I J C T A, 9(37) 2016, pp. 1027-1041

Review and Analysis of TCP Performance in VANET

Kuldeep Kumar[1], Sandeep Kumar Arora[2] and Gurjot Singh Gaba[3*]

ABSTRACT

Vehicular Adhoc Networks (VANETs) makes use of wireless nodes connected to internet over road side units to enhance the operational capability for user based applications. One of the prominent protocol, which contribute a substantial quantity of internet traffic for the majority of web application like file transfers, mail services, message service, etc. was based on transmission control protocol. However, TCP performs not up to the mark under transport wireless situation because of wireless channel errors and high mobility such as interference and fading that results in unexpected packets drop. In this paper, performance of existing TCP variants such as New Reno, BIC, Westwood, Cubic, Vegas, and Compound are analyzed under vehicular traffic environment and variable speed. We have analyzed the behaviour of traditional TCP variants under different circumstances which will be helpful for foundation works in designing of TCP completely for Vehicular environments.

Keywords: TCP, RSU, VANET, CUBIC, BIC, New Reno, Westwood, Vegas, Compound.

I. INTRODUCTION

Vehicular Adhoc Network (VANET) are subclass of Mobile Adhoc Network (MANET), which identify vehicle such as cars act as node and each vehicle are having the transmission capability which are connected to form a networks. To form a network, [13] we create mobile nodes using cars to design a mobile network. VANET is most distinguished vehicular wireless technologies that form wireless communication systems known as Intelligent Transportations System (ITS) [1]. This technology leads to various initiation of research area that increases the security and the efficiency of transportation system in the presence of the ambient condition or traffics in the road condition. This vehicle networks system requirement authorities to govern it with exceeding limit 800 million in the world today, consists of various number of nodes, provided communication range with other vehicle using the short radio signal. Dedicated Short Range Communications (DSRC) (5.850 to 5.925 GHz) with seven allocated non overlapping channels. This network is ad-hoc network, thus using routers as Road Side Units (RSU). So, the road side units work as a router between two or more vehicles and to other network devices. In a VANETs, vehicle will depend on the usual data from the others vehicles [2]. Thus depending upon the received data, further in the future, control decision can be made. If the information is false, then decision can be more dangerous.

As VANET does not have fixed topologies and architecture, [12] it depends upon the moving vehicles and how they are connected in the network. In urban areas and highways VANET follows some fixed path. VANET provides communication to improve the support of wireless products. These products can be PDAs (Personal Digital Assistants, mobiles, laptops, keyless entry devices etc.

In Inter-Vehicular communication, to transmit the traffic related data to the multiple users, it uses the multi-hop configuration i.e. multicast/broadcast network [2]. In intelligent transportation systems, the

[1,2,3] Discipline of Electronics and Communication Engineering, Lovely Professional University, Phagwara, Punjab, India - 144411, *E-mails [1]Kuldeep.etah@yahoo.com, [2]sandeep.16930@lpu.co.in, [3]er.gurjotgaba@gmail.com[*]*

[*]*Corresponding Author*

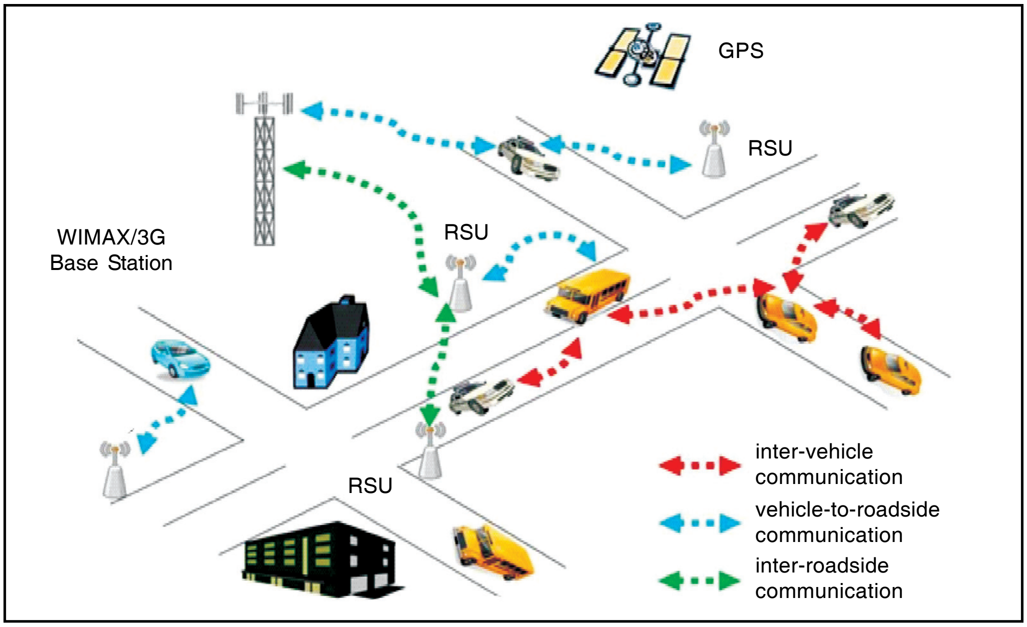

Figure 1: Architecture of VANET

communication activity should be in relevant path in order to avoid the collision in emergency situation as shown in Fig. 1.

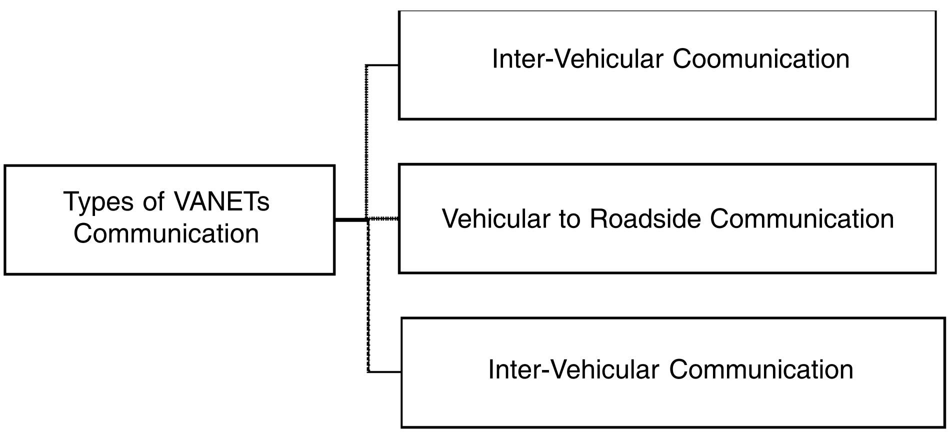

Figure 2: Types of VANETs Communication

According to *Fig. 2.* in Vehicle to Roadside Communication, configurations represent a single hop broadcasts where the road side units (RSU) sends a broadcast message to all prepared vehicle in the surrounding area. For the safety of the vehicle drivers, it uses acknowledgement addresses to solve the problem of native broadcasting.

In Inter roadside communication also known as hybrid communications. In this, vehicle can use current environment or infrastructure to create a network and then broadcast massages information to other ad-hoc networks. Infrastructure can be single-hop or multi-hop depending upon their locations and provides greater flexibility.

(A) VANETs Appications

Safety Related Application: VANET is mainly used [3] for providing safety to the vehicles. This can be done in such methods:

- Collision Avoidance: Many research survey and states that if drivers provided warning messages prior to the collision then 60% accidents can be avoided.
- Cooperative Driving: This method provides the safe driving and can also help from the uninterrupted signals or messages. These users can also get some information regarding curve speed warning, lane change warning etc.

User Based Application: That depends on the users to provide the services to the vehicles.

- Peer –to- peer application: These applications provide important information services like entertainment sources, music and movies can be shared etc. between the vehicles in the network.
- Internet Connectivity: Nowadays, internet has become the most priority in all the fields. Hence, users always want internet to connect with the network. VANET provides the internet connectivity to the vehicle users.

The internet design has penetrated in VANET through V2V gateway or through fixed RSU, since the TCP [14] form bulky of the internet traffics and it perform poorly below extremely high mobile conditions, so analysis is to be created and study to check the activity side of transmission control protocol. The definitive intent of this work is to check the small level behavioural analysis of [11] Standard TCP variant like as BIC, Vegas, Westwood, Compound, New Reno and Cubic under vehicular environments for varied mobility and traffic.

The remaining paper is organised as follows: Section II discusses knowledge based learning algorithm; Section III. comprises simulation, result and analysis; and finally conclusion is done which identifies future scope.

II. IMPLEMENTATION OF STANDARD TCP VARIANTS

The general TCP variant [15] that are carried out in operating system are taken into evaluation and they are widely categorized based totally on their functionality which includes loss based TCP congestion control, delay based TCP congestion control and loss delay based TCP congestion control as shown in Fig. 3. The subsequent are carried out TCP variations that have been experimental analyzed and simulated by means of vehicular surroundings environments.

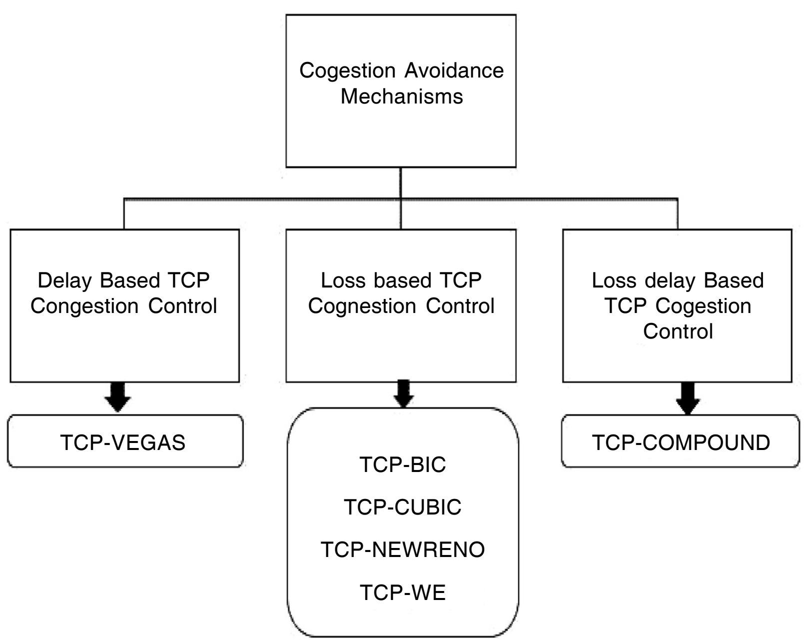

Figure 3: TCP Congestion Avoidance Mechanisms

(A) TCP-VEGAS

TCP-Vegas is one of the best congestion avoidance mechanism that detect packets delay rather than packet loss and faster than TCP Reno and can recover from multiple drop more efficiently. It is fair bandwidth estimation scheme. This algorithm is based on correct or exact calculation of BaseRTT and verifies network capability, mainly [9] TCP Vegas uses Round Trip Time (RTT) estimation. It doesn't rely upon loss packet like as TCP-Reno or TCP NewReno to estimate network capability. As soon as for every Round Trip Time (RTT), TCP-Vegas computes the predicted real throughput and average throughput.

TCP-Vegas doesn't depend on lost packet so as to estimate network capability, another it is uses RTT measurement to work out the availability of network capability. Thus TCP-Vegas carry out superior than TCP New Reno in retransmission lost packet just in case of triple duplicates acknowledgement. For each RTT, TCP Vegas [4] modify the cwnd supported on the distinction between calculate prediction and actual throughput. The particular algorithmic description [11] is given under.

Algorithm

1. Let BaseRTT (Round Trip Time) be the minimum of all measured RTTs (Commonly the RTT of the first packet).
2. If not overflowing the connection, then Expected Rate = Congestion Window/BaseRTT
3. Source calculates sending rate (Actual Rate) once per RTT
4. Source compares Actual Rate with Expected Rate Diff = Expected Rate - Actual Rate
5. if Diff < α (too little extra data) increase Congestion Window linearly in next RTT else if Diff > β (too much extra data) decrease Congestion Window linearly in next RTT
6. else leave Congestion Window unchanged.

(B) TCP-BIC

TCP BIC stands for Transmission Control Protocol Binary Increase Congestion control. TCP-BIC [7] is mainly developed for high speed networks with high latency known as Long Fat Network (LFN) in RFC 1072. In this algorithm, an implementation of TCP with optimize congestion control algorithm. By using binary search algorithm which finds out the maximum throughput where we maintain the window for long period of time. The BIC makes utilization of two congestion control mechanisms in particular binary search increase and additive increase are for substantial and small congestions window. Mainly binary search mechanism can decrease the packet losses and to make sure the RTT fairness and quicker convergence. It integrates with additive increase known as binary increase mainly for making it bandwidth efficient. It makes use of binary increase scheme and it does not increase the round trip time fairness issue of significant TCP, even as accomplishing the better throughput. The certain set of rules description [11] is given under.

Algorithm

1. Some preliminaries

   β multiplicative decrease factor

   Wmax = cwnd size before the reduction
2. Wmin = β*Wmax – just after reduction
3. midpoint = (Wmax + Wmin)/2
4. BIC performs binary search between Wmax and Wmin looking for the midpoint.

```
5.  while (cwnd != Wmax){
    5.1 If ((Wmin – midpoint) > Smax)
        cwnd = cwnd + Smax  → Additive Increase
    5.2 else If ((Wmin – midpoint) < Smin)
        cwnd = Wmax
    5.3 else
        cwnd = midpoint → Binary Search
6.  If (no packet loss)
    Wmin = cwnd
    6.1 else
        Wmin = β*cwnd
    6.2 Wmax = cwnd
        midpoint = (Wmax + Wmin)/2
        }
    6.3 while (cwnd >= Wmax)
        {
7.  If (cwnd < Wmax + Smax)
    cwnd = cwnd + Smin → Slow Start
    7.1 else                                   Max
    cwnd = cwnd + Smax → Additive Increase     Probing
    8. If (packet loss)
        Wmin = β*cwnd
        Wmax = cwnd
        }
```

(C) TCP-CUBIC

In TCP CUBIC which is much lesser aggressive and additional systematic derivative of BIC that is developed for Long Fat Network (LFN), wherein the windows are a cubic role of time for the reason that previous congestion instance, with the variation of points set to the window priority to the instance. TCP CUBIC is implemented with an associate optimize congestion control set of rules for highly bandwidth network with excessive latency. [8] TCP CUBIC is utilized by default Linux kernel from version 2.6.19 and Android.

TCP CUBIC is higher model of BIC Algorithm. Its congestion window growing feature is designed to make simpler and improve the window controller of BIC. This algorithm makes simple to BIC window control and improve RTT fairness and good bandwidth utilization for small bandwidth delay network. Cubic is mainly increasing cwnd slower around loss events. The Cubic characteristic guarantees intra protocols fairness the various competing flow of the same protocols. It is having better throughput while size of buffer is less than bandwidth delay. CUBIC can show off slow convergence by following network disturbances inclusive the begin up of latest flow. The special set of rule description is given underneath [11].

Algorithm

1. If ACK received

   Cwnd = C. $(T-K)^3$+Wmax

   Cwnd is a congestion window.

   Wmax is the window size just before the last window reduction.

   T is the elapsed time from the last window reduction.

   K is updated at the time of last lost event.

   C is a cubic parameter scaling factor.

2. Recovery: Packet Loss

   $K = \sqrt[3]{\beta . W\max/C}$

   β is a constant window multiplication decrease factor

3. Update K:

   K = β.Wmax/C

4. Update Wmax : Wmax= β .Wmax

(D) TCP-NEWRENO

In TCP New Reno we incorporated some minor modifications over the TCP Reno. It is capable to detect multiple packets loss. In this algorithm, mainly we introduce enhanced Fast Recovery (FR) algorithm if we observe that RENO experience which can be avoiding several of the retransmission timeout activities and get recover more than one loss in a single window of data. TCP selective acknowledgement (SACK) possibility was projected to permit receiver to acknowledgements (ACKs) out of order information. TCP NewReno is used that with selective acknowledgement Transmission Control Protocol (TCP) sender could get recover multiple loss more rapidly comparison in to with NewReno. In this algorithm, we modify Reno fast recovery conduct on receipts of a non-duplicates acknowledgement, by identifying stuck between a partial acknowledgements and full acknowledgements.

TCP NewReno [5] is mainly define by RFC 6582 and improvement of retransmission for the duration of FR approaches of TCP Reno. For the period of fast recovery for each duplicate ACK that is return back to TCP NewReno and novel unsent packets from the finish of the congestion window is transfer in to the hold on transmit window occupied. TCP NewReno suffered from the fact that it takes one RTT to detect each packet loss. When the acknowledgements (ACKs) for the first retransmitting section is receive only then we can deduce which others section was not having. The special algorithmic description is given under [11].

Algorithm

1. TCP NewReno = Reno + Enhanced Fast Recovery (FR) Algorithm
2. Fast Recovery (FR) Algorithm:

   Set ssthresh = cwnd/2

   Where cwnd is congestion window.

   ssthresh is Slow start threshold.

3. If (cwnd = ssthresh)

   cwnd = cwnd/2

4. for each duplicate acknowledgement (dup ACK)

dup ACK = dup ACK + 1

5. Maximum Window =Min (cwnd+dup ACK, awnd)
   Awnd is a advertised window from receiver
6. If new data receive acknowledgement (ACK)
   Set dup ACK = 0
   Else
   Exit Fast Recovery
7. When Retransmission timeout (RTO) Expires
   cwnd = 1

(E) TCP-WESTWOOD

TCP Westwood (TCPW) is a congestion control technique [6] mainly design to worked for both wired and wireless network. TCPW works as TCP Reno or NewReno but with little modifications. TCPW losses cause the window to be reset to the sender estimation of the bandwidth delay creation, which is the lowest measure round trip time instance from the observed amount of receiving acknowledgement. TCPW is only sender side simplest modifications to TCP NewReno. So it is better to manage large bandwidth delay with possible packets losses because of transmissions or other errors [4]. TCP Westwood is calculating suitable rate to update the cwnd from the round trip time and received acknowledgement. In this algorithm, congestion window (cwnd) increases or decreases entirely based on estimation of bandwidth. In brief, the information bandwidth estimation is employed to set slow start threshold (ssthresh) and congestion window (cwnd) value with regards to the estimation of bandwidth as shown in *Fig. 4.* The description specific algorithm [11] is given under.

Algorithm

1. TCPW works as TCP Reno but with smallest changes.
   1.1. After 3 ACK losses or congestion occure
        ssthresh = (BWE * RTTmin)/seg_size;
2. If (cwnd > ssthresh) cwnd = ssthresh;
3. Timeout expiration
   ssthresh = (BWE * RTTmin)/seg_size; (minimum 2)
   cwnd = 1;

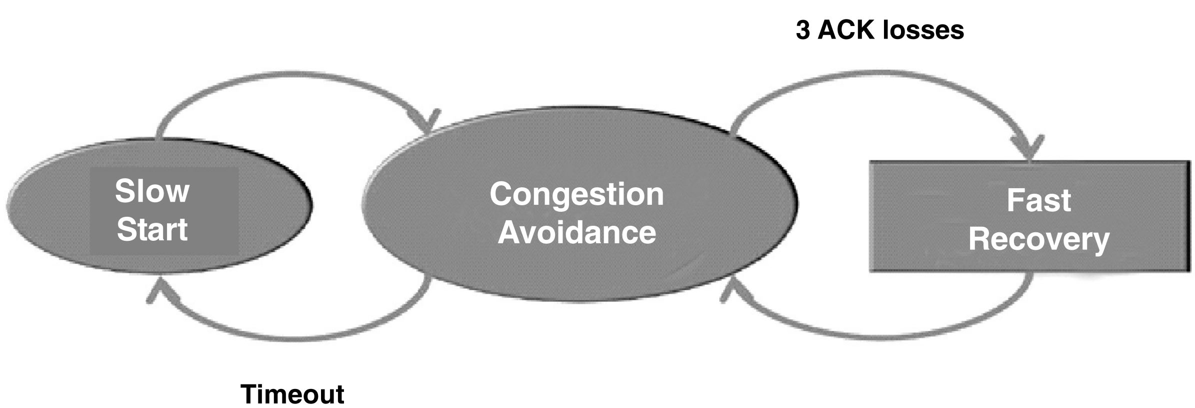

Figure 4: TCP Westwood Algorithm

(F) TCP-COMPOUND

TCP Compound is combined both of loss based and delay based technique of congestion avoidance. It mainly focuses on efficiency and friendliness and maintain the delay window and congestion window used to determine send window. It keeps the congestion avoidance and slow start part. Mainly two phases dwnd (delay window) and cwnd (congestion windows) are the variable to enforce TCP-Compound. But, TCP-Compound can also send both cwnd and dwnd packets in a single Round Trip Time RTT (in place of one packet) whenever dwnd represent the delay window that control delay based element. TCP Compound [10] is mainly having the advantage of fast ramp up, more fair to flow with different RTT and disadvantage must be estimated RTT, which is very challenging. The designated [11] set of description is given below in this algorithm.

Algorithm

1. TCP sending window:
   Win = min (cwnd+dwnd, awnd)
   cwnd- congestion window
   dwnd- delay window
   awnd- advertised window from receiver
2. cwnd is updated as in conventional TCP.
3. if ACK received,
   cwnd (T+1) = cwnd(T) + (1/win(T))
4. if loss detected,
   cwnd(T+1) = cwnd(T)/2
5. Expected=win/base RTT
6. Actual= win/RTT
7. Estimates the bottleneck queue size:
   Diff= (Expected – Actual) *base RTT

III. RESULTS AND ANALYSIS

In this phase, in depth evaluation to be done to observe the TCP behavior factors and its overall performance under the vehicular environments. The simulation setup has been cautiously selected to resemble the practical nature of vehicular surrounding. So many experiments were carried out and the simulations become taken in this pattern, the primary set of experiments is to analyze the behavioral of the TCP for the quantity of nodes with respect to time. For analysis, following three metrics such as average packet drop, average delay and throughput has been taken. These results are carried out in [16] NS2 Simulator and following parameters for NS2 has been selected.

Table I
Simulation Parameters in NS2

Simulation Parameters	
Number of Nodes	50,100
Propagation Model	Two Ray Ground
Antenna Type	Omni Directional Antenna
Routing Protocol	AODV
MAC	802.11
Packet Size	200
Standard TCP Variants	Vegas, BIC, NewReno, Cubic, Compound, Westwood
Simulation Area	1000*1000

(A) Quality of Service for TCP Variants

Quality of Services (QoS) analysis to be made to examine the TCP behavior component for vehicular environment. We have analyzed overall performance of TCP for number of nodes and behavior of the TCP with variation of time. QoS is measured with respect to three metrics i.e. average packet drop, throughput and average packet delay.

1) Average Packet Drop: Packet loss takes place while one or more number of packet of record visiting through a computer networks fails to reaches the destination. Packet loss is commonly due to congestion of network [11]. Packet loss is measured as a percentage of packet losses w. r. t. packet send. The TCP detects packets loss and perform retransmission to make certain reliability of messaging. In TCP connection the packet loss is likewise use to avoid the congestion and decreases the connection of throughput.

As shown in *Fig.6.* we have seen that comparison between average packets drop w.r.t. various number of nodes. Implementation of standard TCP variants such as TCP Binary Increase Congestion (BIC), TCP Vegas, TCP New Reno, TCP Cubic, TCP Compound and TCP Westwood under vehicular environment shows that TCP Vegas is having minimum average packet drop in comparison with TCP Standard variants techniques. TCP Vegas is having low average packet drop because of proactive in nature.

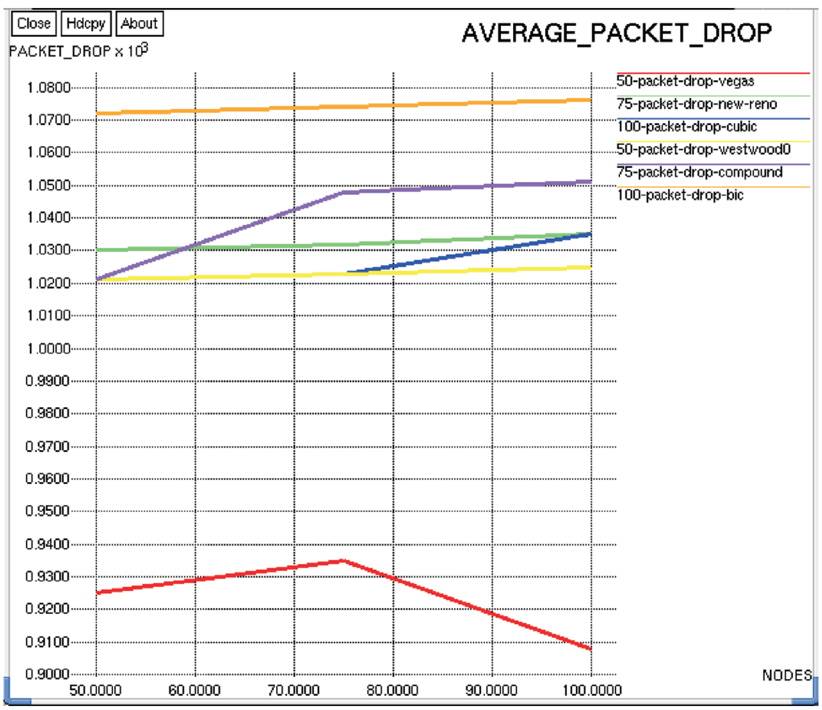

Figure 6: Comparison Between Average Packet Drop Vs Number of Nodes

2) Average Delay: Average packet delay is a measure of overall time taken for an information packet transferred from source nodes to the destination nodes. In this place variety of node is extended, the average delay also increases because of the related delay together with queuing delay, contention delay and rerouting delay additionally will increase for all the standard TCP variant except TCP Vegas. In all TCP variants, it is observed that TCP Vegas is very minimum average delay because of it is pro-active nature.

As shown in *Fig.7.* we have seen that comparison between average delay is with respect to various number of nodes. This shows that the implementation of all standard TCP variants such as TCP Vegas, TCP BIC, TCP New Reno, TCP Westwood, TCP Compound and TCP Cubic. We have observed that TCP Vegas has minimum average delay in comparative to another TCP standard variants techniques.

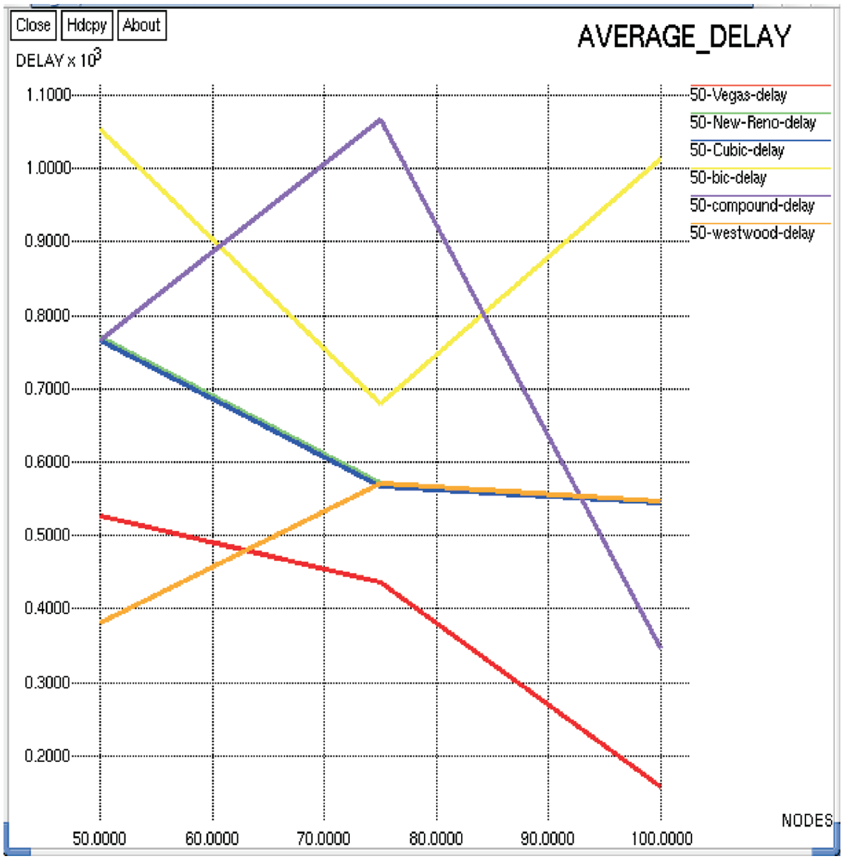

Figure 7: Comparison Between Average Delay Vs Number of Nodes

3) Throughput: Throughput is a measurement of the entire number of efficient packet successful delivery from source to the destination over a time period. Throughput is normally measure in bit per seconds (bps or bit/s), megabit per seconds (Mbps) or gigabit per seconds (Gbps) but sometimes data packets measured in packet per second (pps or p/s) or data packet per time period. The aggregate throughput or device throughput is the summation of data rate which might be deliver to all terminal in a networks. In spite of data transmission, networks throughput is the quantity of data successfully delivered from one region to another region in some given particular time periods.

As shown in *Fig.8.,* we have seen that comparison between throughput is with respect to various number of node shows the implementation of all standard TCP variants such as TCP-Vegas, TCP-Cubic, TCP-NewReno, TCP-Westwood, TCP-Compound and TCP-BIC with respect to throughput of varying number of nodes under the vehicular environment. All the standard TCP variants shows the performance of throughput. TCP Vegas is showing maximum throughput in 100 number of nodes as compared to other TCP variant.

(B) Quality of Services (QoS) for 50 Nodes

1) Throughput: As shown in *Fig. 9.,* throughput of all standard TCP variants w. r. t. variation of particular time. So we have taken first scenario of 50 number of nodes. All the standard TCP variants shows TCP Vegas is showing maximum throughput in comparison with all other TCP variants techniques.

2) Average Delay: In this *Fig.10.,* average delay of all standard TCP variants with respect to variation of particular time. So we have taken first scenario of 50 number of nodes. All the standard TCP variants shows the performance of average delay under the vehicular environments. So in comparison with another TCP variants TCP westwood and vegas techniques is showing minimum average delay.

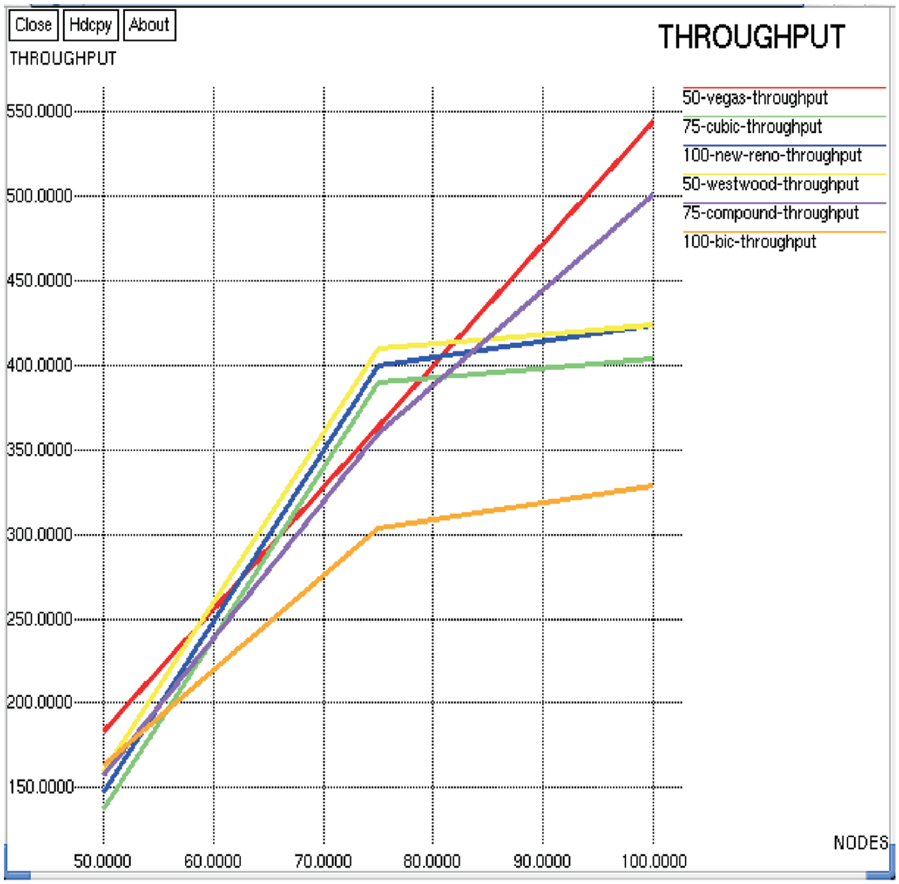

Figure 8: Comparison Between Throughput Vs Number of Nodes

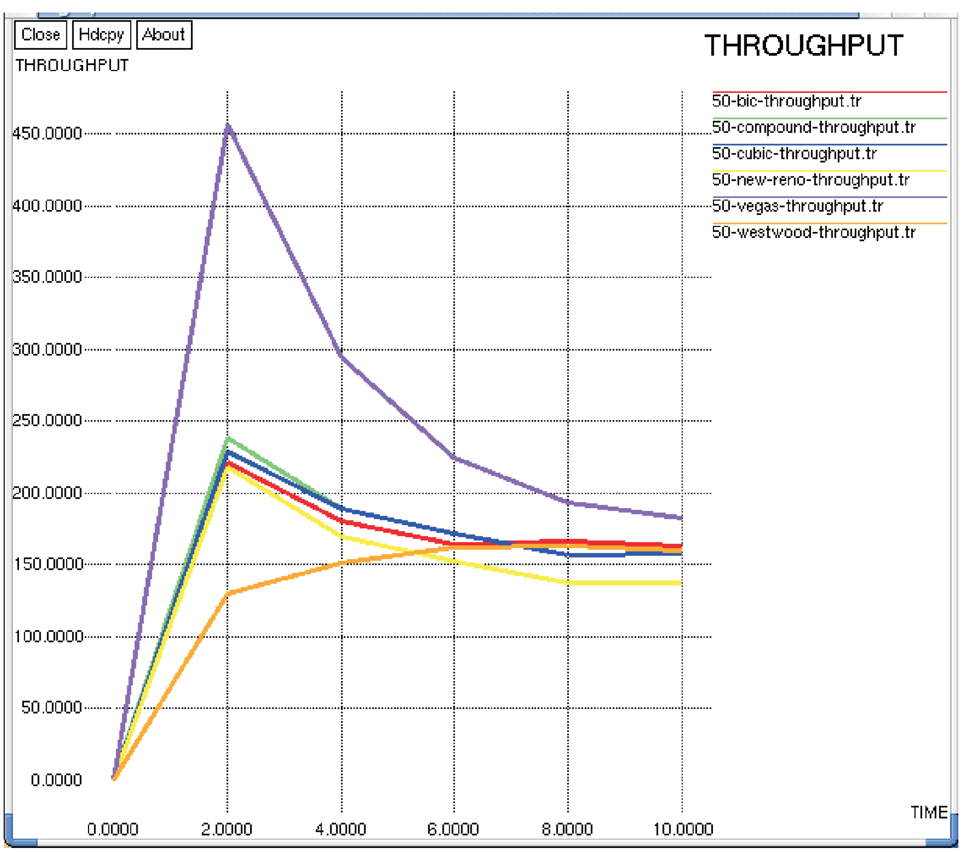

Figure 9: Throughput w.r.t Number of 50 Nodes

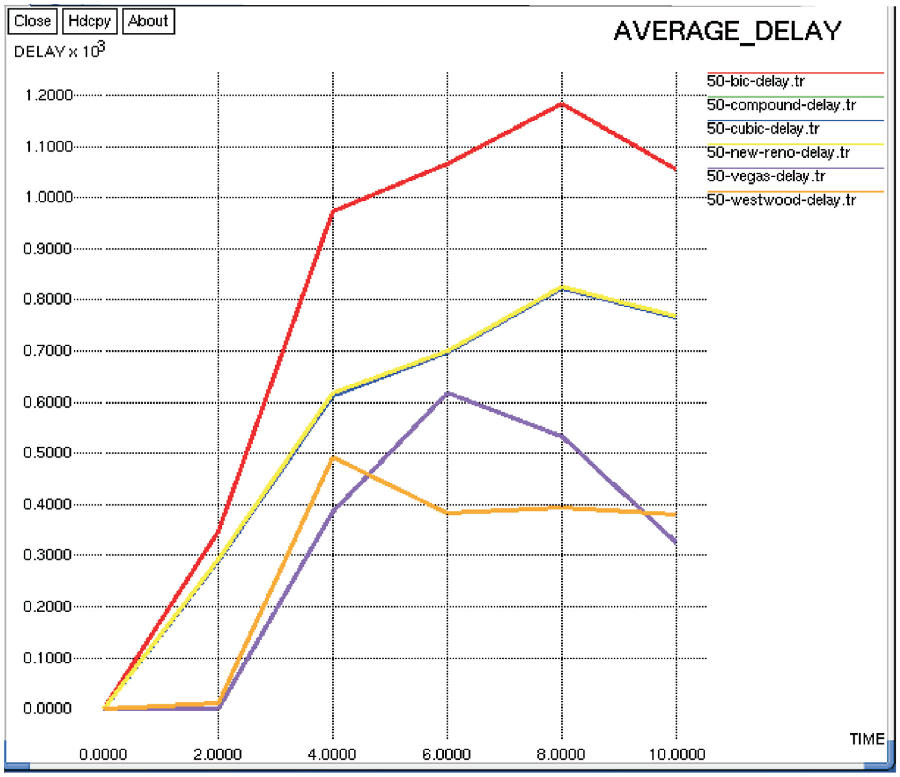

Figure 10: Average Delay w.r.t Number of 50 Nodes

3) Average Packet Drop: As shown in *Fig.11.,* average packet drops for all standard TCP variants with respect to variation of particular time. So we have taken first scenario 50 number of nodes. All the standard TCP variants shows the performance of average packet drop under the vehicular environments has been shown in *Fig.11.* TCP Vegas is showing minimum average packet drop in comparison with all other TCP variants techniques.

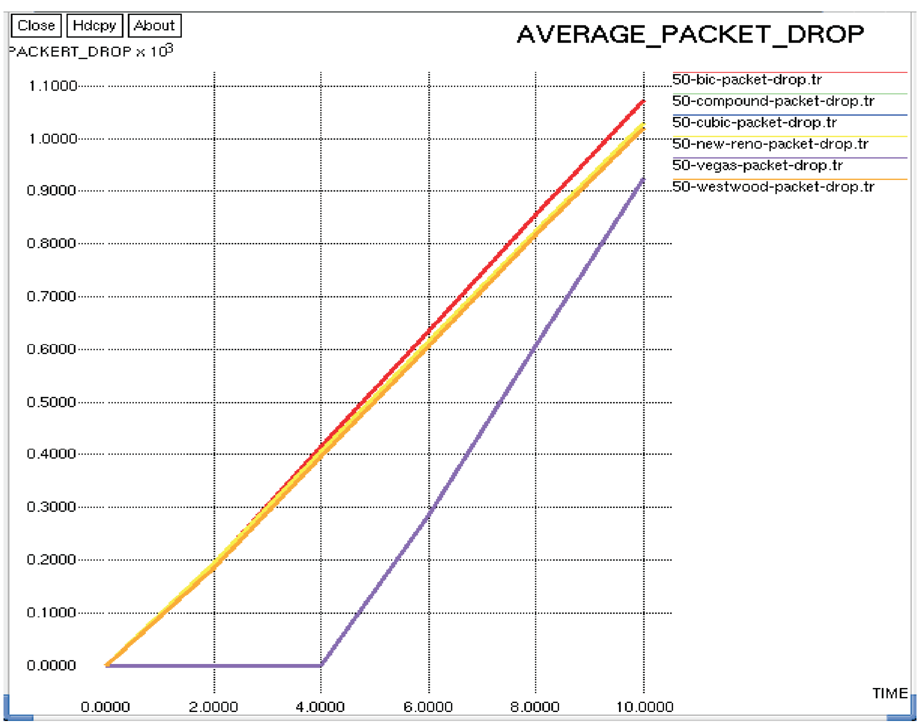

Figure 11: Average Packet Drop w.r.t Number of 50 Nodes

(C) Quality of Services (QoS) for 100 Nodes

1. Average Delay: Fig. 12. shows that average delay of all standard TCP variant w. r. t. variation of particular time. So we have taken second scenario more number of node numerically taken number of nodes 100. All the standard TCP variants shows the performance of average delay under the vehicular environments has been shown in Fig. 12. TCP Vegas is showing minimum average with all other TCP variants technique.

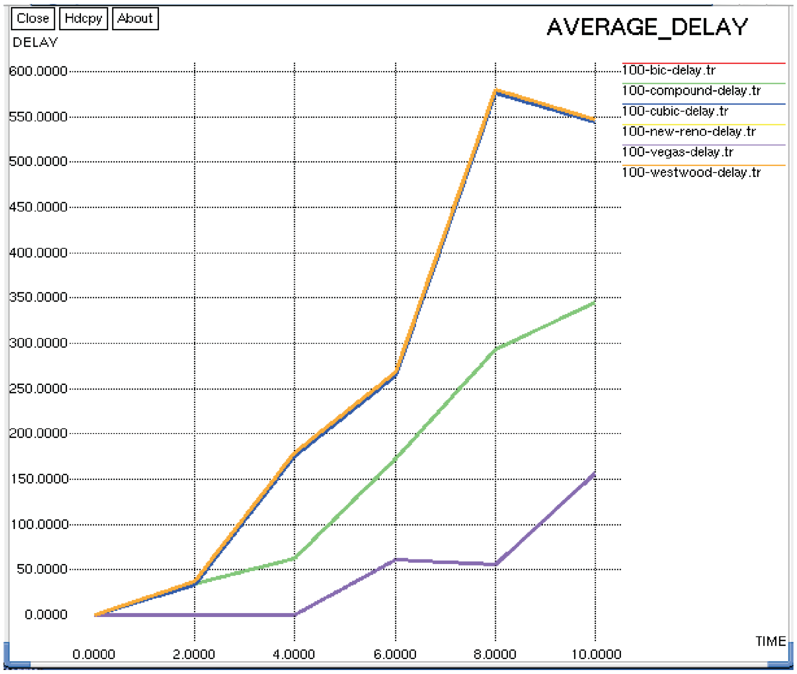

Figure 12: Average Delay w.r.t Number of 100 Nodes

2) Average Packet Drop: As shown in Fig. 13. average packet drops of all standard TCP variant with respect to variation of particular time. So we have taken second scenario 100 number of nodes. All the standard TCP variants shows the performance of average packet drop under the vehicular environments has been shown in Fig. 13. TCP Vegas is showing minimum average packet drop with all other standard TCP Variants techniques.

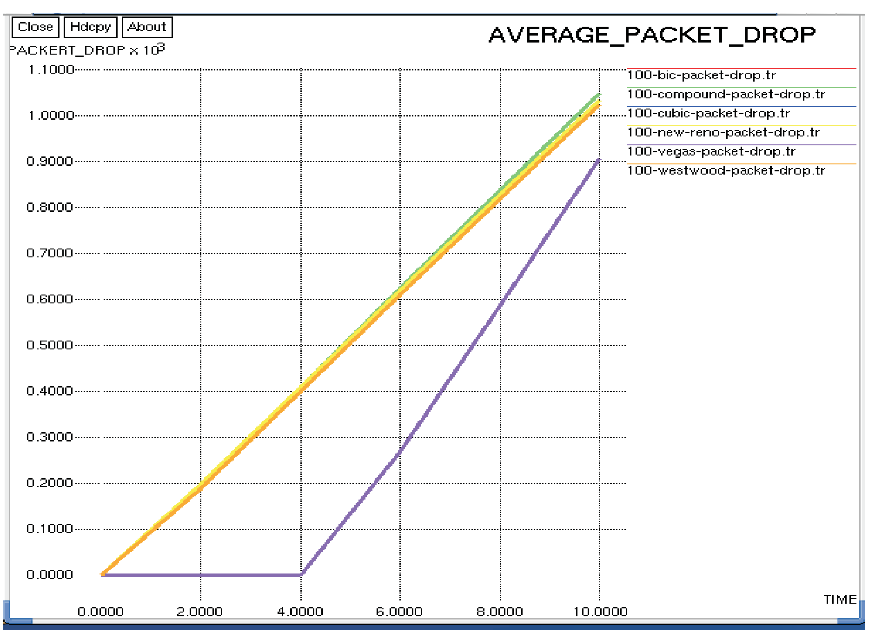

Figure 13: Average Packet Drop w.r.t Number of 100 Nodes

3) Throughput: As shown in Fig. 14. throughput of all standard TCP variant w. r. t. variation of particular time. In second scenario we have taken number of nodes 100. All the standard TCP variants shows the performance of throughput under the vehicular environments has been shown in *Fig.14.* TCP Vegas show maximum throughput with comparative to all standard TCP variants.

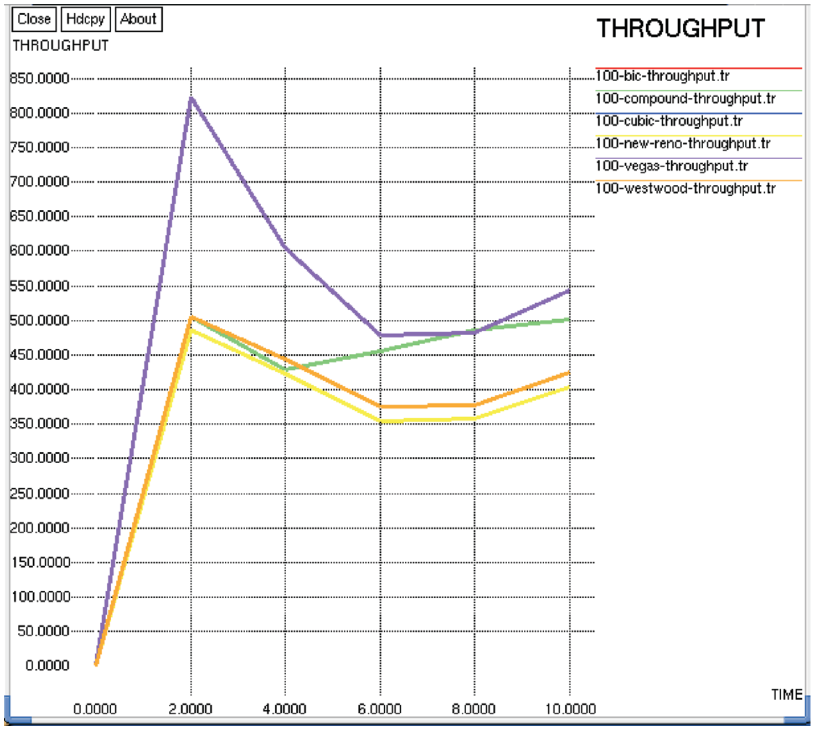

Figure 14. Throughput w.r.t Number of 100 Nodes

CONCLUSION

In vehicular environment, evaluation offers a perfect representation about the vulnerability in implementation of all standard TCP variant. We have seen that Transmission Control Protocol (TCP) overall performance of average packet drop, average delay and throughput for various number of nodes. If you see for vehicular environment, we observed that one of the best technique is TCP Vegas because average packet drops and average delay is minimum in various number of nodes under the vehicular surroundings and throughput is also maximum in TCP Vegas technique. The resulting conclusion become drawn from the analysis for complete scale deployments of internet architecture in Vehicular Ad hoc Networks. Firstly, Design TCP need to be nature in cross layer, so that it will rapidly adapting the vehicular networks conditions and development of TCP with developing congestion notifications at the wireless nodes so that it can readjusting the sending rate. Secondly, TCP must be design to counter the receivers buffering contention and buffer overflowing at the intermediate wireless nodes.

REFERENCES

[1] S. Bhuvaneshwari, G. Divya, K.B. Kirithika and S. Nithya, " A Survey on Vehicular ad-hoc Network" IEEE International Conference on Communications, ICC'00, New Orleans, LA, pp. 1697–1701, 2013.

[2] S.A. Sultan, M.M. AlDoori, A. H. AlBayatti and H. Zedan, "A Comprehensive Survey on Vehicular Ad Hoc network" Elsevier Journal of Network and Computer Applications, vol.37, pp. 380–392, 2014.

[3] S. Zeadally, R. Hunt, Y.S. Chen, A. Irwin and A. Hassan, "Vehicular ad hoc networks (VANETS): status, results, and challenges" © Springer Science+Business Media, LLC, pp. 217-241, 2010.

[4] A. Macura, E. Missoni and Z. Kordic, " Comparison of Westwood, New Reno and Vegas TCP Congestion Control" Published by DAAAM International, Vienna, Austria, EU, pp. 151-154, 2012.

[5] T. Henderson, S. Floyd, A. Gurtov and Y. Nishida, "RFC 6582: The New Reno Modification to TCP's Fast Recovery Algorithm" Internet Engineering Task Force, pp. 1-16, 2012.

[6] C. Casetti, M.Gerla, S. Mascolo, M.Y. Sanadidi and R. Wang, " TCP Westwood: End-to-End Congestion Control for Wired/Wireless Networks" Springer Journal of Wireless Networks, pp. 467 – 479, 2002.

[7] X. Lisong, K. Harfoush and I. Rhee, "Binary Increase Congestion Control (BIC) for Fast Long Distance Networks" In Proceedings of IEEE INFOCOM, pp. 2514-2524, 2004.

[8] S. Ha, I. Rhee and L. Xu, "CUBIC: A New TCP-friendly high-speed TCP variants"ACM SIGOPS Operating Systems Review, pp. 64-74,2008.

[9] L.S Brakmo and L.L. Peterson, "TCP Vegas: end to end congestion avoidance on a global Internet" IEEE Journal on Selected Areas in Communications, vol.13, no.8, 1465-1480, 2006.

[10] K. Tan, J. Song, Q. Zhang and M. Sridharan, "A Compound TCP Approach for High-speed and Long Distance Networks" In Proceedings of IEEE INFOCOM 06, pp. 1-12, 2006.

[11] J.A. Jude, R. Ganesan, "A Comprehensive experimental analysis of standard TCP Variants in vehicular environment" IEEE International conference on computing and communication technologies, pp. 350-355, 2015.

[12] G. Karagiannis, O. Altintas, E. Ekici, G. Heijenk, B. Jarupan, K. Lin, T. Weil, "Vehicular networking: A Survey and tutorial on requirements, architectures, challenges, standards and solutions" IEEE Communications Surveys & Tutorials. pp. 584–616, 2011.

[13] M. Gerla and L. Kleinrock, "Vehicular networks and the future of the mobile internet. El Vehicular networks and the future of the mobile internet" Elsevier Computer Networks, pp. 457–469, 2011.

[14] M. Allman, V. Paxson and W. Stevens, " TCP Congestion Control, RFC 2581" Network Working Group, pp. 1-14, 1999.

[15] C. Callegari, S. Giordano, M. Pagano, T. Pepe, "Behavior Analysis of TCP Linux Variants" Elsevier Computer Networks, vol. 56 ,pp. 462–476, 2012.

[16] E. Hossain and T. Issariyakul, "Introduction to Network Simulator NS2" Published book by Springer Science+Business Media, pp. 1-512, 2008.