iJRASET
International Journal For Research in
Applied Science and Engineering Technology


# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

**Volume:** 5 **Issue:** IV **Month of publication:** April 2017

**DOI:** http://doi.org/10.22214/ijraset.2017.4195

Volume 5 Issue IV, April 2017
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

# Performance Analysis of TCP Variants with Pause Time and Node Speed Variation over AODV and DSR Protocols in NS2

Kadganchi Harshad[1], Prof. Shirgan S.S.[2], Valsangkar Farid[3]
*NBNSCOE, Solapur*

***Abstract*: TCP is most widely used transport layer protocol. Most of the applications such as e-mails, file transfers use TCP due to its reliable communication. There are various mechanisms to control the congestion in the network. The variants of TCP implement slow start, congestion avoidance, fast retransmit and fast recovery algorithms in different ways for congestion control. In this paper, we have simulated four TCP variants namely Tahoe, Reno, New-Reno and Vegas in mobile ad hoc network over AODV and DSR routing protocols. Simulation is done in NS2. Comparison of throughput, end-to-end delay and packet delivery fraction is made against pause time and node speed variation to determine the performance of these four TCP congestion control algorithms.***
***Keywords*: MANET, TCP, Tahoe, Reno, New-Reno, Vegas, AODV, DSR**

## I. INTRODUCTION

A mobile ad hoc network is a continuously self-configuring, infrastructure-less network of mobile devices connected wirelessly. Each device in a MANET is free to move independently in any direction, and will therefore change its links to other devices frequently. Each must forward traffic unrelated to its own use, and therefore be a router. The network topology is peer-to-peer, self-forming, self-healing in nature. An ad hoc routing protocol is a convention, or standard, that controls how nodes decide which way to route packets between computing devices in a mobile ad hoc network. In ad hoc networks, nodes are not familiar with the topology of their networks. Instead, they have to discover it. Typically, a new node announces its presence and listens for announcements broadcast by its neighbours. Each node learns about others nearby and how to reach them, and may announce that it too can reach them. Once the path is established, transport layer protocol (UDP or TCP) is required for data transfer. TCP is connection oriented protocol which is widely used due to its congestion control capability. There are several variants of TCP which have been suggested time to time for providing a solution to the congestion problem. Each variant possesses some special criteria and hence provide different performance outputs under a typical wireless environment.

## II. AD HOC ROUTING PROTOCOLS

### *A. AODV*

In Ad hoc On-Demand Distance Vector Routing protocol [3, 4], to find a route to the destination, the source node sends route request packet (RREQ) over the network. The RREQ packet is flooded and they create temporary route entries for the reverse path through every node it passes in the network. When it reaches the destination a route reply (RREP) is sent back through the same path the RREQ was transmitted. Every node maintains a route table entry which updates the route expiry time. In AODV, all routing packets carry the sequence numbers. These determine freshness of routing information and to prevent routing loops. Whenever a link is broken, a RERR message is used to notify other nodes of the loss of the link. In order to enable this reporting mechanism, each node keeps a precursor list containing the address for each its neighbours that are likely to use it as a next hop towards each destination.

### *B. DSR*

Dynamic Source Routing [5] is a type of source routing. The sender knows the complete hop-by-hop route to the destination. It is based on route discovery and route maintenance procedures. When a node in the ad hoc network attempts to send a data packet to a destination for which it does not already know the route, it uses a route discovery process. Route discovery works by flooding the network with route request (RREQ) packets. Each node receiving an RREQ rebroadcasts it, unless it is the destination or it has a

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

route to the destination. Such node replies to the RREQ with a route reply (RREP) packet that is routed back to the original source. The route carried back by the RREP is cached at the source. Route maintenance is the mechanism by which sender is able to detect, while using a source route to destination, if the network topology has changed such that it can no longer use its route to destination because a link along the route no longer works. When route maintenance indicates a source route is broken, sender deletes the cached route entry and it can attempt to use any other known route to destination, or can invoke route discovery again to find a new route. Route Maintenance is used only when sender is actually sending packets to destination.

## III. DESCRIPTION OF TCP VARIANTS

### *A. TCP Tahoe*

It is the TCP algorithm suggested by Van Jacobson in his paper [6]. TCP Tahoe uses acknowledgements to clock outgoing packets. It also maintains a congestion window to reflect the network capacity. For congestion avoidance Tahoe uses 'Additive Increase Multiplicative Decrease'. A packet loss is taken as a sign of congestion. If three duplicate acknowledgements are received, it is considered as packet loss. Tahoe performs a fast retransmit and saves the half of the current window as a threshold value. It then sets the congestion window to one and starts slow start until it reaches the threshold value. After that it increments linearly until it encounters a packet loss.

### *B. TCP Reno*

This Reno retains the basic principle of Tahoe, such as slow start and the coarse grain re-transmit timer. However it adds some intelligence over it so that lost packets are detected earlier. If three duplicate acknowledgements are received, Reno will perform a fast retransmit and skip the slow start phase by instead halving the congestion window (instead of setting it to one like Tahoe), setting the slow start threshold equal to the new congestion window, and enter a phase called Fast Recovery. In this state, TCP retransmits the missing packet that was signalled by three duplicate acknowledgements, and waits for an acknowledgment of the entire transmit window before returning to congestion avoidance. If there is no acknowledgment, TCP Reno experiences a timeout and enters the slow start state.

### *C. TCP New-Reno*

New-Reno [8] is a slight modification over TCP Reno. It is able to detect multiple packet losses and thus is much more efficient than Reno in the event of multiple packet losses. Like Reno, New-Reno also enters into fast retransmit when it receives multiple duplicate packets; however it differs from Reno in that it doesn't exit fast recovery until all the data which was out standing at the time it entered fast recovery is acknowledged. Thus, it overcomes the problem faced by Reno of reducing the congestion window size multiples times. The fast-retransmit phase is the same as in Reno. The difference is the fast-recovery phase which allows for multiple re-transmissions in New-Reno. TCP New-Reno exits fast recovery after receiving acknowledgement of all unacknowledged segments. It then sets congestion window size to slow start threshold and continues the congestion avoidance phase.

### *D. TCP Vegas*

Vegas [9] is a TCP implementation which is a modification of Reno. It tries to get around the problem of coarse grain timeouts by checking for timeouts at a very efficient schedule. Also it overcomes the problem of requiring enough duplicate acknowledgements to detect a packet loss, and it also suggests a modified slow start algorithm which prevents it from congesting the network. It detects congestion before the packet losses occur. However it still retains the other mechanism of Reno and Tahoe, and a packet loss can still be detected by the coarse grain timeouts.

There are three major changes in Vegas version. First, it keeps track of when each segment was sent and it also calculates an estimate of the Round Trip Time by keeping track of how long it takes for the acknowledgment to get back. Whenever a duplicate acknowledgement is received it checks to see if the difference between current time and segment transmission time is greater than estimated Round Trip Time. If it is then it immediately retransmits the segment without waiting for 3 duplicate acknowledgements or a coarse timeout. Second change in this variant is in congestion avoidance mechanism. It determines congestion by observing a decrease in sending rate as compared to the expected rate, as result of large queues building up in the routers. Third major change is modified slow start. The reason for this modification is that when a connection first starts it has no idea of the available bandwidth and it is possible that during exponential increase it over shoots the bandwidth by a big amount and thus induces congestion. To this end Vegas increases exponentially only every other RTT, between that it calculates the actual sending throughput to the expected

Volume 5 Issue IV, April 2017
ISSN: 2321-9653
**International Journal for Research in Applied Science & Engineering Technology (IJRASET)**

and when the difference goes above a certain threshold it exits slow start and enters the congestion avoidance phase.

## IV. SIMULATION DESCRIPTION

We have designed a wireless ad hoc network in Network Simulator (NS2). We have taken a topology of 50 wireless mobile nodes moving in area 500m x 500m for 100 seconds and having 20 FTP connections. The performance parameters are measured against pause time and node speed. The simulation is done over AODV and DSR protocols. In the first part, with 10m/s node speed the pause time is varied from 0 to 100s in steps of 10s and the results are calculated. In the next part, the node speed is varied from 5 to 50m/s in steps of 5m/s with 0 pause time and the variation of throughput, delay and pdf is measured. Throughput is in kbps, PDF is percentage calculation and End-to-End Delay is in ms. Table I shows the parameters and their values used in simulation.

TABLE I

NODE PARAMETERS IN SIMULATION

| Parameter | Value |
|---|---|
| Channel type | Channel/Wireless Channel |
| Radio-propagation model | Propagation/TwoRay Ground |
| Network interface type | Phy/WirelessPhy |
| MAC type | Mac/802_11 |
| Interface queue type | Queue/DropTail |
| Link Layer Type | LL |
| Antenna | Antenna/OmniAntenna |
| Maximum packet in ifq | 50 |
| Area (x) x (y) | 500x500 |
| Number of mobile nodes | 50 |
| Source type | TCP (Tahoe, Reno, NewReno, Vegas) |
| Simulation Time | 100 seconds |
| Routing protocol | AODV, DSR |

## V. SIMULATION RESULTS AND ANALYSIS

The analysis is done based on the results obtained from the simulation environment. We have used throughput, packet delivery fraction and end-to-end delay as the performance parameters. Fig. 1 to Fig. 12 shows the graphical analysis of throughput, delay and pdf with variation in pause time and node speed.

### *A. Throughput*

Throughput is the measurement of number of packets passing through the network in a unit of time. This metric shows the total number of packets that have been successfully delivered to the destination nodes and throughput improves with increasing nodes density. Throughput can be defined as:

$$\frac{\Sigma \text{ Node Throughputs of Data Transmission}}{\text{Total number of nodes}}$$

### *B. End-to End Delay*

It is time involved in delivery of data packets from source node to destination node. A specific packet is transmitting from source to destination and calculates the difference between send times and received times. Average end to end delay includes all possible delays caused by buffering during route discovery, latency, queuing at the interface queue, retransmission delays at the MAC, and propagation and transfer times of data packets.

Volume 5 Issue IV, April 2017
ISSN: 2321-9653
**International Journal for Research in Applied Science & Engineering Technology (IJRASET)**

*C. Packet Delivery Fraction (PDF)*

Packet delivery fraction is the ratio of packets that are successfully delivered to a destination compared to the number of packets that have been sent by sender. In order to calculate packet delivery fraction we need total number of packets sent and number of received packets.

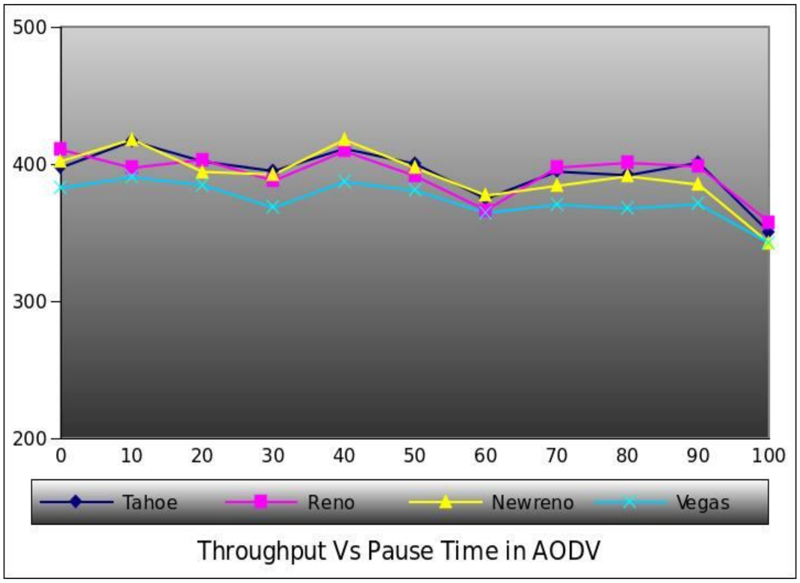



Fig. 1 Throughput Vs Pause time in AODV environment

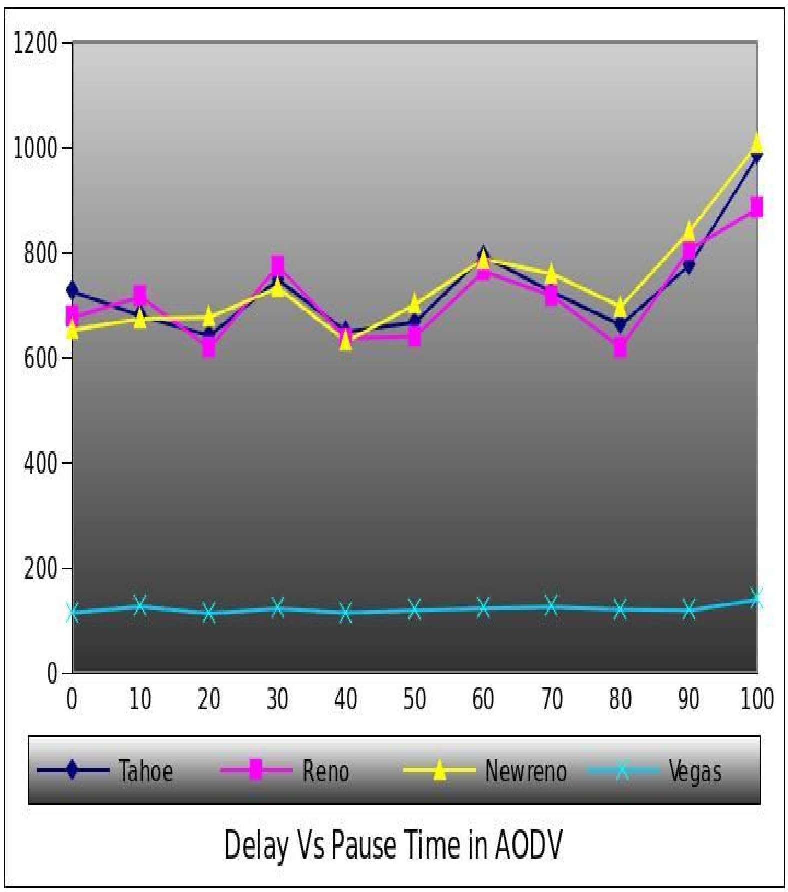



Fig. 2 End-to-end delay Vs Pause time in AODV environment

Volume 5 Issue IV, April 2017
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

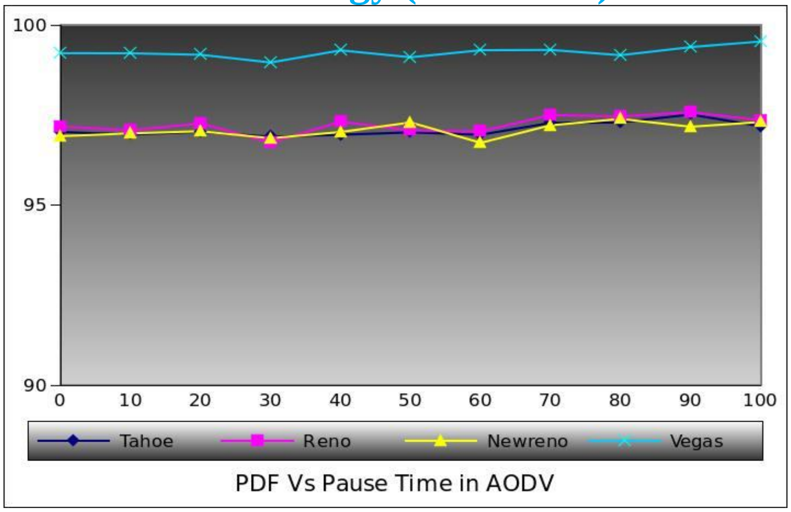



Fig. 3 Packet Delivery Fraction (PDF) Vs Pause time in AODV environment

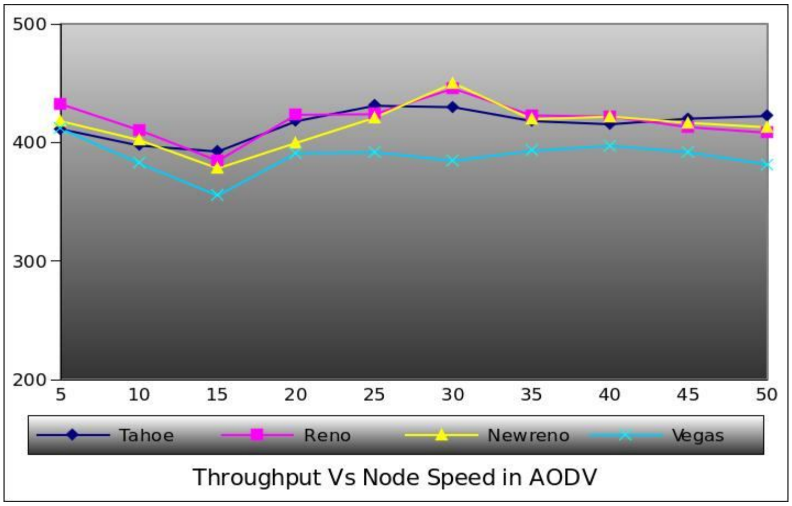



Fig. 4 Throughput Vs Node speed in AODV environment

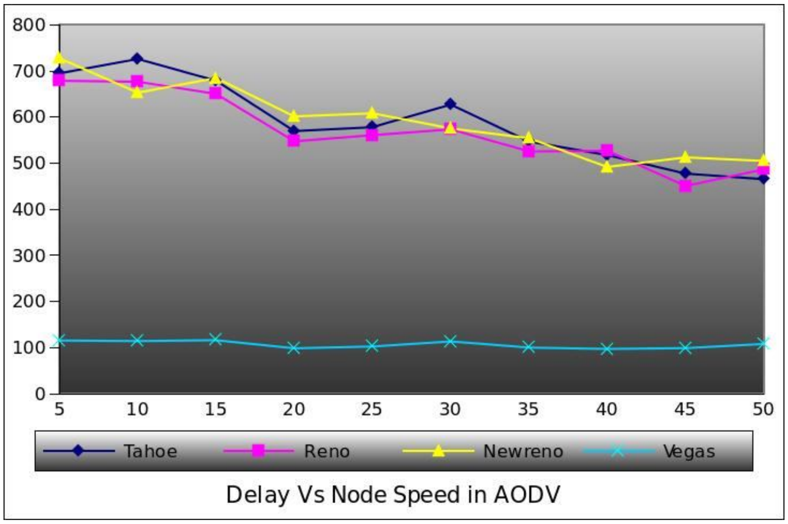



Fig. 5 End-to-end delay Vs Node speed in AODV environment

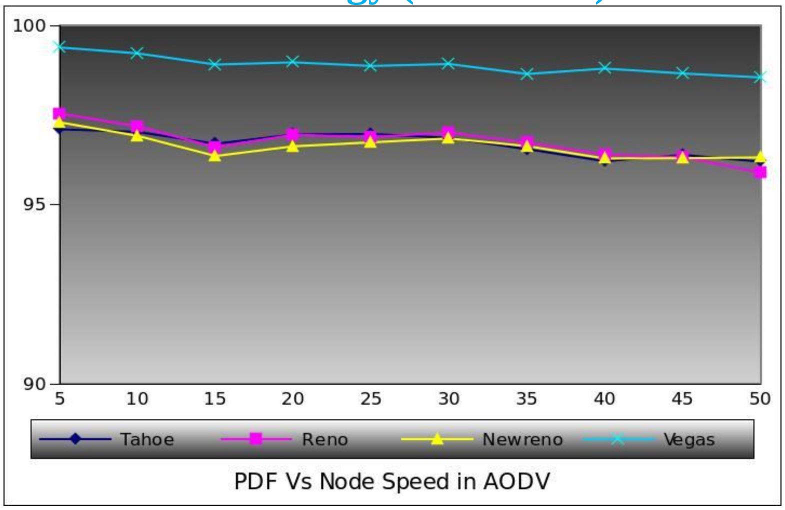



Fig. 6 Packet delivery fraction (PDF) Vs Node speed in AODV environment

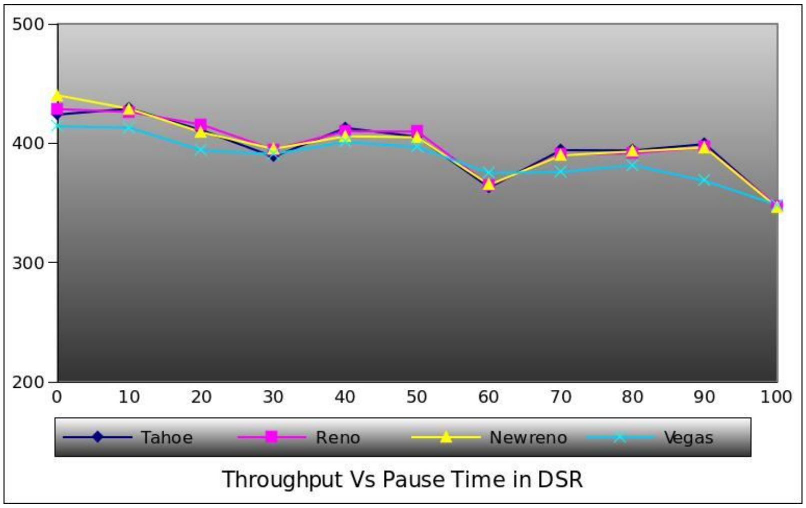



Fig. 7 Throughput Vs Pause time in DSR environment

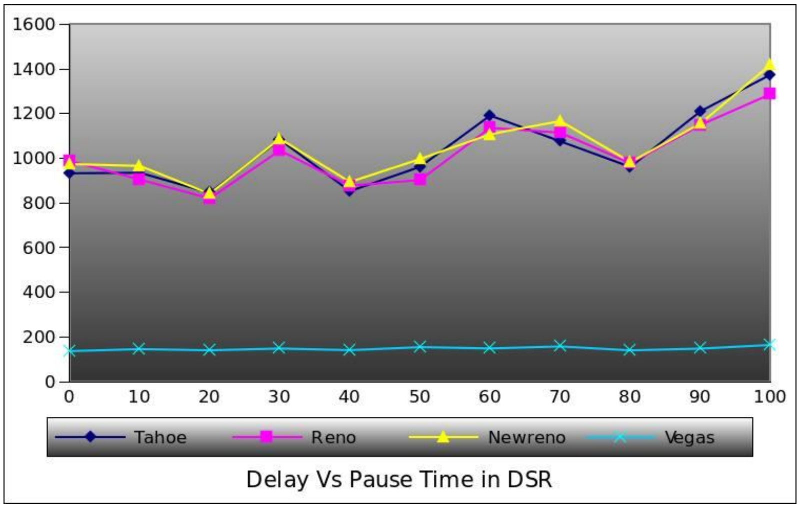



Fig. 8 End-to-end delay Vs Pause time in DSR environment

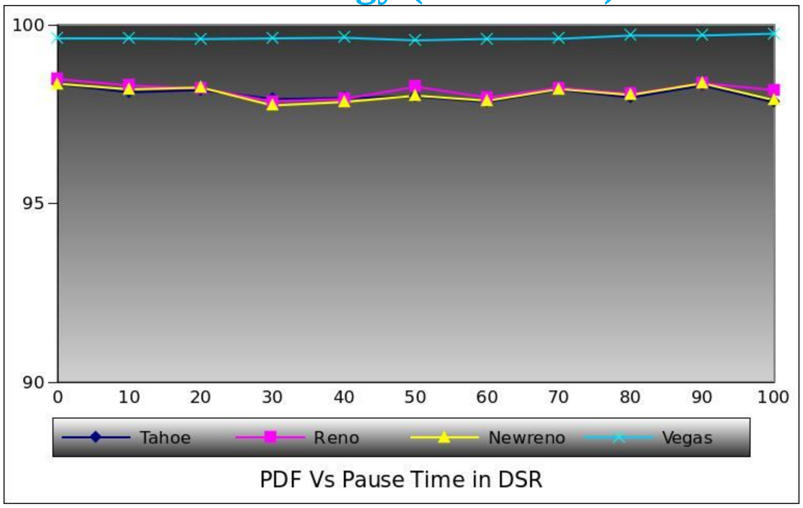



Fig. 9 Packet delivery fraction (PDF) Vs Pause time in DSR environment

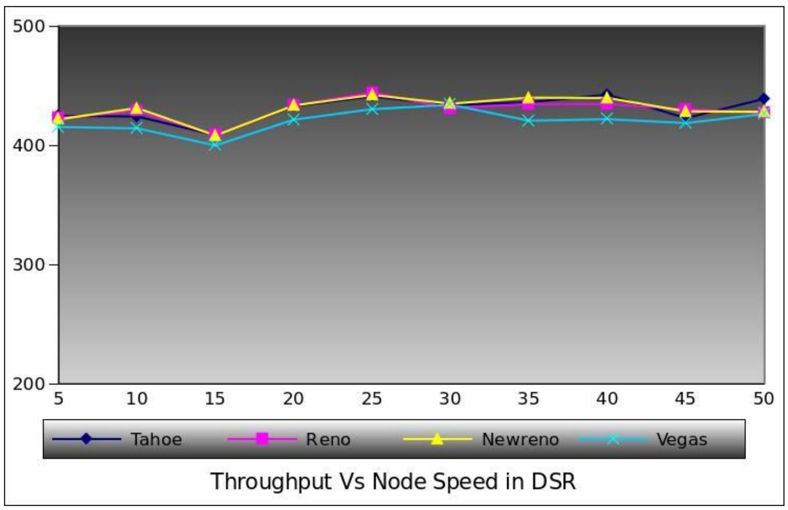



Fig. 10 Throughput Vs Node speed in DSR environment

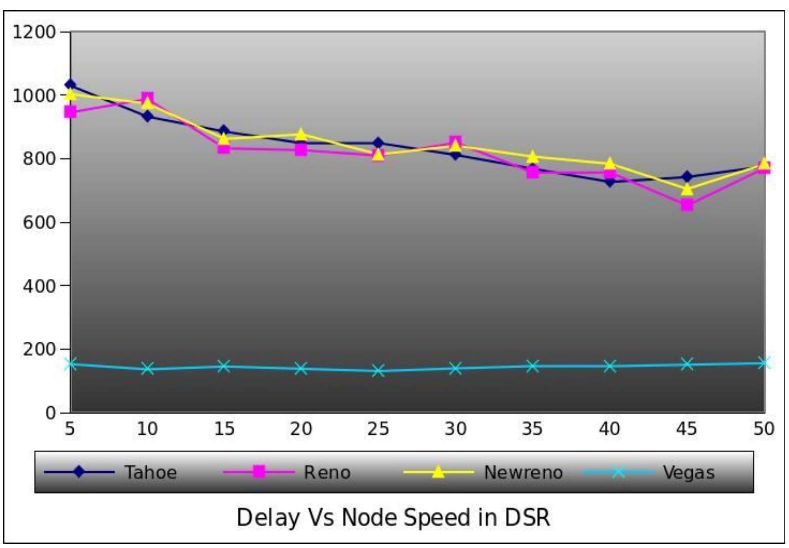



Fig. 11 End-to-end delay Vs Node speed in DSR environment

Volume 5 Issue IV, April 2017
ISSN: 2321-9653
**International Journal for Research in Applied Science & Engineering Technology (IJRASET)**

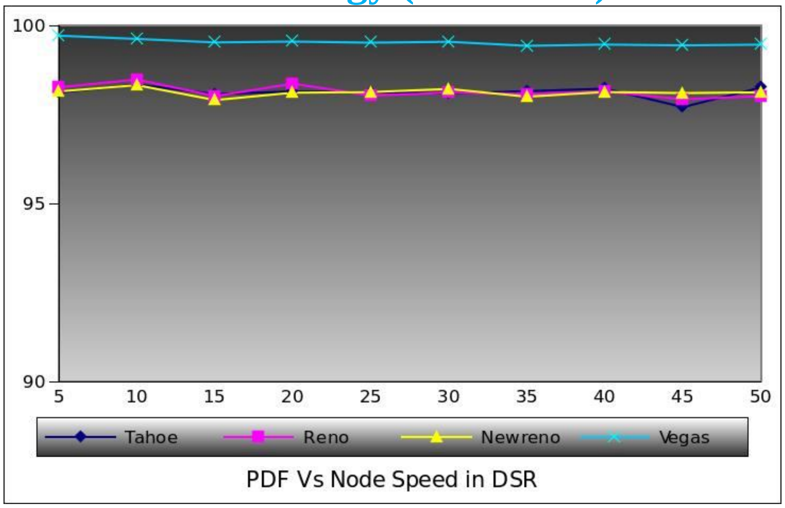


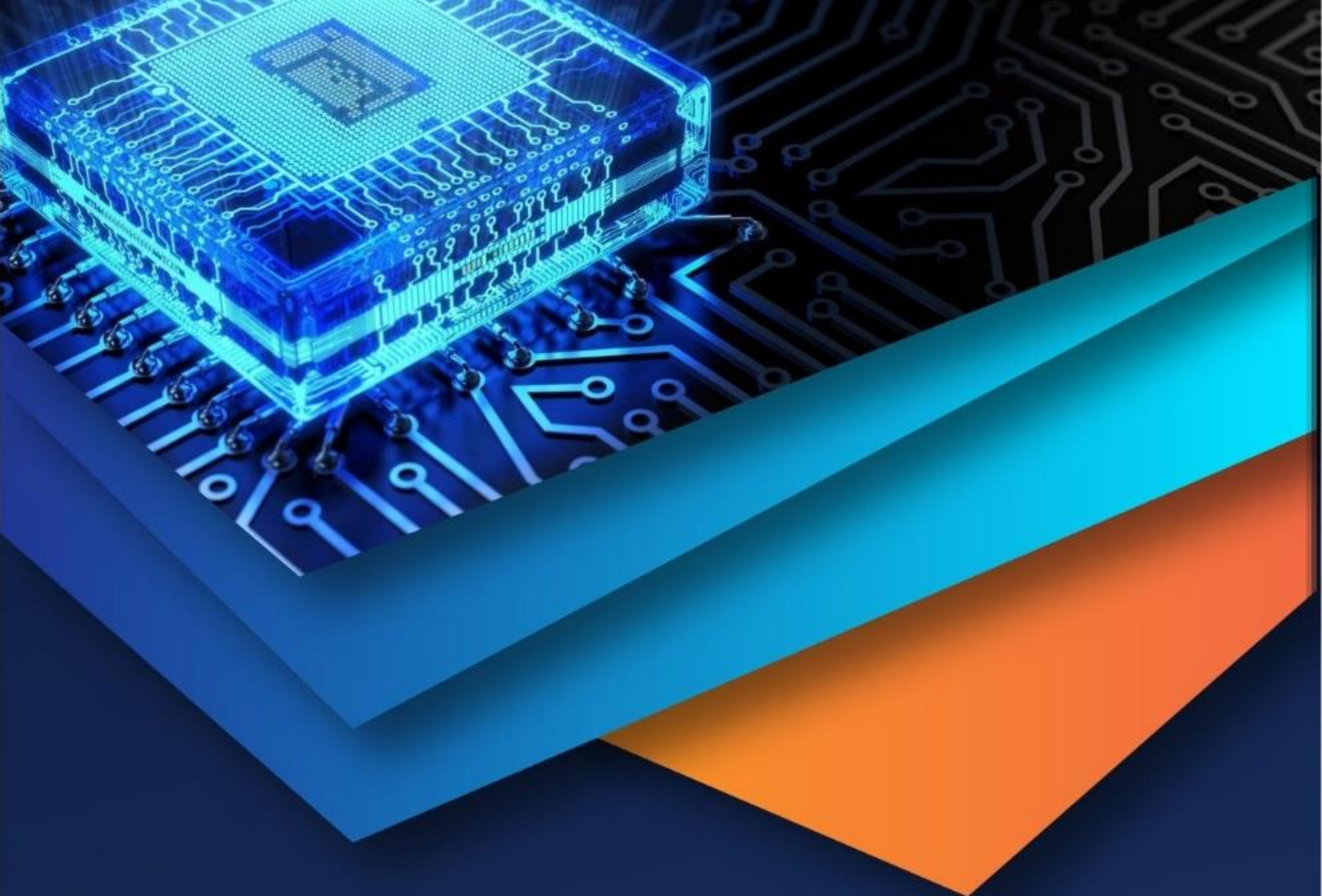

Fig. 12 Packet delivery fraction (PDF) Vs Node speed in DSR environment

## VI. CONCLUSION

From the simulation study we have found that Vegas perform better than NewReno, Reno and Tahoe over AODV and DSR in terms of PDF and Delay. In AODV environment, with pause time variation, throughput variation is similar in Tahoe, Reno, New Reno and Vegas. However, Vegas outperforms in delay and PDF. With increase in node speed, slight decrease found in delay. In DSR environment, with variation in pause time, delay is found to be varying for Tahoe, Reno and New Reno, but it is nearly steady for Vegas. PDF of Vegas in DSR environment is higher than the other three variants, with pause time and node speed variation. While considering two network layer protocols under analysis, delay in AODV is less as compared to delay in DSR for all four variants of TCP. Delay of Vegas is lowest and PDF is highest among all scenarios under study. Due to the major changes involved in Vegas such as modified slow start, new retransmission and congestion avoidance mechanism, it shows improved performance. Thus, TCP Vegas found to be most suitable congestion control algorithm among the four simulated TCP variants.

## REFERENCES

[1] NS-2, The ns Manual (formally known as NS Documentation) available at http: //www. isi.edu/nsnam/ns/doc.
[2] Teerawat Issariyakul, Ekram Hossain, “Introduction to Network Simulator”, Springer Publications
[3] C. Perkins, E. B. Royer and S. Das, "Ad hoc On-Demand Distance Vector (AODV) routing", RFC 3561 , IETF Network Working Group, July 2003.
[4] C. Siva Rama Murthy and B.S. Manoj, "Ad hoc wireless networks: architectures and protocols", Second Edition, Prentice Hall.
[5] D. Johnson, Y. Hu and D. Maltz, 'The Dynamic Source Routing Protocol (DSR) for mobile ad hoc networks for IPv4" IETF Experimental RFC 4728, February 2007
[6] V. Jacobson. "Congestion avoidance and control", on SIGCOMM '88 Symposium proceedings on Communications architectures and protocols.
[7] W. Stevens, TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms, RFC 2001, January 1997, 1-7
[8] S. Floyd, T. Henderson “The NewReno Modification to TCP’s Fast Recovery Algorithm” RFC 2582, Apr 1999.
[9] L. S. Brakmo, L. L. Peterson, "TCP Vegas: End to End congestion avoidance on a global Internet", IEEE Journal on Selected Areas in Communication, vol. 1 3[1 995], pp. 1465-1490
[10] Mandakini Tayade; Sanjev Sharma, "Performance comparison of TCP variants in mobile ad-Hoc Networks", International Journal of Computer Science and Information Security, Vol. 9, No. 3, March 2011 , pp. 165-170
[11] Balveer Singh, "A Comparative Study of Different TCP Variants in Networks", IJCTT International Journal of Computer Trends and Technology, Vol. 4, 2013, pp2962-2966.
[12] J. Postel, "Transmission Control Protocol", RFC 793, September, 1981.
[13] M. Allman, "RFC 2582: TCP Congestion Control" Technical report, 1999.
[14] C. Siva Ram Murthy, B.S. Manoj, "Wireless Technology", 2008.




10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429


# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY