IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-6, Issue-6, June 2018 E-ISSN: 2347-2693

Simulation Windows Size Quadric Increase Congestion Control Algorithm Implementation using NS3 in Wired Computer Networks Scenario

G. Paliwal[1*], S. Taterh[2]

[1]Amity Institute of Information Technology, Amity University, Jaipur, Rajasthan, INDIA
[2]Amity Institute of Information Technology, Amity University, Jaipur, Rajasthan, INDIA

Corresponding Author: gpaliwal@jpr.amity.edu, Tel.: +91-9351342925

Available online at: www.ijcseonline.org

Accepted: 17/Jun/2018, Published: 30/Jun/2018

***Abstract*—**Now these days most of the electronic devices are connected to the internet through wired or wireless computer network. This is very popular networking terminology known as the internet of things IOT, In background each and every internet networking devices are connecting to the backbone networking server. If this backbone network route is congested then IOT devices are suffered from their services and all these things are lacking in performance. The high throughput and low latency internet services are required for each device to do well. There is a number of reasons behind the internet speed. The most important reasons are that the TCP/IP protocol does not completely use the actual channel bandwidth and congestion occurrence during the data transmission. TCP provides the connection-oriented connection. TCP may have a problem with utilizing the full bandwidth of the communication channel. Numbers of congestion control proposals have been suggested to reduce this problem. This paper presents the implementation of a quadric increase congestion control algorithm and it's simulation through the NS3. This algorithm is based on binary increased congestion control algorithm (TCP BIC). In ns3 TCP QIC is separately implemented and tested with the different congestion control algorithms. These congestion control algorithms are TCP Westwood, BIC, NewReno, scalable and Illinois. The performance of the TCP QIC has the significance over the other congestion control algorithms in respect of throughput, goodput, delay variance and round-trip time.

***Keywords*—**Hybrid TCP Illinois, Congestion Control Algorithm, TCP BIC, TCP New Reno, NS3, TCP (Transmission control protocol) , QIC: Quadric Increase Congestion control algorithm.

I. Introduction

Over the past decades, so much work has been already done to enhance the TCP performance and the result of many different congestion control algorithms. The number of congestion control algorithms are already developed and implemented after finding some enhancement result over the existing congestion control algorithms [1]. The authors comment on the heterogeneous resource allocation to achieve the maximum efficiency of the devices for load balancing. The proposed load balancing algorithm compared with others to check their credibility. The load balancing is the major problem to achieve the maximum throughput and make network congestion free but in real life load balancing applied, thereafter congestion occurs and required a best approach to resolve the congestion [2]. In this research article, the authors proposed an algorithm for packet routing in a wireless sensor network to increase the performance of the WSN, in this article the focus to reduce the packet loss rate during routing, reduce energy consumption and increased the throughput[3, 4]. The implementation and comparing them with other congestion control algorithms is troublesome without network simulators. In the real world scenario, it is very difficult to compare algorithm performance on the basis of actual devices it is possible with the simulators it is less time to consume and less expensive and given the moral as actual results. Computer simulator provides valuable insight into potential implementation before extending the considerable effort that may be required to build a technology simulator allow us to measure and compare the existing algorithm as well as we can introduce a new algorithm and compare in light of existing algorithms. It provides a much more flexible way to implement a new algorithm and finding out the related result on the basis of some existing networking topology [5].

This paper presents an implementation of the QIC congestion control algorithm that is based on the TCP BIC. In order to the ns3 TCP BIC that is most widely used by the Linux operating system [6]. I enhance the congestion control algorithm TCP back that use the methodology of the congestion window increase in a binary manner whereas I will change a little bit in this algorithm to check the performance of quadric congestion control window increase the effect is immense to attract our attention.

The rest of this paper is organized as in the following sections the basic of the TCP in section II, section III related work of the TCP congestion control algorithm, section IV simulation setup, and section V have the simulation result matrix the last section is the conclusion of the present quadric congestion control algorithm performance[7].

The TCP having four congestion control algorithms to implementation: First slow start, second congestion avoidance, third fast retransmit and fourth fast recovery explained in details in (RFC2581)[8].

A. Slow start and congestion avoidance algorithm implementation

In the phase of a slow start and congestion avoidance algorithm implementation must be used by a TCP sender data controlling to inject outstanding data into the network. This can be handled by two variables the congestion window (cwnd), this is a sender side limit of data amount transmission before receiving an acknowledgment (ACK) and receiver advertised window (RWND) is the receiver side limit of data amount outstanding for processing. Another variable is slow to start threshold (ssthresh), it is used for determining the congestion avoidance algorithm is used to data transmission control. The initial cwnd size depends on the SENDER MAXIMUM SEGMENT SIZE (SMSS) and cwnd set to less or equal to 2 * SMSS. on receiving every ACK cwnd is increased by full segment size until not detected duplicate ACK. when packet loss detected then set the maximum ssthresh by the maximum value between flightsize/2 and 2*SMSS, where flightsize is the amount of outstanding data in the network. The threshold selection is important for quality of service requirement for the communication network. The intelligent and optimal threshold selection is for both local and global communication schemes. [8]

B. Fast Retransmit/Fast Recovery

The TCP sender should use the "fast retransmit" algorithm to detect and repair loss, based on incoming duplicate ACKs. After receiving 3 duplicate ACKs, TCP performs a retransmission of what appears to be the missing segment, without waiting for the retransmission timer to expire.

1. If 3DUPACK is received, set ssthresh to max (flightsize/2, 2*SMSS)

2. Retransmit the lost segment and set cwnd to ssthresh plus 3*SMSS.

3. For each additional duplicate, ACK received, increment cwnd by SMSS.

4. Transmit a segment, if allowed by the new value of cwnd and the rwnd.

5. When the next ACK arrives that acknowledges new data, set cwnd to ssthresh

This ACK should be the acknowledgment elicited by the retransmission from step 1, one RTT after the retransmission.[8] Figure 1 shows the TCP congestion control algorithm implementation data flow diagram with the different stages.

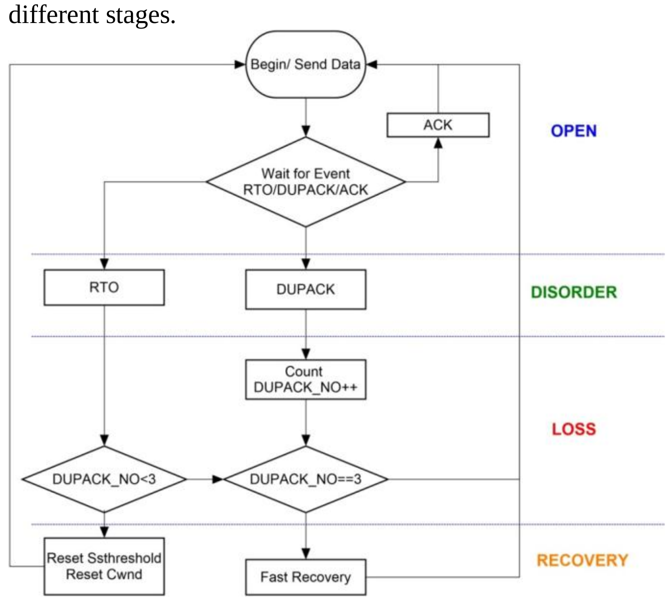

Figure 1. TCP Congestion control algorithm with fast recovery flow representation

In this paper, we propose an improved version of TCP congestion control algorithm, named TCP-ACC based on the TCP BIC. The algorithm TCP ACC is simulated and analysis with the other congestion control algorithms as NewReno, BIC, and Hybrid Illinois. Experimental results obtained using network simulators NS3 that show the performance of the algorithm is significantly improved as compared to other.

The rest of this paper is organized as follows. In section II, we give an overview of the goal of the congestion control algorithms. In section III, we give a description of the NS3 simulation setup and analysis of different algorithm results. In the section, IV proposed TCP-ACC algorithm analysis with others. Section V Finally, we summarize the findings.

II. TCP BASICS AND GOAL OF THE CONGESTION CONTROL ALGORITHM

This provides a connection-oriented connection between sender and receiver. A Connection-oriented communication means TCP send packets between computers and provides acknowledgment number through the sequence number. Unacknowledged TCP packets are retransmitted based on the information about successfully received sequence number and in some cases, round trip time out of the packet is used for retransmission of the packet. TCP congestion control algorithm proposed over the last 30 years roughly to maximize the throughput and avoiding the congestion at the internet routers and provide the services effectively and efficiently. TCP congestion control algorithms have been having some common features these features are on the basis of changing in congestion window acknowledgment received round trip time of a packet. On the basis of these parameters each and every TCP congestion control algorithm wants to provide good throughput and Plus latency time and use the maximum capacity of the channel [8].

TCP congestion window control the number of packets a TCP flow they have in the network at any time the congestion window is set using additive increase multiplicative decrease AMD mechanism that probes the available bandwidth dynamically adapting to changing network conditions the AMD mechanism work well when the sender continuously has data to send as is typically the case for TCP used for bulk data transfer current TCP implementation have a range of the behaviors for starting up after an ideal period some current TCP implementation slow start after an ideal period longer than the RTO estimates as suggested in [8] RFC 2581 while other implementations don't reduce their congestion window after an ideal period. When the congestion window is reduced the slow start threshold remains as a memory of the recent condition window. sending of back to back packets are facing packets out over the period of a round trip time and old congestion window that has not been fully using for some time cannot be trusted as an indication of the bandwidth currently available for the floor we would contain that the mechanism to piss out packet allowed by the congestion window are largely used to determine the appropriate size of the congestion window [9]. The Mechanic to determine the number of RTO in the most recent ideal period could also be implemented by using the timer. Every RTO is check after each packet acknowledgement received.

The congestion control mechanisms are having supervision of data control and data traffic within a network. It avoids congestion during the data transmission and prevents network unfair allocation of any of either processing or network bandwidth capabilities by dropping the data transmission rate of the send packet.

The main goal of the congestion control algorithms are

1. To achieve fairness and better compatibility with widely used protocols.

2. To achieve maximum network bandwidth utilization.

3. To achieve efficient and effective data communication.

4. To sustain high responsiveness.

5. To reduce the amplitude of network congestion overhead [9]

There is little research on some of the new algorithm for high-bandwidth long-delay networks. This paper includes studies the default TCP BIC congestion control algorithm in Linux kernel 2.6.8.[10]. The the internet has recently been used by different type of devices. So that it changes from homogeneous congestion control to heterogeneous congestion control scenario. A few years ago, only the TCP congestion control algorithm mainly used for controlled the Internet traffic. The TCP congestion control algorithm used the AIMD standard Additive-Increase-Multiplicative-Decrease algorithm. However, different TCP algorithms are used to control internet traffics[11]. First of all, we start the analysis of existing congestion control algorithm TCP NewReno, BIC, and Hybrid Illinois according to the following simulation scenario setup into the NS3 network simulator.

III. RELATED WORK

This is very challenging task to design a new protocol that can satisfy all the criteria like in congestion control these criteria are RTT fairness TCP friendliness and scalability [1, 9, 10]. A protocol should adopt its window control depending on the size of the window. TCP BIC the congestion control problem is viewed as a search problem [10]. TCP BIC (binary increase congestion control) is consists of two parts binary search increase and additive increase.

A. Binary search increase

Starting points for this search our current maximum window size and maximum window size usually Wmin and W max are the windows I just before the last recovery. The algorithm rapidly computes the midpoint between Wmax and Wmin set the current window size to the midpoint and check for feedback in the form of packet loss.

If there is a packet loss and then present threshold minimum increments. This technique is called binary search increase allow bandwidth propping to be more aggressive initially when the difference from the current window size to the target window size is large and become less aggressive as the current window size gets closer to the target window size [12].

B. Additive Increase

To ensure convergence at RTT fairness the binary search in trees having an additive increase strategy also. If, windows distance from the midpoint to current is large than increase window size directly to midpoint. If the distance from the current window size to the target is less than the Smax, at that time window increase directly to the target [12].

C. Slow Start

The TCP congestion control algorithm runs a "slow start" strategy to probe for a new maximum, once the current window size is greater than Wmax but smaller than Wmax+Smax. The congestion window increases in each RTT round in steps.

Under the fast convergence, the binary search increase combines in multiplicative decrease converges to fair share search. Since the large window reduces more and multiplicative decrease. The fast convergence strategy the binary search increase after a window reduction new maximum and minimum are set. than Readjust the new condition window maximum size to be the same as the new target window which is the midpoint and then we adjust the target and apply binary search this has an effect to reduce the increasing rate of the larger window allows the smaller window to catch up this strategy called first convergence [12].

The detailed description of TCP BIC algorithm could be found in [13]. A diagram of this algorithm during the Congestion Avoidance phase is shown in figure 2

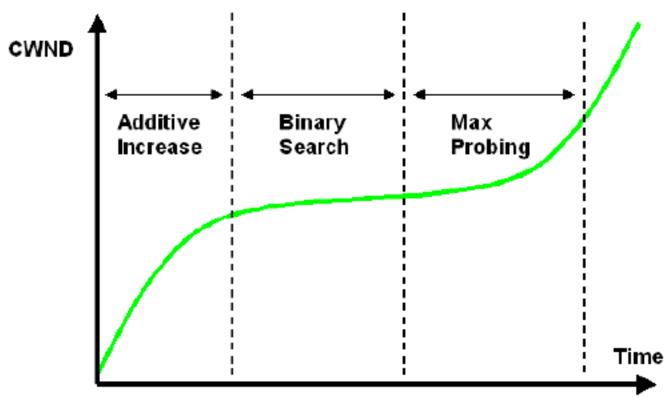

Figure 2. TCP BIC congestion avoidance representation

IV. SIMULATION SETUP IN NS3

In this section, I have evaluated and analyzed the different type of congestion control algorithms as TCP Westwood, BIC, NewReno, scalable, Illinois, and QIC. I am using a simple network scenario and applied through the network simulator NS3. To test the behavior of each Protocol in the bottleneck bandwidth network, I used simple simulation Technology consisting of a single sender and single receiver interconnected to each other through a router at the other end this is shown in Figure 3. Here I create a point to point connection between the sources and sink and simulate according to different bandwidth channel between server to the router and varying the time of simulation. Each TCP congestion control algorithm is analysis on the basis of the congestion window, Round trip time and finally gets the throughput. Table 1 shows the network simulator parameters that used in simulation through NS3. [14].

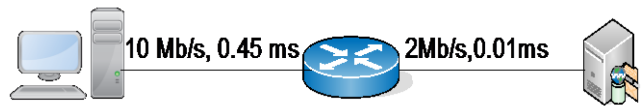

Figure 3. TCP congestion Simulation network Scenario

Table 1 Simulation variables and its values

Parameters	Values
Bottleneck Bandwidth	2 Mb/s
Time of Simulation	50 seconds
Queue Type	PfifoFastQueueDisc
Nodes	2
Delay	0.01 ms
TCP Congestion Control Algorithms	TCP Westwood, BIC, NewReno, scalable, Illinois, and QIC
Access Bandwidth	10 Mb/s
Access Delay	0.45 ms

V. TCP WINDOWS SIZE QUADRIC INCREASE RESULT ANALYSIS WITH OTHERS

On the basis of the simulation result, it is analyzed that the throughput of the TCP QIC having some significance value with comparing to other congestion control algorithms. This is represented by the figure 4. In this graph the throughput representation in increasing order.

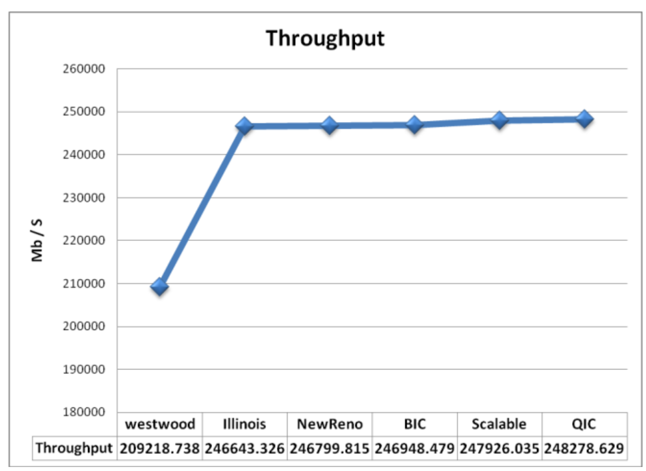

Figure 4. Throughput comparison between congestion control algorithms in increasing order

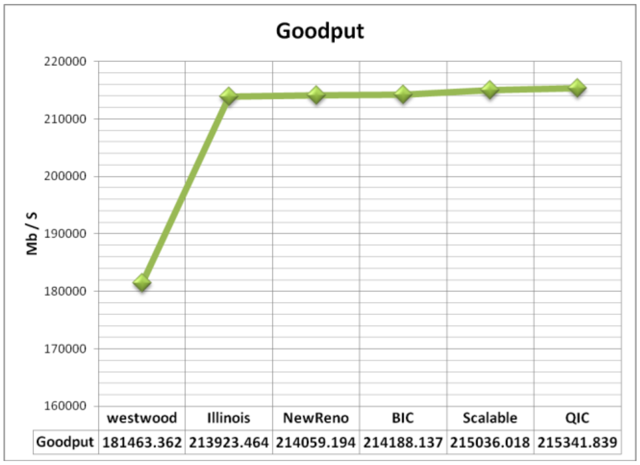

Figure 5. Goodput graph for bottleneck bandwidth 2Mb/s in increasing order

In figure 5 represent the goodput graph. The throughput and goodput result of the simulation 50 seconds for 2 Mb/s bottleneck bandwidth. I have simulated with different cases of bandwidth (2,5,10) and simulation time(25,50,80) but the observation is that the when bandwidth increasing the congestion windows is increasing significantly and congestion does not occur for this scenario that's why here only one simulation results describe.

In the figure 6 shows the end to end delays in transmission. In this graph, the average delay calculated that show the lowest average delay TCP Westwood, NewReno, and Scalable and highest average delay TCP Illinois and BIC. TCP QIC having an average delay between lowest and highest. In this regard, the TCP QIC performance is better than the TCP BIC and Illinois.

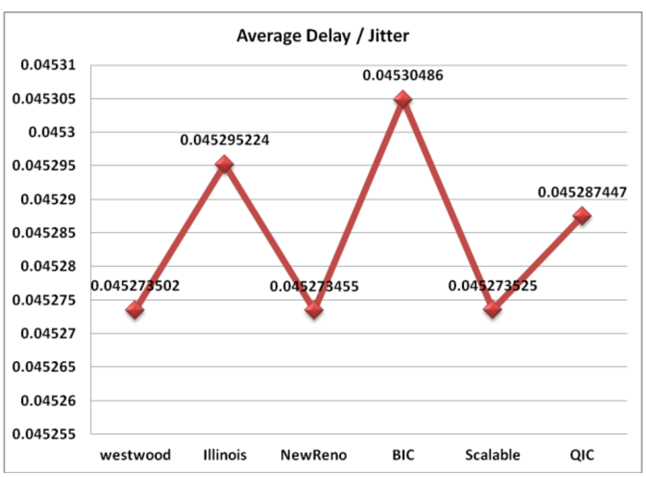

Figure 6. Average end to end delay representation

The following figure 7 shows the number of packets sends by each TCP congestion control algorithm in increasing order these are represented by the graph. It is very simple logic when the number of packets sent by any algorithm then data more transmitted and it shows the maximum utilization of the bandwidth and produced maximum throughput. It is proved by the above and below graphs that the TCP QIC is having an advantage in terms of throughput and data transmission.

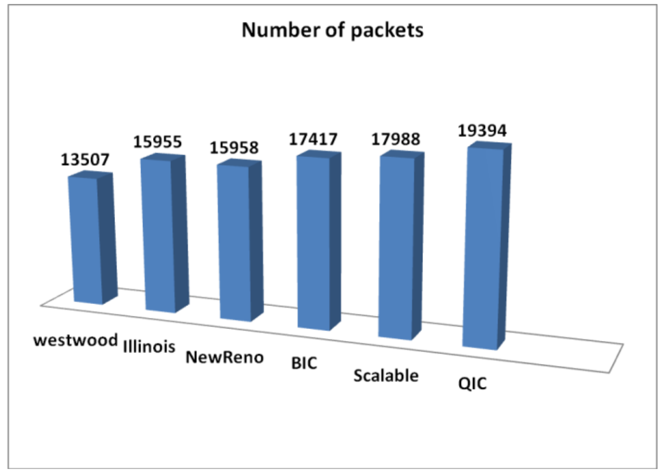

Figure 7. Packet received in 50 seconds simulation

Figure 7 represents the data packet transmission speed that is represented by the lambda. Lambda is the data packet send per second. If data packet sends per second is higher that means the data packed acknowledgment received speed is higher, that show the congestion prevention and congestion control is efficiently handled by the TCP QIC rather than TCP BIC, Westwood, NewReno, Illinois and Scalable.

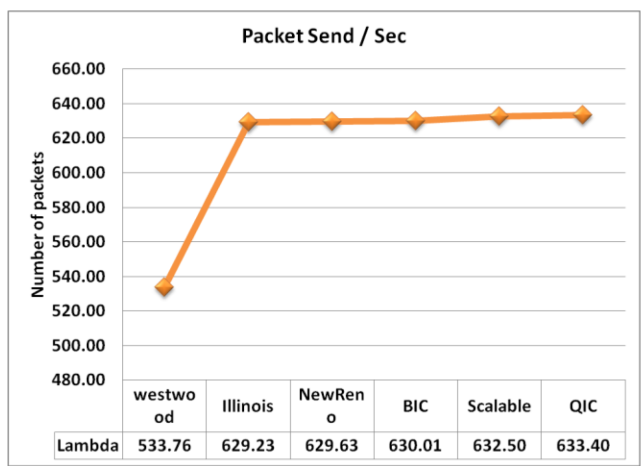

Figure 8. packet send speed

If data packet send speed higher but receiving speed not similar to the sending speed in this situation congestion occur and data packets are dropped and retransmission of same data packets. This is affecting the throughput and average end to end delay.

VI. CONCLUSION

This paper proposes a TCP congestion control algorithm based on the TCP BIC that is a quadric congestion control (QIC). TCP QIC is a simulation of the different parameters values using the NS3 simulator. The simulation metrics are generated after 50-second simulation each of the congestion control algorithms simulated same time. The different kind of graph plotted on the basis of simulation Matrix result that represents the TCP QIC having better throughput and Goodput. TCP Q IC having efficient data packet transmission speed and it also receives a maximum number of packets within a 50 second of simulation time. TCP QIC average delay time is higher from lowest delay time but it is lower than the highest delay time. Finally, I say that the TCP QIC having better performance than the other TCP congestion control algorithms as Westwood, Illinois, NewReno, BIC and scalable.

REFERENCES

[1] K. Nagori, *et al.*, "Common TCP Evaluation Suite for ns-3: Design, Implementation and Open Issues," in *Proceedings of the Workshop on ns-3*, 2017, pp. 9-16.

[2] H. J. A. Nasir, *et al.*, "Load balancing using enhanced ant algorithm in grid computing," in *Computational Intelligence, Modelling and Simulation (CIMSiM), 2010 Second International Conference on*, 2010, pp. 160-165.

[3] H. J. A. Nasir, *et al.*, "Enhanced Ant-Based Routing for Improving Performance of Wireless Sensor Network," *International Journal of Communication Networks and Information Security (IJCNIS)*, vol. 9, 2017.

[4] A. M. Kishk, *et al.*, "Proposed Jamming Removal Technique for Wireless Sensor Network," 2015.

[5] P. Yang, *et al.*, "TCP congestion avoidance algorithm identification," *IEEE/Acm Transactions On Networking*, vol. 22, pp. 1311-1324, 2014.

[6] B. Levasseur, *et al.*, "A TCP CUBIC implementation in ns-3," in *Proceedings of the 2014 Workshop on ns-3*, 2014, p. 3.

[7] G. Paliwal and S. Taterh, "A Topology Based Routing Protocols Comparative Analysis for MANETs."

[8] M. Allman, *et al.*, "RFC 2581: Tcp congestion control, April 1999," *Obsoletes RFC2001 [48]*, 2001.

[9] M. Allman, "TCP congestion control with appropriate byte counting (ABC)," 2003.

[10] W. Hua and G. Jian, "Analysis of TCP BIC Congestion Control Implementation," in *Computer Science & Service System (CSSS), 2012 International Conference on*, 2012, pp. 781-784.

[11] G. Paliwal and S. Taterh, "Impact of Dense Network in MANET Routing Protocols AODV and DSDV Comparative Analysis Through NS3," in *Soft Computing: Theories and Applications*, ed: Springer, 2018, pp. 327-335.

[12] L. Xu, *et al.*, "Binary increase congestion control (BIC) for fast long-distance networks," in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, 2004, pp. 2514-2524.

[13] S. Feyzabadi and J. Schönwälder, "Identifying TCP Congestion Control Mechanisms Using Active Probing," *Computer Science Department, Jacobs University Bremen, Germany*.

[14] L. A. Grieco and S. Mascolo, "Performance evaluation and comparison of Westwood+, New Reno, and Vegas TCP congestion control," *ACM SIGCOMM Computer Communication Review*, vol. 34, pp. 25-38, 2004.

Authors Profile

Mr. Girish Paliwal is an Assistant Professor in Amity Institute of Information Technology at Amity University Rajasthan. He has more than 17 years of academic & teaching and research experience. During his academic career, he has held various positions such as member of Board of Studies, Head of examination etc. He has appointed as paper setter and an external examiner for the viva-voce exams of B.Tech / M.Tech, MCA students of various universities. He has provided the guidance to the projects and Internship. He has published various research papers in national (2) and international journals (10) and also actively participated in various conferences (10), workshop and Seminars (25). His research areas related to data communication networking, Mobile ad hoc networking, Wireless sensor networking, congestion control algorithms and vehicular ad hoc networking. He is editorial member and reviewer of a national and international journal. He has qualification SET, PhD*, M.Tech(IT), MCA.

Dr. Swapnesh Taterh is an Associate Professor in Amity Institute of Information Technology at Amity University Rajasthan. He has more than 15 years of academic & teaching experience. During his academic career, he has held various positions such as member of Board of Studies, Centre superintendent for conducting examination etc and currently as a Programme Coordinator in the department of AIIT. He has provided the guidance to the research scholar and taking the classes in Ph.D. course work. He has published various research papers in national and international journals and also actively participated in various conferences, workshop and invited for Invited Talk in Conference's. He guided research scholar in the areas related to software security, cryptography and Networking and submitted project's in Department of Science & Technology, Rajasthan. He is editorial member and reviewer of a national and international journal

