Estimating Performance of TCP Alternatives in Wireless Environment

Neha Bathla
Assistant Professor,
Department of Computer Sc. & Engg. Yamuna Institute of Engg. & Tech., Gadholi, Yamuna Nagar.
nehabathla1985@gmail.com
Amanpreet Kaur
Assistant Professor,
Department of Computer Sc. & Engg., Yamuna Institute of Engg. & Tech., Gadholi, Yamuna Nagar
aman_preet_k@yahoo.co.in
Gurpreet Singh
Dean Academics, Head (CSE/IT)
Department of Computer Sc. & Engg., Yamuna Institute of Engg. & Tech., Gadholi, Yamuna Nagar.
gps_ynr@yahoo.com

----------ABSTRACT----------
Today's wireless internet or MANETs has become popular in these years due to the vast growth in the number of mobile computing devices and high demand for continuous network connectivity in spite of physical locations. In this paper, with the help of simulator we will study the effects of TCP variants in AODV environment which is a Multihop wireless network and compare their performance on different parameters like throughput, number of packets send, number of packets dropped, delivery ratio, average delay, average jitter using NS2.

Keywords: TCP Reno, New Reno, Sack, Vegas, NS2.

I. INTRODUCTION

TCP (Transmission Control Protocol) is used for a reliable end-to-end byte stream over an unreliable internet environment. The major responsibilities handled by TCP are firstly, it divides the message into the number of packets and on the other end reassembling the packets into complete message that IP manages on the network.TCP works on the fourth layer of OSI model Today's Internet traffic mostly uses TCP for many applications like HTTP (for Web Browsing), FTP (for file transfer) or SMTP (for Electronic Mail Transfer). TCP was designed to provide a secure and reliable transfer of information over an unreliable network. For reliable transport services, TCP users must establish a connection-oriented session with one another. Connection establishment is performed by using a "three-way handshake" mechanism. In three way handshake firstly, the connection establishment secondly the data transfer and in end connection termination. Three-way handshake synchronizes both ends of a connection by allowing both sides of users to agree upon initial sequence numbers. This mechanism also ensures that both sides are ready to transmit data and know that the other side is ready to transmit as well. This is necessary because packets are not transmitted or retransmitted during connection establishment or after session termination.

MANET is an infrastructure-less network of mobile devices that are connected without wires. MANET or mobile Ad hoc network is a collection of mobile nodes that are dynamically and randomly located where interconnections between nodes are able of changing on a frequent basis. Each mobile device independently in a wireless network is freely to move in any direction. MANETs consist of a self configuring, self-forming, self-healing network rather than a mesh network that has a central controller. In MANETs, in order to alleviate communication within the network, a routing protocol is used to discover routes between nodes. MANETs basically are a kind of wireless Adhoc network that uses a routable networking environment. After the route is establish, either connection oriented protocol (TCP) or connection less protocol (UDP) is necessary to transfer the actual data packets. Due to its reliable transfer of information, TCP and its variants play a important role in data transfer over wireless network. Though similar studies have been carried out earlier but this paper provides a concise view of the comparative performance of four TCP variants over AODV routing protocol.

II. OVERVIEW OF ADHOC ON DEMAND DISTANCE VECTOR ROUTING PROTOCOL

The AODV protocol is the most widely adopted and well known reactive routing protocol that the routes are created only when they are needed [18]. The mobile devices or nodes in the network exchange the routing packets between them when they want to communicate with each other and maintain only these established routes. AODV defines three message types-RREQs, RREPs, RERRs, RREQ messages are used to initiate the route finding process. RREP messages are used to finalize the routes. RERR messages are used to notify the network of a link breakage in an active route. Every node maintains a route table entry which updates the route expiry time [3][2]

III. TCP CONGESTION CONTROL ALGORITHMS

Four Congestion Control Algorithms**:** These algorithms are defined in [6] and [13]. Slow Start, Fast Retransmit, Fast Recovery, Congestion Avoidance.

Slow Start: One of the algorithms used in TCP congestion control is slow start is used to control the congestion inside the network. This algorithm is based on the idea that the size of congestion window (cwnd) starts with one maximum segment size (MSS). It is also known as the exponential growth phase. During the exponential growth phase, slow-start works by increasing the congestion window exponentially means each time the segment is send its corresponding acknowledgment is received. It increases the window size by the number of segments acknowledged. This happens until either an acknowledgment is not received for some segment or a predetermined threshold value is reached. If a loss event occurs, TCP assumes that it is due to network congestion and takes steps to reduce the load on the network. Once the threshold has been reached, TCP enters the linear growth (congestion avoidance) phase. At this point, the window is increased by 1 segment for each RTT (round trip time). This happens until a loss event occurs. Although the strategy is referred to as "Slow-Start", its congestion window growth is quite aggressive, more aggressive than the congestion avoidance phase [6].

Fast recovery: In this algorithm when the congestion occurs on the network the congestion window (cwnd) size must be decreased. The only way to guess that congestion has occurred on the network is by the need to retransmit a segment. Retransmission occurs by one of the two cases when three duplicate acknowledgements are received or when a timer times out. In both cases the window size of the threshold is dropped to one half. If three dup. ACKs are received there is weaker possibility of congestion on the network that segments have been lost, but some segment after that may have reached successfully. This process is called the fast recovery. In the Fast Recovery algorithm, during Congestion Avoidance mode, when packets (detected through 3 duplicate ACKS) are not received, the congestion window size is reduced to the slow-start threshold, rather than the smaller initial value.

Fast Retransmit: Another algorithm used in TCP is fast retransmission. In this algorithm, the fast retransmission occurs when three duplicate acknowledgements are acknowledged or when a timer times out. When a duplicate acknowledgement is received the sender does not wait for a retransmission timer to expire before transmitting the segment. This process is called fast retransmit algorithm [13].In fast retransmit stage, once TCP gets duplicate ACKs it adopts to resend the segment, where no waiting time is required for the segment timer to expire. This process will speed up the recovery of segment losses. In fast recovery, when the segment is lost, the TCP attempts to keep the existing flow rate without returning to slow-start. The fast retransmit mechanism was initially introduced in TCP Tahoe [13].

Congestion Avoidance: In this algorithm the congestion window increased additively rather than exponential increase. When the slow start reaches the threshold value the slow start stops and additive increase (congestion avoidance) phase begins. Congestion avoidance is used to slow down the transmission rate. However, there may be a point during Slow Start that the network is forced to drop one or more packets due to overload or congestion. In the Congestion Avoidance algorithm the reception of duplicate ACKs can implicitly signal the sender that network congestion occurs. The sender immediately sets its transmission window to one half of the current window size. If congestion was indicated by a timeout, the congestion window is reset to one segment, which automatically puts the sender into Slow Start mode. If congestion was indicated by duplicate ACKs, the Fast Retransmit and Fast Recovery algorithms are invoked. [13][10][11].As data is received during Congestion Avoidance, the congestion window is increased. However, Slow Start is only used up to the halfway point where congestion originally occurred. This halfway point was recorded earlier as the new transmission window. After this halfway point, the congestion window is increased by one segment for all segments in the transmission window that are acknowledged. This mechanism will force the sender to more slowly grow its transmission rate, as it will come close to the point where congestion had previously been detected [12].

IV. VARIANTS OF TCP

There are numbers of variants of TCP named as:

1. Reno
2. New Reno
3. Sack
4. Vegas

Reno: TCP Reno was suggested by Van Jacobson. During the transmission, when 3- duplicate acknowledgements received, it will halve the congestion window, perform a fast retransmit, and enters a phase called fast recovery. If a timeout event occurs, it will enter into the slow-start. TCP Reno was used to recover from a single packet loss, but it is not used when multiple packets are dropped from window of data [9].The drawback of TCP Reno is that it does not work well when multiple packets are lost from single window of data.

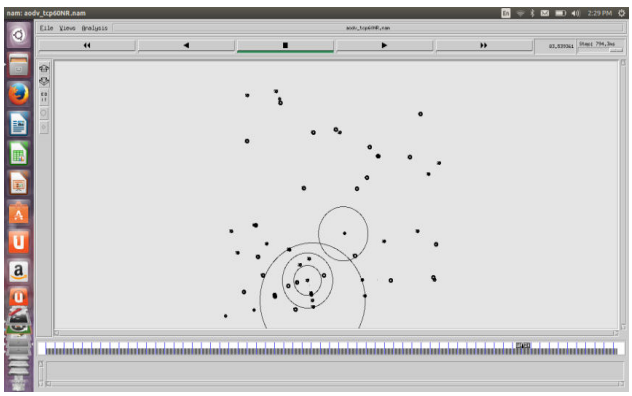
New Reno: TCP New Reno makes from the Reno + recovery of multiple packet loss.TCP New Reno modifies Fast Recovery algorithm and tries to improve the TCP Reno's performance when multiple packets are lost. In TCP New Reno, the ack. that arrives after retransmission (partial ack.) could indicate that a second loss occur.TCP New Reno works by assuming that the packet that immediately follows the partial ACK received at fast recovery is lost, and retransmit the packet. However, this might not be true and it affects the performance of TCP [5].

Sack: Sack requires that segments should acknowledge selectively [18]. Thus each ACK. has a block which describes which segments are being acknowledged. The main difference between SACK TCP and Reno TCP implementations is in the behavior when multiple packets are dropped from one window of data. SACK sender maintains the information which packets is missed at receiver and only retransmits these packets. When all the outstanding packets at the start of fast recovery are acknowledged, SACK exits fast recovery and enters congestion avoidance.

Vegas: TCP Vegas is a TCP congestion avoidance algorithm that emphasizes packet delay, rather than packet loss, as a signal to help determine the rate at which to send packets [14][15][18].TCP Vegas detects proactive measure to encounter congestion. It does not depend on solely on packet lost as a sign of congestion. It detects congestion before it occurs while Reno, New Reno detects congestion only after it has actually happened via packet loss. Three major changes by TCP Vegas are New Retransmission mechanism, Congestion Avoidance Modified Slow-Start. TCP Vegas is better than Reno because it doesn't wait for 3-duplicate packets so it can retransmit sooner. It is better from TCP New Reno in sense that its congestion avoidance mechanism to detect incipient congestion is very efficient and utilizes network resources very efficiently [16] [7] [8] [4].

V. SIMULATION ENVIRONMENT AND RESULTS ANALYSIS

In this simulation there is scenario in which we use 60, 80, 100 nodes in wireless networks. In this investigation we use routing protocol AODV (Adhoc on demand distance vector) with different types of TCP variants Reno, New Reno, Vegas, Sack based on ad-hoc wireless network of 60,80,100 nodes. The investigation involves the measurement of different parameters like Throughput, Delivery Ratio, Number of packet send, Number of packet dropped, Average delay, Average jitter of the network in each of the TCP variants. Finally the result achieved AODV routing protocol with TCP variants no of nodes in the network will be accessed.



We discussed the results of simulated scenario of TCP variants Reno, New Reno, Vegas, Sack with different parameters using Ns-2 simulator

Throughput: Throughput is defined as the total amount of data received by destination node from the source node divided by the total time it takes from the destination to get the last packet and it measures is bits per second (bit/s or bps).

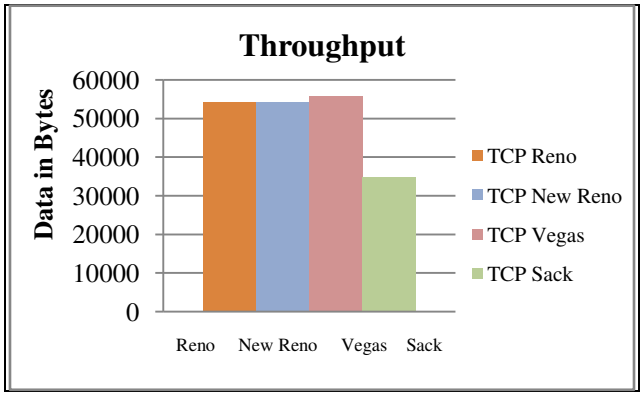


Figure 1: Throughput of 60 nodes

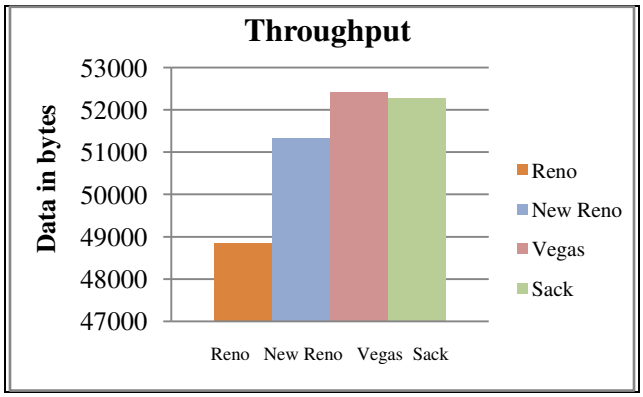


Figure 2: Throughput for 80 nodes

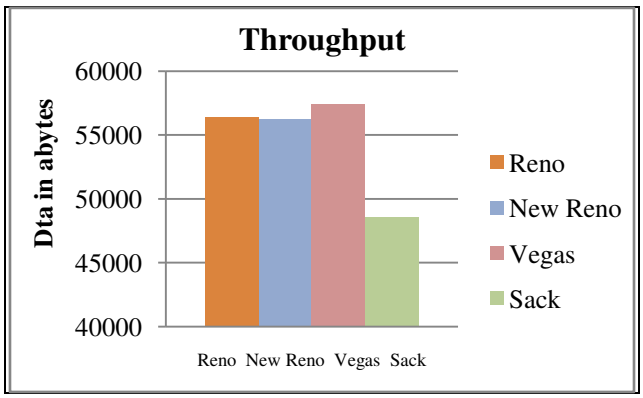


Figure 3: Throughput for 100 nodes

For calculating the throughput, various variants of TCP were made to execute for different number of nodes. The experiments were performed for 60, 80 and 100 number of nodes in wireless environment to conduct the readings from the generated trace files in NS-2. Figures 1, 2 and 3 show the throughput obtained from the topology of 60, 80, 100 nodes in

terms data in bytes. These shows that Vegas has better throughput as compared to others has low throughput value.

Number of packet send: Rate of transmission of packets.

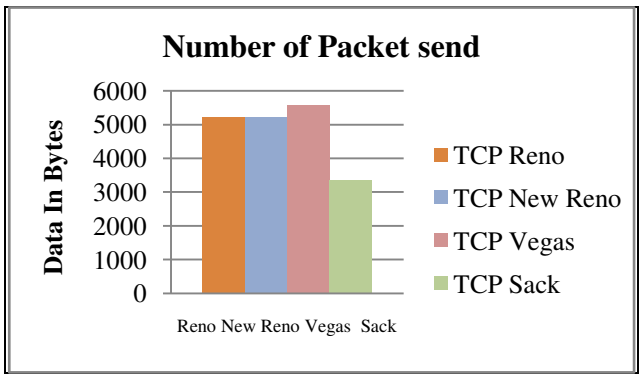


Figure 4: Number of packet send of 60 nodes

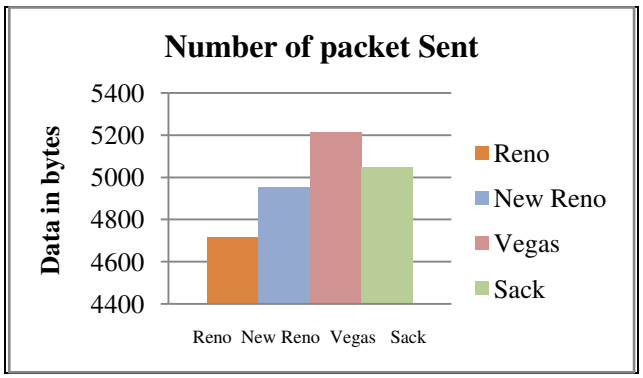


Figure 5: Number of packet send for 80 nodes

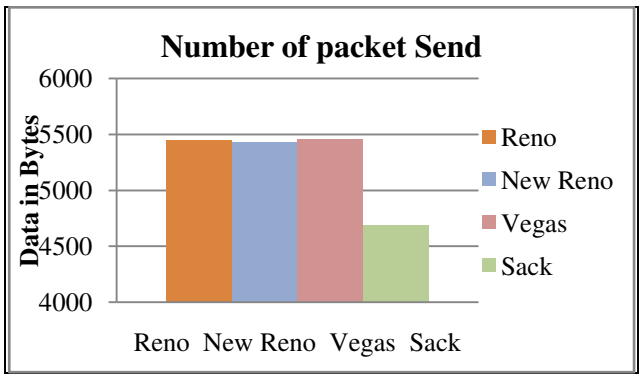


Figure 6: Number of packet send for 100 nodes

Fig 4, 5, 6 shows the number of packet send for 60, 80,100 nodes in terms of bytes. This shows that data rate of Vegas is better than other variants.

Number of packet Dropped: Failure of one or more transmitted packets to arrive at their destination.

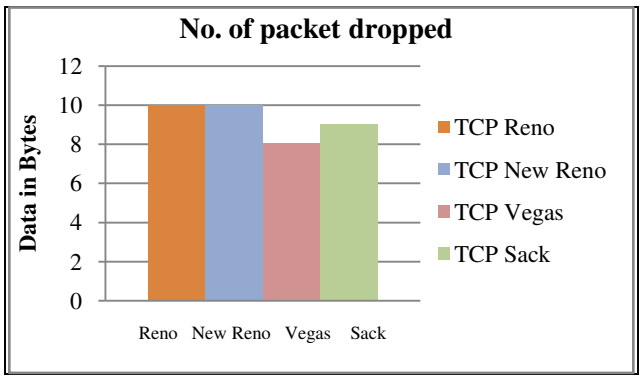


Figure 7: Number of packet dropped for 60 nodes

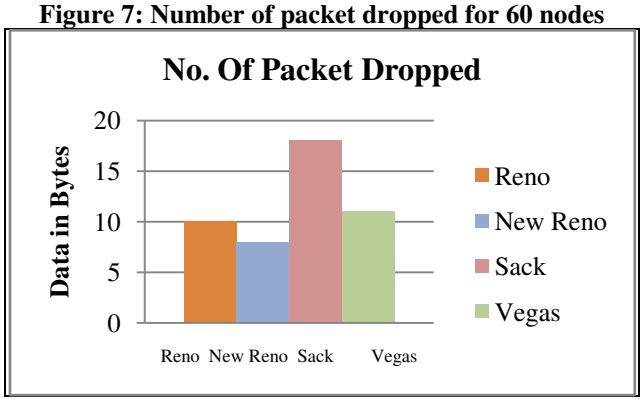


Figure 8: Number of packet dropped for 80 nodes

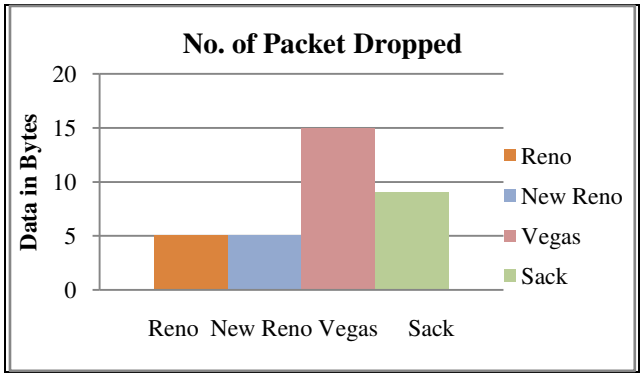


Figure 9: Number of packet dropped for 100 nodes

Fig 7, 8, 9 shows that number of packet dropped for 60, 80,100 nodes in terms of bytes. This shows that if the number of packet increased Vegas has more no. of packet dropped as compared to other variants.

Delivery Ratio: Packet delivery ratio is the ratio of total packets sent by the source node to the successfully received packets by the destination node.

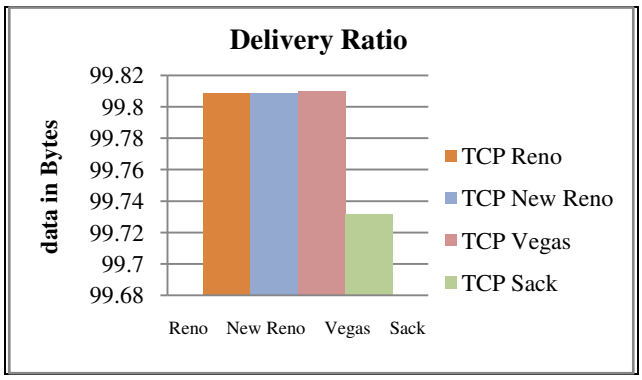


Figure 10: Delivery Ratio for 60 nodes

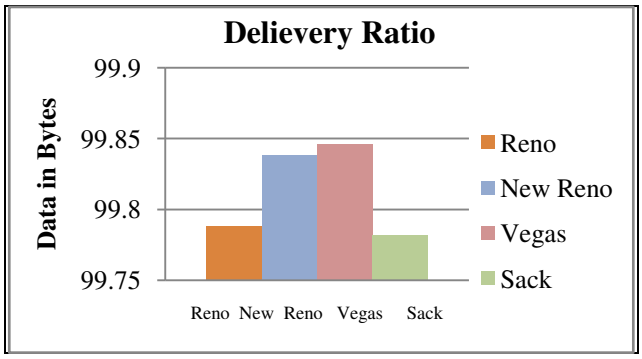


Figure 11: Delivery Ratio for 80 nodes

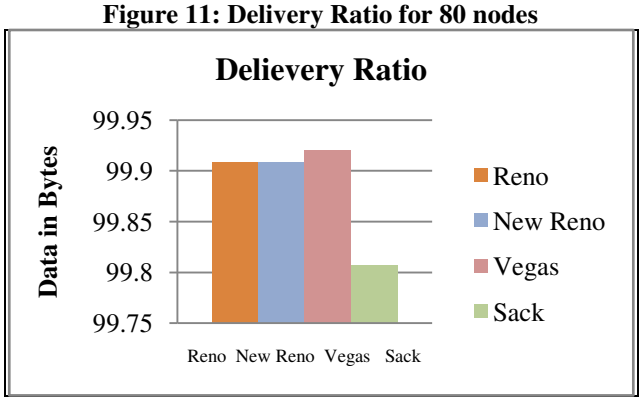


Figure 12: Delivery Ratio for 100 nodes

Fig 10, 11, 12 shows the delivery ratio of TCP in terms of bytes. This shows that TCP Vegas has better delivery ratio than other variants.

Average Delay: Average end-to-end delay is the time interval when a data packet generated from source node is completely received to the destination node.

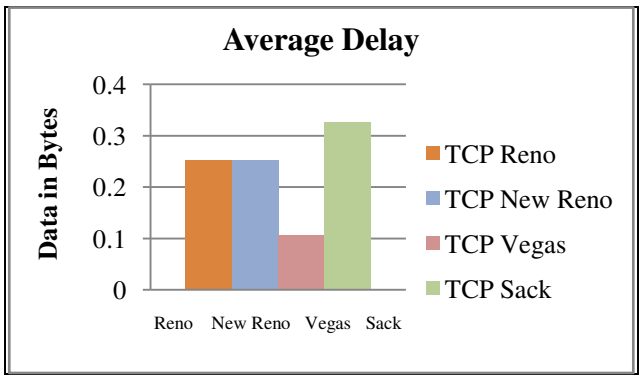


Figure 13: Average Delay for 60 nodes

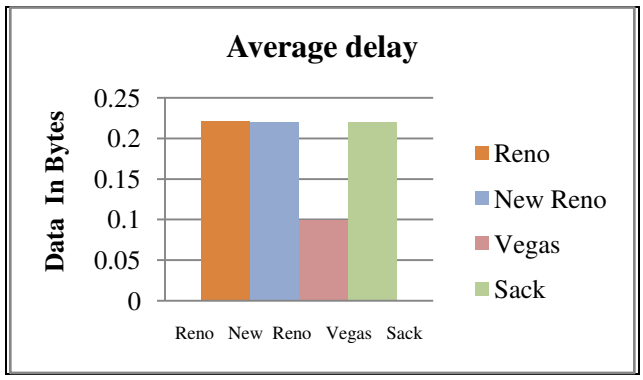


Figure 14: Average Delay for 80 nodes

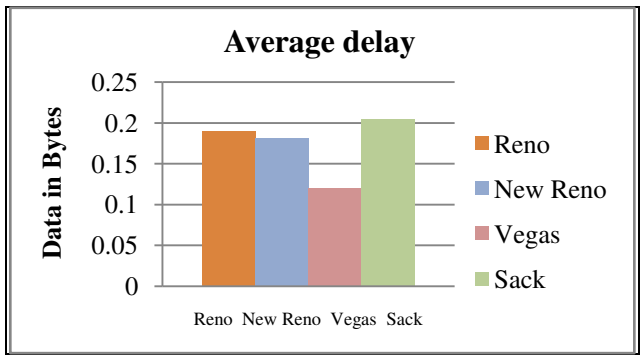


Figure 15: Average Delay for 100 nodes

Fig. 13, 14, 15 shows the average delay of all the TCP variants. This shows that the Vegas has less delay as compared to other variants.

Average Jitter: Jitter is the time variation between subsequent packet arrivals; it is caused by network congestion, timing drift, or route changes. It must be as low as possible for an efficient protocol.

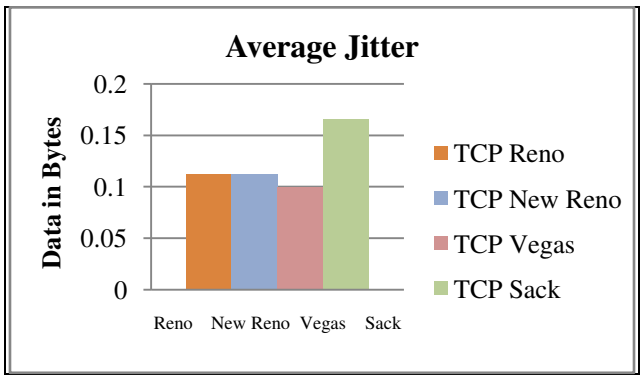


Figure 16: Average Jitter for 60 nodes

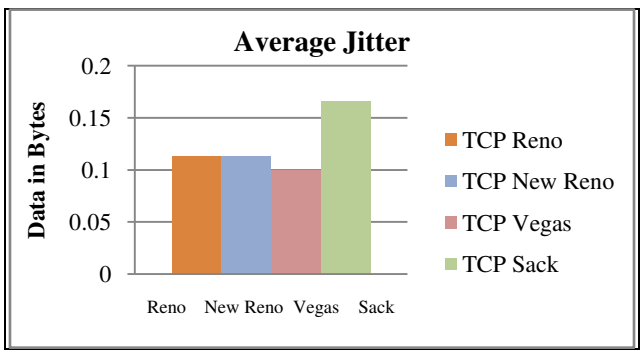


Figure 17: Average Jitter for 80 nodes

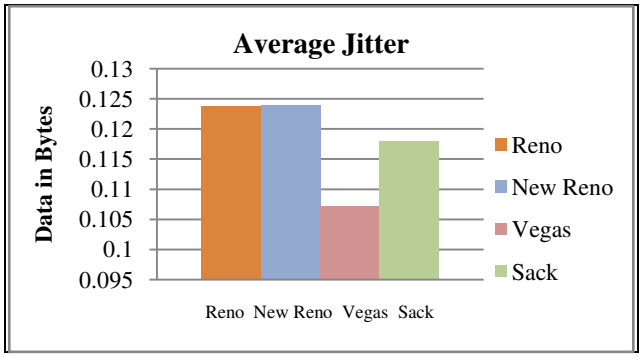


Figure 18: Average Jitter for 100 nodes

Fig. 16, 17, 18 shows the average jitter of all the TCP variants. This shows that the Vegas have less time variation as compared to other TCP variants.

VI. CONCLUSION

This paper compares the TCP variants performance using AODV routing protocol on NS2 simulator with different parameters Throughput, Number of packet send, Number of packet dropped, Delivery Ratio, Average Delay, Average Jitter. Simulation results shown through graphs represent overall performance of TCP variants with AODV routing protocol. From the obtained results shown by graphs, we can say that TCP Vegas shows highest efficiency and performs best.

REFERENCES

1. Yuvaraju B., N. Niranjan, N Chiplunkar "Scenario Based Performance Analysis of Variants of TCP using NS2-Simulator" International Journal of Advancements in Technology,1(2) 2010.
2 Sofiane Hamrioui, Jaime Lloret, Pascal Sofiane Hamrioui, Jaime Lloret, Pascal, "Performance in Mobile Ad hoc Networks", Volume5, Issue 4, 2013
3 C. Perkins, E. B. Royer and S. Das, "AdHoc On-Demand Distance Vector (AODV) routing", RFC 3561, IETF Network Working Group, July, 2003
4 Abed Ghassan,Ismail Mahamod, Jumari Kasmiran, "Characterization and observation of TCP-Vegas performance with different parameters over LTE(Long Term Evaluation) networks", Academic journals 2011.
5 Fall, K. and Floyd, S. "Simulation-based Comparisons of Tahoe, Reno and SACK TCP", ACM SIGCOMM Computer Communication Review 1996.
6 Haining Wang, Hongjie Xin, Douglas S. Reeves, Kang G, Shin, "A simple refinement of slow-start of TCP congestion control", Fifth IEEE symposium on Computers and Communications (ISCC-2000), Pages. 98-105, 03-06 July, 2000.
7 Larry L. Peterson, Lawrence Brakmo, Sean W. O'Malley, "TCP Vegas: New Techniques for Congestion Detection and Avoidance", supported in part by National Science Foundation Grant IRI-9015407 and ARPA Contract DABT63-91-C-0030, Pages 1-18, 16 February, 1994.
8 Rayadurgam Srikant "TCP-Vegas", The mathematics of Internet congestion control, Ed. 1, Pages 61-63, 2004, ISBN:0817632271
9. Gurpreet Singh, Dinesh Kumar, Amanpreet Kaur, "Simulation based comparison of performance metrics for various TCP extensions using NS-2", Proceedings of IEEE sponsored International Conference on Innovative Technologies (ICIT-09), organised by PDM College of engineering, pages 106-110, 18-19 June, 2009.
10. Amanpreet Kaur, Gurpreet Singh,Baninder Singh, "Evaluation of congestion control variants of TCP by Strolling propagation delay in NS-2",International journal for Applied Engineering and Research,5,pp.655-658, April, 2011.
11. V. Jacobson, "Congestion avoidance and Control", SIGCOMM - Symposium on Communication Architecture and protocols Volume 18 Issue 4 pp-314-329,August 1988
12 V. Jacobson "Modified TCP Congestion Control and Avoidance Algorithms". Technical Report, April 30, 1990.
13 W. Steven, "TCP slow start, congestion avoidance, Fast retransmit & fast recovery algorithm", IETFRFC, January 1997.
14 Z. Wang and J. Crowcroft, "A New Congestion Control Scheme: Slow Start and Search (Tri-S)". ACM Computer Communication Review, 21(1):32–43, Jan. 1991.
15 Qing Gao, Qinghe Yin, "Adaptive Vegas: A Solution of Unfairness Problem for TCP Vegas", Information networking: convergence in broadband and mobile networking: Springer, Volume 3391, pages 132-141, 2005.
16 Richard J. La, Jean Walrand, and Venkat Anantharam, "Issues in TCP Vegas", Technical report by department of Electrical Engineering and Computer Sciences, University of California at Berkeley, 13 July 1998.
17 Neha Bhatla, Amanpreet Kaur, Gurpreet Singh, "Congestion Control Techniques in TCP: A Critique", in the proceedings of 3rd National Conference of Advances and Research in Technology (ART-2014), Pages 45.1-45.5, 8-9 March, 2014.
18 Neha Batla, Amanpreet Kaur, Gurpreet Singh, "Relative Inspection of TCP Variants Reno, New Reno, Sack, Vegas in AODV", International Journal of Research in Engineering and Applied Sciences (IJREAS), Volume 4, Issue 5, pages 1-12, May 2014.