PERFORMANCE ENHANCEMENT OF TCP VEGAS OVER WIRELESS NETWORK

Mansi Vaidya[1]
Department of EC Engineering
V.V.P. Engineering College
Rajkot, India
vaidya.mansi121@gmail.com

Jignesh H. Joshi[2]
Department of EC Engineering
V.V.P. Engineering College
Rajkot, India
joshijh@gmail.com

Abstract—**In wireless network packet loss is the major issue. So it is necessary to find out the reason behind packet loss whether it is due to congestion or bit error. TCP gives congestion avoidance algorithm to reduce the loss of packets which are due to congestion. Congestion avoidance mechanism that TCP Vegas uses is quite different from that of TCP Tahoe and TCP Reno. Vegas is not able to identify the reason behind the packet loss whether it is due to congestion or bit error. Unnecessarily size of congestion window is decreased when loss is occurred due to bit error. In this paper, the error model is introduced in different variants of TCP. The calculation of dropped packets is done for TCP New Reno, Vegas and Sack. Results are taken for different error rates.**

Index Terms— Mobile ad-hoc network, TCP, TCP Vegas, fairness of bandwidth.

I. INTRODUCTION

The TCP protocol is used by the mainstream of the network applications on the Internet. TCP performance is strongly influenced by its congestion control algorithms that limit the amount of transmitted traffic based on the probable network capability and utilization. TCP is commonly used protocol of Transport layer, which proposals a trustworthy byte stream service. Transparent segmentation and reassembly of user data are provided by TCP and it also handles flow and congestion control. TCP packets are cumulatively acknowledged as they arrive in sequence. But the TCP packets which are received randomly at the receiver causes duplicate acknowledgments while the procedure of transmission. The concept of duplicate ack is advantageous in TCP. In [1] the TCP assumption that all losses are due to congestion becomes quite problematic over wireless links. In order to adapt TCP to wireless network, perfections have been proposed in the literature to help TCP to give difference among the different types of losses. Indeed, in mobile or static Ad hoc networks losses are not always due to network congestion, as it is mostly in the case in wired networks.

TCP is reliable end to end protocol because TCP is trying
to provide reliable data transmission between two entities. Transmission Control Protocol (TCP) is the predominant Internet protocol and it carries approximately 90 percent of internet traffic in today's heterogeneous wireless and wired networks. TCP primary purpose is to provide a connection oriented reliable data transfer service between different applications to be able to provide these services on top of an unreliable communication system. TCP needs to consider data transfer, reliability, flow control, multiplexing, congestion control and connection management. Strategy of several variants of TCP is based on properties of wired network.

In [2], the complications in wireless network is associated. In this wireless network, the principle problem associated with TCP lies in execution is that to protect the segments from congestion in case of harms that are not persuaded by network congestion. Since bit error rates are very low in wired networks, nearly all TCP versions now a day assume that packets losses are due to congestion. When detection of lost packets are done, either by timeout or by multiple duplicated ACKs, TCP tries to avoid or to control the congestion and will decrease the rate of flow.

TCP over mobile networks recognised ensuing problems:

1. Signal attenuation and Multipath fading are the actual reasons for generating lossy channels.
2. Causes of Route irregularity: Bandwidth asymmetry, loss rate asymmetry, Direction asymmetry
3. Direction-finding failures
4. Power controls

Now a days TCP is heavily used in internet. So after the use of TCP, it is possible to provide reliability in mobile networks is necessary in order to achieve a smooth integration with the wired Internet. The various congestion avoidance algorithms are included in [3] and [4].

In [5], the explanation about the various variants of TCP are provided. Effective procedure done by Tahoe is that, when the process is in running state, it protects the packets. There is no any measurement related to bandwidth like Vegas. As compared to Tahoe, Vegas performs better because it is much more robust in the sense of the lost packets. It can detect and retransmit lost packet much sooner than timeouts in Tahoe. Only one lost packet is detected and retransmitted before timeouts by the TCP Reno. There is no any maturity level is performed to reduce the size of congestion window. Bandwidth utilization is better than Tahoe. Modification in the avoidance algorithm fast recovery and fast retransmission is

Reactive type TCP Variants	**Proactive type TCP Variant**
Based on Congestion detection	Based on congestion avoidance
Congestion is detected after causing it	Congestion is detected before causing it
As a feedback signal, packet loss is used	As a feedback signal packets delay is used
Use Corse - grained timers	Use Fine - gained timers
TCP Tahoe, TCP Reno, TCP New Reno are examples	TCP Vegas is an example
Less accurate Round Trip Time estimation	More accurate Round Trip Time estimation

kept in TCP New Reno. Whenever multiple packets are lost from a window during transmission, the wait for a retransmit timer is eliminated. TCP Vegas doesn't have a clear cut advantage over SACK TCP. Only few fields are there where it appears to outperform SACK. Vegas gives better utilization of bandwidth and lesser congestion as compared to SACK. TCP Vegas flattens out its sending rate at the optimal bandwidth utilization point thus including stability.

This paper is organized as follows: Introduction is provided in Section I. The overview of basic algorithm of Vegas is described in Section II. Section III includes the literature survey on Vegas. Simulation and results are included in section IV. Finally section V presents the conclusion and future work.

II. OVERVIEW OF THE BASIC ALGORITHM TCP OF VEGAS

Vegas is an exceptional operation that interoperates with any other legal implementation of TCP, and all changes are controlled to the sender side. The key algorithms that were broadcasted in [5] by Vegas are:

a. Vegas- New retransmission mechanism
The calculation of RTT is done using the recorded time when the packet is sent and when an ACK arrives. The perfect RTT and the decision of retransmission is taken as follows:
- After acceptance of the duplicate acknowledgement, if the fresh RTT is found greater than RTO, the retransmission is occurred by the Vegas without having wait for the third duplicate acknowledgment.
- After receiving non duplicate acknowledgement Vegas treats the receipt of certain ACKs as a trigger to find if a timeout should happen.

b. Vegas – New congestion avoidance mechanism
The critical indication of TCP Vegas was using packet delay as congestion pointer instead of packet loss which have been used for congestion indication in loss based congestion control algorithms. The adjustment of congestion window is done in TCP Vegas by given set of rules:

$$CWND = \begin{cases} CWND+1 & \text{if } delta<\alpha \\ CWND & \text{if } \alpha<delta<\beta \\ CWND-1 & \text{if } \beta<delta \end{cases}$$

Where
- Delta is calculated as the difference between expected rate and actual rate
- Expected rate is the ratio of cwnd (t) to the base rtt, where cwnd (t) is defined as current congestion window size and base rtt is defined as the minimum of RTT of related connection.
- Actual rate, which is calculated as the ration of cwnd (t) to rtt, where rtt is the actual roundtrip time.
- α and β are parameters which their ideals are typically set as 1 and 3, respectively.
  The indication of underutilizing the available bandwidth is delta<α, means the actual rate is smaller than it is expected, the congestion window linearly increased. Opposite to that the congestion window linearly decrease when delta>β.the congestion window remains same when delta is in between α and β.

c. Vegas – Modified slow start mechanism
During slow start to be able to detect and avoid congestion, the exponential growth is allowed by Vegas only every other RTT. In between, the congestion window stays fixed so a valid comparison of the expected and actual rates can be made. The γ threshold – Vegas changes from slow start made to linear increase/decrease mode when the actual rate falls down to expected rate by certain amount.

III. REVIEW OF LITERATURE

In this we will discuss about the research work of different authors, how they use the protocol to solve the problems like unfairness, bandwidth consumption and long delay which degrades its performance.

In [7] authors proposed an end host adaption to the congestion avoidance mechanism of original TCP Vegas. TCP Arta Vegas, manages a better concert than TCP Vegas in both instantaneous and non-simultaneous situations. Authors also looked at the fairness problem associated with TCP-Vegas. The simulations designated that TCP-Arta Vegas is able to compete better against TCP Reno, while at the same time it will also increase the throughput of TCP Vegas.

As described in [8] to develop the compatibility and fairness of Vegas, Adaptive Vegas which is based on the adaptive parameters, has been presented. By the experiment results the authors illustrated the fairness of adaptive Vegas algorithm coexisting with Reno in actual network scenario. As compared to Vegas the compatibility of Adaptive Vegas is better than Vegas, while the main features of Vegas on packet loss rate, delay and delay jitter remain.

In [9] authors analysed the problems when TCP Vegas is applied in mobile ad hoc network, offered a new algorithm TCP Vegas-HA which includes the modified congestion control algorithm based on TCP Vegas, TCP Vegas-HA can significantly increase the network throughput. The empirical values may be changed due to the different network scenario which means there is a requirement to test and gain the empirical values for each particular network before the application of TCP Vegas-HA, so TCP Vegas-HA will be more perfect if an automatic test scheme is designed. A new extenttechnique of base RTT using the realistic value is assessed. The result of simulation express that the algorithm has decentflexibility with route buffers changing while hops differs, and can successfullyrise the TCP Vegas throughput in mobile ad hoc network.

In [10] due to the need for accurate Base RTT estimations in MANETs, we proposed TCP-Vegas-ad hoc. It allows a TCP sender to acquire a new base RTT without the use of additional explicit control messages whenever an RC occurs. When an intermediate node detects an RC, its occurrence is piggybacked into the IP header of an on-the-fly packet toward the TCP receiver. It enables the TCP receiver to inform the TCP sender of the RC through its TCP ACK, which finally allows the TCP sender to acquire a new Base RTT. With the help of using ns-2, it can be shown that TCP-Vegas-ad-hoc has given comparatively better results than the typical TCP-Vegas. In particular, it improved the performance greatly (up to 20%) at high node mobility. Although our work evaluated TCP Vegas- ad hoc by using OLSR and AODV as its underlying proactive and reactive routing protocols.

In [11] authors proposed the simulation results taken from evaluating of TCP Vegas over large bandwidth delay network model with different parameters. Rendering to numerous studies, TCP Vegas does not accomplishgreater efficiency than other TCP variants, causes much rarer packet retransmissions, and is not based against the connections with longer RTTs. The authors proved that the important parameters of congestion window algorithm for TCP Vegas could not produce a competitor performance with default parameters values, especially in high bandwidth large delay networks, but when the values of alpha and beta changed, the high performance notified to the degree of vulnerability.

In [12] authors said that The modified TCP Vegas (Vegas - A) is able to obtain a fairer share of the network bandwidth when competing with other TCP flows. Vegas – A preserves the properties of Vegas that have made it a noteworthy protocol. The main idea in Vegas-A is that, rather than fixing α and β they be made adaptive. At the start of connection, α is set to 1 and β to 3. These values however change dynamically depending on the network conditions. The slow start and congestion recovery algorithms of Vegas-A are the same as that of Vegas. Vegas – A overcomes the rerouting limitations of Vegas and is able to adapt to the changes in RTT and router faster; Vegas – A connections do not suffer from the unfairness towards old connection and unfairness against higher bandwidth connections problems of Vegas. It was also shown that even though Vegas-A is different from Vegas, the basic congestion control algorithm still remains as effective as Vegas in decreasing the average queue occupancy and packet retransmission.

In [13] authors proposed Vegas-W, which considers the special features of wireless ad hoc networks and improves legacy Vegas in four aspects: fractional window support, slower start, moderate congestion avoidance and *Wth* update, respectively. The simulation results show that Vegas-W obtains higher throughput than Vegas up to 87% and outperforms FeW up to 27%. The main reasons lie in Vegas's large minimum congestion window, large reset slow start threshold and aggressive window increase policy. To fix these problems, we propose a modified TCP protocol based on TCP-Vegas for multi hop ad hoc networks, called Vegas-W. We extend the congestion window to fraction; change the probing mechanisms of legacy TCP-Vegas in both slow start and congestion avoidance and update slow start threshold tracking the stable window. Extensive simulation results under a variety of scenarios show that Vegas-W can improve the throughput up to 87% over legacy TCP-Vegas and up to 27% over FeW, which is another improved algorithm based on TCP-New reno scenarios.

IV. PROBLEM DEFINITION

As per the literature survey, many modifications have been done in the algorithm of Vegas.But still there is a requirement to improve the packet delivery ratio in Vegas.Vegas tries to avoid the congestion after the loss of packets. Basic idea of Vegas is that when Vegas found congestion it will decrease the size of congestion window as per algorithm. There is no any technique from which the identification is possible that whether the packet loss is due to congestion or bit error. So if packet loss is due to bit error than also Vegas will decrease the congestion window. So the problem is there are unnecessarily packets are dropped. So some modification has been needed in Vegas to solve this problem. To improve the throughput of the system using TCP Vegas, is the objective of this research work. This improvement can be achieved by doing some modification in ordinary Vegas. The main work related to this is to differentiate the congestion and bit error. So the main thing which is to be carried out is to modify the Vegas. This scheme is achieved by setting the R_B bit by the receiver after the packet loss is occurred.Whenever the packet is lost and sender receives dupack, it will check the status of R_B to decide whether the size of congestion window needs to decrease or not.

V. SIMULATION RESULTS

Network Simulator Version 2 is the abbreviation of NS 2. It is an open-source event-driven simulator designed specifically for research in computer communication networks. The ns Simulator covers a very large number of applications, of protocols, of network types, of network elements and of traffic models called as “simulated objects”. Basically the NS2 is collection of all protocols. Protocols are written in .cc file and the .h file is used to change in header format of protocol. NS2 is kernel which is a part of operating system. For any protocol after compiling the .cc file, .o file is generated which shows the output file of protocol.

In any variant of TCP, if we want to calculate the no of packets which are lost due to error or congestion, the error model is used. Error model used in the propose work, simulates link-level errors or loss. This model will simulate it by mainly two methods: Indicating the error flag of segment or The segment is dumped to a drop target. For each and every models, the unit of error can be stated in term of packet, bits, or time-based. Fig. a shows the analysis of dropped packets in Vegas. If 13 packets are dropped during transmission, among them 7 packets are lost due to congestion and 6 packets are lost because of error as shown in fig. a. But as per the algorithm of Vegas, the congestion window is decreased for the lost packets. If packets are lost due to error then there is no need to decrease the size of congestion window. Vegas will decrease the cwnd without distinguish the reason behind packet loss. So to improve the throughput of Vegas, it is compulsory to avoid the decrement in cwnd for lost packets which are due to error.

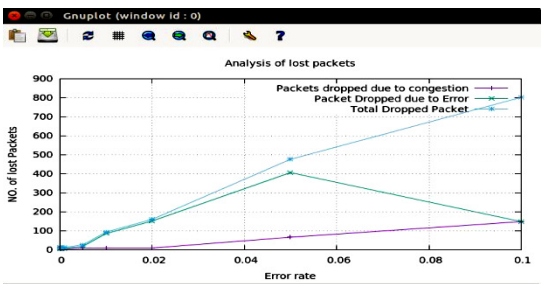

Fig. a

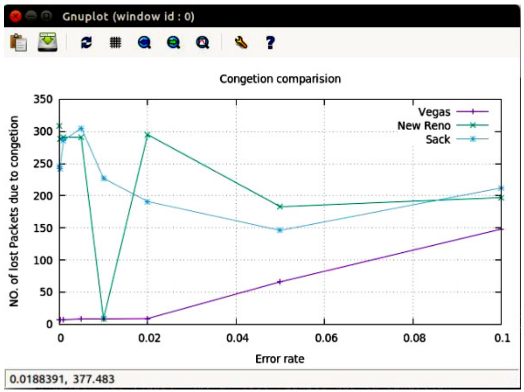

Fig. b

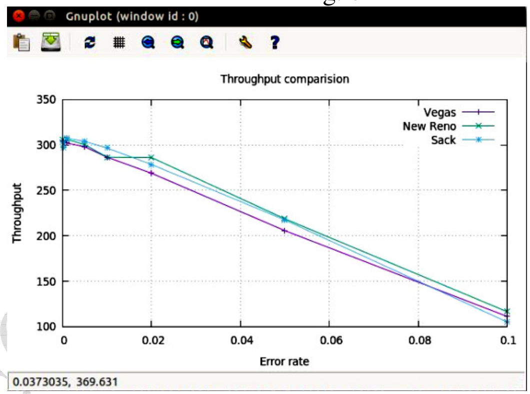

Fig. c

Fig. b shows the no. dropped packets due to congestion for different error rates for Vegas, New Reno and Sack. It can be shown that as compared to New Reno and Sack, dropped packets due to congestion is very less in Vegas. Throughput analysis is shown in fig. c. comparatively throughput of Vegas is less than New Reno and Sack. So to improve the throughput of Vegas it is compulsory to pretend the decrement in cwnd for dropped packets due to error.

VI. CONCLUSIONS

After introducing error model in simulation it is possible to calculate the no. of total dropped packets, packets dropped due to congestion as well as error for different TCP variants. TCP Vegas gives better results compared to TCP New Reno and Sack against congestion. But among the total dropped packets, there are much more packets are dropped due to error. That’s why the performance of Vegas will degrade. New Reno and Sack gives better result for throughput as compared to Vegas. So the conclusion is that Vegas plays a remarkable role to avoid the congestion with its best congestion control algorithm.

VII. FUTURE WORK

To get the remarkable improvement in throughput of Vegas, it is necessary to pretend the decrement in cwnd for lost packets which are due to error. Watch dog timer can be used at the receiver side to catch the lost packets due to error by marking R_B bit as 0 or 1 in header of packet.

REFERENCES

[1]. A. A. Hanbali, E. Altman, and P. Nain, "A survey of TCP over ad hoc networks," Common. Surveys Tuts., vol. 7, no. 3, pp. 22–36,Third Quarter 2005.

[2]. Dr.(Mrs)Saylee Gharge, Mr. Ajinkya Valanjoo, "Simulation based Performace Evaluation of TCP Variants and Routing Protocols in Mobile Ad-hoc Network", IEEE International Conference on Advances in Engineering & Technology Research (ICAETR-2014), August 01-02,2014,Dr. Virendra Swarup Group of Istitutions Unnao, India.

[3]. Dr. Neeraj Bhargva, Dr. Ritu Bhargva, Manish Mathuria, Shilpi Gupta, Kamal Kumar Jyotiyana, "Analysis of Different Congestion Avoidance Algorithms", IRACST-International Journal of Computer Networks Communication (IJCNWC), ISSN:2250-3501 Vol.3, No 1, February 2013.

[4]. RFC 2001 TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms.

[5]. Hardik K.Molia, Prof. Rashmi Agrawal, "A conceptual Exploration of TCP Variants", Emerging Technology Trends in Electronics, Communication and Networking (ET2ECN) 2014 2nd International Conference.

[6]. Suhas Waghmare, Aditya Parab, Pankaj Nikose, Prof. S.J.Bhosle, "Comparative analysis of different TCP variants in a wireless environment" IEEE 2011.

[7]. Nima Afraz, Morteza Analoui, "TCP-Arta Vegas:Improving the Fairness of TCP-Vegas", 2015 23rd Iranian Conference on Electrical Engineering (ICEE).

[8]. Yongjun Guo, Xiaoping Yang, Rui Wang and Jiwei Sun," TCP Adaptive Vegas: Improving of TCP Vegas Algorithm", IEEE 2014.

[9]. Li Huang, Jian-de Lu, "Hops Adaptive Algorithm for TCP Vegas in Mobile Ad Hoc Network", 2011 International Conference on Mechatronic Science, Electric Engineering and Computer August 19-22, 2011, Jilin, China.

[10]. Dongkyun Kim, Hanseok Bae, and C. K. Toh," Improving TCP-Vegas Performance Over MANET Routing Protocols", IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, VOL. 56, NO. 1, JANUARY 2007.

[11]. Wei Zhou, Wei Xing, Yongchao, Jianwei Zhang, "TCP Vegas-V: Improving the performance of TCP Vegas", 2012 Automatic Control and Artificial Intelligence(ACAI 2012), International Conference, Xiamen.

[12]. K.N.Srijith, Lillykutty Jacob, A.L.Ananda, "TCP Vegas-A:Solving the Fairness and Rerouting Issues of TCP Vegas", Computer Communications, pp. 429-440,(2005).

[13]. Lianghui Ding, Xinbing Wang, Youyun Xu, Wenjun Zhang, Wen Chen, "Vegas-W: An Enhanced TCP-Vegas for Wireless Ad Hoc Networks", 978-1-4244-2075-9/08/$25.00 ©2008 IEEE.

[14]. Joel Sing and Ben Soh, "TCP New Vegas: Improving the Performance of TCP Vegas Over High Latency Links", Proceedings of the 2005 Fourth IEEE International Symposium on Network Computing and Applications (NCA'05) 0-7695-2326-9/05 $20.00 © 2005 IEEE.

[15]. M.Jehan, Dr.G.Radhamani & T.Kalakumari, "VEGAS: Better Performance Than Other TCP Congestion Control Algorithms on MANETs", International Journal Computer Networks (IJCN), Volume (3) : Issue (2) : 2011.

[16]. L. Brakmo and L. Peterson, "TCP Vegas: End to end congestion avoidance on a global internet," *IEEE J. Sel. Areas Commun.*, vol. 13, no. 8, pp. 1465–1480, Oct. 1995.

[17]. R. J. La, J. Walrand, and V. Anantharam, "Issues in TCP Vegas," UCB/ERL Memorandum, No. M99/3, Jan. 1999, Univ. Calif. Berkeley.

[18]. M. Allman, V. Paxson and W. R.Stevens, ''TCP congestion control'', in IETF RFC, 2581, 1999.

[19]. S.McCanne and S. Floyd, ns Network Simulator. http://www.isi.edu/nsnam/ ns/.

[20]. Lawrence S. Brakmo, Sean W. O' Mally, Larry Peterson. TCP Vegas: New Techniques for Congestion Detection and Avoidance[C]. IEEE/ACM Transactions on Networking, 1994: 1024-1035.