transmission process, which is done by number of nodes. The sender sends packets to receiver, and receives ACK, when one of these packets was lost for any reason, the receiver send two ACKs stating that the packet was not received then the sender immediately resend the missing packet via E new Reno algorithm and repeats these steps for each lost packet in the network. The main idea in making two duplicate acknowledges instead of three duplicate acknowledges reducing the time required to retransmit the lost packets. A way must be found to improve general throughput in this mechanism.

REFERENCES

[1] N. Parvez, A. Mahanti, and C. Williamson, "TCP NewReno: Slow-but- Steady or Impatient?" IEEE International Communications Conference, Vol.3 (2), pp. 716-722, June 2006.

[2] Dong Lin and H.T. Kung, “TCP Fast Recovery Strategies: Analysis and Improvements”, Harvard University Cambridge, MA 02138 USA.

[3] Hanaa A. Torkey, Gamal M. Attiya and I. Z. Morsi, "Performance Evaluation of End-to-End Congestion Control Protocols," Menoufia journal of Electronic Engineering Research (MJEER), Vol. 18, no. 2, pp. 99118, July 2008.

[4] MOHAMMED ASSAAD & Djamal Zeghlache, “TCP performance over UMTS-HSDPA System”, Auerbach Publications Taylor & Francis Group, Boca Raton New York.

[5] Hanna Torkey, Gamal Attiya, Ibrahim Z. Morsi, “Modified Fast Recovery Algorithm for Performance Enhancement of TCP-New Reno “*International Journal of Computer Applications (0975 – 8887) Volume 40– No.12,* February 2012.

[6] Guangjie Han, Maode Ma, “Connecting sensor networks with IP using a Configurable tiny TCP/IP protocol stack”, in Proceedings of the 6th International Conference on Information, Communications & Signal Processing, 10-13 Dec. 2007, pp. 1-5.

[7] Zhou, Chuan-Sheng, Chong, Fu, “Implementation of a General Reduced TCP/IP Protocol Stack for Embedded Web Server”, in Proceedings of the 3rd International Conference on Intelligent Information Hiding and Multimedia Signal Processing, IIHMSP 2007, 26-28 Nov. 2007, Vol. 2, pp. 377-380.

[8] Ri-Kun Liao, Yue-Feng Ji, Hui Li, “Optimized Design and Implementation of TCP/IP Software Architecture Based on Embedded System”, in Proceedings of the Fifth International Conference on Machine Learning and Cybernetics, Dalian, 13-16 August 2006, pp. 590-594.

[9] L. A. Grieco and S. Mascolo, “Performance evaluation and comparison of Westwood+, New Reno and Vegas TCP congestion control”, ACM Computer Communication Review, April 2004, Vol. 34(2).

[10] Cheng Peng Fu, Liew, S.C., “TCP Veno: TCP enhancement for transmission over wireless access networks”, IEEE Journal on Selected Areas in Communications, Vol. 21, Issue 2, Feb. 2003, pp. 216-228.

[11] A. Dunkels, “Full TCP/IP for 8-bit architectures”, in Proceedings of the First International Conference on Mobile Systems Applications and Services (MobiSys-03), San Francisco, CA, USA, 2003, pp. 85- 98.

[12] A .Dunkels, T. Voigt, J. Alonso, H. Ritter, and J. Schiller, “Connecting Wireless Sensornets with TCP/IP Networks”, in Proceedings of the 2nd International Conference on Wired/Wireless Internet Communications, February, 2004.