A Review of Comparative Analysis of TCP Variants for Congestion Control in Network

Pooja Chaudhary
Department of Computer
Science & Engineering
Ajay Kumar Garg Engineering College,
Ghaziabad, Uttar Pradesh, India

Sachin Kumar, PhD
Department of Computer
Science &Engineering
Ajay Kumar Garg Engineering College,
Ghaziabad, Uttar Pradesh, India

ABSTRACT
TCP is a consistent, association- oriented and extensively used end-to-end transport protocol in the computer network. It gives data in the structure of byte streams, start the connection and it is used in many applications that depend upon secured delivery of information. TCP assigns array to establish their integrity and deliver operation measures from timeouts and retransmissions to provide accuracy. Many analysis affiliate with the computer network processes showed that the accurate characteristics of traffic possess the capacity of time-scale invariant. Such an impact is produced by the specific character of file allocation on servers, their dimensions ahead with an ordinary behavior of users. It was introduce that the data streams, which originally do not show autonomous-analogy properties after being processed at the host server and an effective grid elements, start exhibit the distinct signs of autonomous-analogy. It produces quick buffer overwhelm even with using low factor. If no action is taken to eliminate the arriving traffic then the queues on the maximum weighted boundary will grow repeatedly and finally increase the size of the buffers at the identical nodes This paper presents a comparison of TCP variants for Congestion Control in network concerning the basis of various performance metrics such as end-to-end wait, throughput, queue dimension and packet delay rate using Network Simulator-2 (NS-2). The conclusion show that in high congested network, Vegas does best while in low cohesive network Reno gives best result.

Keywords
Congestion Control, TCP Tahoe, TCP Reno, TCP New Reno, TCP Vegas.

1. INTRODUCTION
TCP is a consistent, association-oriented transport protocol in the computer structure. It provide data in the structure of byte streams, start the connection and it is used in many applications that depends upon secured delivery of information. TCP assigns array to establish their integrity and deliver operation measures from timeouts and retransmissions to provide accuracy.TCP is a complicated transport layer protocol containing four network algorithms: Slow start, Congestion avoidance, Fast retransmit and Data recovery algorithms. Slow Start algorithm is a process to control the transmission rate. This is adept through the return rate of acknowledgements against the recipient. During a new setup is initiated, the Congestion Window (CWND) is fixed to one segment. Every period an acknowledgement (ACK) is taken, the congestion window is maximized by one segment for each Round Trip Time (RTT). The sender can send up to the least of the congestion window and the advertised window. Although the congestion window greater a threshold named Slow Start Threshold (SSTHRESH), it gets in congestion avoidance phase. Congestion avoidance is the method which deals with lost packets. In Congestion avoidance phase a retransmission clock quit or the reception of identical acknowledgement can naturally notable the sender that a network congestion situation is arriving. The sender instantly fixed its transmission window to one halve of the current window dimension. If congestion was designated by a timeout, the congestion window is restart to one segment that necessarily put the sender into slow start mode. If congestion was designated by identical acknowledgement, the Fast recovery algorithms and Fast retransmit algorithm are used.

Fast retransmit algorithm makes usage of identical acknowledgement to discover packet loss. In Fast retransmit, during an acknowledgement packet is received a congestion window is fixed to three, TCP sender is adequately assured that the TCP packet is lost and will retransmit the packet beyond waiting for retransmission clock. Fast recovery is approximately connected to retransmit the packet .In Fast recovery algorithm, TCP sender will not arrive in the slow start phase rather it will exactly decrease the congestion window by halve and boost the congestion window by estimating the convenient congestion window. When an acknowledgement of current data is received, it restore to congestion avoidance phase. This appropriate case may cause fast buffer uniform with low use determinant. Suddenly the queues on the maximum loaded lines will build endlessly and in the end exceed the width of the buffers at the equivalent nodes. This leads to the known fact that the packets retransmit to the nodes with complete buffers will be reboot and therefore are to be reentering and that in change effect in wasting of network resources.

Traffic management in TCP examines the reality of two autonomous methods: Delivery control regulated by the recipient using the window specification and Congestion control regulated by the sender for employed the congestion window and slow begin method. The first method oversees the recipient input buffer and the second method registers the channel congestion, hence it helps to decrease the level of traffic. The Congestion Window(CWND) and slow start method gives resolve the full loading of the virtual connection and decreasing the packet loss in case of overloading in the network. Few modification and extensions are registered uniformly in TCP in an exertion to determine the problems arriving in the development of the protocol or to improve its functions for systems with focused area. The aim of this paper is to handling an approximate analysis of TCP Variants on the basis of various performance metrics and to identify which Variants perform best in different types of network

2. CONGESTION

Congestion is a situation in which node or link carries so much data that deteriorate the performance of network structure such as packet deficit, queuing interruption and block new connections. In TCP, congestion appears when sender receives three duplicate acknowledgments or when a packet loss arrives, resulting in dissipation of resources.

Congestion can also be originated during the following situations.

- When routers are too slow to execute a tasks
- Packet arrival rate exceeds the outgoing link capacity.
- Overloading on the network
- When insufficient memory to store arriving packets
- The output capacity of router is less than the sum of input
- When speed of processor is slow

2.1 Congestion Control

Congestion control and congestion prevention is a process to compromise with lost packets. In this process, there are two notion of packet loss: the receipt of identical acknowledgement and a packet loss arriving. Hence loss of packet indicates the congestion elsewhere in network between source and destination. In case of congestion avoidance mechanism, it predicts when the congestion will occur and accordingly decrease the rate through which hosts sends the data just before the packets start being discarded.TCP congestion control mechanism response to packet deficit by descend the number of unsigned data portion concede in the network. It diffuse congestion in bottleneck link by reducing congestion window sizes .Two approaches are used to control congestion in network: Open loop and Close loop approach. Open loop prevents congestion by guarantee that the system never arrives in a congested state. It use retransmission policy, window policy and acknowledgement policy to prevent congestion before it happens. Closed loop approach guides the connection and to identify either the connection is congested or not and where the definite area and equipment are included. It tries to remove the congestion after it happens.

Wang, Zhiming et al. (2016) describe an available Round Trip Time (RTT) and packet loss rate may differ over many orders of magnitude, which characterizes the discrepancy of the computer network. To manage the heterogeneity in congestion control, they propose a modified TCP scheme. They introduces an adaptive increase factor to the growth function to make sure that the screen development rate matches the path condition and this enhance factor measures the trail condition utilizing a custom function of this available bandwidth and minimum RTT. It adopts an adaptive queue threshold in the loss category scheme to increase the accomplishment of Transmission Control Protocol (TCP) over lossy links.

Ou, Shih-Hao et al. (2016) describe the growth of cellular information technologies, mobile devices are usually assembled with several network interfaces and equipped to accept transport that is arriving protocols such as Multiple path Transmission Control Protocol (MTCP). This protocol is exclusively beneficial for computer network applications that are flow from analytical delay and bandwidth requirement. In this paper, they suggest a combined congestion control and scheduling algorithm that acknowledge disorderly packet transmission. It is attain by flexible window pairing, congestion unfairness, and delay awareness packet ordering. The algorithm is executed in kernel Linux for actual world experiments.

Domanski, Adam et al. (2016) design a TCP congestion control algorithms in order to improve an online communication security. The performance of standard New Reno algorithm is set along with the function of TCP Vegas that tries to avoid congestion by decreasing the congestion window (CWND) dimension for previous packets loss. Acquired outcome represent that TCP Vegas is a impartial algorithm but it has issues with the assignment of accessible bandwidth.

Abolfazli Elham et al. (2016) proposed various types of Transmission Control Protocol containing Reno, Tahoe and TCP Vegas. Although TCP Reno is trusted for guided media, TCP Vegas is proved to have better performance in wireless surroundings. In TCP Reno, to attain a queue that is specific at the intermediate routers, Active Queue Management (AQM) schemes are utilized. An AQM scheme marks the packets during the routers which can be an intermediate state, TCP Reno source uses that mark to regulate its speed. However, in TCP Vegas such scheme is not useable. A design which is utilized at the origin and adapts the TCP Vegas guidelines dynamically is proposed. This scheme plays the role of AQM in TCP RENO and TCP Vegas. This scheme will be applied to the model that is linearised of Vegas.

Radhika Mittal et al. (2015) describe data center transports try to deliver low latency messaging together with high throughput. They show that the packet is single is measured as Round Trip Time (RTT) at hosts, is an effective congestion signal with no need for switch feedback. First, they show that advances in NICs hardware have made RTT measurement possible with microsecond accuracy and these RTT are sufficient to estimate switch queuing. They describe how timely can adjust transmission rates using RTT gradients to keep packet latency low. They implement the layout in host software running over NICs along OS-circumvent facility.

Yang, Peng et al.(2014) emphasize on Internet traffic had been primarily controlled by the Reno that is traditional whereas it is now controlled by numerous different TCP algorithms such as Reno, Cubic, and Compound TCP (CTCP).In this paper, they first propose a mechanism called TCP Congestion Identification and Avoidance Algorithm (CIAA) for earnestly pinpointing the TCP algorithm . CIAA can identify all default TCP algorithms (Reno, Cubic, and CTCP) and most non failure TCP discovery of large operating system families.

Zhou, Keren et al. (2014) proposed TCP Vegas congestion control approach, various drawbacks arise such as utilization of low bandwidth and not achieve fairness whenever channel is allocation with Reno TCP. They reconsider these issues and introduce a modification approach called DYNAMIC Vegas.

It dynamically adopt slow start algorithm and modifies the increase or decrease rate in congestion avoidance phase according to distinct network environment. Many investigations shows that in a single connection environment, it action is same as Vegas. In multiple connection environments, it attains big championship beyond TCP Reno and achieves impartial throughput of entire senders in the end of communication.

Wu Qing-Rui et al. (2014) proposed an enhanced Congestion Control method called TCP NewBR, which calculate bandwidth by utilizing the bottleneck connection operation and preferred exact time interval of acknowledgement to enhance the certainty of usable bandwidth estimation. In addition, they also modify the fast recovery and fast retransmission algorithms based on the queue length of bottleneck link. Simulation results in show that TCP New BR can get more significant throughput and accurate bandwidth estimation than traditional TCP Westwood and more fairer than Reno TCP.

Winstein, Keith et al. (2013) emphasize on new strategy to end-to-end congestion control on a several user network environment. Instead of manually develop every endpoint feedback to congestion indicator, like in conventional protocols; they construct a program called Remy which provides congestion control algorithm that execute at the endpoints. In this strategy the protocol constructor determine their earlier knowledge or previous expectation about the network and aim that the algorithm will tries to provide maximum throughput and small queuing delay.

Alfredsson, Stefan et al. (2013) examine the communication between congestion control in TCP and buffering in wireless networks. Comprehensive calculations have been accomplished in commercial 3 Generation, 3.5 Generation and 4 Generation wireless networks, with a mixture of large and small TCP flows using the Cubic TCP, New Reno congestion control scheme. The conclusion acknowledges that the completion times during the short flows maximize intensely while simultaneous large flow of traffic is recommended. This is caused by high buffer holding from the large flows. In addition, for 3.5 Generation and 3 Generation the finishing times are presented to be rely upon the congestion control algorithms occupied for the circumstances flows, with Cubic leading to extremely larger finishing times.

3. TCP VARIANTS

3.1 TCP Tahoe

TCP Tahoe operation used a number of current algorithms and processing to previous application. TCP Tahoe current algorithm consist of Slow Start phase, Congestion Avoidance phase and Fast Retransmit phase. Nonetheless, it differs from different TCP scheme via utilizing a modified circular shuttle time estimator. An adjustment is included by the refinements to the Round Trip Time Estimator (RTTE) used to fixed retransmission timeout code. Tahoe fixed slow start threshold (SSTHRESH) to partial the congestion window (CWND) and fixed CWND to 1. It arrives in the slow start phase later an acknowledgement of the retransmitted packet has been achieved. If the dimension of the CWND is not boost to the current SSTHRESH, congestion window dimension increases exponentially. Later slow start sender arrive in Congestion avoidance phase and onwards the dimensions of CWND rises by using 1/CWND for each and every get acknowledgement.

3.2 TCP Reno

The TCP Reno utilization maintained the improvement included into Tahoe, but modified the Fast Retransmit procedure to include Fast Recovery , the latest algorithm prevents the communication route from going vacant later fast retransmit, through avoiding the used to slow start phase to refill it later a specific packet loss. Fast recovery phase accomplish by presumption each identical acknowledgement received produce a distinct portion carrying left the pipe. Thus, in Fast Recovery the TCP sender is ready to estimates the number of dominant data.

Fast Recovery is listed by a TCP sender later getting an initiative threshold of duplicate acknowledgement. This threshold code is usually fixed to three. When the threshold of duplicate acknowledgement is taken, the sender retransmits one packet and minimize its congestion window by one halve. Rather of slow start phase, as is accomplish by a TCP Tahoe sender, the Reno sender need of additional duplicate acknowledgement to clock subsequent outgoing packets. Reno Fast Recovery algorithm is optimized as the situation when a specific packet is discarded from a window of data. The Reno sender retransmits only one discarded packet per round trip time. TCP Reno extremely enhance upon the action of TCP Tahoe whenever a specific packet is discarded from a window of data packet but can undergo with performance issue when several packets are dropped from a window of data packets. This issue is easily designed in our simulator during a TCP Reno connection with a big congestion window deteriorate a burst of packet deficit after slow-beginning in a connection with drop-tail gateways or additional gateways that decline to guide normal the queue capacity..

3.3 TCP New Reno

TCP New Reno is a conversion of TCP Reno. It is capable to find when several packet losses are arising in the network. It is more effective than Reno in the case of several packet losses is arises in the computer network. New Reno enhances retransmission as the rapid reformation phase of TCP Reno. It used an extended Fast Recuperation (FR) algorithm in order to solve the timeout problem where a couple of packets are misplaced from the same window. Congestion control elements of New Reno and Reno TCP are same. TCP New Reno characterize a Full Acknowledgement (FA) from a Partial Acknowledgement (PA) by using TCP-Reno fast restoration conduct after it receives a non-replica ACK. FA acknowledges all of the tremendous segments on the beginning of Fast Recuperation. Nonetheless PA acknowledges only one of the tremendous knowledge. New Reno incorporates a little variation in the Reno TCP algorithm at the sender side that discard Reno wait for a retransmit timer when several packets are lost from the similar window .This change corporate the sender nature over Fast Recovery process.TCP New Reno can transmit new packets at the extent of the congestion window over fast recovery process. While TCP enters fast recovery it maintains the high outstanding unrecovered packet classification order. When this classification number is return,TCP restoration to the congestion avoidance case. New Reno TCP is an improved version of Reno that avoids multiple degradation of the congestion window while multiple segments from the same window of data get lost.

3.4 TCP Vegas

TCP Vegas implementation is a modification of TCP Reno, it is important to conclude the available bandwidth and progressively adapt the best guidelines. Vegas algorithm estimates the buffering that does arise in reach the system and controls the rate affiliate with appropriate flow. This algorithm is absolutely capable to regulate and decrease the flow rate since the packet loss arise. The dimension is maintained whereas it determines an estimate of round trip time, the transmission period of the packets throughout the information medium to the destination node and back for packets that have been sent previously. This mechanism identifies that the network is near to overload and decrease the diameter of the window. If RTT is decreased, the sender can examine that the network has conquered the congestion. TCP Vegas is more accurate than TCP Reno; it does not wait for duplicate acknowledgement.

It define two thresholds value a and b.

If Diff < a, TCP Vegas increases Congestion Window (CWND) linearly during next RTT.

If Diff > b, TCP Vegas decreases the CWND linearly.

If a < Diff < b, TCP Vegas leaves the CWND.

4. SLOW START

Slow start algorithm is introduced for the implementation of TCP Variants. It is process used by the sender to control the transmission ratio. This is accomplished through the arrival rate of acknowledgements against the receiver node. When a TCP connection first begins, the slow start algorithm begins a congestion window to one segment that is the maximum segment dimension started by the receiver node while the establishment of connection. While an acknowledgements are restored by the receiver node, the congestion window maximize by one segment for every restored acknowledgement. In this mechanism the sender can send the minimum of the congestion window and the announced window of the receiver, also known as transmission window. Every time an acknowledgement is received, congestion window is maximized through one segment. At once the sender begins by transmitting one segment and stay for its acknowledgment. While an acknowledgement is received, the congestion window is maximized from one to two and two segments can be sent. While each of these two segments is recognized congestion window is maximized to four. This gives an exponential development of window. At few point the capability of the computer network can be arrived maximal and an intermediary router will begins to drop the packets. This is the time the sender recognizes that its congestion window is too bigger that provide the indication of Congestion in the network. Immediately it minimizes the sending rate and reduced CWND to one and starts process again.

5. SIMULATION AND ANALYSIS

A simulative network consisting of several senders, receiver and the router that was establish to regulate the simulation in the community using Network Simulator-2 (NS-2). The bottleneck of the system was the router and the output channel. The TCP operational algorithms is the buffer sizes of the router as well as the receiver and the bandwidth at the router output have been modified within the course of numerical research. The traffic into the network under the analysis provides a autonomous-analogy process that is random user-specific guidelines. Queue length, packet delivery rate, end-to-end wait, throughput are the primary metrics that are used in our simulation.

5.1 Performance Metrics

5.1.1 Throughput:

It is describes as the percentage of maximum rate of production to the given period of time. It can also be pronounced that it is the capacity of number of sent packets.

$$\text{Throughput} = \frac{\text{P}}{\text{T}}$$

5.1.2 Packet Delivery Ratio:

It is represented as the fraction of the number of packets that are initiated from source node and received at destination node.

$$\text{P.D.R} = \frac{\text{SP} - \text{RP}}{\text{SP}} * 100$$

5.1.3 Mean Queue Length:

It is defined as the moderate number of packets in the system. It can also be said that the ordinary number of packets waiting in the system to get delivered.

N = mean (expected) number of customer

= 0 × P[k Packets in system] + 1 × P[1 customer in system] + 2 × P[2 Packets in system] + ...

= P {k = 0, 1...} k × P [k Packets in system] (definition of "expected value")

= P {k = 0, 1...} k × pk

5.1.4 End -to -End Delay

It is describe as the complete time taken by the packet to transit from the source node to the destination node.

Table 1. Throughput Comparison of TCP Variants

S.No	Time	Tahoe	Reno	New-Reno	Vegas
1	10	10	10	10	10
2	20	500	500	510	550
3	30	1000	1000	1100	1100
4	40	1300	1200	1100	1000
5	50	3300	3200	3100	3000
6	60	5000	7000	9000	11000

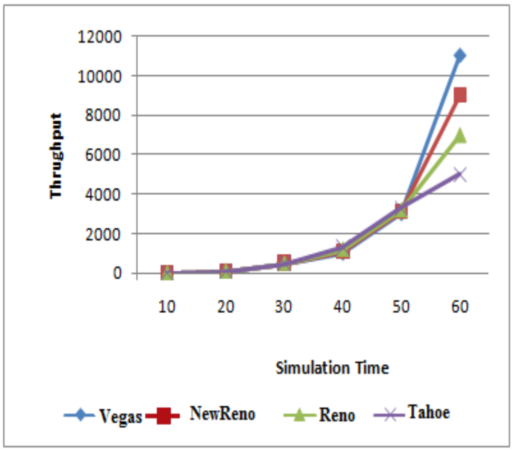

Fig 1.Throughput Comparison of TCP Variants

Figure1 shows that TCP Vegas exhibit higher throughput than other TCP Variants. TCP New Reno performs well when multiple packet loss is arriving in the network. TCP Reno is efficient to retrieve small number of packet loss; however it still undergo from performance problems when multiple packets are discarded from the window. New Reno tries to improve the TCP Reno performance when a multiple packets are lost by transforming the recovery algorithm. TCP Tahoe is a very simple algorithm and exhibit very low throughput.

Table 2. Packet Delivery Ratio vs. TCP Variants

S.No	Time	Tahoe	Reno	New-Reno	Vegas
1	10	1.00	2.00	3.00	4.00
2	20	0.80	1.75	2.50	3.45
3	30	0.72	1.50	2.25	3.15
4	40	0.72	1.50	2.32	3.35
5	50	0.65	1.25	2.00	2.85
6	60	0.60	1.00	1.75	2.55

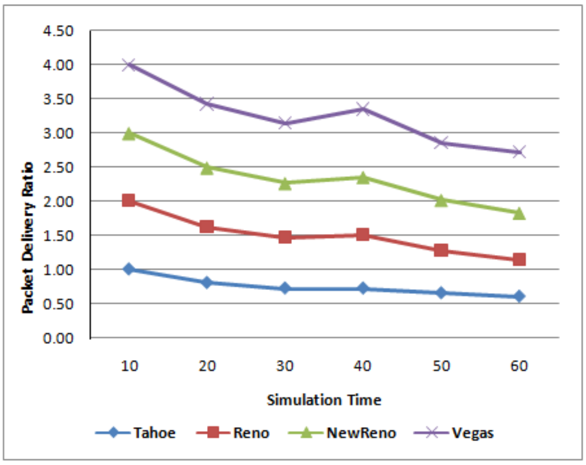

Fig 2. Packet Delivery Ratio vs.TCP Variants

Figure 2 shows that simulation time values ranges from 10, 20, 30, 40, 50, 60 in which the TCP Tahoe and TCP Reno is nearly in the smallest position than other TCP Variants. TCP New Reno in this case is quite increase the performance of network but the result of TCP Vegas is approximately high which is completely improved the performance of network. TCP Reno is efficient to retrieve small number of packet loss; however it still undergoes from performance problems when multiple packets are discarded from a window of data. TCP New Reno tries to improve the TCP Reno performance when a multiple packets are lost by transforming the recovery algorithm.

Table 3. Mean queue length vs. Simulation time

S.No	Time	Tahoe	Reno	New-Reno	Vegas
1	10	9702	9604	8918	5096
2	20	11,466	11,270	9310	5488
3	30	11,622	11,270	9506	6860
4	40	14,700	14,406	10,000	6870
5	50	14,994	15,484	11,760	7546
6	60	15,950	16,000	13,450	8051

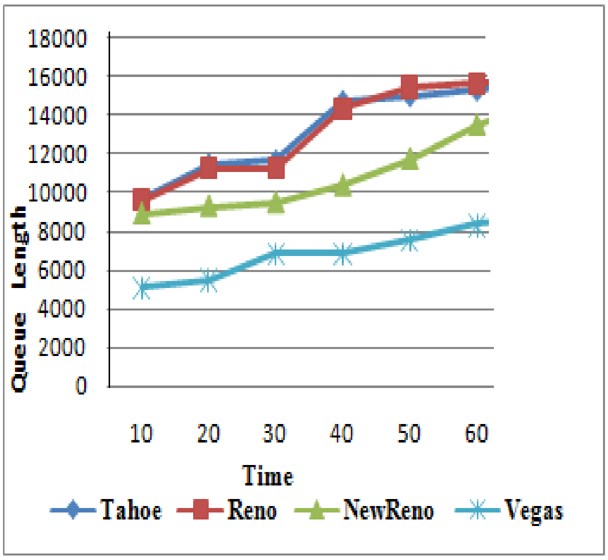

Fig 3. Mean Queue Length of various TCP Variant

Figure 3 shows that TCP Vegas connections continue to decrease their queue length. On the other hand, TCP Reno connections persist to increase their queue length before the buffer becomes full. The comparative analysis of TCP Variants was based on the amount of lost data, buffer capacity of the router, utilization of channel and performance of network.

Table 4. The number of received packets for added delay

S.No	Time	Tahoe	Reno	New-Reno	Vegas
1	10	6448	7001	6934	7268
2	20	6141	6667	6603	6943
3	30	5834	6395	6273	6523
4	40	5716	6146	6146	6148
5	50	5325	5726	5726	6106

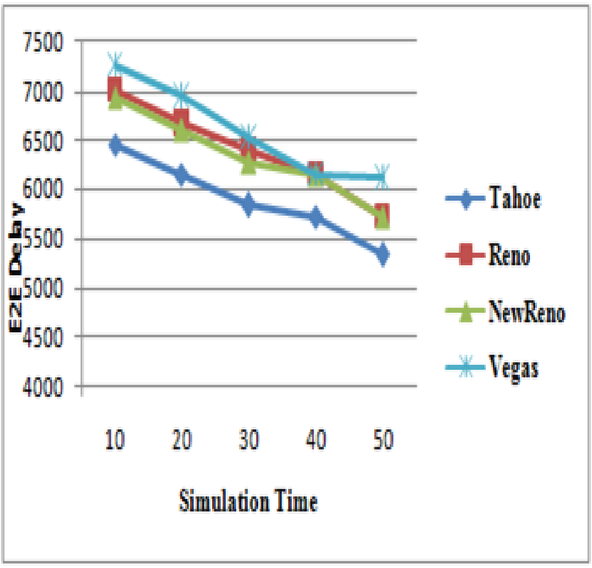

Fig 4.The number of received packets for added delay

Figure 4 show that TCP Tahoe, TCP Reno, TCP New-Reno, is uniformly maximizing the performance of the network by minimizing the delay of packets. TCP Vegas minimize the performance by maximize the value of End-to-End delay. TCP-Vegas protocol tries to implement a number of enhancements such as more processing that is advanced evaluation of RTT. Vegas give best performance in high congested network.

6. CONCLUSION AND FUTURE SCOPE

In order to reduce the problem of congestion in network, various TCP variant have been used. Nonetheless, it is important to know which TCP variant is suitable in which network. Here, we analyzed the working of four TCP variant, viz. TCP Tahoe, TCP Reno, TCP New Reno and TCP Vegas on the basis of results. It can be concluded that TCP Tahoe is a very simple variant, it take a complete timeout to discover a packet loss. It does not send an instant acknowledgement, every time a packet is lost it waits for a timeout and the pipeline is vacate. This cause a high bandwidth loss and decrease the transmission rate.TCP Reno perform well during the number of packets loss is small. While in multiple packet loss, it does not perform too well and it performance is same as TCP Tahoe. In multiple packet loss, TCP New Reno performs best. The problem with TCP New Reno is that it take one Round Trip Time (RTT) to detect packet loss.TCP Vegas perform best in high congested network, it detect congestion before it happen. But still there are many problem in TCP Vegas, when TCP Vegas share a same bottleneck link with TCP Reno, it create fairness, rerouting problem.

In future, we propose a new Bandwidth Estimation scheme and also compare TCP Variants with this new Bandwidth Estimation scheme in order to enhance the performance of network. This new scheme is also used for solving TCP slow start problem, in which there is a definite lag in TCP start up, we will try to remove this lag be novel bandwidth estimation strategy. It has an improved slow-start phase and realizes a novel congestion avoidance phase.

7. REFERENCES

[1] Wang, Zhiming, et al. "TCP congestion control algorithm for heterogeneous Internet." Journal of Computer Applications and Networks 68 (2016): 56-64.

[2] Ou, Shih-Hao et al. "Out-of-order transmission facilitates Congestion and scheduling control for multipath TCP." Cellular Communications and Mobile Computing Conference (CCMCC), 2016 International. IEEE, 2016.

[3] Domanski, Adaman, Domanska, J. Pagano, M., & Czachorski, T. (2016, October). The Liquid Flow Similarity of the TCP RENO and VEGAS Congestion Control Mechanism. In International Symposium on Computer and Information Sciences (pp. 193-200). Springer International Publishing.

[4] Abolfazli Elham and Vahid Shah-Mansouri (2016). Dynamic adaptation of queue levels in TCP Vegas-based networks. Electronics Letters, 52(5), 361-363

[5] Radhika Mittal, N. Dukkipati, Emily Blem, Hassan Wassel, M.Ghobadi, A.Vahdat & David Zats (2015, August). TIMELY: RTT-based Congestion Control for the Datacenter. In Computer Communication Review (Vol. 45, No. 4, pp. 537-550). ACM SIGCOMM.

[6] Peng, Yang, et al. “Identification of TCP congestion avoidance algorithm." IEEE/ACM Transactions on Networking 22.4 (2014): 1311-1324

[7] Zhenwei Zhu, Yu Qian, Zhou, and Liu Wenjia, Keren. "Dynamic Vegas: Efficient Congestion Control Mechanisms." In International Conference on Information Technology and Computer Science, pp. 333-340. Springer India, 2014.

[8] Wu Qing-Rui, Jie Hong & Nan Ding (2014). An Enhanced TCP Congestion Control Algorithm Based on Estimation of Bandwidth in Heterogeneous Networks. Journal of Communications, 9(10),2014

[9] Winstein, Balakrishnan Hari and Keith."TCP ex machina: Computer-Originated Congestion Control." In Computer Communication Review, vol. 43, no. 4, pp. 123-134.ACM SIGCOMM,(2013)

[10] Alfredsson Stefan, Garcia, Del Giudice , Brunstrom Anna, Luca De Cicco, & Saverio Mascolo. (2013, June). An Impact of TCP Congestion Control on Bufferbloat in Wireless Networks. In World of Mobile Networks, and Wireless Multimedia (WoMoNWM) (2013 IEEE 14th International Symposium and Workshops on a (pp. 1-7). IEEE

[11] Wang, Jingyuan, Jiangtao Wen, Yuxing Han, Chao Li, Jun Zhang and Zhang Xiong. "CUBIC-FIT: A tremendous performance and TCP Cubic friendly congestion control algorithm." IEEE Communications Letters 17, no. 8 (2013): 1664-1667

[12] Mark Allman, Vern Paxson, and Ethan Blanton. TCP congestion control. No. RFC 5681. 2009.

[13] Lar Saleem-ullah and Xiaofeng Liao. "An initiative for a classified bibliography on TCP/IP congestion control." Journal of Computer Applications and Network 36, no. 1 (2013): 126-133

[14] J.Chicco, D.Collange and A. Blanc, "An imitation study of new TCP variants," in 2010 IEEE International Conference on Computers and Communications (IICCC), pp. 50–55, 2010.