International Journal of Computer Applications (0975 – 8887)
Volume 167 – No.13, June 2017

EARRA: Enhanced Adaptive Rate Response Adjustment Technique for Congestion Control in Networks

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

Internet has evolved into dynamic network where users use wired or wireless connection technologies. The dissimilarity of bandwidths, interruption and error rates in network are increased on Internet. TCP is the common denominator for several services thus by modifying TCP the need of applying solutions locally can be reduced. Now the researchers have focuses in designing an enhanced bandwidth estimation technique which are efficient to utilized bandwidth. This paper proposed modified Slow-start mechanism, called an Enhanced Adaptive Response Rate Adjustment (EARRA) bandwidth estimation technique to improve the startup performance in wireless networks and estimating the available bandwidth (ABW) of an end-to-end network path more accurately and less intrusively In this paper EARRA technique use Eligible Rate Estimation (ERE) mechanism that adaptively and repeatedly resetting ssthresh during the slow-start phase. By adjusting to network situation throughout the startup section, sender is able to grow the congestion window fast without incurring uncertainty of overloaded buffer. This technique adjust the ssthresh dynamically , readjust the slow start threshold according to the current carrying capacity of the network and determine the sending end whether is in the slow start phase or congestion avoidance phase based on the new size of ssthresh. This allows the volumes of data packets that sent from the sending end keep changes along with the change of the ability of network capacity. The performance of proposed EARRA technique is evaluated by throughput, packet delivery ratio, latency through simulation set, which evidence better performance in comparison to different tcp variants.

Keywords

Congestion Control, Wireless Network, TCP Variants, Slow Start, EARRA**.**

1. INTRODUCTION

The Transmission Control Protocol (TCP) [9] has been generally utilized in today's network. The protocol maintains reliable information transport by beginning a connection among the transmitting and receiving nodes. The sender start a timeout method when transmit a packet to the recipient node. The sender consistently records the Round Trip Time (RTT) for its packets via to estimate the relevant timeout period. Through the receiver node, each obtained packet is recognized inevitably or definitely to the sender. Whenever the sender does not get an acknowledgment for a specified packet at the equivalent timeout duration terminate, the packet is considered to be deficit and subject to Retransmission.

A Congestion Window (CWND) along dynamically adapted dimension is used by the protocol to manage the traffic flow from the sender to the receiver. Internet behavior is deeply rely upon TCP congestion control operation. Though the number of packets is deliver to the network is much more than the volume of network, congestion arise in the network and by that packets are discarded .TCP congestion control goal to average the transmitting rate of flows for preventing congestion. Various versions of TCP Variants [10] are appeared to enhance the performance of the wireless network.

The primary objective of TCP was to conveniently use the accessible bandwidth in the network and to prevent overloading in the network and the resulting packet deficit by suitably inhibit the transmitter sending rates. Congestion in the network is considered to be the fundamental reason for packet deficit .Therefore the performance of TCP is frequently inadequate while used in wireless networks and needed several enhancement techniques Now we are dealing with broadband networks, the requirement of speedy slow start stage is understandable if we want to implement better utilization of network. This is the main focus area of our research and inspires us to enhance the slow start algorithm in TCP Variants. [10]

In this paper we proposed an Enhanced Adaptive Response Rate Adjustment (EARRA) bandwidth estimation technique to enhance the slow startup performance in wireless network. This proposed approach used Eligible Rate Estimation (ERE) process that adjusting and regularly rebuilding SSTHRESH during the slow start stage**.** This will help in enhancing the performance of network and also compare this technique with other TCP variants [9] on the basis of various performance metrics.

The rest of the paper is systematized as follows: Section 2 gives brief overview of TCP Congestion Control algorithm phase; Section 3 discusses some related work and motivates the need for our approach. Section 4 describes our proposed work, section 5 show performance evaluation, section 6 presents the simulation results. Section 7concludes the paper.

2. TCP CONGESTION CONTROL

TCP congestion control algorithm has four phases: 1) Slow Start 2) Congestion Avoidance 3) Fast Retransmit 4) Fast Recovery.

2.1 Slow start

TCP uses slow start algorithm [3] to restrict transmission ratio of the sender. This stage has been adept by accepting the ratio of the acknowledgment from receiver. Though TCP start a connection, the slow start procedure set cwnd to 1 segment. During this stage cwnd = MSS (Maximum Segment Size). While an ACK is recover by receiver, the cwnd maximized by 1segment for every acquired acknowledgment .This stage is really not so much slow, because each time when ACK received, congestion window maximized at dual rate, i.e. when sender gets first ACK, sender maximized cwnd by 2 segments, when sender receive other two acknowledgement, sender increase cwnd by 4segments, so on. At threshold level cwnd extent at highest level and packets deficit will produce and sender goes into congestion avoidance phase. [5][11]

2.2 Congestion Avoidance

A point during slow start that network is forced to drop one or more packets due to congestion. If this appear, Congestion avoidance phase is used [3][11] . In congestion avoidance process, the sender knows about deficit of packets due to congestion during duplicate ACK obtained by sender. The sender immediately reduces the cwnd by one half of current window size, but to at least two segments. If timeout arise due to congestion, decrease congestion window to 1 segment, which necessarily set the sender into slow start mode. However in this phase slow start is only use up to the halfway point where congestion originally occurred. After this halfway point, cwnd is increased by one segment. If the congestion was noticed by DUPACK (duplicate acknowledgment), starts fast retransmission and fast recovery algorithm.

2.3 Fast Retransmit

When DUPACK received by sender, it does not know the actual reason that the segment was lost or simply that segment was delayed. Typically no more than one or two duplicate ACKs should be received when simple segment has been delayed. But when more than two DUPACKs received by the sender, it is a strong indication that at least one segment has been lost due to congestion. When three DUPACK are received, the sender does not wait for time out and immediately retransmit lost segment. This procedure is called fast retransmission. [3]

2.4 Fast Recovery

With the help of duplicate acknowledgement, the sender know about another segments obtain favorably at receiver. This is a strong indication that serious congestion may not happen and loss of the segment due to delayed. So instead of decreasing window abruptly by going all the way into slow start, the sender only arrive in Congestion avoidance mode. The sender does not fix cwnd to 1 segment as in slow start phase, but proceed transmission with larger window and continuous incrementing. This allows better throughput and performance under moderate congestion.

3. RELATED WORK

There have been several solutions proposed for enhancing the performance of TCP in wireless networks. This section briefly summarizes the existing research related to the work done in this paper. The related work is classified into two parts..The first part describes the existing tcp variants and second part describes the existing bandwidth estimation technique.

3.1 Existing TCP Variants

3.1.1 TCP Tahoe

TCP Tahoe[1] [9]was initiated with 3 congestion control algorithms specially Slow Start , Congestion Avoidance , Fast Retransmit .Different parameters for congestion control were added like Congestion Window (CWND)) and Slow Start Threshold Window (SSTHRESH). The dimension of congestion window change as per the acceptance of acknowledgements for the deliver packets. Failure to get an acknowledgement since the end of the defined RTT duration is elucidated as packet deficit as congestion through TCP. The next for TCP is to perform fast retransmit where in if the transmitter get 3 duplicate acknowledgement for the same packet, then it retransmits the packet without waiting for the timeout. Entering fast retransmit, it sets the ssthresh to half of the current cwnd and decrements the cwnd to 1. This leads to TCP entering the slow start stage and the cwnd increments exponentially for every ACK it receives for sent packet until the ssthresh is reached. Thereafter, TCP enters congestion avoidance where cwnd increments linearly for every received ACK. The window size is taken as minimum of congestion window and advertized window. The problem with TCP Tahoe is that it holds an entire timeout interval to identify packet losses. When the packet loss is identified, the performance of TCP Tahoe is too slow. Due to the fact that the flow transmission reduced.

3.1.2 TCP Reno

Reno [1] [9] [10] has same congestion control algorithms as Tahoe with an addition of fast recovery. In fast recovery, ssthresh and new cwnd is set to half of the current cwnd instead of setting the cwnd to 1. Thus Reno skips slow start and directly enters congestion avoidance. However, it undergo with performance problem when several packets are released from a window as it discovers only small packet losses

3.1.3 TCP New Reno

New Reno [1] [10] [4] was introduced as Reno does not perform well when there are multiple packet drops in the same window. Unlike Reno, New Reno does not come out of the fast recovery unless and until it receives ACKs for all the packets that were present in the window while entering fast recovery.

3.1.4 TCP Vegas

Vegas [1] [9] [12] implements congestion avoidance rather than first detecting the congestion and then taking steps to decrease congestion on the channel. Vegas basically calculate a base RTT and compare it with RTT of packet with recently received ACK. If compared RTT is much smaller than base RTT it increases its sending window and if RTT is greater than the base RTT, it decreases its sending window. TCP Vegas is consider as balanced congestion control technique, nonetheless, shortcomings such as small bandwidth utilization and not gain fairness while sharing connection with TCP Reno.

3.1.5 TCP Dynamic Vegas

TCP Dynamic Vegas [12] is the modification of TCP Vegas that modify the basic TCP Vegas in Slow-Start and Congestion Avoidance stage. It dynamically chooses slow start algorithm and adjust decrease/increase rate in congestion avoidance phase according to specific network environment.

3.2BANDWIDTESTIMATION TECHNIQUE

3.2.1 Trains of Packet Pairs (TOPP)

TOPP sends packet streams and gradually increases the stream rate to calculate the available bandwidth. It is based on sending many pairs of equal sized packets with a starting transmission rate to a receiver.

3.2.2 An Adaptive Response Rate Adjustment mechanism (AARRA)

In the actual environment performance, the network structure is consistently redesign, and the delay of wireless connection is Excessive, so the estimated bandwidth may be distorted. It effect the transmission rate if put it correctly, so it is necessary to create a smoothness around the evaluated value, in which process we can also make it reply quickly when the modification of bandwidth. This technique is formed to conclude the assessed value of bandwidth on the basis of original time adaptation of network. [13]

4. PROPOSED WORK

This paper aims to develop an Enhanced Adaptive Response Rate Adjustment (EARRA) bandwidth estimation technique for congestion in networks. In this paper, we propose a simple sender-side only modification, which we call Enhanced Adaptive Response Rate Adjustment bandwidth estimation technique to enhance the startup performance in wireless networks and for evaluating the available bandwidth (ABW) of an end-to-end network route more accurately. In this paper EARRA technique takes advantage of the Eligible Rate Estimation (ERE) mechanism that adaptively and repeatedly resetting *ssthresh* during the startup phase, both connection startup, and after every coarse timeout .By adapting to network conditions during the startup phase, a sender is able to grow the congestion window (*cwnd*) fast without overflowing the bottleneck buffer. Astart is more robust to small buffer sizes, avoiding premature termination of Slow-start, as well as multiple losses and the resultant coarse timeout.

EARRA combined an inline available bandwidth probing over TCP scheme namely inline measurement TCP into the TCP slow start mechanism. The slow start threshold ssthresh can thus be set adaptively with the measured bandwidth. It recommends an effective way to evolve the congestion window from its primary value to the slow start threshold ssthresh rapidly and smoothly. It includes TCP pacing to fill the connection pipe smoothly.

5. PERFORMANCE EVALUATION

The proposed EARRA performance is evaluated through NS-2 version 2.34 is used. NS-2 is an object oriented simulator. NS uses two distinct languages. NS is written in C++ and in frontend use tcl interpreter. The comparison results are also shown with the help of graphs

In the simulation setup, the proposed EARRA technique is simulated and is compared with existing TCP Variants. In the simulation scenario, 22 nodes are taken which are randomly deployed in a region of 1000m × 1000m. The simulation time is 150sec. We also generate 22 TCP connections between random senders and receivers .To analyze the performance of the proposed technique ten experiments are performed on the basis of the equal time interval. The simulation traffic is CBR. Figure 2–4 present the simulation results for the proposed work along with existing technique. For the results the mean queue length, throughput and packet delivery ratio, latency is calculated. The simulation parameters are shown in Table 1
The sample screen shot for 22 nodes is shown in fig 1. The proposed EARRA technique is compared with the TCP Variants. The following performance metrics are mainly considered according to which the performance is evaluated.

5.1Packet Delivery Ratio (PDR):

PDR is obtained by the dividing the number of packets received by the number of packets sends by the source nodes. This performance evaluation metric indicates the data delivery reliability.

5.2Throughput

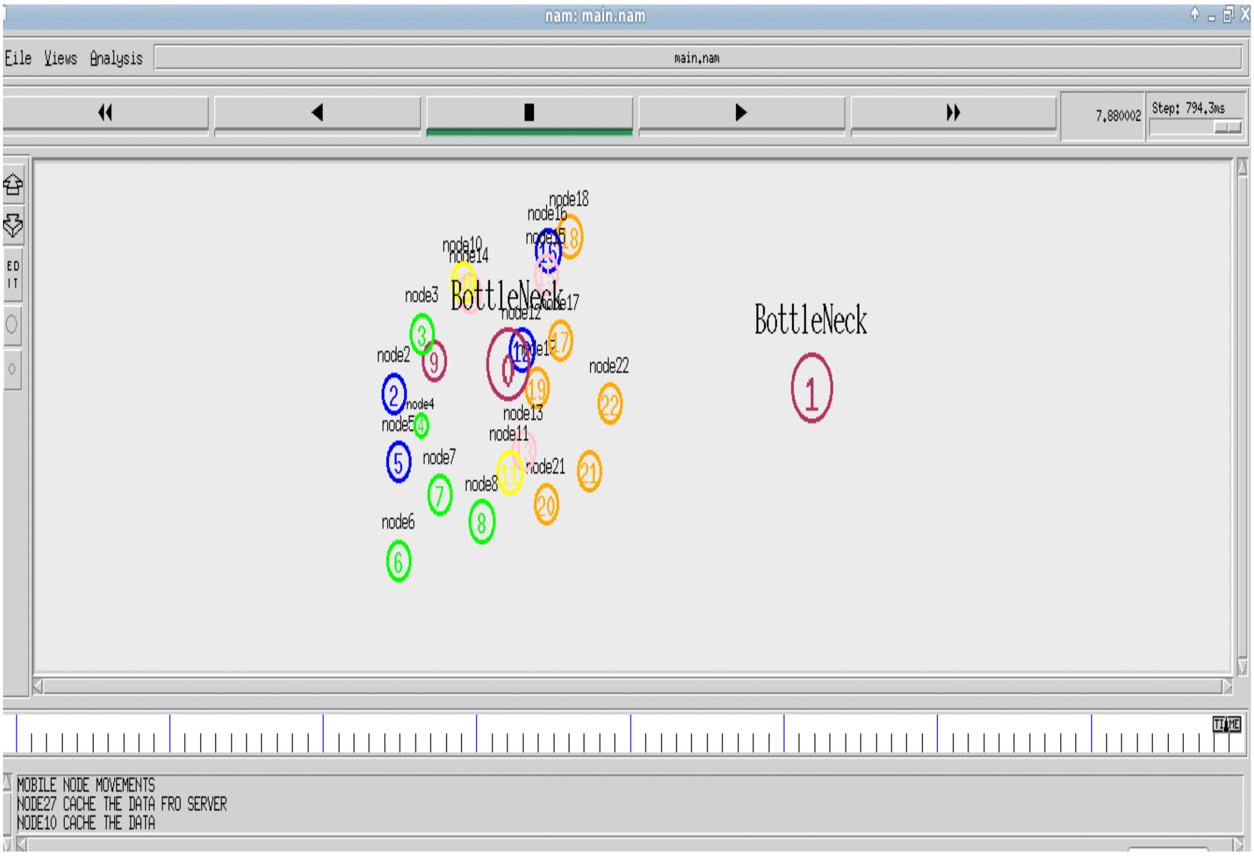
Throughout is defined as the total number of packets received in unit time. This performance evaluation metric indicates the data transfer, Higher the data transfer higher the value of throughput metric.

5.3Latency

It refers to time interval or lag when a system element is waiting for another system element to do a task.

Table 1. Simulation Parameter for evaluation of routing protocols

S.No	**Parameters**	**Values**
1	Simulation used	NS-2.34
2	Dimension of Simulation Area	1000m x 1000m
3	No of nodes	22
4	Simulation time	150s
5	MAC protocol	802.11
6	Traffic Type	CBR
7	Average speed	11.40m/s
8	Maximum connection	10
9	Transport Layer	TCP
10	Congestion Window Type	TCP Tahoe, TCP Reno, TCP New Reno, TCP Vegas, TCP Dynamic Vegas
11	Packet size	512 bytes
12	Mobility of node	random



MOBILE NODE MOVEMENTS
NODE27 CACHE THE DATA FRO SERVER
NODE10 CACHE THE DATA

Fig 1. Sample screenshot of 22 nodes

6. RESULTS

In the simulation setup, the proposed EARRA technique is simulated and is compared with TCP Variants. In the simulation scenario, 22 nodes are taken which are randomly deployed in a region of 1000m × 1000m. To analyze the performance of the proposed technique ten experiments are performed on the basis of the equal time interval. Figure 2–4 present the simulation results for the proposed work along with existing technique. For the results the mean queue length, throughput and packet delivery ratio, latency is calculated.

Figure 2 shows the result for throughput. Throughput of the proposed algorithm, EARRA is high in comparison to the TCP Variants. So EARRA performs better in context to throughput from the other TCP Variants.

Figure 3 shows the result for packet delivery ratio (PDR). The graph shows that the proposed algorithm, EARRA has high packet delivery ratio in comparison

Figure 4 shows the result for network latency. The proposed algorithm EARRA has less latency in comparison to other tcp variants.

Table 2. Throughput

Network Load	**2**	**5**	**10**	**20**	**30**	**50**	**100**
EARRA	5.3	5.1	4.8	5	4.2	4.4	4
Dynamic Vegas	5.2	5.1	4.2	4	3.7	3.5	3.3
New Reno	4.5	4.3	4.2	4.1	4	3.5	3.4
Reno	2.5	2.4	2.2	2	1.7	1.3	1.2
Tahoe	1.5	1.3	1.1	0.9	0.8	0.7	0.6

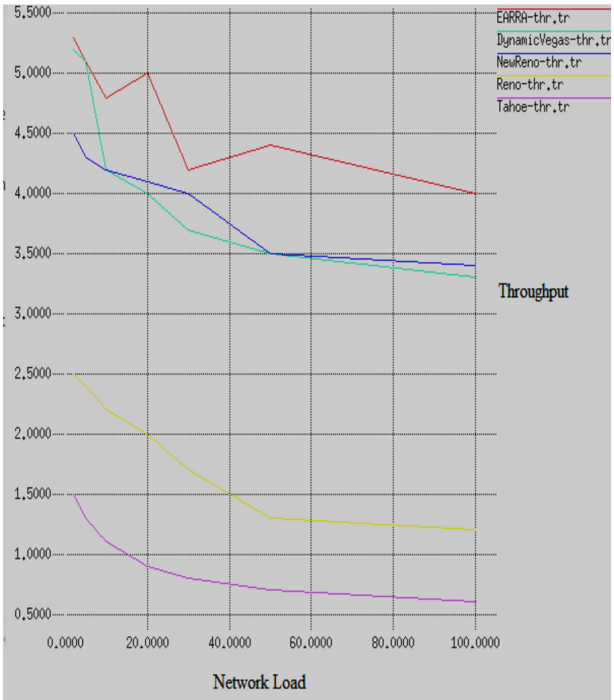


Fig 2. Throughput

Table 3.Packet Delivery Ratio

Simulation Time	**10**	**20**	**30**	**40**	**50**	**60**
Tahoe	0.99	0.800	0.720	0.720	0.650	0.600
Reno	0.99	0.820	0.750	0.780	0.620	0.540
New Reno	0.98	0.870	0.790	0.850	0.740	0.680
Vegas	0.99	0.940	0.880	1.000	0.850	0.900
EARRA	0.97	0.960	0.920	1.000	0.890	0.950

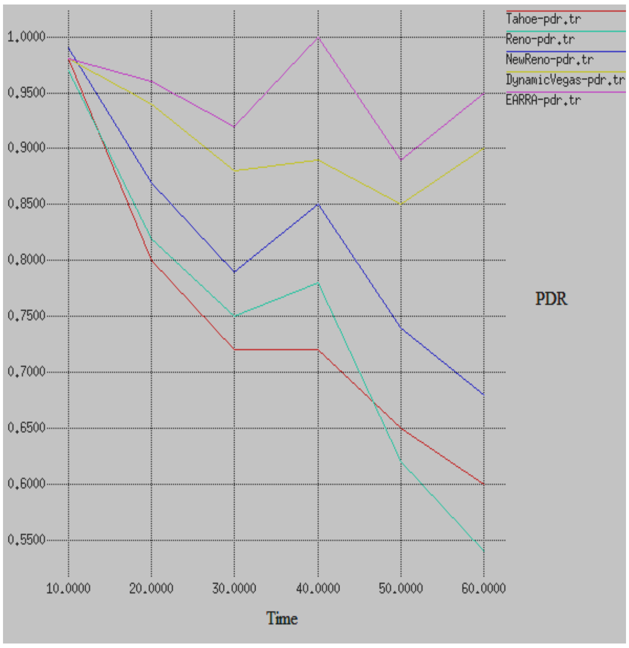


Fig 3. Packet Delivery Ratio (PDR)

Table 4. Latency

Simulation Time	**10**	**20**	**30**	**40**	**50**	**60**
Tahoe	1	0.9	0.91	0.88	0.88	0.86
Reno	0.81	0.8	0.78	0.76	0.75	0.74
New Reno	0.78	0.72	0.72	0.7	0.65	0.63
Dynamic Vegas	0.6	0.6	0.58	0.56	0.5	0.48
EARRA	0.6	0.6	0.6	0.59	0.42	0.4

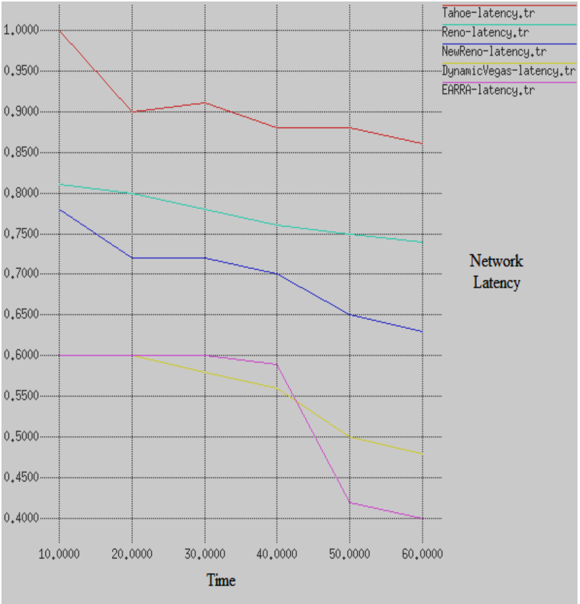


Fig 4. Latency

7. CONCLUSION AND FUTURE SCOPE

This paper explains the comparative analysis of existing TCP Variants. The paper also discusses the concept of various bandwidth estimation techniques for congestion control in wireless network. The paper also presents the Enhanced ARRA (EARRA) Bandwidth estimation technique. In this paper, the performance of EARRA is compared with existing TCP Variants. NS-2.34 is used for performing the simulation. The comparison results show that EARRA has higher packet delivery ratio in comparison to other TCP variants. The throughput of EARRA is better than other TCP variants and network latency is less in EARRA technique.

In future we compare our proposed technique with different routing protocol by using various performance metrics such as throughput, mean queue length, jitter, delay etc.

8. REFERENCES

[1] Yew, R. Ahmad R. et al., “Performance Evaluation of TCP Vegas versus different TCP Variants in Homogeneous and Heterogeneous Networks”, International Journal of Computer Science and Information Security (IJCSIS), Vol. 14, No. 3, March 2016.

[2] Subramani B., T. Karthikeyan. “A Review on Congestion Control.” International Journal of Advanced Research in Computer and Communication Engineering Vol. 3, Issue 1, January 2014.

[3] Zeng Xiaoping, Chen Li et al., "TCP congestion control mechanism for heterogeneous network connection." Journal of Computer Applications and Networks, 2016, pp: 56-64

[4] Attiya G., Z. Morsi et al., “Modified Fast Recovery Algorithm for Performance Enhancement of TCNewReno.” International Journal of Computer Applications Vol. 40, No.12, February 2012.

[5] Ganiyu A. Oyeyinka, Olusegun Folorunso, Aderounmu, "TCP Window Based Congestion Control Slow-Start Approach,"Communications and Network, Vol. 3, pp.85-98, May2011

[6] Chary, Leonard Kleinrock et al.,” Improving Host-to-Host Congestion Control Protocols by Dynamic Bandwidth Estimation of the Network, International Journal of Computer Applications (0975 – 8887) Volume 104, No.1, October 2014.

[7] Francis, B., Narasimhan, V., Nayak, A. and Stojmenovic, I. (2012), “Techniques for Enhancing TCP Performance in Wireless Networks”, International Conference on Distributed Computing Systems Workshops IEEE, DOI 10.1109/ICDCSW, pp. 222-230.

[8] Prasanthi. S and Sang-Hwa Chung “An Efficient Algorithm for the Performance of TCP over Multi-hop Wireless Mesh Networks” 609-735, IEEE 2010.

[9] Babatunde O. Oyenike Mary Olanrewaju2 , Isaiah O. Adebayo3,Transmission Control Protocol and Congestion Control : A REVIEW OF TCP VARIANTS.International Journal of Computer Science and Information Security (IJCSIS), Vol. 14, No. 3, March 2016

[10] Ghulam Kassem Fozia Hameed TCP Variants: An Overview2010 Second International Conference on Computational Intelligence, Modeling and Simulation.

[11] 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 .

[12] 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.

[13] Wang X., K. Zheng, et al.,”Novel Quick Sort (QS) approach for optimization of TCP “in proceeding of Springer science business media New York, 2015

[14] Luigi Fratta, fabio Martignon Bandwidth Estimation Schemes for TCP over Wireless Networks IEEE Transactions on Mobile Computing, VOL. 3, NO. 2, APRIL-JUNE 2004.

[15] NetworkSimulator: www.isi.edu/nsnam/ns/doc/ns_doc.pdf

9. AUTHOR PROFILE

Pooja Chaudhary, is a Research scholar, M.Tech student at the Department of Computer Science and Engineering,A.K.G.E.C college. She has published a paper in ICCCA- 2017 sponsored by IEEE and also presented an article in National Conference on Emerging Computer Technologies COMTECH-2016. Her current research includes congestion control, networking, multiconnection wireless network, dynamic programming approach.

Dr. Sachin Kumar is a Ph.d, Associate Professor at the Department Computer Science, A.K.G.E.C College. His current research interest includes data mining technique, wireless sensor network, fuzzy time series, and congestion control. He is an Editorial Committee member of 2013 3rd IEEE International Advance Computing Conference.