RELATIVE INSPECTION OF TCP VARIANTS RENO, NEW RENO, SACK, VEGAS IN AODV

Neha Bathla*

Amanpreet Kaur**

Gurpreet Singh***

ABSTRACT:

Wireless internet has become popular in these years due to the enormous growth in the number of mobile computing devices and high demand for uninterrupted n/w connectivity regardless of physical locations.In this paper, through 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, delivery ratio, number of packets send, number of packets dropped, average delay, average jitter using NS2.

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

*M.Tech. Scholar(CSE), Deptt. of Computer Sc.& Engg.Yamuna Institute of Engg. & Tech.Gadholi,Yamuna Nagar

**Assistant Professor(CSE), Deptt. of Computer Sc.& Engg.Yamuna Institute of Engg. & Tech.Gadholi,Yamuna Nagar

***Dean Academics & Head (CSE), Deptt. of Computer Sc.& Engg.Yamuna Institute of Engg. & Tech.Gadholi,Yamuna Nagar

INTRODUCTION:

TCP(Transmission Control Protocol) was designed to provide a reliable end-to-end byte stream over an unreliable internetwork. An internetwork differs from a single network because different parts may have wildly different topologies, delays,packet sizes and other parameters. TCP is also known as a connection-oriented protocol, which means that a connection is established and maintained until such time as the message or messages to be exchanged by the application programs at each end have been exchanged. TCP is responsible for ensuring that a message is divided into the packets that IP manages and for reassembling the packets back into the complete message at the other end. In the Open Systems Interconnection (OSI) communication model, TCP is in layer 4, the Transport Layer. Today's Internet traffic uses predominately TCP, as for applications like HTTP for Web Browsing, FTP for file transfer or SMTP for Electronic Mail Transfer. The performance perceived by users of these Internet applications depends largely on the performance of TCP.TCP provides a secure and reliable transfer of information. To use reliable transport services, TCP hosts must establish a connection-oriented session with one another. Connection establishment is performed by using a "three-way handshake"mechanism. Three-way handshake synchronizes both ends of a connection by allowing both sides to agree upon initial sequence numbers. This mechanism also guarantees that both sides are ready to transmit data and know that the other side is ready to transmit as well. This is necessary so that packets are not transmitted or retransmitted during session establishment or after session termination. Each host randomly chooses a sequence number used to track bytes within the stream it is sending and receiving. Therefore it is used by most of the existing Internet applications today and more than 90 percent of all data transfers use TCP.

MANET or mobile Ad hoc network is a collection of mobile nodes that are dynamically and arbitrarily located where interconnections between nodes are capable of changingon a continual basis. In order to facilitate communication within the network, a routing protocol is used to discover routes between nodes. After path establishment, either connection oriented protocol TCP or connection less protocol UDP is necessary to transfer the actual data packets. Due to its reliability, TCP and its variants play a crucial role in datatransfer over MANET. Though similar studies have been carried out earlier but this paper provides a succinct view of the comparative performance of four TCP variants over four different routing protocols.

Overview of simulated routing protocol:

AODV- Ad hoc On-Demand Distance Vector Routing protocol (AODV) is an on demand routing protocol. In this protocol to find a route to the destination, the source node floods the network withRouteRequest packets. The RREQ packets create temporary route entries for the reverse path through every node it passes in the network. When it reaches the destination aRREPLY is sent back through the same path the RREQ was transmitted. Every node maintains a route table entry which updates the route expiry time [3][2][9]

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:Slow-start algorithm is used to control congestion inside the network. It is also known as the exponential growth phase.During the exponential growth phase, slow-start works by increasing the TCP congestion window each time the 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 offered 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. 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]Before slow-start was introduced in TCP, the initial pre-congestion avoidance phase was even faster.

Fast recovery: There is a variation to the slow-start algorithm known as Fast Recovery, which uses fast retransmit followed by Congestion Avoidance. 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: When a duplicate acknowledgement is received the sender does not know if it is because a TCP segment was lost and received out of order at the receiver. If the receiver can reorder

the segments it should not be long before the receiver sends the latest expected acknowledgement. If more than two duplicate acknowledgement are received by the sender, it is a strong indication that at least one segment has been lost.TCP sender assume that the enough time has passed for all segments to be properly reordered by the fact that receiver has enough time to send three duplicate acknowledgements. When three or more duplicate acknowledgement are received the sender does not wait for a retransmission timer to expire before retransmitting 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: During the initial data transfer phase of a TCP connection the Slow Start algorithm is used. 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. If this happens, Congestion Avoidance is used to slow the transmission rate [17]. However, Slow Start is used in conjunction with Congestion Avoidance as the means to get the data transfer going again so it doesn't slow down and stay slow.In the Congestion Avoidance algorithm a retransmission timer expiring or the reception of duplicate ACKs can implicitly signal the sender that a network congestion situation is occurring. The sender immediately sets its transmission window to one half of the current window size (the minimum of the congestion window and the receiver's advertised window size), but to at least two segments. 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 FastRecovery 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 approach the point where congestion had previously been detected [12].

Variants of TCP: There are numbers of variants of TCP named as:

1.TCP Reno

2.TCP New Reno

3.Sack

4.Vegas

TCP Reno: When triple duplicate ACKs received, it will halve the congestion window, perform a fast retransmit, and enters fast recovery. If a timeout event occurs, it will enter slow-start, same as TCP Tahoe. TCP Reno is effective to recover from a single packet loss, but it still suffers from performance problems when multiple packets are dropped from a window of data[9].

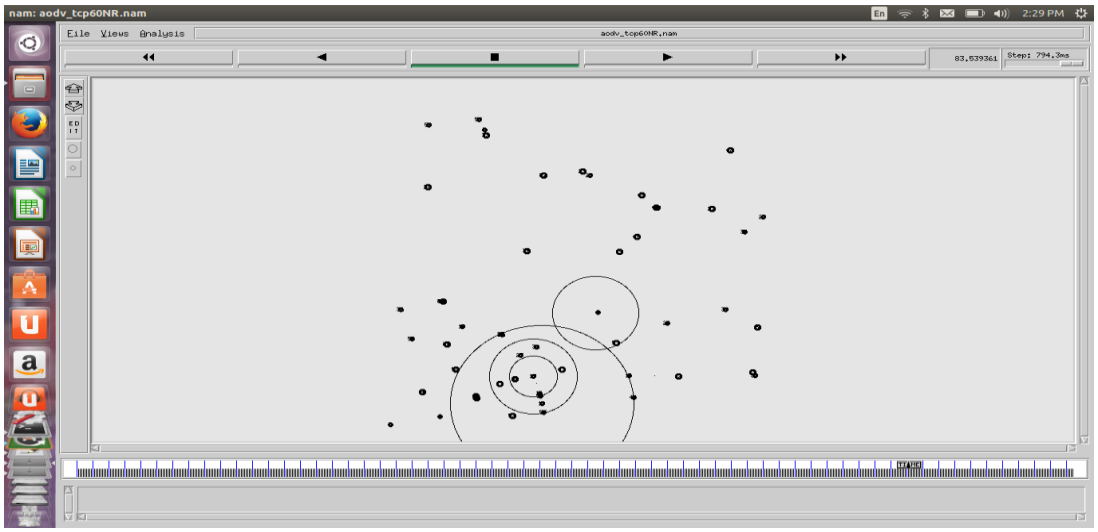
TCP New Reno: TCP NewReno modifies Fast Recovery algorithm tries to improve the TCP Reno's performance when a burst of packets are lost. In TCP NewReno, a new data ACK is not enough to take TCP out of fast recovery to congestion avoidance. Instead it requires all the packets outstanding at the start of the fast recovery period are acknowledged.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].

TCP Sack: TCP Sack adds a number of Sack blocks in the TCP packet, where each Sack block acknowledges a non-contiguous set of data has been received. 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 [17].

TCPVegas: TCP Vegas is a TCP congestion avoidancealgorithm that emphasizes packet delay, rather than packet loss, as a signal to help determine the rate at which to send packets[14][15].TCP Vegas detects congestion at an incipient stage based on increasing Round-Trip Time (RTT) values of the packets in the connection unlike other flavors like Reno, NewReno, etc., which detect congestion only after it has actually happened via packet drops. The algorithm depends heavily on accurate calculation of the Base RTT value. If it is too small then throughput of the connection willbe less than the bandwidth available while if the value is too large then it will overrun the connection. A lot of research is going on regarding the fairness provided by the linear increase/decrease mechanism for congestion control in Vegas. One interesting caveat is when Vegas is inter-operated with other versions like Reno. In this case, performance of Vegas degrades because

Vegas reduces its sending rate before Reno as it detects congestion early and hence gives greater bandwidth to co-existing TCP Reno flows[16][17][8][4].

Simulation Environment and Results Analysis: In thissimulation there is scenario in which we use 60 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 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

Table 1: Tabular Representation of TCP Variants

	TCP Reno	TCP NewReno	TCP Vegas	TCP Sack
Throughput	54159.2	54159.2	55880	34710.4
No.of packet send	5223	5223	5588	3351
No.of packet dropped	10	10	8	9

Delivery Ratio	99.80	99.808	99.810	99.73
Average Delay	0.2520	0.1052	0.105	0.3263
Average Jitter	0.1123	0.1123	0.099	0.165

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).

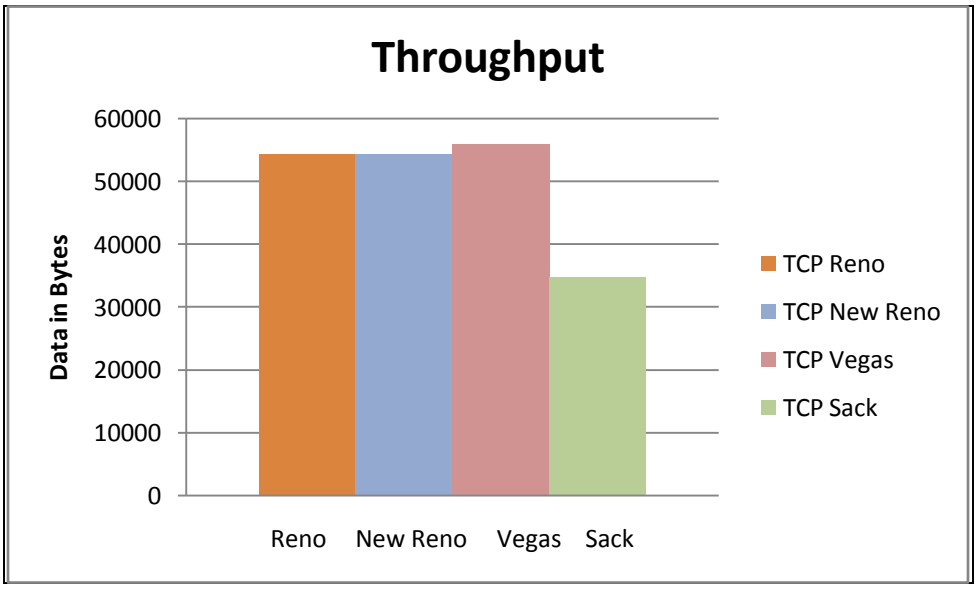


Figure 1: Throughput

Fig 1.shows the throughput in terms data in bytes.This shows that Vegas has better throughput as compared to others has low throughput value.

Number ofpacket send: Rate of transmission of packets.

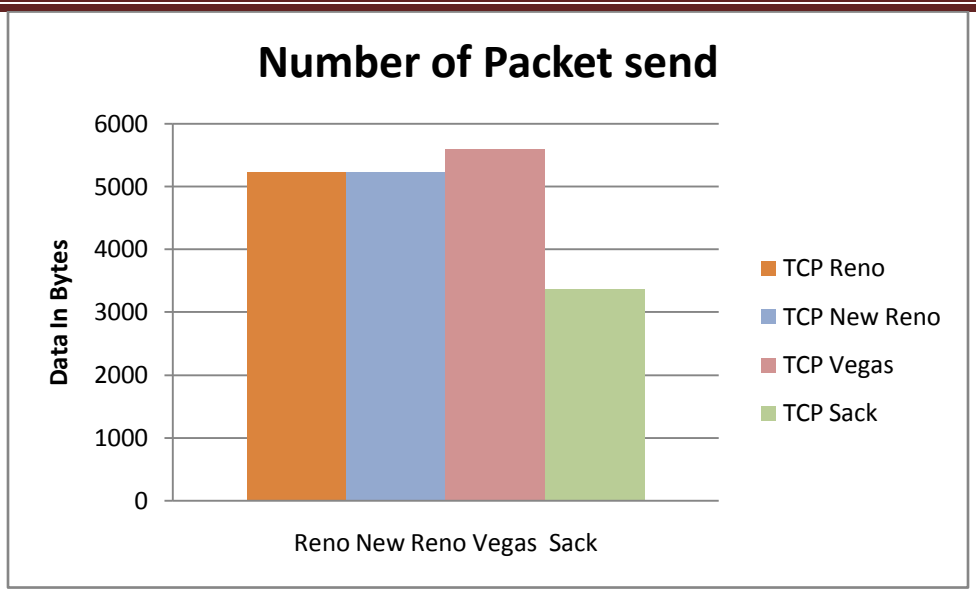


Figure 2:Number of packet send

Fig 2 shows the number of packet send 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.

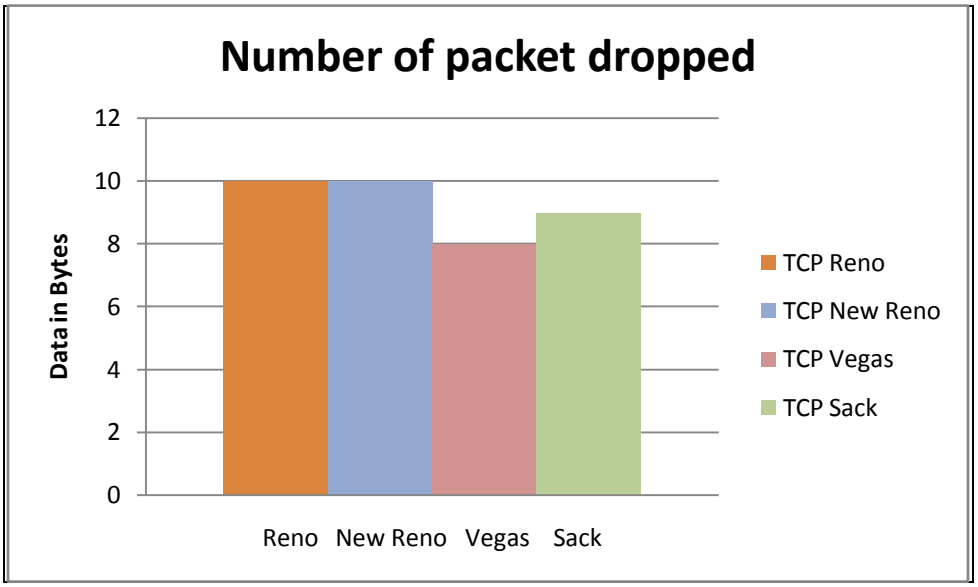


Figure3: Number of packet dropped

Fig 3 shows that number of packet dropped in terms bytes.This shows that vegas has less number 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.

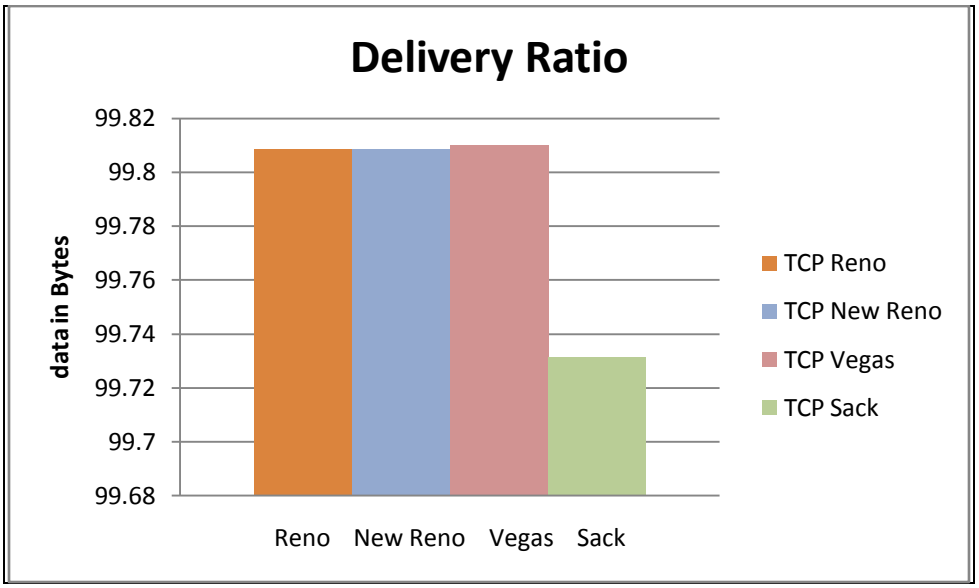


Figure4: Delivery Ratio

Fig 4 shows the delivery ratio of TCPin 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.

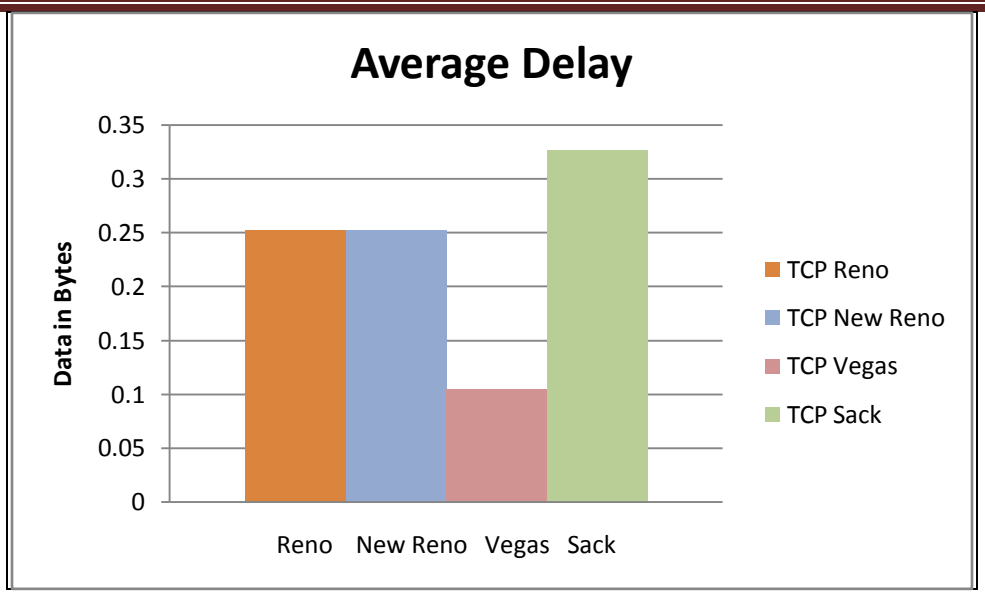


Figure6: Average Delay

Fig 5 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.

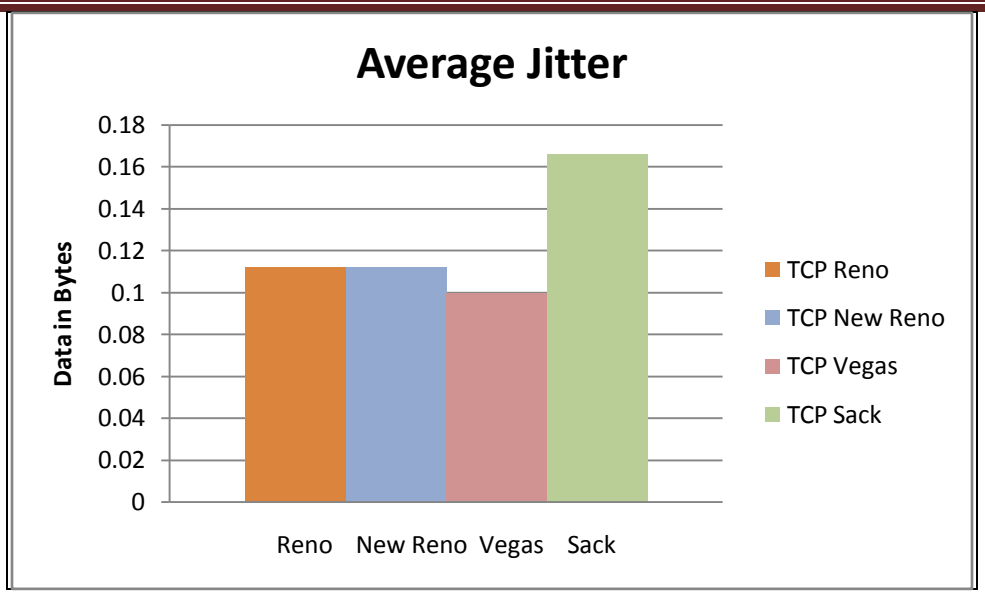


Figure 6: Average Jitter

Fig 6 shows the average jitter of all the TCP variants.This shows that the Vegas has less time variation as compared to other TCP variants

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 droppeSd,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.This can also be verified from the data represented through table 1.

REFERENCES:

1. Allman, M., Paxson, V. and Blanton, E. 2009. TCP Congestion Control. RFC 5681.
2 B.S.Manoj, and C. Siva Rama Murthy 11 March 2012,“Adhoc wireless networks:architectures and protocols”, Second Edition, Prentice Hall.
3 C. Perkins, E. B. Royer and S. Das,July 2003. “AdHoc On-Demand Distance Vector (AODV) routing”, RFC 3561, IETF Network Working Group,
4 C. Siva Ram Murthy and T. Venkatesh, TCP Vegas An Analytical Approach to Optical Burst Switched Networks.

5 Fall, K. and Floyd, S. 1996. Simulation-based Comparisons of Tahoe, Reno and SACK TCP. ACM SIGCOMM Computer Communication Review.

6 H., et al.,Wang et al.2000, A simple refinement of slow-start of TCP congestion control, 2000, pp. 98-105.

7 LarryL. Peterson, Lawrence Brakmo, Sean W. O'Malley1994TCP Vegas: New Techniques for Congestion Detection and Avoidance

8 Rayadurgam Srikant TCP-Vegas The mathematics of Internet congestion control

9. Subir Kumar Sarkar and T. G. Basavaraju"Adhoc wireless networks:architectures and protocols", Second Edition, Prentice Hall.

10. Van Jacobson,August 1988.. Congestion Avoidance and Control. Computer Communications Review, Volume 18 number 4, pp. 314-329.

11. V.JacobsonAugust1988 "Congestion Avoidance andControl".SIGCOMM Symposium no Cummunication Architecture and protocols. Volume 18 Issue 4, Pages 314-329

12 V.Jacobson 30 Apr 1990 "Modified TCP Congestion Control and Avoidance Alogrithms".Technical Report.

13 W. Steven,January 1997. TCP slow start, congestion avoidance, Fast retransmit & fast recovery algorithm, IETFRFC 2001.

14 Princeton University - Technical Reports - Computer Science - Understanding TCP Vegas: Theory and Practice

15 Adaptive Vegas: A Solution of Unfairness Problem for TCP Vegas Information networking: convergence in broadband and mobile networking By Cheeha Kim

16 University of California at Berkeley - Issues in TCP Vegas.

17 NehaBhatla, 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.