DOI 10.4010/2016.1335
ISSN 2321 3361 © 2016 IJESC

Research Article Volume 6 Issue No. 5

An Enhanced Congestion Control with Bit Error Identification for TCP-Vegas

Dimple Bharad[1], Tosal M. Bhalodia[2], Shubham Sharma[3]
Student[1], Assistant professor[2], Assistant professor[3]
AITS Rajkot, India
bharaddimpler@gmail.com[1], tmbhalodia@aits.edu.in[2], sssharma@aits.edu.in[3]

Abstract:
TCP is a connection oriented and reliable protocol of Transport layer. That is used for wireless communication and for that different types of TCP variants are available like TCP-TAHOE, TCP-RENO, and TCP-NEW RENO etc. But only TCP-VEGAS use better bandwidth estimation and so that it has better throughput, congestion control, delayed network and mobility control and stability than other TCP variants in heterogeneous networks. In recent year TCP-ARTA VEGAS and TCP VEGAS THRESOLDBASED CONGESTION CONTROL significantly improved the fairness of bandwidth allocation and bit error rate in heterogeneous network but this does not identified or resolve the problem when multiple packet loss and not identified proper reason for packet loss in the network. TCP-Vegas unnecessarily decrease the congestion window size. In our work we will consider the bit error and multiple packet loss problems of TCP-VEGAS and propose algorithm for improve throughput and bandwidth allocation and provide identification about packet loss reason using RED and ECN notification over heterogeneous network.

Keywords: congestion control, congestion avoidance, bit error identification TCP-VEGAS,TCP-RENO,TCP-NEWRENO

1. INTRODUCTION

Tcp is used everywhere over wired or wireless link. it read as RFC 793.the most popular transport layer protocol is TCP. it is reliable end to end byte stream connection-oriented protocol.TCP provide segmentation of data and control over congestion.TCP generate acknowledgment for unordered or congested packet. Duplicate acknowledgement show packet loss in the network. Retransmits the packet based on time out value by estimating RTT (round trip time) A prime concern for TCP is congestion. Congestion occurs when routers are overloaded with traffic that causes their queues to build up and eventually overflow, leading to high delays and packet losses. Since most Internet traffic is carried by extremely reliable wired links, TCP assumes that all losses indicate congestion. Therefore, when losses are detected, besides retransmitting the lost packet.

First sender buffer all data before the transmission, allocate sequence number for each packet it conform delivery of data packet. Sender block data packet until conformation and un acknowledgement packet is lost. TCP used sliding window protocol. when there are many users and user demands for shared bandwidth and control flow control it called congestion collapse.

Internet performance is depending on transmission control protocol (TCP) congestion control mechanism. When total number of packets sent to the network is greater than the network capacity, congestion occurs in the network so packets are dropped. TCP congestion control aims to moderate the sending rate in order to avoid congestion. Several versions of TCP are improving the performance of the basic TCP algorithm. TCP variants can be classified as packet loss based algorithms are TCP-RENO and TCP-NEWRENO. Second is delay based algorithm like TCP-VEGAS[1].mainly TCP provide connection between two nodes and also take decision about in which way transmission ,formation and routing perform on packet data at receiver side[1].

TCP LAYERS:

1) Link layer :-mostly used in local communication network.
2) Internet Layer:-used in LAN and wide area network.
3) Transport layer:- used for communication among all nodes.
4) Application layer :- for transmits application, messages and more .provide communication between clients and server[2].

Congestion can be defined as a network state in which the total demand for resources, e.g. bandwidth, among the competing users, exceeds the available capacity leading to packet or information loss and results in packet retransmissions (Papadimitriou, 2011). At the time of congestion in a computer network there will be a simultaneous increase in queuing delay, packet loss and number of packet retransmissions. In other words congestion refers to a loss of network performance when a network is heavily loaded.

Congestion control mainly divided in to four class like slow start, congestion avoidance, fast retransmission and fast recovery[3]. Congestion means overflow of network.

2. ACTIVE QUEUE MANAGEMENT

Congestion control is one of the problems at the internet routers and it adversely affects the performance of the TCP. When the queue buffer of an internet router becomes full

the router starts dropping the packets and this practical solution is called drop-tail mechanism. A simple solution to solve this problem is by increasing the buffer size at the congested link.
This solution leads to increased implementation cost, high queuing delay and delay jitter. An Active Queue Management scheme called RED has been proposed to reduce this problem however its solution is not satisfactory for real applications. RED uses an exponentially weighted moving average (EWMA) to calculate the average queue size and compute probabilities in order to mark packets for dropping decision. Another called Adaptive RED and denoted by "ARED" to solve the problem.

3. TCP-VEGAS

The founder of TCP-Vegas was Brakmo, O'Malley & Peterson in 1994.TCP-Vegas uses the estimated difference between the EXPECTED and ACTUAL throughput. It shows the total available bandwidth in the network, regulate the transmission rate based on this value avoid congestion. Basic algorithm says that when the network is free from congestion actual and expected diff is close. If network is congested then actual throughput smaller than actual. TCP-Vegas adjust its cwnd by following equations.

$$\mathbf{cwnd=} \begin{cases} \mathbf{cwnd + 1} & \mathbf{diff<\alpha} \quad \mathbf{1)} \\ \mathbf{cwnd - 1} & \mathbf{diff>\beta} \quad \mathbf{2)} \\ \mathbf{cwnd} & \mathbf{otherwise} \quad \mathbf{3)} \end{cases}$$

$$\mathbf{diff = ExpectedRate - ActualRate} \qquad \mathbf{4)}$$
$$\mathbf{ExpectedRate = cwnd(t) / BaseRTT} \qquad \mathbf{5)}$$
$$\mathbf{ActualRate = cwnd(t) /RTT} \qquad \mathbf{6)}$$

BaseRTT is the minimum encountered RTT of the connection, cwnd (t) is used for current congestion window size and the actual RTT called round trip time. α and β are parameters whose values are typically set to 1 and 3 respectively.
TCP-Vegas is capable to detect congestion in the network early level and to prevent periodic packet losses that usually happen in TCP Reno[1].

- When diff < α it means the actual throughput is less, hence sender consider underutilization the available bandwidth and linearly increase the window size.
- When diff > β in case Vegas will decrease window linearly and when diff is between α and β congestion window remain unchanged.

TCP-VEGAS calculate the time difference for every packet sent and calculate round trip time on each acknowledgment received .if the difference between current and last packet time is large then it retransmits the packet by avoiding 3dupACK[1].

4. EXPLICITE CONGESTION NOTIFICATION (ECN)

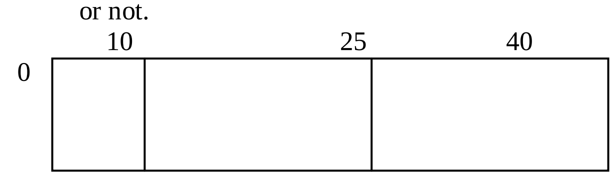
Explicit Congestion Notification (ECN)[25] is an extension proposed to Random Early Detection (RED). RED is an active queue management mechanism in routers, which detects congestion before the queue overflows and provides an indication of this congestion to the end nodes. A RED router signals incipient congestion to TCP by dropping packets probabilistically before the queue runs out of buffer space.

RED router operates by maintaining two levels of thresholds minimum (minth) and maximum (maxth). It drops packets probabilistically if and only if the average queue size lies between the minth and maxth ECN relies on extension to RED, which marks a packet instead of dropping in when the average queue size is between minth and maxth.

5. ENHACED VERSION OF TCP-VEGAS

TCP-ARTA VEGAS improve fairness of tcp-vegas but does not detect any bit error in the network.
Some time connection will be fail or packet loss because of not only congestion but reason for that is bit error in the network packet. Because of that TCP all the time decrease their congestion window size when packet is loss and assume that is congestion. Without considering the quantity of the delta(diff).
So we describe an algorithm for Modified TCP-Vegas with dynamically adjustment of two parameters alpha and beta.

- First packet is sent from the sender to the receiver side and when the packet reach up to R2(router) it will apply the method APDATED RED . .
- In that case the sender identifies that packet loss due to the Bit error and so it is not required to decrease the congestion window size.
- Each time Vegas calculates the total RTT .If the packet has no bit error and no congestion then packet transmit successfully.
- Modified Vegas calculates total number of n_ rtt that satisfies the condition diff<α. Store this value and increment window like that.
- First packet is sent from the sender to the receiver side and when the packet reach up to R2(router) it will check the condition using APDATED RED. calculate average length of that queue find if congestion occur or not.



In MODIFIED RED first decide two threshold value like minimum and maximum as 10% and 70 % of the total queue length.

We assume that if minimum value is 10 and maximum value is 40 the we find

$$\mathbf{avg= \frac{maximum + minimum}{2} = \frac{10+40}{2} = \underline{25}}$$

- **Δ = maximum –minimum (40-10=30)**
- **Δ-avg**
- **Δ+avg**

From this if value is less than the average queue length then not required to give congestion notification.

If value will be more than average queue length then one ECN (explicit congestion notification) sent by receiver to sender.

ECN used for notify end system about congestion. it required some change in both header IP and TCP. In IP header add one field with two bits is used for notification to end system.

From this we can remove multiple packet drops. If network has no congestion but packet loss due to any bit error then receiver send higher bit to sender. So sender does not required to unnecessary decrease CWND SIZE.

For TCP two new flag in the reserved field of the header specify by sender and receiver .receiver notify to sender about bit error (1 and 0).

Using this we get proper identification about reason for packet loss and save multiple packet loss and unnecessary cwnd decrement.

6. PROPOSED ALGORITHM

STEP 1: client initially set cwnd = 10, and at particular time client sends frame-0 and server stores frame-0 inside buffer and send back positive ACK.

STEP 2:client sends next successive frames without waiting of acknowledgment because cwnd set to 10.

STEP 3:if cwnd reach at threshold level then packet start to drop due to the congestion issue due to queue mechanism.

STEP 4:so in our propose work we modified the RED mechanism and use two threshold parameters.

STEP 5:APDATED RED indicate congestion issue to source in our work we send ECN notification to source before dropping packet.

STEP 6: if sender get positive ACK from the receiver then calculate total no. of RTT that satisfied diff<α cwnd increment as it is.

STEP 7: if packet loss due to congestion then MODIFIED RED send ECN notification and based on that cwnd decrease by 2.

STEP 8: if packet loss due to bit error then two flag set as high =1 and low=0 for bit error notification sent to sender .so sender does not required to decrease cwnd and sender increase cwnd as per algorithm. In that case the sender identifies that packet loss due to the Bit error or due to congestion. so it is not required to decrease the congestion window size if packet loss due to Bit error.

Using this algorithm we can improve the throughput, PDR and delay in case of multiple packet loss over wireless link for that we can use NS2 for simulation and try to introduce different amount of error using Uniform Error model .

7. SIMULATION WORK

This simulation perform using NS2 simulator. Using this simulator we tack different graphs of TCP-variants like Tahoe, Reno, Newreno and Vegas.

The intention of this modification to TCP Vegas has been the achievement of better fairness while preserving throughput approximately the percent of increment in fairness has been observed show that the performance improvement.

7.1 Throughput comparison

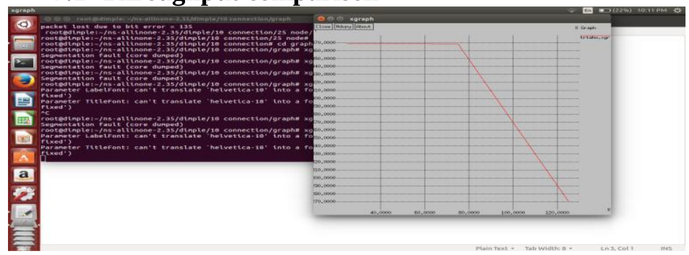


Fig 1.Tahoe Throughput

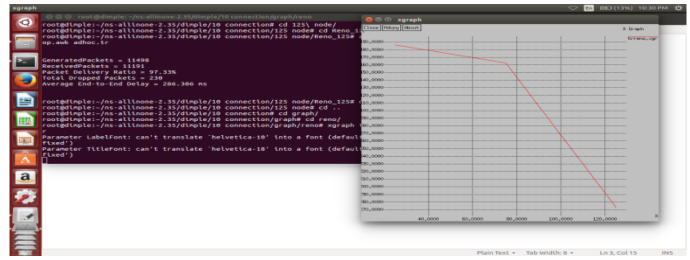


Fig 2 Reno Throughput

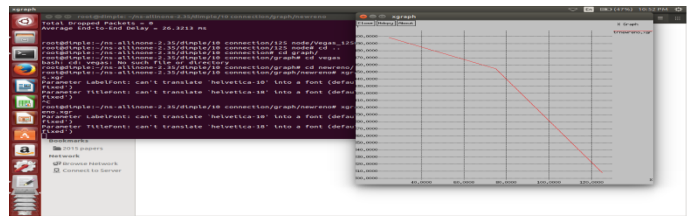


Fig 3.New reno Throughput

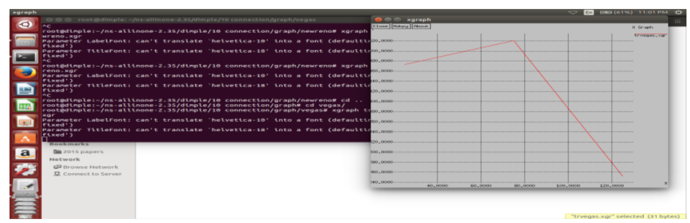


Fig 4.Vegas Throughput

7.2 PDFCOMPARISION

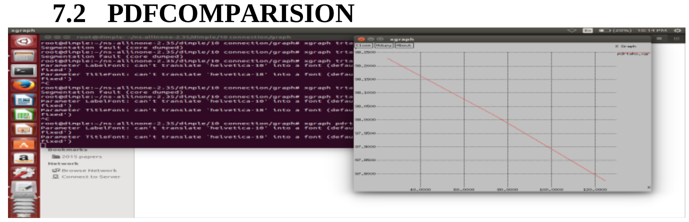


Fig 5. Tahoe PDF

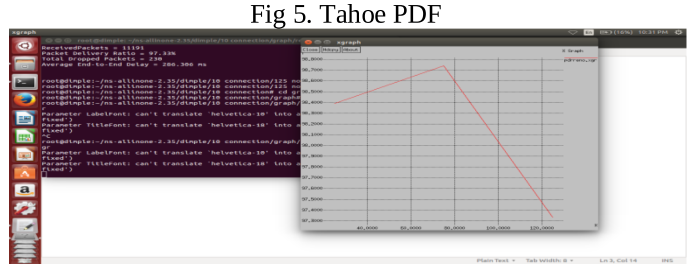


Fig 6 Reno PDF

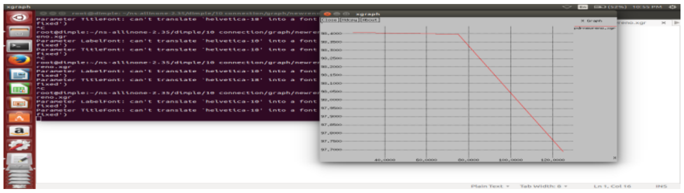


Fig 7. NewReno PDF

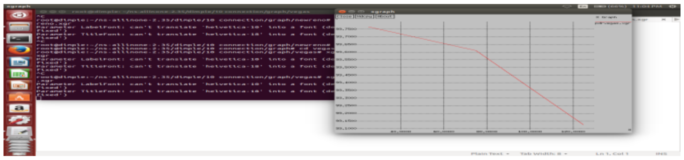


Fig 8.Vegas PDF

Nodes	PDF of Tahoe	PDF of Reno	PDF of Newreno	PDF of Vegas
25	98.2283	98.3912	98.4080	**99.7655**
75	98.0119	98.7415	98.3975	**99.6094**
125	97.7729	97.33	97.6878	**99.1247**

Fig 9. PDF comparison

Nod es	Throughp ut of Tahoe	Throughp ut of Reno	Throughp ut of New reno	Throughp ut of Vegas
25	469.20	486.40	498.69	472.95
75	468.46	462.72	454.74	**520.45**
125	270.08	272.44	307.68	251.99

Fig 10. Throughput comparison

7.3 BIT-ERROR COMPARISION

Error rate	Throughput	PDR in %	Delay	Good put
0.01	293.80	0.9990	236.039	147.64
0.02	277.67	0.9986	209.77	139.45
0.03	261.00	0.9961	228.54	131.20
0.04	250.67	0.9944	236.081	126.02
0.05	223.57	0.9921	289.226	112.42

TABLE .Error comparison

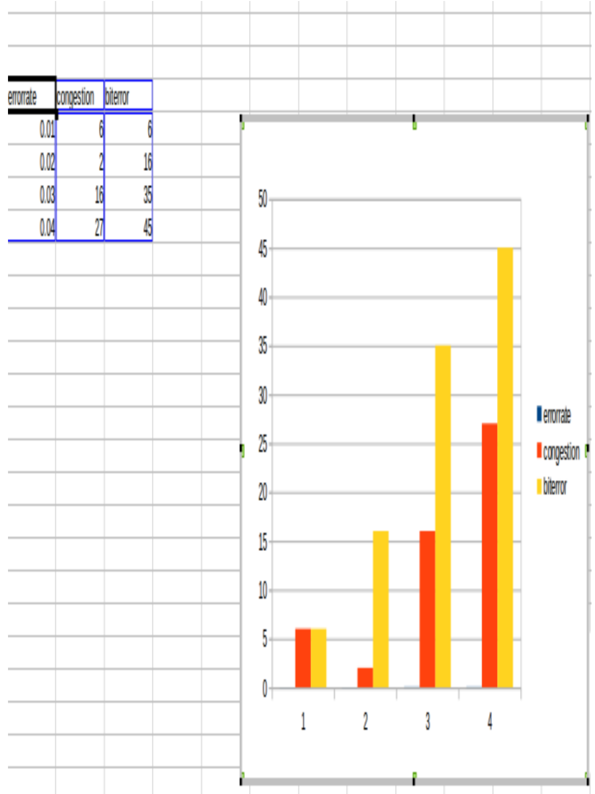


Fig 11. Comparison between congestion loss and error loss

7.3 QUEUE SIMULATION

NODE	Throughput	PDF in %	Good put
7	313.11	97.54	302.47
14	435.68	99.78	420.88
21	452.39	99.49	437.03

TABLE .RED result

NODE	Throughput	PDF in %	Good put
7	307.81	99.67	297.36
14	435.68	99.78	420.88
21	451.03	99.63	435.71

TABLE .Modified RED

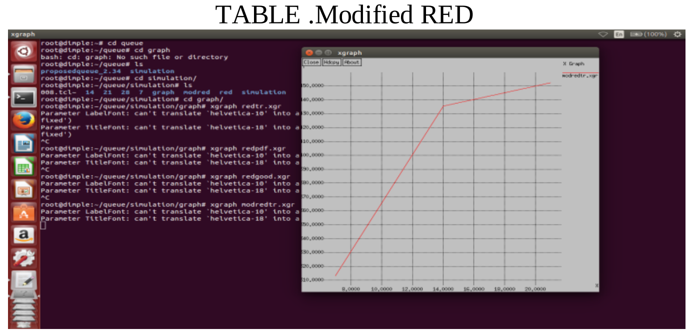


Fig 12.Throughput of modified RED

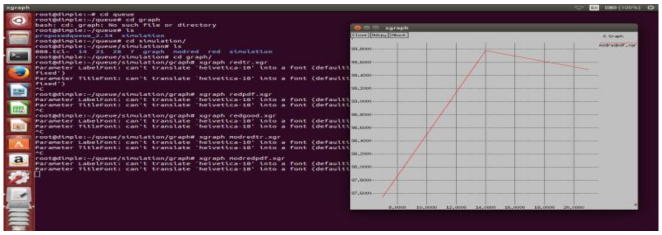


Fig 13 PDF of modified RED

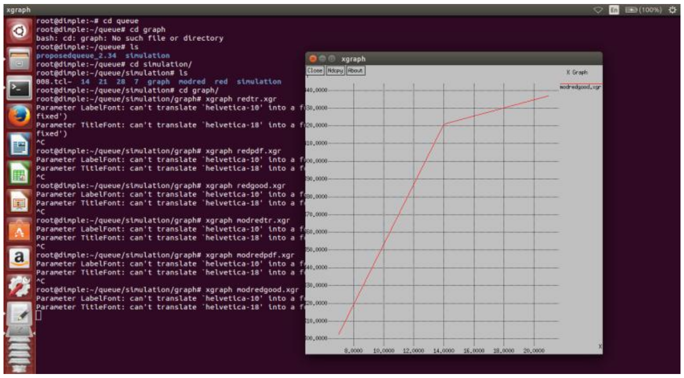


Fig 14.goodput of modified RED

CONCLUSION

In this work we looked at the identification problem associated with TCP-Vegas for that our modified protocol identified the causes of packet loss like bit error or congestion.
And improve throughput, Delay and PDR over heterogeneous network. By this we will get proper identification about packet loss and no reason for packet loss and reduce unnecessary cwnd decrement. We may try to get better result by reducing the congestion in network**.**

REFERENCES

[1] VegasShahram Jamalia, Neda Alipasandib, Bita Alipasandic, "TCP Pegas: A PSO-based improvement over TCP Vegas"2 APRIL 2015.

[2]IETF RFC 2581 M. Allman,Dr. Neeraj Bhargava, Dr. Ritu Bhargava, Manish Mathuria, Shilpi Gupta, Kamal Kumar Jyotiyana, "Analysis of Different Congestion Avoidance Algorithms", IRACST,ISSN: 2250-3501,Vol.3, No1, February 2013.

[3]Nima Afraz,Morteza Analoui,"TCP-Arta Vegas: Improving the fairness of TCP-Vegas", ICEE, 2015.

[4]Jin-Ru Chen,Yaw-Chung Chen, "Vegas Plus: Improving the Service Fairness" VOL. 4, NO.5, MAY 2000.

[5]Joel Sing,Ben Soh," TCP New Vegas: Improving the Performance of TCP Vegas Over High Latency Links" 2005.

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

[7]Lianghui Ding, Xinbing Wang, Youyun Xu, Wenjun Zhang, Wen Chen,"Vegas-W: An Enhanced TCP-Vegas for Wireless Ad Hoc Networks" No. 60702046), ICC 2008.

[8]Yi-Cheng Chan, Chia-Tai Chan, and Yaw-Chung Chen," An Enhanced Congestion Avoidance Mechanism for TCP Vegas",VOL. 7, NO. 7, JULY 2003.

[9]Sukant Kishoro Bisoy, Prasant Kumar Pattnaik, Amardeep Das, Mohit Ranjan Panda," The Impact of Delayed ACK on TCP Variants Protocols in Wireless Network",2014.

[10]Yi-Cheng Chan, Chia-Liang Lin,Chia-Tai Chan, Cheng-Yuan Ho," Improving Performance of TCP Vegas for High Bandwidth-Delay Product Networks" ISBN 89-5519-129-4 FEB 2006.

[11]Kolawole I. Oyeyinka1, Ayodeji O. Oluwatope2, Adio. T. Akinwale3, Olusegun Folorunso3, Ganiyu A. Aderounmu2, Olatunde O. Abiona4," TCP Window Based Congestion Control Slow-Start Approach" 2011 SciRes.

[12] Ghassan A. Abed *, Mahamod Ismail, Kasmiran Jumari," Exploration and evaluation of traditional TCPcongestion control techniques"2012 Sciencedirect.

[13]Hamzeh Qabaja, Marwan Bikdash," Performance Comparison of AutoRED under different TCP Variations"2015 April IEEE.