International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 3, Issue 12, December 2014

COMPARATIVE ANALYSIS OF TCP VARIANTS FOR MOBILE ADHOC NETWORKS

Arshdeep Singh[1], Savina Bansal[2], R.K.Bansal[3]

[1]*Research Scholar, Deptt of Electronics & Comm. Engg, GZS-PTU Campus, Bathinda* 151001 *(Punjab), INDIA*
[2,3]*Professor, Deptt of Electronics & Comm. Engg, GZS-PTU Campus, Bathinda 151001 (Punjab), INDIA*

Abstract - **Due to the reliability of Transmission control protocol, almost all the traffic data make use of this TCP protocol in wired and wireless network. In mobile ad hoc network, when different traffic applications use TCP protocol at transport layer, then it is seen that it does not offer reliable services. The reasons for packet drop in wireless networks may be other than wired networks. This paper analyzes the performance of three TCP variants with the variation in the node size and mobility rates. Both real time and non real time applications are used. This paper concludes that TCP Reno is the efficient and robust TCP variant, when it is compared with TCP New Reno and SACK using OPNET simulator.**

***Keywords* – MANET, New Reno, Performance, Reno, Sack, TCP.**

I. INTRODUCTION

In wireless local area network (wlan), nodes have fixed infrastructure. Due to this mobility becomes a major drawback in such networks. To overcome this problem, mobile ad hoc networks (MANET) are introduced. In MANET, nodes may operate as independent entities and they may communicate via multiple wireless hops without any centralized control and pre existing network. Nodes in MANET may be fixed or mobile in nature and may combine randomly with each other to form arbitrary topologies. Due to quick deployment of MANET, it is mainly used in disaster hit and military areas [1].

Congestion plays an important role in communication network, it occurs when a router or node carrying so much data that its quality of service (QOS) deteriorates and causes queuing, delay, packet loss or blocking of new connections. Packet drop in wired network is mainly assumed due to network congestion. To avoid network congestion, there are different TCP congestion control algorithms are designed that are slow start, congestion avoidance, fast retransmit and fast recovery [2, 3]. Congestion window and receiver window are the two flow control algorithms, which are used. When new connection is established, TCP begins the slow start algorithm by setting the congestion window to one segment. The size of the congestion window is incremented by one for each ACK received by the sender. Hence, one, two and four packets are sent in first, second and third round trip time respectively. When the network is congested then TCP enters into congestion avoidance algorithm. Slow start phase have exponential growth, whereas congestion avoidance linear in nature. In fast retransmit algorithm, sender retransmits the packet when it receives three duplicate ACKs. Fast recovery algorithm is an advancement of congestion control mechanism. After fast recovery phase, congestion avoidance phase is invoked instead of slow start phase. There are different TCP variants are developed in order to deal with network congestion. TCP Reno, New Reno and Sack are the popular variants of TCP which are described below.

TCP Reno uses four congestion control algorithms that are slow start, congestion avoidance, fast retransmit and fast recovery. In previous variant, TCP Tahoe, complete timeout is required to detect a packet loss. Therefore, when every time a packet is lost, it waits for timeout and pipeline is emptied that results major cost in high bandwidth delay product link. To overcome this problem, TCP Reno is introduced with modification in the algorithm. In this variant, there is no need to empty the pipeline. TCP New Reno is an improvement on the TCP Reno version in order to solve the timeout problem when multiple packets are lost from the same window. Rest of the congestion control components is same for both TCP New Reno and TCP Reno. SACK stands for selective acknowledgement, in this TCP variant acknowledgement is only provided for the selective segments which have been received successfully. Therefore, only those segment are retransmitted which are not acknowledged. TCP Sack uses same fast recovery algorithm as in the case of TCP Reno that initializes congestion avoidance algorithm even for a single packet loss.

Above mentioned TCP algorithms are designed mainly to deal with network congestion. TCP performance is found to be degraded in MANET [4 - 6] because performance of TCP in MANET is highly influenced by other factors. These factors include high bit Error Rate, path asymmetry impact, network partitioning, hidden and exposed impact and likewise [7]. These factors may cause packet drop and TCP unnecessary invokes congestion control algorithms. Additionally mobility of the node is considered as the major reason for route failure and re-establishment is also needed instantly in that case. And when new route establishment takes longer time than (Retransmission Time out) RTO of the sender, then TCP assumes that packet is lost due to network congestion. In this paper, to analyze the impact of above factors on the performance of TCP, different TCP variants performance is analyzed with the variation in node size and mobility rates. Both real time (Video conferencing) and non-real time application (Email) make use of TCP protocol for transmission. Various performance metrics included in this paper are download response time, wlan delay, retransmission attempt and packet delay variation.

Mohamadreza Najiminaini et.al [8] have compared the different TCP variants using Opnet simulator. In their work, performance of different TCP variants is performed for fixed trajectory using non- real time application. They concluded TCP Reno as efficient TCP variant for wireless connection. However, their work is limited to single mobile node for wireless network. Different researches [9 - 11] have been carried out to compare TCP variants, though most of the work is limited to non-real time applications, limited node sizes and mobility rates with application size is just limited to few Kilobytes (50 Kb). In this paper, performance of different TCP variants is analyzed under variable node sizes and mobility rates for real {for high resolution video} and non-real time applications (Video Conferencing and Email). The paper is organized as follows-different performance metrics and simulation parameters are given in the section II followed by section III that presents result and performance analysis w.r.t. scalability and mobility on various TCP variants. Section IV concludes the work.

II. Simulation AND Performance Metrics

TCP variants analysis is carried out using discrete event simulation software OPNET [12]. The simulation parameters are shown in table 1.

A. *Simulation Parameters*

Table 1: Simulation Parameters

Area	100 x 100 sq. m
Routing Protocol	AODV
TCP variants	Reno, New Reno, Sack
Node size	6, 8, 10
Mobility Rate	10, 20, 30 m/s (constant with pause time 100sec and start time 10 sec)
Application	email (high load with size 50,000 bytes) video conferencing (High Resolution Video frame size 128 x 240)
Performance metrics	wlan delay, download response time, retransmission attempt, packet delay variation
Node Placed	Randomly

B. *Performance Metrics*

To evaluate the performance of different TCP variants, wlan delay, video packet delay variation and email download response time are used which are defined as:

i. Wlan Delay (sec)

The end-to-end delay is the time needed to traverse from the source node to the destination node in a network. It represents the end to end delay of all the packets received by the wireless LAN MACs of all WLAN nodes in the network and forwarded to the higher layer.

ii. Download Response Time (sec)

Download response time is defined as the time elapsed between sending a request and receiving back the response packet, which is measured between the time a client sends a request to the server and the time it receives back a response packet.

iii. Retransmission Attempts (packets/sec)

Retransmission attempts can be defined as the total number of retransmission required by all WLAN MACs in the network that have been lost or damaged due to a link failure. It also shows the number of packets failed in the process, which, in effect, requires retransmission.

iv. Packet Delay Variation (sec)

It is the variation among end to end delays for video packets. End to end delay for a video packet is measured from the time it is created to the time it is received. It is generally measured for real time applications e.g. Voice, Video- Conferencing.

III. Result AND Performance Analysis

Results are obtained using different performance metrics under variation in node size and mobility rates.

A. *Performance w.r.t. Node size*

Performance of different TCP variants is analyzed with target applications and node size variation from 6 to 10 with AODV as the routing protocol. Node speed is set to 10 m/s. Fig 1 represents download response time for different TCP variants under node size variation. When 6 users are communicating, TCP Reno has download response time equals to 7.33 sec which is least as compared to other variants. Similar trends could be seen when node size is increased to 8 and 10. So, TCP Reno outperforms TCP New Reno which in turn performs better than Sack. Wlan delay for different TCP variants tends to increase when node are added to the MANET environment (Fig. 2) e.g. percentage increment in delay for TCP Reno, New Reno and Sack is found to be 2%, 4% and 8 % respectively, when node size is increased from 6 to 8. At each node size, TCP Reno has lower value of wlan delay, whereas other two variants have almost similar performance.

Download response time and Wlan delay are highly influenced by TCP's congestion window. Larger the congestion window, shorter would be the download response time [8]. It is gathered from the results obtained that the performance of all the TCP variants degrades when more nodes are added in the network. In the presence of large number of users (10), the network becomes more congested due to the establishment of more links among the users, which causes TCP to unnecessarily invoke the counterproductive measures as described in section I.

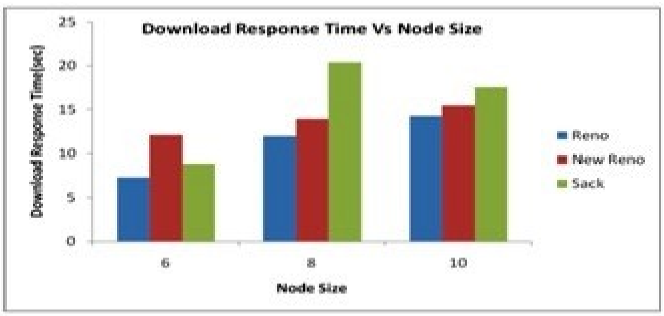

Fig. 1: Average download response time w.r.t. node sizes

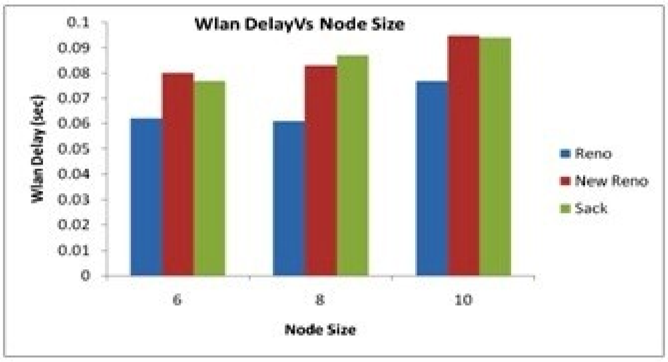

Fig.2: Average wlan delay w.r.t. node sizes

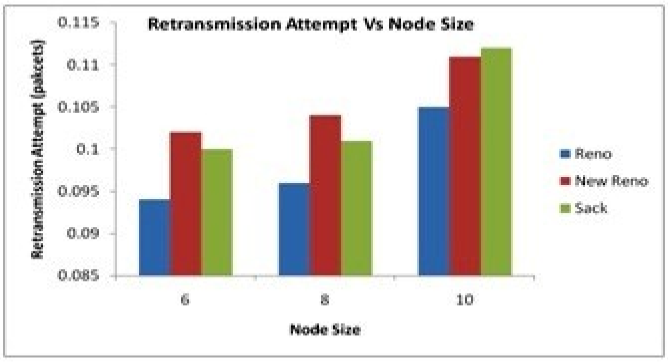

Fig.3: Retransmission attempt w.r.t. node sizes

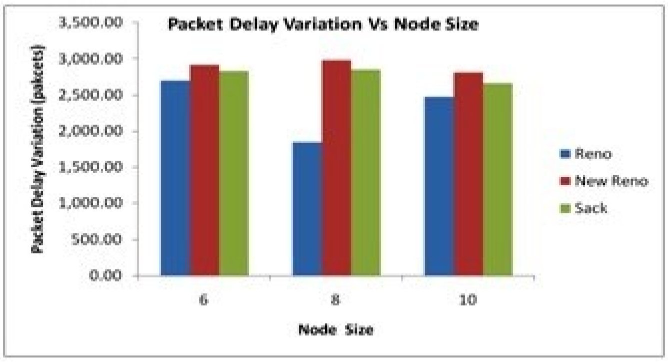

Fig.4: Packet delay variation w.r.t node sizes

Retransmission attempts (Fig. 3) are increased with increasing node size for all the three TCP variants indicating rise in the packet drop. TCP may have to take several retransmissions to deliver a single packet to its destination successfully. The retransmission attempts are maximum, when 10 users are communicating. But overall, TCP Reno performs better than other two in each scenario reflecting that TCP Reno is better able to handle this issue. Performance in terms of packet delay variations for real time application is presented in Fig. 4. Usually, these applications use UDP rather than TCP protocol owing to timeliness requirements. Results obtained as shown in Fig 4, do not reflect any consistent trend w.r.t. node size variations. Though the variations observed were minimum for the TCP Reno among all the three variants.

B. *Performance w.r.t. Node speed*

Performance of TCP variants with changing mobility rates with node size as 8 are shown in Fig 5 to 8. Fig 5 – 7 represent download response time, wlan delay and retransmission attempts respectively. All the three performance metrics get affected with the variation in the mobility. But significant improvement could be seen in case of TCP Reno. When node speed is varied from 20 to 30m/s, the percentage variations in download response time for TCP Reno, New Reno and Sack are 43.6 %, 3.8 % and 28.48 % respectively. Similarly, the percentage variations in wlan delay for above three variants are 35.5 %, 5.6 % and 7.8%. In case of retransmission attempt, same trend could be seen as 19.7 %, 4.6 %, 0.9 %. The value of above mentioned performance metrics tends to (improved) decrease in case of Reno and New Reno, whereas in case of TCP SACK opposite trend could be seen. Variation by the high percentage rate leads to conclude that performance of TCP Reno is highly improved among all the three.

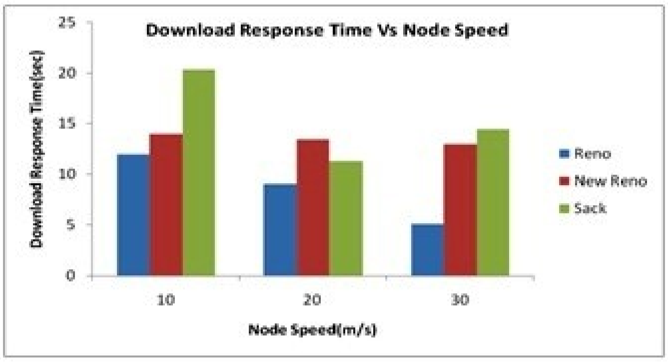

Fig.5: Download response time w.r.t. node speeds

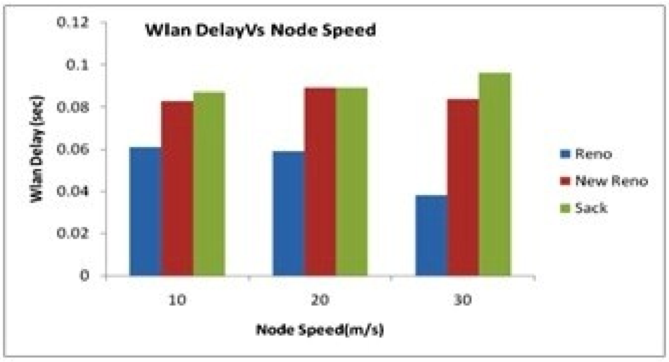

Fig.6: Wlan delay w.r.t. node speeds

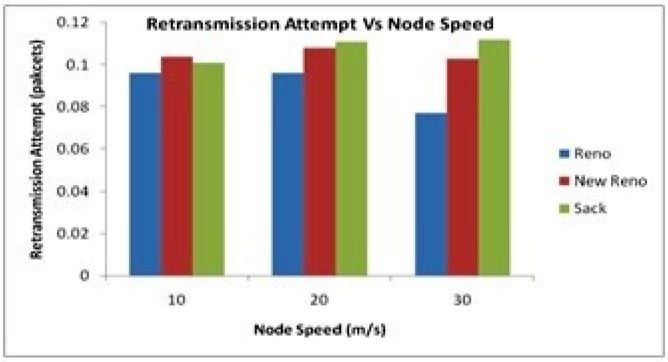

Fig.7: Retransmission attempt w.r.t. node speeds

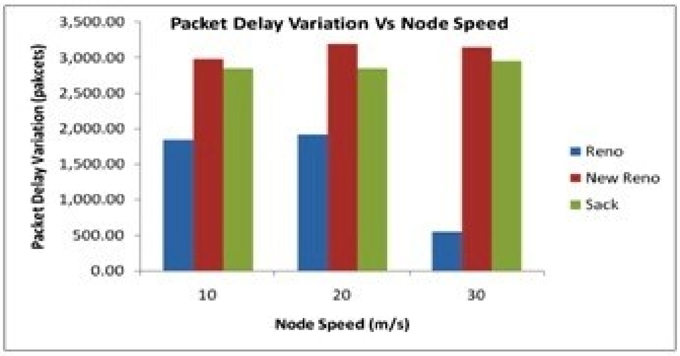

Fig.8: Packet delay variation w.r.t. node speeds

With increase in mobility rates, frequent changes of node position occurs which results change in the link states that leads to more packet losses. As TCP variants are not mainly designed for handling link failure due to mobility, so it is interesting to observe the behavior of these variants under different mobility rates. It is gathered from the results obtained that change in mobility rate results in a variety of reactions to three TCP versions. Performance of TCP Reno improves when the speed is increased to 30 m/s. It appears that the motion helps the network to discover routes and to achieve a better connectivity. So, increasing node velocity does not seem to work as a degradating factor for the TCP performance in a MANET.

Performance of different TCP variants w.r.t. mobility variation for real time application is represented in Fig. 8. Significant influence on the performance of TCP Reno could be seen, especially when mobility is varied to 30 m/s., whereas, the performance of other two variants is affected only by smaller extents.

IV. CONCLUSIONS

This paper aims to analyze the impact of scalability and mobility on the performance of TCP variants. It is concluded from the results that with increasing node size, the performance of all the three TCP variants get adversely affected in general. However, the performance of TCP Reno is better than New Reno and Sack. Whereas, with changes in mobility rates, TCP New Reno and Sack did not respond much but TCP Reno showed improvement in all the performance metrics with increasing mobility. It is worth noting that for real time application, packet delay variation improved (reduced) significantly with higher mobility for TCP Reno.

REFERENCES

[1] S S Manaseer, "On backoff mechanisms for wireless mobile ad hoc networks", Ph. D thesis, the Faculty of Information and Mathematical Sciences at University of Glasgow, pp. 1-156, Scotland, 2009

[2] W Stevens, "TCP slow start, congestion avoidance, fast retransmit, and fast recovery algorithms", Network working group, RFC 2001, Jan.1997 (https://tools.ietf.org)

[3] W Stevens, M. Allman and V. Paxson,"TCP congestion control", Network working group, RFC 2581, April 1999 ((https://tools.ietf.org)

[4] M Hassan and R Jain, "TCP performance", Communications Magazine, IEEE, vol.39, no.4, pp.51, Apr 2001

[5] X Chen, H Zhai, J Wang and Y Fang, "TCP performance over mobile ad hoc networks", Canadian Journal of Electrical and Computer Engineering, IEEE, vol.29, no.1/2, pp.129-134, Jan.-April 2004

[6] K C Leung, Li and V O K, "Transmission control protocol (TCP) in wireless networks: issues, approaches, and challenges," Communications Surveys & Tutorials, IEEE, vol.8, no.4, pp.64- 79, Fourth Quarter 2006

[7] O Bazan, U Qureshi, M Jaseemuddin and H M El-Sayed, "Performance Evaluation of TCP in mobile ad-hoc networks", Proc of 2nd International Conference on Innovations in Information Technology, IIT'05, pp. 175-185, Toronto, Canada, 2005

[8] M Najiminaini, L Trajkovic and L Subedi, "Performance evaluation of TCP Tahoe, Reno, Reno with SACK, and New Reno using OPNET modeler", in Simon Fraser University Vancouver, British Columbia, Canada, 2009

[9] Md. N I Khan, R Ahmed and Md. T Aziz, "A survey of TCP Reno, New Reno and Sack over mobile ad-hoc network", International Journal of Distributed and Parallel Systems, vol.3, no.1, January 2012

[10] Yuvaraju B N and N N Chiplunkar, "Scenario Based Performance Analysis of Variants of TCP using NS2-Simulator", International journal of advancements in technology, vol 1, no. 2, Oct. 2010

[11] S Waghmare, A Parab, P Nikose and Prof S J Bhosale, "Comparative Analysis of different TCP variants in a wireless environment", Proc. of 3rd International conference on Electronics Computer Technology, vol 4, pp. 158-162, April 8-10, 2011, Kanyakumari, India

[12] OPNET Simulator, (http://www.opnet.com)

Brief profile of authors:

Arshdeep Singh did his Bachelor from Sant Longowal Institute of Engg & Tech (Deemed Univ) in 2011. He is currently studying as research scholar in Giani Zail Singh - PTU Campus, Bathinda. His area of interest includes mobile ad hoc networks.
You can email him at: arsh [dot] longowal [at] gmail [dot] com.

Savina Bansal did Bachelor of Science and later Bachelor of engineering both from Punjab University Chandigarh. She did Masters' degree from Thapar Inst of Engg & Tech (Deemed Univ) and PhD from IIT Roorkee (India) in parallel & distributed computing. She started her professional career with Punjab state government enterprise *PUNWIRE in 1988* and later shifted to technical education with TIET (now Thapar University) Patiala in Punjab and is currently Professor at GZS-PTU campus Bathinda. She specializes in irregular interconnection networks, energy efficiency & multiprocessor scheduling, fault-tolerant real-time computing, wireless-sensor-networks and grid computing. She has more than 25-papers in Int'l Journals. She is Fellow IE(I), IETE and member CSI, ACCS and ISTE.
You can contact her at: savina [dot] bansal [at] gmail [dot] com

Rakesh Kumar Bansal did Bachelor and Masters' of engineering both from Thapar Institute of Engg & Tech (Deemed Univ). Later, he did PhD (Engg) from Punjabi Univ Patiala (India). He started his professional career in technical education with TIET (Now Thapar University) Patiala in 1986 and is presently Professor with Giani Zail Singh - PTU Campus, Bathinda in Punjab. His areas of interest include instrumentation applications, fault-tolerant real-time scheduling in parallel systems and wireless-sensor-networks. He has more than 20-papers in Int'l Journals.
He is available at: drrakeshkbansal [at] gmail [dot] com

