Volume: 5 Issue: XI Month of publication: November 2017
DOI:

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor:6.887
Volume 5 Issue XI November 2017- Available at www.ijraset.com

Congestion Control Techniques In Transport Layer For Wired Connections

Sweeti Sah[1], Aman Verma[2], Jitendra Kurmi[3]
[1]*M.Tech Student, BBAU, Vidya Vihar Raibareily Road Lucknow*
[2]*M.Tech Student, BBAU, Vidya Vihar Raibareily Road Lucknow*
[3]*Asst. Professor, BBAU, Vidya Vihar Raibareily Road Lucknow*

Abstract:Today we are able to use the Internet successfully all over the world because of Congestion Control Techniques working properly. There are various types ofTCP Congestion Control Techniques including Tahoe, Reno, New Reno, SACK, Vegas and these techniques are different from each other[1]. When any packet is being lost or the timeout occurs, these techniques come into role and what is the effect on throughput, efficiency, performance when compared with TCP Vegas. When congestion occurs at any router their role is to follow certain protocol, follow some algorithm for following another path and information about that packet is stored in the storage media and the simulator which is used for implementing this is ns2 installed in Ubuntu 16.04 version.

Index terms: Efficiency, Throughput, TCP Tahoe, TCP RENO, TCP NEW RENO, TCP SACK, TCP Vegas[2].

I. INTRODUCTION

TCP is connection oriented end to end transmission protocol[3].Reliability of packet is ensured of receiving the acknowledgment segment within the timeout interval by the receiver node. Packet loss can be because of the delay, timeout, buffer overflow and etc. We assume the loss due to network is minimal, but due to buffer overflow is more at router[4]. So these techniques are introduced to deal with congestion and how they react and take appropriate action and improve throughput, efficiency.

A. *There are few components*

1) *Slow Start:* The congestion window start with size=1 and grows on exponentially until it reaches its threshold value.
2) *Additive Increase Multiplicative Decrease (AIMD):* When congestion window size reachesa threshold value, then it decreases the congestion window multiplicatively further do linear increment.
3) *Fast Retransmit:*After receivingthree duplicate acknowledgement, the lost data packet is resent to receiver.
4) *Fast Recovery:*This is the component which tells recovery has taken place.

TCP Tahoe = Slow Start + AIMD +Fast Retransmit
TCP Reno = Slow Start + AIMD +Fast Retransmit+ Fast Recovery
TCP New Reno = TCP Reno + Partial ACK
TCP SACK = TCP RENO + Selective ACK
TCP Vegas = Focused on RTT

II. TCP TAHOE

A. *Suggested by Van Jacobson in 1988[4].*

Start with slow start mechanism, congestion window size=1. Network capacity can be determined by a congestion window[4].As we send data packet we get an acknowledgement, then increment the congestion windowsize and keep on sending the data until reaches threshold value and move into congestion avoidance phase, there it keep on sending the data and after getting an acknowledgment it just increment congestion window=congestion window +1/ congestion window and we keep on sending unless loss or time out occurs.After getting three duplicate acknowledgement it moves into Fast Retransmit state and send the missing packet. Set the threshold value as congestion window/2 and congestion window=1, move to slow start phase[5]. In case of timeout in slow start and congestion avoidance phase, it moves into Retransmission timeout phase, when all acknowledgementsare received from retransmission timeout phase to slow start phase for whatever packet is being sent. The process repeat and so on.

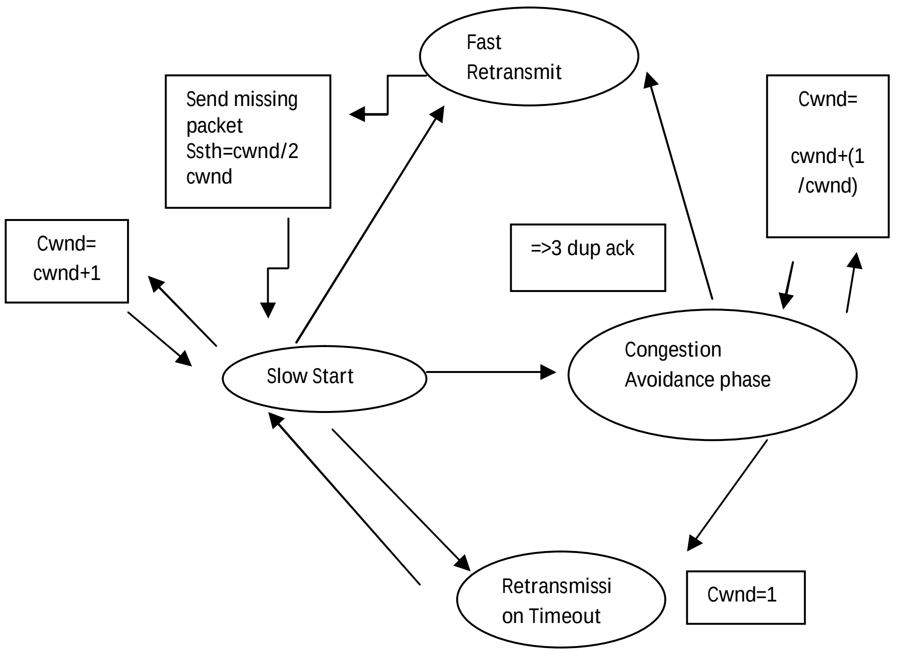

Figure 1: TCP Tahoe

III. TCP RENO

First few steps of TCP Reno are same as TCP Tahoe. When it is in Fast Retransmit state[8], it moves immediately to Fast Recovery state and set threshold=congestion window/2 and congestion window =threshold, after that sends missing packet.

From Fast Recovery state after receiving duplicate acknowledgment, it increases congestion window=congestion window+1 and keep on sending the data and move to congestion avoidance phase when there are no duplicate acknowledgements left by setting congestion window=threshold.

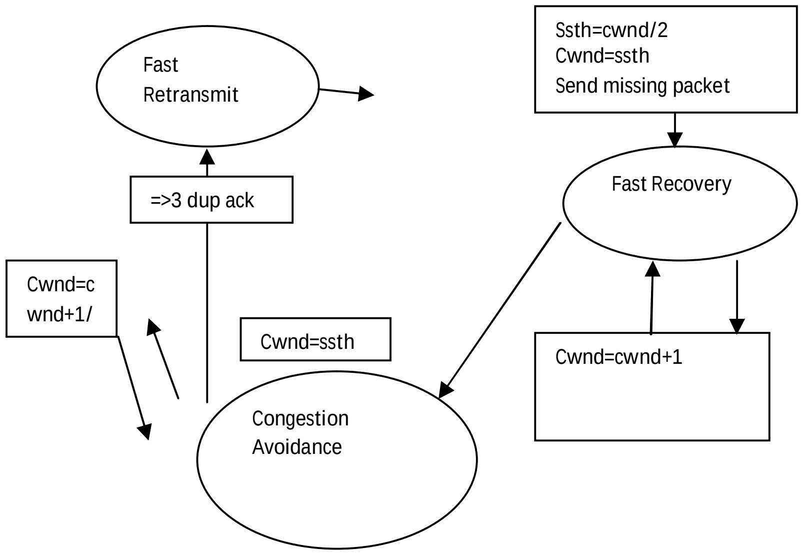

Figure 2: Block Diagram of modification done in TCP Tahoe

IV. TCP NEW RENO

It extends fast Recovery state phase and remain in a Fast Recovery state until all data in the pipe before detecting three duplicate acknowledgement are acknowledged.Able to avoid the problem of multiple packet loss problem.

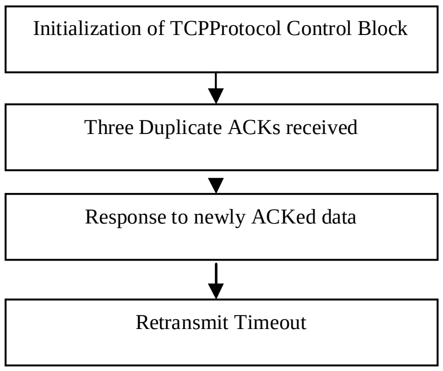
Initialization of TCPProtocol Control Block
Three Duplicate ACKs received
Response to newly ACKed data
Retransmit Timeout

Figure 3: Block Diagram of Reno

V. TCP SACK (SELECTIVE ACKNOWLEDGEMENT)

It reports non continuous block of data. After the detection of packet loss, more than one lost packet can send in one Round Trip Time. Acknowledgement of packet is done selectively for maximum utilization.After entering into Fast recovery state, the data which are outside the network will be initialized by some variable.It will set congestion window as half the current size[6]. For every acknowledgement that is being received by receiver there is a decrement of pipe by one. When congestion window goes larger the pipe than it will check for the last segment, so that it can be resent back. If there are no segments outside the network, then new segment will be send[7]. Thus, more than one segmentcan send in one round trip time. When both the end nodes support the SACK option, then only TCP Connections established.

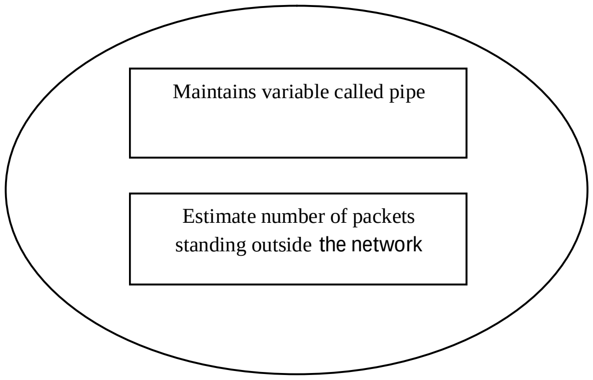
Maintains variable called pipe
Estimate number of packets standing outside the network

Figure 4: Fast Recovery State of TCP SACK

VI. TCP VEGAS

A. Introduced by Brakmo and Peterson in 1994[12].

It is proactive in nature. It detects early packet loss. It is more efficient than all the above mentioned and here it does not require three duplicate acknowledgement for detection of packet loss. It does not wait for three duplicate acknowledgement[6]to send the lost packet.It keeps the track of all the segment that is being sent and also calculate the estimation of Round Trip Time by keeping the track that how much time it is going to take to receive an acknowledgement back.

TCP Vegas is different compared to other implementation during Congestion Avoidance phase. Instead of detecting the congestion by the loss of a segment, it detects congestion by decreasing the actual sending rate compared to theexpected sending rate.

B. Difference= (Expected –Actual) Base RTT[11]

Where,

Expected = (CWND/Base RTT) [11]

Actual = (CWND/RTT) [11]

Now based on the difference the sender is going to update the window sizeaccordingly.

VII. COMPARISION

	SOLUTION	PROBLEM
TCP TAHOE	• First, the congestion control technique to detect packet loss • This has created the base of all congestion control techniques • Grow exponentially then after a timeout or packet loss grows linearly.	• Complete Timeout Interval to detect Packet Loss • Cumulative ACK • Cwnd=1 when packet loss • Inefficient • Pipeline emptied
TCP RENO	• Cwnd=Cwnd/2 • Immediate ACK • Packet loss is detected earlier, whenever there is three duplicate ack, i.e. the sign of one packet loss. • After Fast Retransmit state, it enters into Fast Recovery state. • The pipeline is full • Efficient than Tahoe	When multiple packets are getting loss then it is not efficient.
TCP NEW RENO	• Detect Multiple Packet Loss • Extends Fast Recovery Phase until all data in thepipe before detecting three duplicate ACK are acked. • Partial ACK	Just one packet loss can be detected in one round trip time
TCP SACK	• Within one round trip time more than one packet can be send that is being lost. • Not acknowledged cumulatively but selectively. • The sendersends only those segments selectively that is actually being lost.	Not Easy
TCP VEGAS	• Overcomes the problem of getting three duplicate ACK for One packet loss. • Proactive • Efficient • Detects Congestion before Packet Loss • Detects Multiple Packet Loss faster	• Cannot Compete with more aggressive TCP Reno connection • Rerouting path may change propagation delay • Adjust sending rate. • Performance may degrade in asymmetric networks • Treats all packet loss as Random loss.

VIII. OPERATIONS DONE BY ROUTER

The router has an incoming buffer for receiving the data packet, andthe outgoing buffer for sending the data packet to another router. When any type of congestion occurs on it. Router follow the Border Gateway Protocol in which it maintains a routing table and store details about the packet like its source address, destination address, MTU and etc. It is divided into two parts-

A. Interior Bgp

All the routes are within a single autonomous system.

B. Exterior Bgp

When router from one autonomous system communicate with router of another autonomous system.

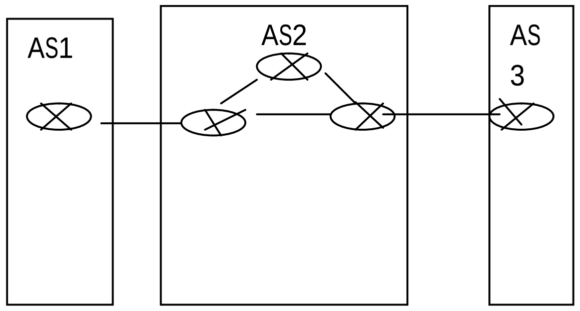

Figure 5: Block Diagram of Border Gateway Protocol

Routers follow the shortest path algorithm and the information are stored in RIP (Routing Information Base) [9]

IX. SIMULATION

A. Operating System: Ubuntu 16.04
*B. Software:*NS-2 (Network Simulator), NAM (Network Animator), Xgraph
First, we have created NS simulator
After creating the simulator, we will create output files, and create nodes. Here we have considered 10 nodes

LINK	FROM NODE	TO NODE	RATE	TIME
Duplex	N0	N1	150mb	5ms
Duplex	N1	N2	150mb	5ms
Duplex	N2	N6	150mb	5ms
Duplex	N2	N3	10mb	2ms
Duplex	N2	N4	20mb	4ms
Duplex	N2	N5	100mb	10ms
Duplex	N3	N7	8mb	5ms
Duplex	N4	N7	2mb	1ms
Duplex	N2	N7	10mb	1ms

Duplex	N7	N8	10mb	10ms
Duplex	N8	N9	100mb	10ms
Duplex	N8	N10	8mb	1ms

Figure 6: Structural Detail of Simulation

Above table describes the structural representation of nodes with their communication link, bandwidth rate and time at which it is going.

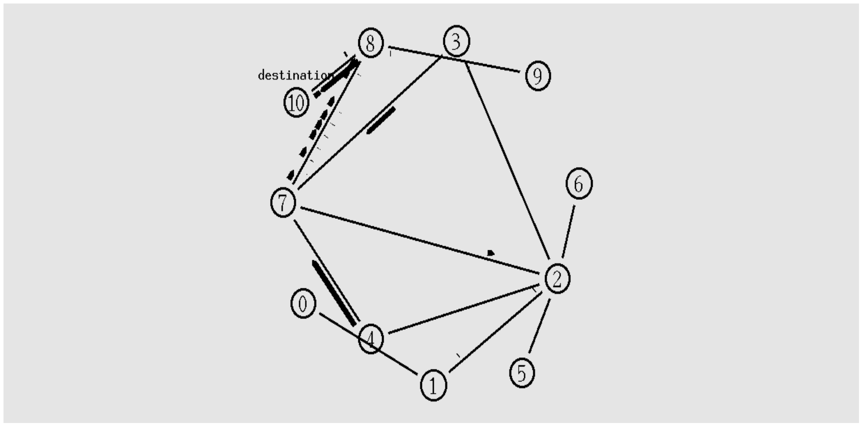

Figure 7: NS2 Simulation in NAM

Above is the table showing nodes are connected toeach other and transferring data packets to the destination node having a certain bandwidth rate and time.

X. EXPERIMENTAL RESULT

Below is the xgraph generated, Calculating the Bandwidth and writing it to files fo, f1, f2, f3, f4, f5, f6, f7, f8, f9.
Out1.tr, Out2.tr, Out3.tr, Out4.tr, Out5.tr, Out6.tr, Out7.tr, Out8.tr, Out9.tr are the Output files.
Where, .tr extension is the trace file.

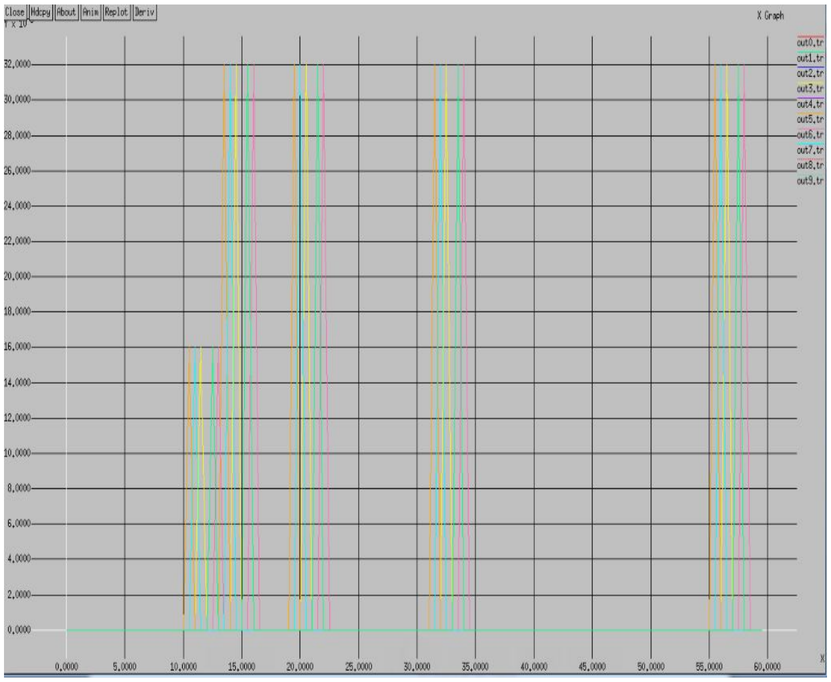

Figure 8: XGraph Result

TRAFFIC SOURCE	STARTING TIME	ENDING TIME
Source0	10.0	50.0
Source1	10.0	50.0
Source2	10.0	50.0
Source3	10.0	50.0
Source4	10.0	50.0
Source5	10.0	50.0
Source6	10.0	50.0
Source7	10.0	50.0
Source8	10.0	50.0
Source9	10.0	50.0

Figure 9: Traffic Source Time

At 60.0 the simulation ends.
Hence, in the above simulation, we have created nodes, set the TCP Vegas Agent then passed the exponential traffic over the network.
After passing the traffic we have calculated the bandwidth in a Mbit/s and write it to file. We created the traffic Sink and attach to destination node.

XI. PROPOSED SOLUTION

The proposed solutionscan be implemented using ns2.The Performanceof TCP Vegas can be improved when symmetrical network is used and by creating an algorithm that could distinguish between packet loss and random loss.
In the algorithm of TCP Vegas there is a calculation of Expected Sending Rate and Actual Sending rate. Additionally, we can calculate the time of Actual Sending the data packet, and estimating the Expected Time of sending and receiving the data packet when it'sstored in the buffer.

XII. CONCLUSION

Hence TCP Vegas is more efficient than TCP Tahoe, Reno, New Reno, Sack by improving the throughput as it detects the packet loss before it occur. The throughput achieved by TCP VEGAS is between 31 and 71 %[10]and by extending the re-transmission mechanism of RENO. TCP Vegas conserve bandwidth by transmitting too high at data rate[7]. And also when the connection starts TCP Vegas has no idea of available bandwidth.
The Simulation Result displays the data at which rate they are going.

REFERENCES

[1] http://materias.fi.uba.ar/7543/dl/Paper_Congestion_Algorithms.pdf
[2] Mohammad A. Mikki. "Optimal TCP and QueuingDiscipline for Web Proxy Servers", 2009 2nd International Conference on Computer Science and itApplications, 12/2009
[3] https://www.ietf.org/rfc/rfc793.txtRFC: 793TRANSMISSION CONTROL PROTOCOL DARPA INTERNET PROGRAM PROTOCOL SPECIFICATION September 1981

[4] V. Jacobson. "Congestion Avoidance and Control". SIGCOMM Symposium no Communication Architecture and protocols.
[5] https://www.researchgate.net/profile/Jon_Mark/publication/3156633_Robust_CrossLayer_Design_of_WirelessProfiled_TCP_Mobile_Receiver_for_Vertical_Handover/links/553d78f90cf29b5ee4bcc4a3/Robust-Cross-Layer-Design-of-Wireless-Profiled-TCP-Mobile-Receiver-for-Vertical-Handover.pdf
[6] A Comparative Analysis of TCP Tahoe, Reno,New-Reno, SACK and Vegas.
[7] K. Fall, S. Floyd "Simulation Based Comparison of Tahoe, Reno and SACK TCP".
[8] V. Jacobson "Modified TCP Congestion Control and Avoidance Algorithm". Technical Report 30, Apr 1990.
[9] http://www.erg.abdn.ac.uk/users/gorry/course/inet-pages/router-opn.html
[10] http://cseweb.ucsd.edu/~rbraud/jsac.pdf "TCP Vegas: End to End Congestion Avoidance on a Global Internet"
[11] http://www2.ensc.sfu.ca/~ljilja/ENSC835/Spring02/Projects/bian_zhang.hilary/Hilary_and_Bian_Report.pdf ENSC 835-3: NETWORK PROTOCOLS AND PERFORMANCE CMPT 885-3: SPECIAL TOPICS: HIGH-PERFORMANCE NETWORKS
[12] TCP Variations_ Tahoe, Reno, New Reno, Vegas, Sackpublished by Lydia Shipler