Performance Investigation of TCP variants in Mobile Ad Hoc Networks

Suveg Moudgil
Ph. D Research Scholar (CSE Deptt),
MMEC, MMU, Mullana Ambala, India
Suvegmodgil1@gmail.com

Dr. Sanjeev Rana
Prof. (CSE Deptt),
MMEC, MMU, Mullana, Ambala, India
Dr.sanjeevrana@mmumullana.org

Abstract - Transmission Control Protocol (TCP) is a reliable, connection oriented, congestion control protocol used by most of the networks. TCP was planned for traditional wired networks but recently there has been an increase in its placement over mobile ad hoc networks. We have investigated TCP variants in mobile ad hoc networks in this paper**.** We have employed the network simulator (NS2) for performance investigation. Simulations are performed in various Manet scenarios with different mobility speeds and no. of connections. Packet delivery ratio, throughput and end to end delay are the key metrics which are observed to investigate TCP variants performance.

Keywords*- MANET, TCP, TCP Reno, TCP Sack, TCP-Vegas.*

*****

I. INTRODUCTION

Mobile ad hoc network is a group of mobile nodes that can communicate with each other without the need of fixed infrastructure [1]. The mobile nodes also acts as router to forward the packets to enable communication among different nodes of MANET. The nodes in the network are free to move independently in any direction, leave and join at any time. Multi hop routing, dynamic topologies, distribute working, bandwidth and energy constrained operations and limited physical security are characteristics of MANET [2]. Network scalability, routing overhead, mobility induced route changes, limited bandwidth, packet loss due to transmission errors**,** variation in link and node capabilities and hidden terminal problem are the key challenges which are required to be met before implementing MANET.

The Transmission control protocol does fine over traditional wired networks. However, performance of TCP reduces considerably in MANETs due to inherent features of MANETs including node mobility, shared & time varying wireless channels and medium collision [3], [4], [5]. In Traditional TCP, main assumption is that packet loss is because of the congestion where as in MANET, besides congestion, different causes for packet loss includes tremendous MAC contention**,** route failure and so forth. In MANET TCP misunderstands each packet loss as congestion and activates its congestion control procedures.

In this paper, we investigated the performance of various TCP variants including TCP Tahoe, TCP Reno, TCP New Reno, TCP Fack, TCP Sack, and TCP Vegas in different MANET scenarios with various load and node mobility conditions. Remaining part of the paper is arranged as follows. Section II discuss the TCP congestion control mechanisms in brief with the TCP variants of interest. Section III covers the simulation parameters and performance investigation of TCP variants. Section IV concludes the paper with discussion of the result.

II TCP CONGESTION CONTROL MECHANISMS

To deliver packets in the network TCP has end to end congestion control. TCP utilizes congestion window to restrict the data to be sent at a given amount of time. TCP increases congestion window by a little after receiving each ACK. [6]. It presumes FIFO queuing in routers. In TCP basic version there is no method for congestion control. Various variants of TCP are proposed in the literature including TCP Tahoe, TCP Reno, TCP New reno, TCP Fack, TCP Sack, TCP Vegas etc. which deals with the problem of congestion control in TCP.

TCP Tahoe

Van Jacobson suggested Tahoe TCP congestion control algorithm [7]. It suggested 'slow start' a method that is executed when a TCP connection begins or restarts following a packet loss. The size of congestion window is doubled for each transmission until it faces congestion. In case of congestion, sending rate is lowered and congestion window is lowered to one and make a new beginning again. Tahoe discovers packet loss by timeouts. Sender is informed about congestion based on the packet loss.

TCP Reno

TCP Reno attempt to find lost packets ahead of time and the pipeline is not vacated every time in the event of packet loss [8]. TCP Reno needs that we get acknowledgement immediately whenever a segment is received. TCP Reno proposed 'Fast Re-Transmit' algorithm. Reception of 3 duplicate ACK's is taken as a sign of segment loss**.** Retransmission of segment take place immediately without pausing for timeout. It try to retransmit the segment with almost full pipe. TCP Reno does not decrease the congestion window to 1 because it vacates the pipe**.** It starts into 'Fast Re Transmit' algorithm.

 ISSN: 2454-4248
01 – 04

TCP New Reno

TCP New Reno is capable to discover multiple packet losses [9]. It also admit to fast retransmit as Reno but it stays in fast recovery phase till all the unacknowledged data from one window of data is acknowledged. TCP New Reno beats the issue of multiple reduction of cwnd from one window data by averting expiration of RTO multiple times. It leaves Fast recovery when every data in the window is acknowledged.

TCP Sack

TCP with Selective acknowledgments is a modification of TCP Reno [11]. TCP Sack detects multiple lost packets & retransmit more than one lost packet per RTT**.** TCP Sack needs selectively acknowledgement of segments. It transmit a new packet if no outstanding packets are there. Therefore in one RTT more than one lost segment can be transmitted.

TCP Fack

TCP forward acknowledgement [15] is TCP Sack extension. It utilizes Sack information to have precise control in inserting data into the network. It is attained by explicitly calculating the total no. of bytes of outstanding data in the network. It take most forward selective acknowledgement sequence number as a indication that all the previous acknowledged segments were lost and it permits betterment in recovery of losses considerably.

TCP Vegas

TCP Vegas is an extension of TCP Reno. It proposes a method which verifies for timeouts at a very efficient schedule [11]. It maintain record of each segment that when it was sent and it compute RTT estimate by keeping track of time the acknowledgement takes to get back. TCP Vegas decides congestion by a reduction in sending rate in contrast to the expected rate. TCP Vegas does not utilize the segment loss to indicate congestion. It raises exponentially after every RTT and it compute the actual sending throughput to the expected. TCP Vegas exits slow start when the difference passes some threshold and start congestion avoidance phase.

III SIMULATION AND RESULTS

A. Network Environment

Network Simulator (NS 2.35) [12] is used to perform the investigation of performance of different TCP variants under different network scenarios with changing the network load and node mobility. In NS 2.35, back end language is C++ and front end language is TCL**.** In our model, the wireless terrain is 1500 x 1500 meters. No. of nodes is 50. The transmission range of all nodes is equal. The packet size is 512 bytes. AODV routing protocol [13] is used. TCP protocol variants Tahoe, Reno, New Reno, Sack, Fack, and Vegas are used as the transport protocol. The simulated traffic is ftp type in the simulation.

TABLE 1 SIMULATION PARAMETERS

Parameter	Value
Wireless Terrain	1500x1500
No. of Nodes	50
MAC protocol	MAC 802.11
Routing Protocol	AODV
TCP Protocols	Tahoe, Reno, New Reno, Sack, Fack, Vegas
Traffic Type	FTP
Packet Size	512
No. of Connections	5,10,15,20,25
Node Mobility Model	Random Way Point
Max. speed of node movement (m/s)	5,10,15,20,25,30
Simulation Time	250 seconds

B. Performance Evaluation

We have investigated the performance of various TCP variants under different network load and node mobility conditions.

i) Under different network load conditions

Packet delivery ratio, throughput and average end to end delay are considered to evaluate the performance of the various TCP variants. Analysis is done on the basis of changing the network load in form of connections in the range between 5 and 25 in multiple of 5.

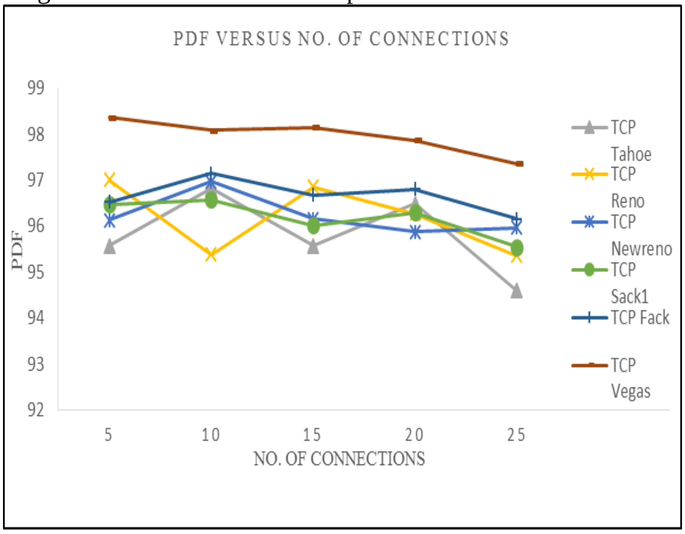

Figure 1.1 PDF of various TCP variants with different no. of connections

Figure 1.1 shows the PDF of various TCP variants with different no. of connections. As we increase the no. of connections PDF of various TCP variants degrades due to congestion. Performance of TCP Vegas is greater than other TCP variants.

ISSN: 2454-4248
01 – 04

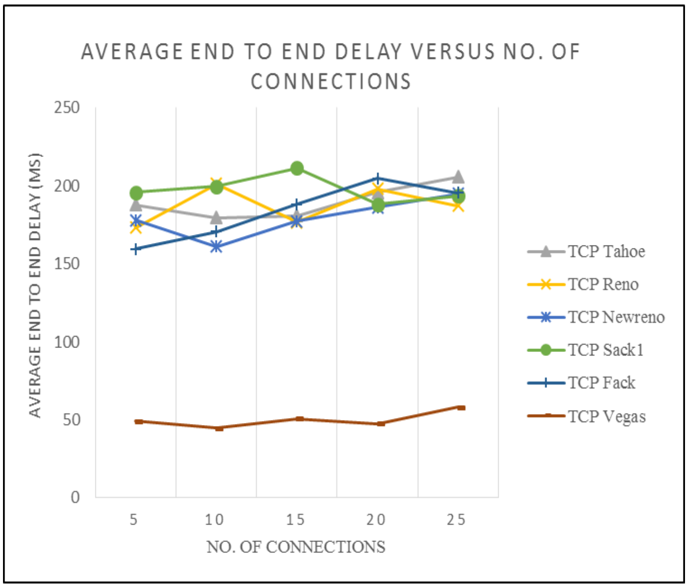

Figure 1.2 Average end to end delay of various TCP variants with different no. of connections

In Figure 1.2, average end to end delay of various TCP variants with different no. of connections. Average end to end delay rises with the rise in no. of connections due to congestion. TCP Vegas has lowest average end to end delay among its peers and it does not increase much with the increase in no. of connection.

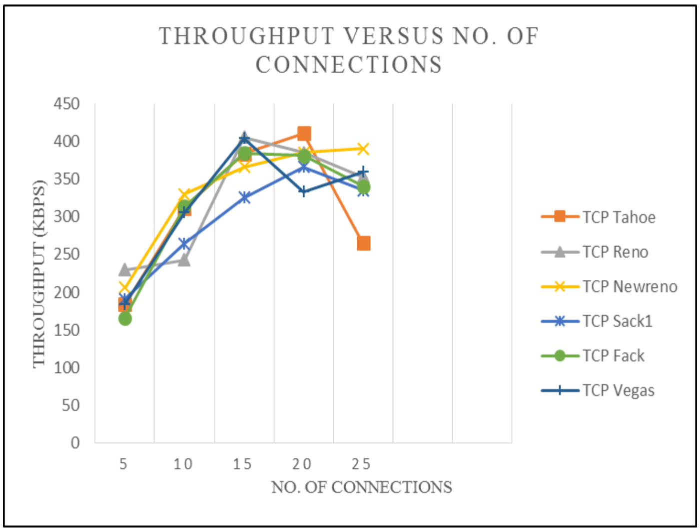

Figure 1.3 Throughput of various TCP variants with different no. of connections

Figure 1.3 shows the throughput of various TCP variants with different no. of connections. Throughput of various TCP variants rises with the rise in the no. of connections. When we have the maximum connections that is 25, throughput of TCP Tahoe decreases sharply due to congestion. The throughput of all TCP variants is optimal, when the no. of connections is between 15 and 20 in the proposed MANET conditions.

ii) Under different node mobility conditions

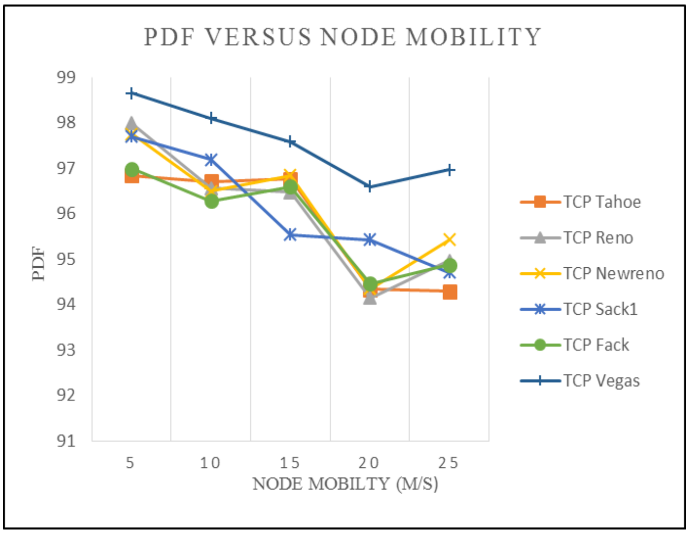

Figure 1.4 PDF of various TCP variants with different node mobility speeds

Figure 1.4 shows the PDF of various TCP variants with different node mobility speeds in the range of 5 m/s to 25 m/s in the multiple of 5. As we increase the mobility of nodes, the performance of all TCP variants decreases and PDF of TCP Vegas is highest among its peers in all the cases of node speeds.

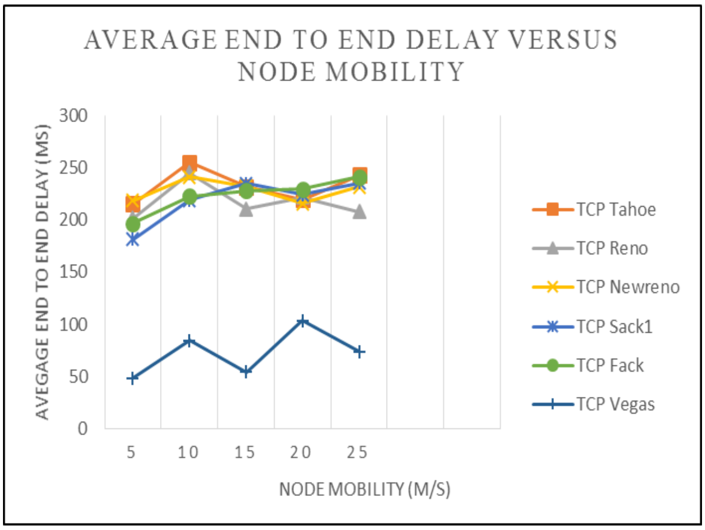

Figure 1.5 Average end to end delay of various TCP variants with different node mobility speeds

Figure 1.5 shows average end to end delay of various TCP variants with different node mobility speeds in the range of 5 m/s to 25 m/s in the multiple of 5. Overall average end to end delay increases as we increase the node mobility speed. TCP Vegas has lowest average end to end delay in its peers.

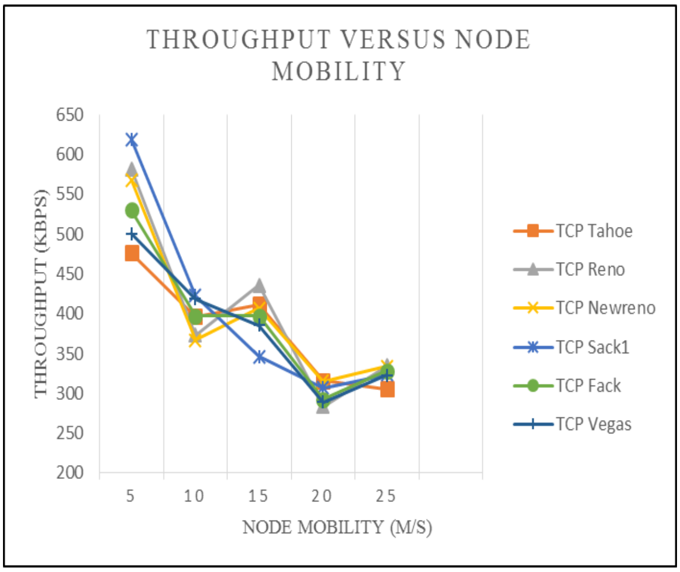

Figure 1.6 Throughput of various TCP variants with different node mobility speeds

Figure 1.6 shows average end to end delay of various TCP variants with different node mobility speeds in the range of 5 m/s to 25 m/s in the multiple of 5. Throughput of all TCP variants id highest at node mobility of 5 m/s. As we increase the node mobility, throughput overall decreases in all TCP variants.

IV CONCLUSION

In this work, we have investigated the performance of TCP variants in MANET under different load and node mobility conditions. We have used ns-2.35 simulator for doing simulation. For node mobility, random way point model have been used with the node movement in the range of 5 m/s to 25 m/s in the multiple of 5. For providing different load conditions, we have changed the no. of connections from 5 to 25 in the multiple of 5. Performance of various TCP variants is analyzed in these scenarios by using different performance metrics which include packet delivery ratio, average end to end delay and throughput. TCP Vegas perform better in terms of better throughput, minimum end to delay and packet delivery ratio due to its estimation of RTT in different scenarios. In the future work, we would extend our work with other TCP variants in different network conditions.

REFERENCES

[1] Loo J, Mauri JL, Ortiz JH, editors. Mobile ad hoc networks: current status and future trends. CRC Press; 2016.
[2] Murthy CS, Manoj BS. Ad hoc wireless networks: Architectures and protocols, portable documents. Pearson education; 2004.
[3] Al Hanbali A, Altman E, Nain P. A survey of TCP over ad hoc networks. IEEE Communications Surveys and Tutorials. 2005 Jul 1;7(1-4):22-36.
[4] Fu Z, Meng X, Lu S. How bad TCP can perform in mobile ad hoc networks. In Computers and communications, 2002. Proceedings. ISCC 2002. Seventh international symposium on 2002 (pp. 298-303). IEEE
[5] Gerla M, Tang K, Bagrodia R. TCP performance in wireless multi-hop networks. In Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA'99. Second IEEE Workshop on 1999 Feb 25 (pp. 41-50). IEEE.
[6] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981
[7] V. Jacobson, "Congestion avoidance and control", ACM SIGCOMM 88 pp 273-288
[8] V. Jacobson, "Modified TCP Congestion Control and Avoidance Algorithms", Technical Report 30 Apr, 1990.
[9] S. Floyd, T. Henderson, "The New Reno Modification to TCP's Fast Recovery Algorithm", RFC 2582, Apr 1999.
[10] 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. 13, 1995 (pp 1465-1490).
[11] M. Mathis, J. Mahdavi, S Floyd, A. Romanow, "TCP selective acknowledgement options", RFC 2018, 1996.
[12] Fall K, Varadhan K. The network simulator (ns-2). URL: http://www. isi. edu/nsnam/ns. 2007.
[13] C. Perkins and E. Belding Royer, "Ad hc On-Demand Distance Vector (AODV) Routing", IETF, RFC 3561, 2003
[14] T. Camp, J. Boleng and V. Davies, "A Survey of Mobility Models for Ad Hoc Network Research", Wireless Communication and Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications, Vol. 2, No. 5, p. 483-502, 2002.
[15] Mathis M, Mahdavi J. Forward acknowledgement: Refining TCP congestion control. In ACM SIGCOMM Computer Communication Review 1996 Aug 28 (Vol. 26, No. 4, pp. 281-291). ACM.