International Journal of Integrated Engineering, Vol. 9 No. 3 (2017) p. 1-6

Enhanced FAST TCP by Solving Rerouting Problem

Shahram Jamali [1,*], Neda AliPasandi[2]

[1] Computer Engineering Department, University of Mohaghegh Ardabili, Ardabil, Iran
[2]Sama technical and vocational training college, Islamic Azad University, Ardabil Branch, Ardabil, Iran

Received 27 February 2017; accepted 19 June 2017, available online 20 June 2017

Abstract: Delay-based congestion control algorithms inability to recognize increased RTT related to rerouting from increased RTT related to congestion is their most serious problem which has serious effect on their throughput. FAST TCP is one of delay-based TCP variants that although outperforms other TCP variants in high bandwidth-delay product networks, but suffers from several problems that inhere in its procedure to estimate trip delay. The most serious of these problems is rerouting. When rerouting occurs and round-trip time (RTT) of the new path is longer than RTT of the old path, the throughput of FAST TCP decreases sharply. Because FAST misinterprets the increased RTT as result of the network congestion and consequently decreases its own window size. This paper solves this problem by considering the relationship between sending rate and observed RTT. The simulation results show the effectiveness of proposed solution to solve rerouting problem while simultaneously preserves FAST TCP prominent primitive features.

Keywords: FAST TCP, Rerouting, Congestion Control, RTT

1. Introduction

Congestion control of internet plays an important role in availability and stability of internet network. Nowadays, Transmission Control Protocol (TCP) is the dominant protocol among the Internet protocols. After the congestion collapses [1] in 1986 TCP was first proposed and implemented to prevent the long run congestion collapses. Afterwards several versions of TCP are emerged to enhance the primitive TCP performance [2]. Many works like [3-7] evaluate and analyze performance of TCP versions in various environment. TCP versions categorized as two classes: loss-based and delay-based. For example, TCP Reno [8], which is the most used version, uses packet loss as the solely sign of congestion, and react to congestion. As another example FAST TCP [9, 10], which is one of the most prominent TCP version in high bandwidth-delay product networks, attempts to prevent congestion by regulation of the transmission rate based on RTT difference.

Many researches [11, 12] investigate FAST TCP performance and demonstrate that FAST TCP is promising in terms of stability, throughput and fairness, but it suffers from several problems that are due to its procedure to estimate round trip propagation delay [13, 14] which is represented by BaseRTT. The most serious of these problems is rerouting which causes throughput degradation of it. If FAST TCP estimation problem is solved, it can be the most used version in high bandwidth-delay product networks

To solve rerouting problem of FAST TCP, this paper proposes a solution that considers the relationship between sending rate and observed RTT. The simulation results that is done by ns2 simulator [15] show that the proposed solution, which is named RRFAST (Rerouting Recognizer FAST), is effective to overcome rerouting problem while simultaneously preserves FAST prominent primitive features.

The rest of this paper is organized as follows. Section 2 gives FAST TCP algorithm and its rerouting problem. Section 3 describes related works. Section 4 presents our algorithm to recognize rerouting. Section 5 presents the simulation results, and finally section 6 summarizes this work.

2. FAST TCP and rerouting problem

FAST TCP was designed for high bandwidth-delay product networks which originally was developed at the Netlab, California Institute of Technology. Actually the main idea of FAST TCP congestion algorithm was derived from TCP Vegas congestion algorithm.

A FAST TCP flow attempts to keep a constant number of packets in the queues. The number of packets in queues is estimated by measuring the difference between the encountered round trip time (RTT) and the round trip propagation delay (BaseRTT). Actually FAST end host estimates BaseRTT of the path because has no way of knowing it. FAST TCP estimation procedure simply sets the minimum of encountered RTT as the round trip propagation delay. Then FAST TCP periodically adjusts the congestion window size (w(t+1)) based on the average RTT and average queuing delay as follows [9,10]:

$$w(t+1) = min\,(2w(t), (1-\gamma)w(t) + \gamma\left(\frac{Base_{RTT}}{avg_{RTT}}w(t) + \alpha(w(t), q(t))\right)) \quad (1)$$

Where w(t+1) is the target congestion window size after adjusting, w(t) is the current congestion window size, $\gamma \in (0,1]$, BaseRTT is the minimum RTT encountered so far, avgRTT is the average RTT since first packet of the

**Corresponding author: alipasandi@yahoo.com*

penerbit.uthm.edu.my/ojs/index.php/ijie

session, and α is a constant, and q(t) is the average end-to-end queuing delay.

So in this mechanism, if the estimated number of packets in queues is low, the sending rate is increased, while if the estimated number of packets in queues is high, the sending rate is decreased.

Estimated number of packets in queues is based on encountered RTT, so whatever encountered RTT is longer than the BaseRTT FAST TCP end host guesstimates that the congestion level is high and so decreases sending rate. Similarly if encountered RTT is close to the BaeRTT meaning that the traffic of the network is light. So in this way FAST TCP detects network congestion in the early level of congestion and successfully prevents periodic packet loss that usually happens in loss based congestion control algorithms like TCP Reno.

Many researches investigate FAST TCP performance and demonstrate that FAST TCP is promising in terms of stability, throughput and fairness, But it suffers from several problems that are due to its procedure to estimate BaseRTT. Estimation procedure of BaseRTT is very simple and fails to recognize the true round trip propagation delay when rerouting occurs. Although if the new route has a shorter propagation delay, this does not make any significant problem for FAST TCP because presumably some packets will encounter shorter round trip delay and BaseRTT will be updated. But, if the new route has a longer propagation delay, the FAST TCP end host cannot distinguish whether the growth in the round trip delay is as a result of network congestion or a rerouting has been occurred. Without this knowledge FAST TCP misinterprets the increased RTT as a result of network congestion and consequently decreases its own window size, which makes the throughput of FAST TCP decrease remarkably.

3. Related work

Several works are proposed to solve rerouting problem of delay-based congestion control algorithms but they have some downsides and limitations. The proposed algorithm in [16] detects rerouting by monitoring of TTL value of two end-hosts. But this strategy does not properly work when the RTT of the new route is longer than that of the old route and concurrently the new route and the old route have equal hop count. Vegas-W which was proposed in [17], attempts to solve rerouting problem in GEO satellite network by assumption that any sharp change in RTT values is as a result of rerouting. For do this, they introduced two new parameters, but finding proper values for those has remained an open problem.

In [18] the authors used a machine-learning technique known as the Experts Framework to estimate the round trip propagation delay. Although they have shown that their proposed algorithm has higher accuracy in estimating the RTT than the standard algorithm used in most TCP implementations, but they do not experiment their proposed algorithm effect on rerouting.

Authors of [9] introduced a new discrete-time link model that fully captures the effect of self-clocking and compared it with the traditional continuous-time model. They also studied stability issue of FAST TCP. A same study was conducted in [19] based on a continuous-time dynamic model of FAST TCP.

The proposed algorithm in [11] use priority to some packets to estimate accurately the round trip propagation delay. This solution maybe improve fairness and reduce queuing variations but do not solve rerouting problem.

Also authors of [20] studied FAST TCP Hopf bifurcation control problem with RED gateway. It analyses stability condition of the congestion control model by using linear stability analysis. These analysis plays an important role in setting guiding system parameters for controlling the FAST TCP and RED model.

Also authors of [12] designed a New TCP Based algorithm on FAST TCP for Datacenter. Their proposed method updates congestion window based on the queuing delay. It maintains a certain queue length at the switch buffer for each flow and keeps the total queue length below the buffer size. Although this method avoids the droptails over switch buffer, but does not consider rerouting problem.

Author of [21] by proposing of two refinement on RTT estimation process try to improve behavior of TCP in wireless local area networks. They implemented the proposed solution in MAC and TCP layers.

4. Proposed solutions

To overcome the rerouting problem mentioned in section 2, this paper proposes an algorithm which is named RRFAST that cautiously monitors the encountered RTT after decreasing the window size. We know when level of congestion is high, encountered RTT is much longer than the BaseRTT. Hence FAST TCP end host decides on decreasing the window size in the hope that congestion is resolved and encountered RTT becomes close to the BaseRTT. This works well if the increased RTT is due to the network congestion. But if the increased RTT is due to a new longer route, this does not work, and whatever FAST end host decreases its sending rate, the encountered RTT never becomes close to the BaseRTT. To check whether this increased RTT is due to congestion or due to rerouting, RRFAST first decreases its sending rate and then checks the observed RTT, when the reduced sending rate leads to decreased RTT it means that the network faces with congestion otherwise it faces with rerouting. When RRFAST detects rerouting, it resets the BaseRTT. The reset method is the same as used in [22]. The reset method sets BaseRTT to very high value and since then updates it with the minimum of encountered RTT. Fig. 1 shows the flowchart of proposed algorithm.

Published by UTHM Publisher
http://www.uthm.publisher.edu.my/ojs/ijie

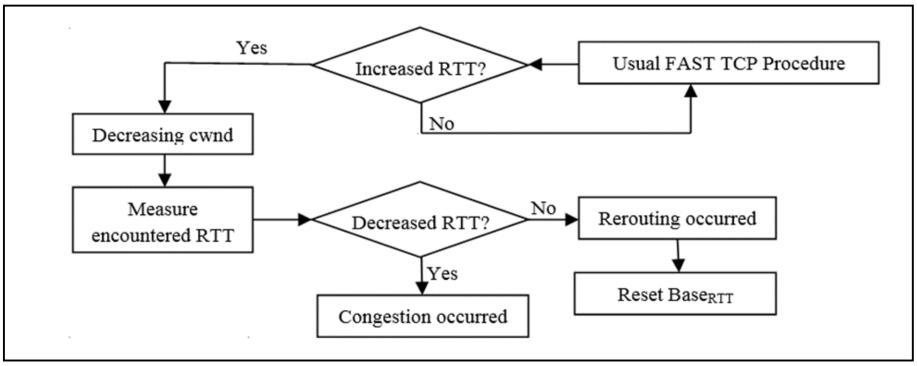

Fig. 1 flowchart of proposed algorithm

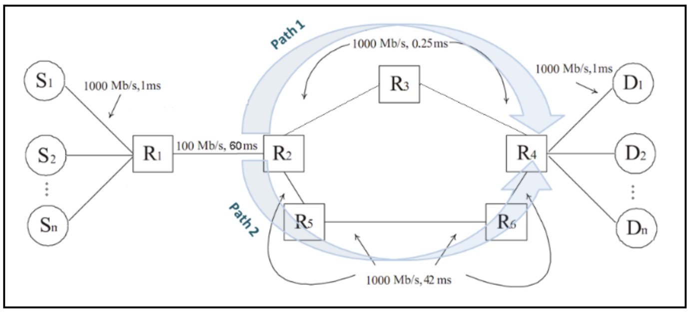

Fig. 2 Network simulation topology

The concept of RRFAST is that decreasing the window size is in the hope that the encountered RTT becomes close to the BaseRTT. So if this expectation is not fulfilled, RRFAST gets reduction in the window size had been futile. Hence it distinguishes increased RTT is as a result of a new longer route.

5. Simulation results

The ns2 simulator is used to evaluate and compare the RRFAST performance with FAST TCP. To evaluate the RRFAST, we implement it by some modifications on FAST TCP module which was developed at NetLAB of Caltech university. Then it is run over the network shown in Fig. 2 and is compared with FAST TCP algorithm in presence of the rerouting. In Fig. 2, Si, Di and Ri represent sender hosts, destinations and routers respectively. Sender Si communicates with destination Di. The bandwidth and propagation delay of each link are labeled on it. The size of data packet is 1 Kbytes, and the size of ACK is 40 bytes. The FIFO service discipline is employed for queue, and queue buffer size is 50 packets, for both original FAAST TCP and RRFAST TCP. We conducted two scenario to compare the proposed algorithm and original algorithm in different situations. It is notable to mention that ten flows are active throughout the simulations, and the results is for one of them as a sample flow.

5.1 Scenario 1 to compare behavior of algorithms when new route is longer

In the first simulated scenario, rerouting occurs at t=30th second. During first 30 second the packets are routed through P1, and then after 30th second the packets are rerouted to the P2 path which propagation delay of it is more than P1 path. Round trip propagation delay of P1 and P2 path are 125ms, 376ms respectively.

As be seen in Fig. 3, which shows the encountered RTT of the sample flow, at 30th second when the packets are rerouted to the new longer path, and the encountered RTT suddenly increases, FAST TCP flow sharply decreases its cwnd because assume that any increases in the encountered RTT is due to congestion. Fig. 4 compares the cwnd of the FAST TCP and RRFAST TCP of the sample flow. As be seen in Fig. 4, RRFAST flow at the beginning decreases the cwnd, but when recognizes rerouting occurrence, resets the BaseRTT a little later and increases cwnd. Fig. 5 shows the BaseRTT of the sample FAST TCP and RRFAST TCP flow.

Published by UTHM Publisher
http://www.uthm.publisher.edu.my/ojs/ijie

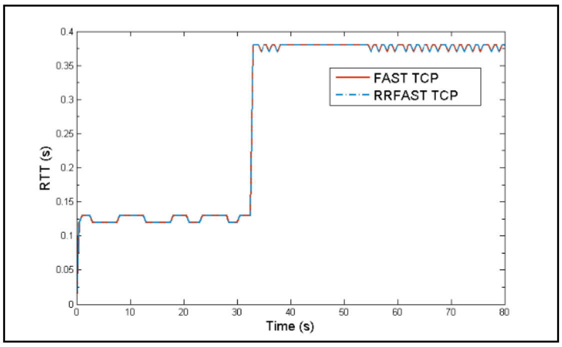

Fig. 3 Comparison of RTT in scenario.

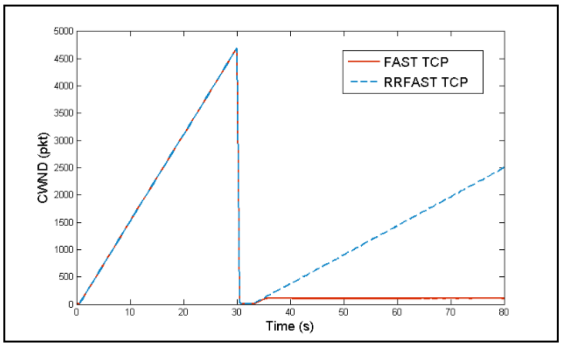

Fig. 3 Comparison of cwnd in scenario 1

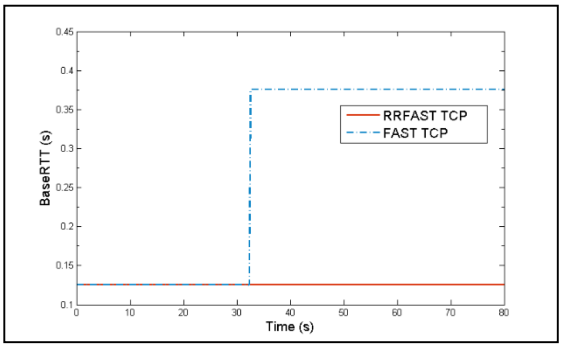

Fig. 4 Comparison of BaseRTT in scenario 1

The effect of true BaseRTT on flow throughput and bottleneck utilization is shown in Fig. 6 and Fig.7. As be seen in these figures the proposed algorithm improves throughput and bottleneck utilization.

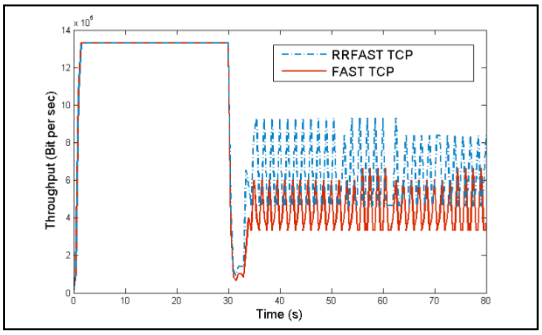

Fig. 5 Comparison of throughput in scenario 1

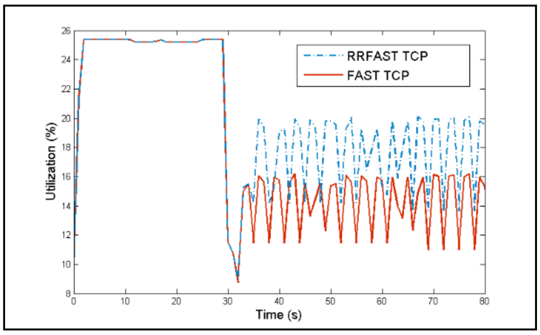

Fig. 6 Comparison of bottleneck utilization in scenario 1

5.2 Scenario 2 to compare behavior of algorithms when new route is shorter

This scenario is designed to demonstrate that RRFAST and FAST have same behavior in other situation. We conducted many experiments, but due to space limitation only one of them is presented. In this scenario like scenario 1 rerouting occurs at t=30th second, but the new route has less propagation delay than old route. During first 30 second the packets are routed through P2, and then after 30th second the packets are rerouted to the P1 path.

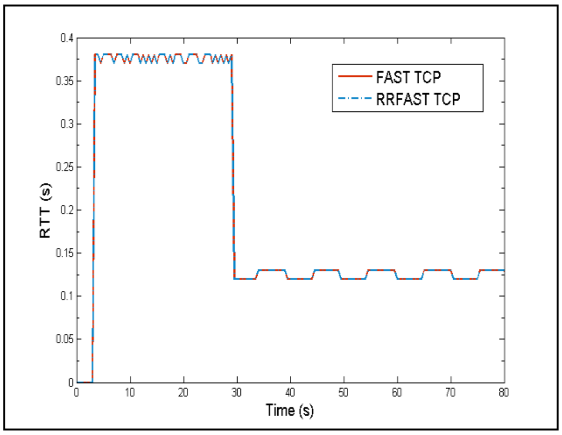

Fig. 8 Comparison of RTT in scenario 2

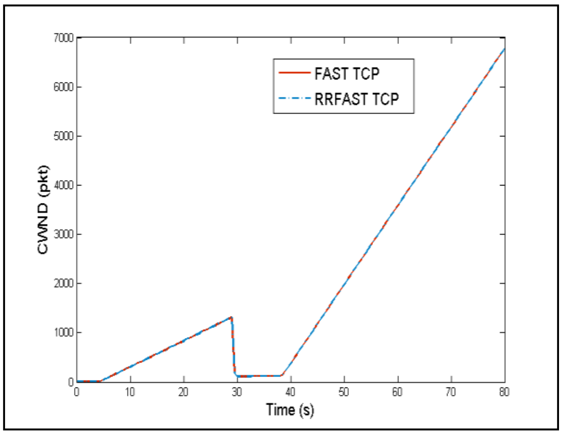

Fig. 9 Comparison of cwnd in scenario 2

Published by UTHM Publisher
http://www.uthm.publisher.edu.my/ojs/ijie

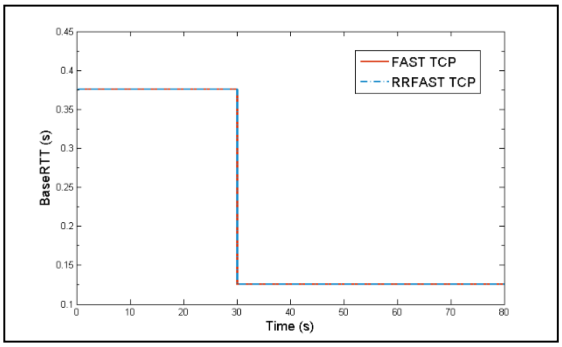

Fig. 10 Comparison of BaseRTT in scenario 2

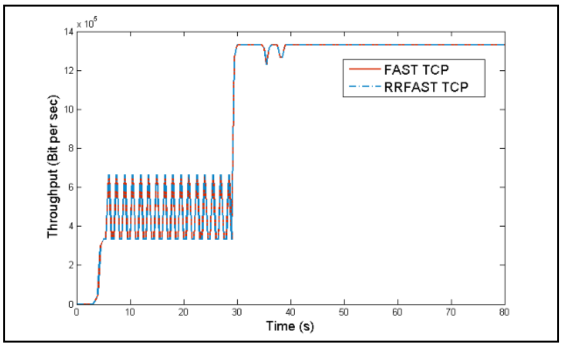

Fig. 11 Comparison of throughput in scenario 2

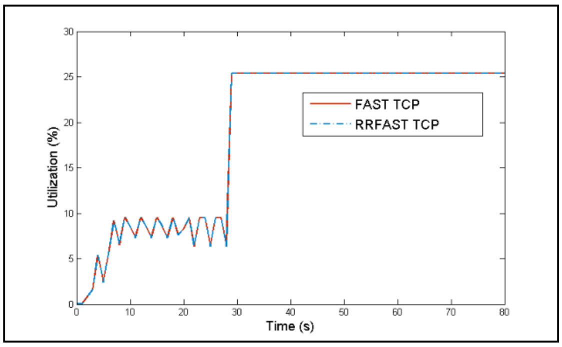

Fig. 12 Comparison of bottleneck utilization in scenario 2

As has been noted in other researches, if the new route has a shorter propagation delay, this does not make any significant problem for FAST TCP because presumably some packets will encounter shorter round trip delay and BaseRTT will be updated. The simulation results is in compliance with this issue. The Fig. 8-12 show that encountered RTT, cwnd, BaseRTT, throughput and bottleneck utilization of the proposed and original algorithm in normal situations are same. So the proposed algorithm preserves FAST TCP prominent primitive features and moreover solves rerouting problem.

But although our algorithm solves rerouting problem, the development of an efficient algorithm for more accurate estimates of BaseRTT seems be essential, and this will be our future work.

6. Summary

This paper proposes a new solution, named RRFAST, to overcome rerouting problem of FAST TCP. The main concept of RRFAST is that decreasing the window size is in the hope that the encountered RTT becomes close to the BaseRTT. So if this aim is not fulfilled, RRFAST gets reduction in the window size had been futile. Hence it distinguishes increased RTT is as a result of a new longer route. Simulation results show that RRFAST solves FAST throughput degradation problem in the new longer path and achieves high throughput in comparing with FAST. However, the existence of an efficient algorithm for more accurate estimates of BaseRTT seems be essential, and our future work is to develop an algorithm that solves all problems due to inaccurate estimation of BaseRTT.

References

[1] Agarwal, A.K. Biofuels (alcohols and biodiesel) applications as fuels for internal combustion engines. *Progress in Energy Combustion science*, Volume 33, (2007), pp. 233-271.

[2] S. Lar, X. Liao, An initiative for a classified bibliography on TCP/IP congestion control, Journal of Network and Computer Applications, vol. 36, no. 1, pp.126-133, 2013.

[3] Chaudhary, Pooja, and Sachin Kumar. "A Review of Comparative Analysis of TCP Variants for Congestion Control in Network." International Journal of Computer Applications 160.8 (2017).

[4] Shinko, Ilir, et al. "Performance analysis of different architectures and TCP congestion-avoidance algorithms using WMN-GA simulation system." Journal of High Speed Networks 23.2 (2017): 163-173.

[5] Lara-Cueva, Román, et al. "Performance evaluation of the new algorithm of the TCP protocol for a long distance wireless link in the Galapagos Islands." Engineering Summit, II Cumbre Internacional de las Ingenierias (IE-Summit), 2016 IEEE International. IEEE, 2016.

[6] Nguyen, Truc Anh N., Siddharth Gangadhar, and James PG Sterbenz. "Performance Evaluation of TCP Congestion Control Algorithms in Data Center Networks." CFI. 2016

[7] T. Lukaseder, et al. "A Comparison of TCP Congestion Control Algorithms in 10G Networks." Local Computer Networks (LCN), 2016 IEEE 41st Conference on. IEEE, 2016.

[8] MA. Alrshah, M. Othman, B. Ali, Z. Mohd Hanapi, Comparative study of high-speed Linux TCP variants over high-BDP networks, Journal of Network and Computer Applications, vol. 43. pp. 66-75, 2014.

[9] Wei, David X., et al. "FAST TCP: motivation, architecture, algorithms, performance." IEEE/ACM Transactions on Networking (ToN) 14.6 (2006): 1246-1259.

[10] Wang, Jiantao, et al. "Modelling and stability of FAST TCP." Wireless communications. Springer New York, 2007. 331-356.

[11] Xue, Lin, et al. "A study of fairness among heterogeneous TCP variants over 10Gbps high-speed

Published by UTHM Publisher
http://www.uthm.publisher.edu.my/ojs/ijie

optical networks." Optical Switching and Networking 13 (2014): 124-134.

[12] Ding, Dawei, et al. "Hopf Bifurcation Control in a FAST TCP and RED Model via Multiple Control Schemes." Journal of Control Science and Engineering2016 (2016).

[13] Zheng, Feng, Yongfeng Huang, and Donghong Sun. "Designing a new TCP based on FAST TCP for datacenter." 2014 IEEE International Conference on Communications (ICC). IEEE, 2014.

[14] Liang, Wei, and Sulei Xu. "Smoothly estimate the RTT of fast TCP by ARMA function model." Wireless Communications, Networking and Mobile Computing (WiCOM 2014), 10th International Conference on. IET, 2014.

[15] DARPA/VINT, LBNL, XEROX, UCB AND USC/ISI (2001), Network simulator ns- 2, Information available at http://www.isi.edu/nsnam/ns/. Code available at http://www.isi.edu/nsnam/dist/

[16] CY. Ho, YC. Chen, CY. Ho, Improving Performance of Delay-Based TCPs with Rerouting, Communications Letters IEEE, vol. 11, no. 1, pp. 88-90, 2007.

[17] SP. Deshmukh, SS. Pawale, TCP Vegas rerouting detection and improving Performance, International Journal of Wired and Wireless Communications, vol. 1, no. 1, pp. 11-14, 2012.

[18] J. Qu, An Enhanced TCP Vegas Algorithm Based on Route Surveillance and Bandwidth Estimation over GEO Satellite Networks, Proceedings of International Conference on Measuring Technology and Mechatronics Automation, vol. 1, pp. 464-467. IEEE, 2010.

[19] Nunes, Bruno AA, et al. "A machine learning approach to end-to-end rtt estimation and its application to tcp." Computer Communications and Networks (ICCCN), 2011 Proceedings of 20th International Conference on. IEEE, 2011.

[20] Koo, Kyungmo, Joon-Young Choi, and Jin S. Lee. "Parameter conditions for global stability of FAST TCP." IEEE Communications Letters 12.2 (2008): 155-157.

[21] P. Dalal, et al. "Refining TCP's RTT dependent mechanism by utilizing link retransmission delay measurement in Wireless LAN." International Journal of Communication Systems 30.5, 2017.

[22] N. Alipasandi, S. Jamali, An improvement over TCP Vegas by solving rerouting problem, AWERProcedia Information Technology and Computer Science, vol. 1, 2013.

Published by UTHM Publisher
http://www.uthm.publisher.edu.my/ojs/ijie