Performance Enhancement Of TCP For Wireless Network

GJCST Classification (FOR)
C.2.5, C.2.1

[1]Pranab Kumar Dhar, [2]Mohammad Ibrahim Khan,[3]P.M. Mahmudul Hassan

***Abstract*-Transmission Control Protocol (TCP) is one of the core protocols of the Internet Protocol Suite which provides reliable, ordered delivery of stream of bytes from a program on one computer to another program on another computer. TCP assumes congestion which is the primary cause of packet loss and uses congestion control mechanisms such as Tahoe, Reno and New Reno to overcome this congestion in wireless network. These TCP variants take longer time to detect and recover packet loss. In order to improve retransmission scheme, we propose a modified version of New Reno that outperforms previous TCP variants because of utilizing faster retransmission scheme as well as transferring more packets to the destination.**

Keywords-TCP/IP, Wireless Network, Transport protocol, Internet, Congestion Control.

I. Introduction

Today Internet is different from a single network. Because many parts in the world have different topologies, bandwidths, delays, packet sizes, and other parameters. TCP is a connection-oriented packet transfer protocol that ensures communication between two hosts. The idea behind this reliability and ordered packet delivery is that the sender does not send a packet unless it has received acknowledgment from the receiver that the previous packet or group of packets which already sent has been received [1-2]. Because of the good performance of TCP, most networks of current traffic use this transport service. It is used by the applications such as telnet, World Wide Web (www), ftp (file transfer protocol), and e-mail [3-4].Because of wide use of Internet as well as the widespread use of TCP by the majority of the network applications, it should be needed to improve the congestion detection and avoidance mechanism of TCP. Starting from the series congestion collapse on October 1986, many researchers developed and implemented different versions of TCP such as TCP Tahoe and TCP Reno and TCP New Reno [2]. In this paper, the mechanisms used by TCP Reno and TCP New Reno for controlling congestion on the network, are studied and analyzed using a Network Simulator known as NS2 [7-9]. From the simulation results we observe that TCP New Reno provides better performance than TCP Reno.Thus, in order to enhance the performance of TCP during network congestion, we propose a modified version of New Reno.The rest of the paper is organized as follows. Section II presents the background information regarding TCPvariants. Section III introduces our proposed modified New Reno. Section IV summarizes and discusses the experimental results and compares the performance of our Proposed New Reno with Reno and New Reno. Finally, section V concludes this paper.

Corresponding author[1]-Assistant Professor Department of Computer Science and Engineering Chittagong University of Engineering and Technology (CUET) Chittagong-4349, Bangladesh Cell: +88-01818-000112
About[2&3]- Department of Computer Science and Engineering, Chittagong University of Engineering and Technology (CUET), Chittagong-4349, Bangladesh

II. TCP Variants

TCP is a component of TCP/IP Internet protocol suite. However, it is definitely considered as an independent, general purpose protocol since it can be used by other delivery systems.TCPprotocol makes very minor assumptions about the underlying network, for example it is possible to employ TCP over single network just like an Ethernet or even over complex networks such as the global Internet [5]. It is the dominant transport protocol over both wired and wireless links [6]. At the end of 1980s, a congestion control algorithm was proposed by Van Jacobson which is known as TCP Tahoe and today‘s Internet stability is mainly based on it. The first modified version of TCP Tahoe was the TCP Reno and then followed by other flavors or variants. The design of congestion control algorithm developed by Van Jacobson is based on the end-to-end principle and has been fairly successful from keeping the Internet away from congestion collapse. The following is a list that shows some of the TCP flavors or variants.

1) Tahoe TCP

The Tahoe TCP algorithm includes Slow-Start, Congestion Avoidance and Fast Retransmission. This algorithmincludes a modification to the round-trip time estimator usedto set retransmission timeout values. This is notsuitable for high band-width product links because it takes acomplete timeout interval to detect a packet loss and in fact, in most implementations it takes even longer time becauseof the coarse grain timeout.

2) Reno TCP

Reno improves the performance of Tahoe by introducing a Fast Recovery Phase. This phase activates after fast retransmission when three duplicate packets are lost. The parameters are initialized as follows:
Slow start threshold = 1/2 Congestion window;
Congestion window = Slow start threshold;
The Reno TCP works as follows:
(i) Slow start threshold and congestion window are both resized to reduce the transmission rate and drain the network congestion. In addition, we do not need to begin from slow start again.

(ii) Until a _non-duplicate packet' is received, a temporary congestion window is used during Fast Recoveryand isincreased by one message for every new received duplicatepacket; this allows new packet transmissions during Fast Recovery operation.
(iii) When the sender receives acknowledgment about the retransmission of the lost packets which are received successfully, the Fast Recoveryphase is terminated. Then CongestionAvoidancephase is initialized and congestion windowsize starts to grow from its updated congestion window size.But the problem with TCP Reno is that if the drop packets are multiple then the first information about the packet loss comes when receiver receives the duplicate acknowledgments. But the information about the second packet which is lost will come only when the sender receives the acknowledgment for the retransmitted first packet after one Round Trip Time (RTT). Thus, it does not provide good performance when multiple packets are dropped from a single window of data [6].

3) New Reno TCP

To overcome the limitations of TCP Reno, New Reno has been introduced. New Reno can be able to detectmultiple packet losses. For this reason, it is muchmore efficient than Reno in case ofmultiple packet losses.Like Reno, New Reno also entersinto fast retransmission process when it receives multiple duplicate acknowledgments. However, it differs from Reno when it does not recover fast until the acknowledgement is received. Thus, it overcomes the problem of Reno by reducing the congestion window in multiples times.The fast-transmit phase of New Reno is similar as Reno. The difference is that New Reno allows multiple re-transmissions in the fast recovery phase.When New Reno enters in fast recovery phase it calculates the maximum outstanding segment. The fast-recovery phase proceeds like Reno, however when a fresh acknowledgement is received, it considers two cases:
(i) If it sends acknowledgement to all outstanding packets, in that case it exits from Fast Recovery phase and sets congestion window to slow start threshold and continues to process congestion avoidance like Tahoe.
(ii) If the acknowledgement is partial in that case, it indicates that the next packet that is in line has lost and it re-transmits that packet again and sets the number of received duplicate acknowledgement to zero.
It exits from fast recovery stage when it sends acknowledgement to all the data available in the window.
The main problem of New Reno is that it takes one RTT to detect each packet loss. When the acknowledgement for the first retransmitted packet is received, after then we can detect the other lost packets.

III. Proposed Modified New Reno TCP

Proposed modified New Reno extends the retransmission mechanism of New Reno. It keeps track the packet transmission and it also estimates the RTT by calculating the time needed to get acknowledgment from the receiver. When a duplicate acknowledgment is received it calculates the time difference between current time and packet transmission time. If it is greater than RTT, then it immediately retransmit the packet without waiting for three duplicate acknowledgments or a coarse timeout. In order to recover multiple packets drop, the modified New Reno records the highest sequence number of the packet in a single window before retransmit the lost packet. If acknowledgment of the retransmitted packet does not cover the highest sequence number of the packet in a single window, then retransmit the indicated packet again. The remaining procedures of New Reno are unchanged in Modified New Reno.

IV. Simulation Results and Discussion

1) Design of Simulation Structure

In this section, we design the general process of the simulation including the topology of the Network and the simulation program to be used. Figure 1 shows the over-all simulation process.

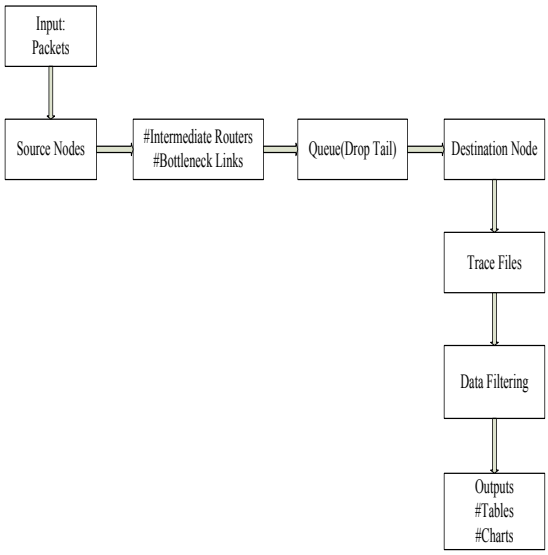

Fig. 1. Block diagram of the over-all process of the simulation

2) Design of Network Topology

The three algorithms of TCP congestion control mechanisms such as TCP Reno, TCP New Reno, and TCP Modified New Reno are represented and evaluated using the following network topology.

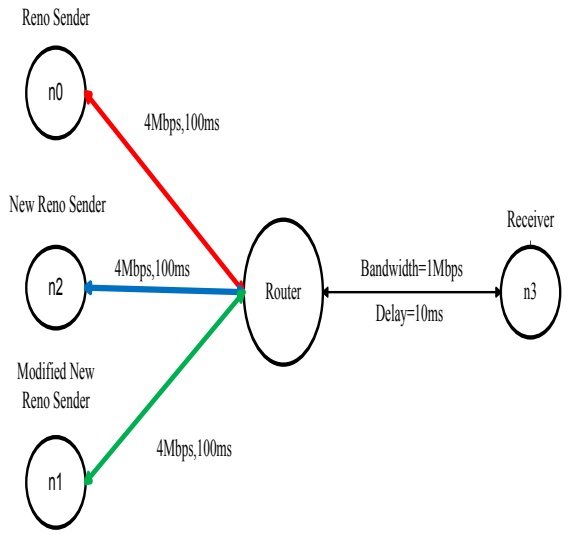

Fig. 2. Topology layout used for simulation

Using the above topology layout, three different sources of TCP are allowed to send FTP traffic to the destination. Each of the TCP source has a connection link of 4Mbps, (Mega bits per second), bandwidth and 10ms, (milli-seconds), delay to its nearest router, and Router. The bottleneck link from Router to node 3 (n3) has a bandwidth of 1Mbps and 10ms delay time. The bottleneck is suitable for evaluating the performance of the algorithms for congestion control and congestion avoidance mechanisms.

As all the sources should be pass through a Router, the queue size is bounded to a limit of 4(not fixed), or above and Drop-Tail queuing mechanism decides which packets will be discarded. Each TCP source is allowed to send FTP traffic with packet size of 460 bytes and the maximum queue size is 30 packets for the duration of 30 sec. Such traffic setup is suitable for collecting data from simulation on a chosen interval over a given bandwidth configuration.

In this paper, simulation results are measured and analyzed using some performance metrics such as number of received packets, number of acknowledgements, number of dropped packets, and throughput.In this simulation, the total packet size is 460 bytes and the maximum queue limit is 30 and also the duration of total simulation is 30 sec. Now if we increase the queue limit more than 30, then more packets will be waited in queue and delay of packet transmission will be increased. This is because we can not transmit all packets to the receiver with in 30 sec. In order to obtain better output we should keep queue limit as small as possible.From Figure 3, we observed that when queue limit is 4 our proposed Modified New Reno obtains the maximum number of received packets. If the queue limit is small, all TCP variants including our proposed New Reno also drop packets as shown in Figure 4. In our proposed New Reno, we have used a new retransmission mechanism that quickly recovers the lost packets, and almost all packets are transmitted to the receiver with in 30 sec. Thus, our proposed New Reno achieves higher number of received packets than Reno and New Reno. When queue limit is 8 the time delay of packet transmission is increased than previous queue limit.

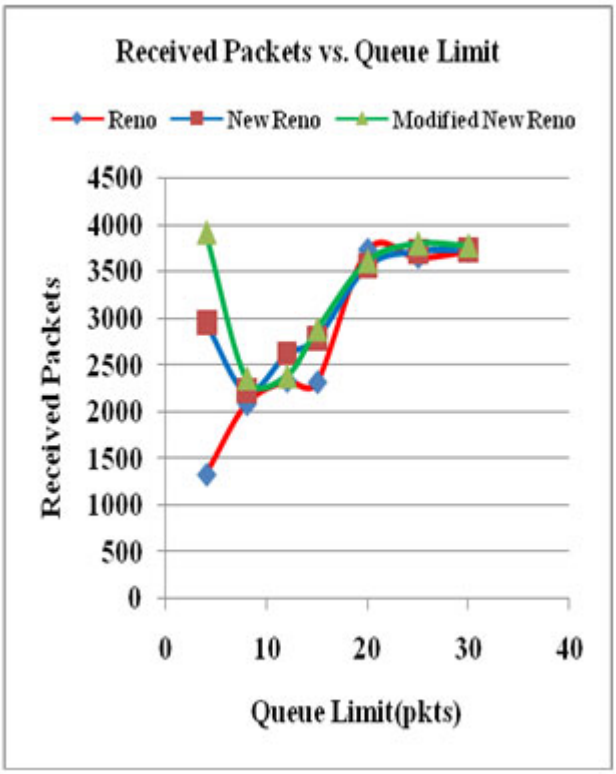

Fig. 3. Received packets vs. Queue limit

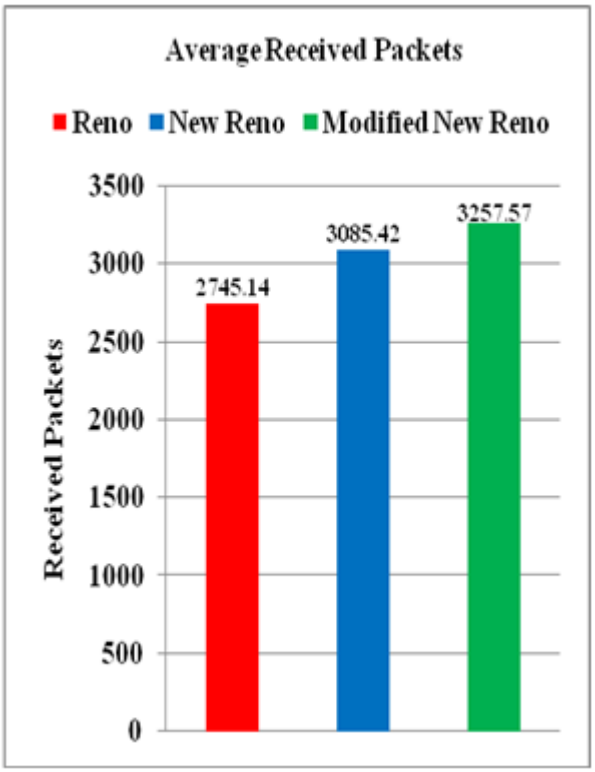

Fig. 4. Comparison of average received packets

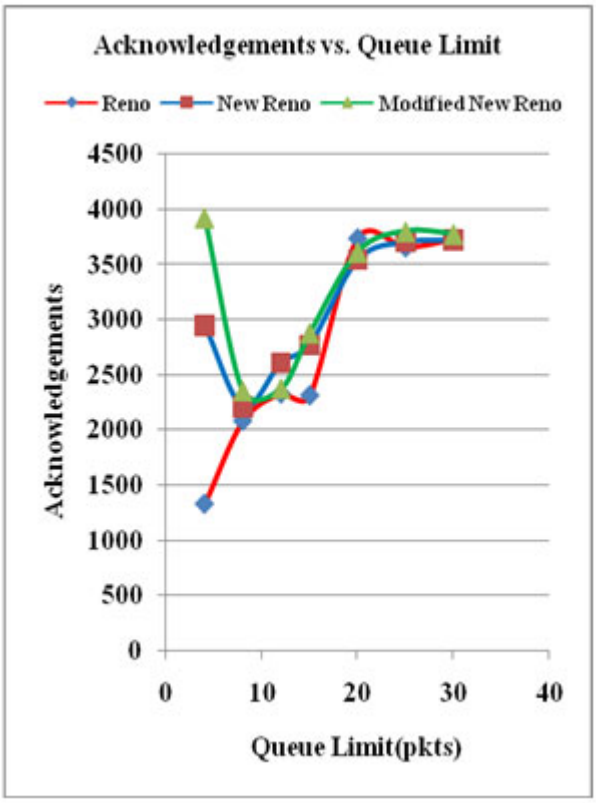

Fig. 5. Acknowledgements vs. Queue limit

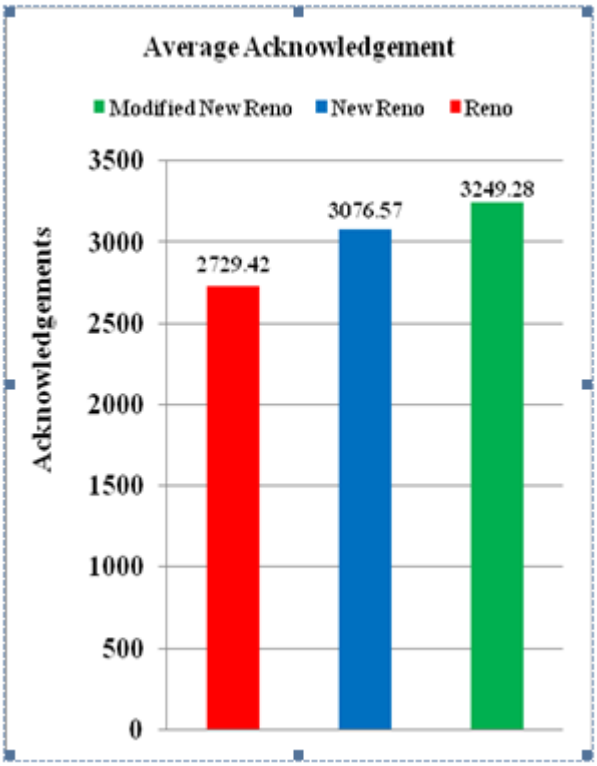

Fig. 6. Comparison of average acknowledgement

Our proposed system has faster retransmission mechanism; however, it has higher delay time for packet retransmission which causes network congestion as like Reno and New Reno. For this reason all packets can not be transmitted to the destination within 30 sec. Thus we achieve lowest number of received packets. When queue limit is 12, New Reno achieves higher output because of its steady retransmission scheme. When queue limit is 15, our proposed New Reno achieves higher number of received packets. When queue limit is 20 as compared to total packet size of 460 bytes, the delay of packet transmission in queue is increased and the three TCP variants have almost same number of received packets. For queue limit of 25, the three TCP variants have all most same output but our proposed New Reno achieves a little bit higher output than previous TCP variants and these outputs will be continued for higher queue limits.

Thus, we can say that if we use large queue limit then the delay of packet transmission from queue is increased. For this reason, we cannot transmit all packets to receiver with in 30 sec as compared to queue limit of 4 in which we can transmit maximum number of packets. Thus, we should limit queue size as small as possible to obtain better output.

From Figure 5 it is seen that acknowledgements send by receiver at different queue limits using Modified New Reno is greater than Reno and New Reno. Figure 6 shows that average acknowledgements send by receiver using Modified New Reno is higher than Reno and New Reno.Figures 7 and 8 show that the number of dropped packets using Modified New Reno is less than Reno and New Reno. If we consider the throughput (Good-put) of Reno, New Reno and Modified New Reno as shown in Figure 9, we observe that the throughput at different queue limits using Modified New Reno is better than Reno and New Reno. If we calculate the average throughput as shown in Figure 10, we also observe that Modified New Reno has better average throughput than Reno and New Reno.Thus, we can conclude that our proposed Modified TCP New Reno speeds up the performance of TCP by improving the end-to-end throughput.

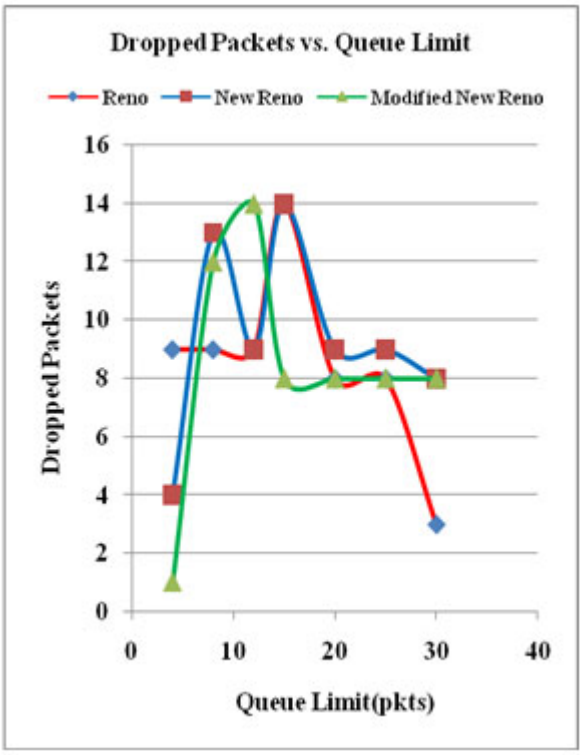

Fig. 7. Queue limit vs. Dropped packets

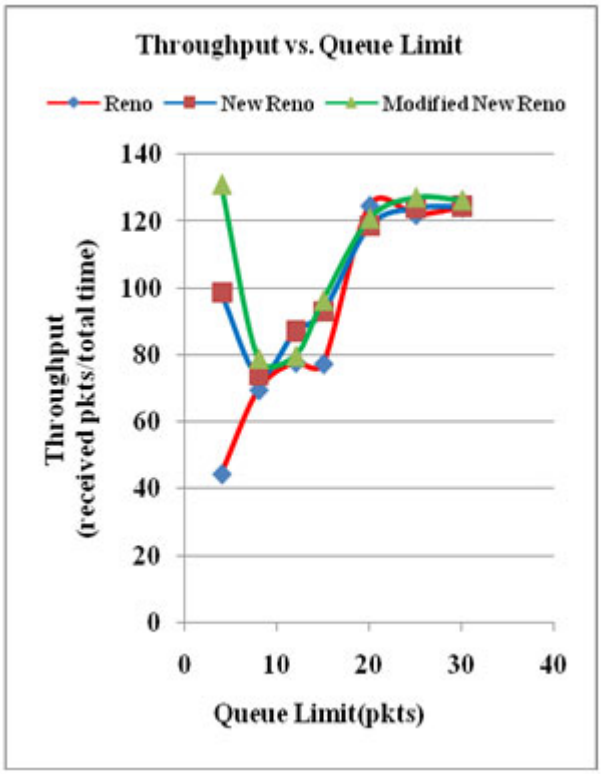

Fig. 9. Throughput vs. Queue limit

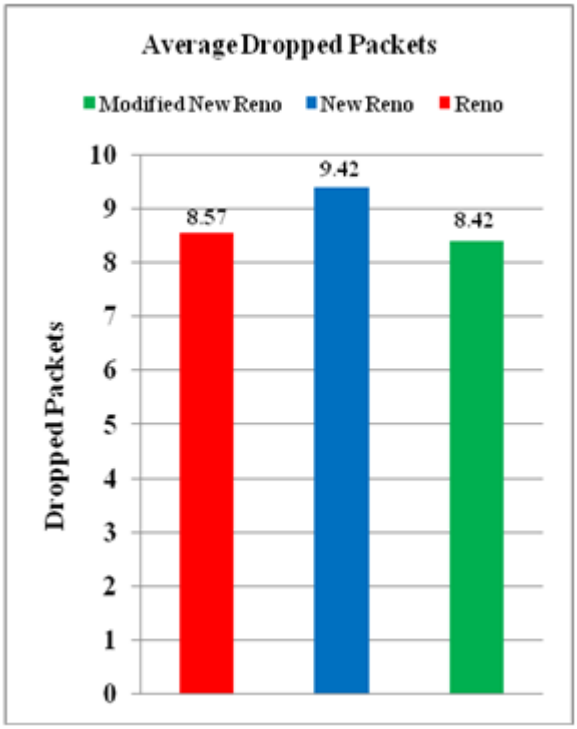

Fig. 8. Comparison of average dropped packets

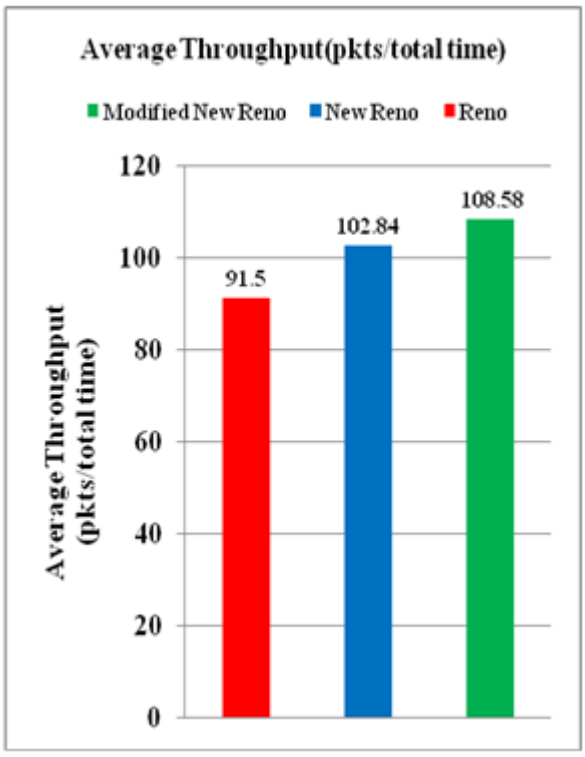

Fig. 10. Comparison of average throughput

V. Conclusion

In this paper, we have presented a modified version of New Reno to improve the TCP performance in wireless network when congestion occurs. Simulation results indicate that Modified New Reno outperforms Reno and New Reno in terms of received packets, dropped packets and throughput. This is because our proposed scheme does not have to always wait for 3 duplicate acknowledgements. For this reason, it can retransmit quickly and does not reduce the window size too much as like Reno. In addition,it prevents many of the coarse grained timeouts of New Reno as it does not need to wait for 3 duplicate acknowledgments before it retransmits a lost packet. Finally, we can say that our proposed New Reno can be a suitable candidate for TCP congestion control.

VI. References

1) Chuck Semeria, "Supporting Differentiated Service Classes: TCP Congestion Control Mechanisms," Juniper Networks, 2000.
2) Van Jacobson and Michael J. Karels, "Congestion Avoidance and Control," November, 1988.
3) Transmission Control Protocol (TCP) http://www.erg.abdn.ac.uk/users/gorry/course/inet-pages/tcp.html
4) TransmissionControlProtocol.http://en.wikipedia.org/wiki/Transmission Control_Protocol.
5) D.E.Commer, Internetworking TCP/IP: Principles, Protocols, and Architecture, Upper saddle, New Jersey: Prentice Hall, 2006.
6) M. Ghaderi —TCP-Aware Resource Allocation in CDMA Networks" In Proc. ACM MobiCom, Los Angeles, USA. 2006.
7) Network Simulator (NS-2) web site: http://www-mash. cs.berkeley.edu/ns
8) The Network Simulator (NS-2) January 6, 2009.
9) Thenetworksimulator-ns-2, http://www.isi.edu/nsnam/ns