International Journal of Engineering Research and Development (IJERD) ISSN: 2278-067X
Recent trends in Electrical and Electronics & Communication Engineering (Page 37-44)
(RTEECE 08th – 09th April 2016)

Impact of False RTT on the Efficiency of TCP-NJ in WLAN

Utsav K Gandhi[1], Purvang D Dalal[2]
[1]*Department Of Electronics And Communication, Dharmsinh Desai University, India.*
[2]*Department Of Electronics And Communication, Dharmsinh Desai University, India.*

Abstract:- *Over few decades, Transmission Control Protocol (TCP) has been employed for the major popular wireless Internet applications. However, TCP has faced major performance issues when it operates in the wireless networks. The classical explanation is that the radio losses are seen as congestion by TCP, and therefore TCP attempts recovery with detrimental transmission rate in absence of any support for loss discrimination. Considering economical gain behind use of wireless Internet services, it become essential for TCP to provide rated application performance and hence lots of efforts are put on in last decade to improve wireless TCP performance. TCP New Jersey uses Timestamp-based Available Rate Estimation (TARE) algorithm. Additionally it utilises the Explicit Congestion Notification (ECN) from an intermediate router to rule out any kind of over rate estimation. In this paper, the performance of TCP New Jersey is investigated over a WLAN. In order to vary the congestion level in the network the simulations are performed with variations in the self-similar TCP traffic. The transmission errors on wireless links are introduced using a well-known two state markov error model. The objective is to investigate performance issues of TCP New Jersey. The simulation results showed that the RTT variations causes inaccuracy in ARE, which is RTT dependent. The results presented are generated using extensive simulations under ns-2.*

Keywords:- *TCP, New Jersey, TARE, ECN, markov error model, WLAN, RTT*

I. INTRODUCTION

TCP is being used as a highly reliable end-to-end protocol for transporting applications. TCP was originally designed for wired networks where the random Bit Error Rate (BER) is negligible and actually assumed that packet losses are due to congestion in the network. The performance degradation of TCP in wireless and wired-wireless hybrid networks is mainly due to its lack of the ability to differentiate the packet losses caused by network congestion from the losses caused by wireless link errors. TCP uses congestion control and avoidance algorithms which degrades end-to-end performance in wireless systems. The TCP sender behavior of adjusting the sending rate of data packets is triggered by the self-clocking acknowledgement (ACK) sent by the corresponding receiver after successfully receiving the data packet. When packet loss occurs at a congested link due to buffer overflow at the intermediate router, either the sender receives duplicate ACKs (DUPACK) or the sender's Retransmission Time Out (RTO) timer expires. These events activate the sender's congestion control mechanism by which the sender reduces the size of its transmission window, or congestion window (cwnd) in TCP terminology, resulting in a lower transmission rate to relieve the link congestion. This is a reactive congestion control scheme, in which the action is triggered by the sender's self-induced congestion. Such TCP sender behavior works fairly well in the wired networks. However, in wired/wireless heterogeneous networks, high BER, fading and blackout become non-negligible factors for packet losses. Standard TCP's congestion control and congestion avoidance mechanisms based on the assumption that all packet losses are due to congestion become incapable of handling the mixed packet losses. TCP without modification exhibits throughput degradations when used in wired/wireless heterogeneous networks.

II. TCP NEW JERSEY & LOSS RECOVERY MECHANISM

TCP New Jersey differentiates wireless losses and congestion losses and in addition to this it also responds fairly to the network congestion losses due to integration of ECN. It consists of two key components, the Timestamp-based Available Rate Estimation (TARE) algorithm and the Congestion Warning (CW) router configuration. CW marks all the packets when the average queue length exceeds a threshold. This marking scheme leaves the TCP sender, which receives the marks, to decide its window adjustment strategy. The marking of packets by the CW configured routers helps the sender of the TCP connection to effectively differentiate packet losses caused by network congestion from those caused by wireless link errors. The purpose of CW is to convey a simple image of the bottlenecked queue to the sender. TARE is a TCP-sender-side algorithm that continuously estimates the bandwidth available to the connection and guides the sender to adjust its transmission rate when the network becomes congested [1].

TCP New Jersey adopts slow start and congestion avoidance from New Reno, but implements the rate-based cwnd control procedure based on TARE. It operates as follows. If an ACK is received without the CW mark, it proceeds as New Reno. If the received ACK or the third DUPACK is marked with the CW bit, it calls the rate control procedure to adjust the window size. When the third DUPACK is received without the CW mark, TCP New Jersey concludes that the packet drop is caused by a random error, and therefore it enters the fast retransmit without adjusting the window size. TCP New Jersey combines TARE and CW so that the TCP sender could set its cwnd to a more sensible value when congestion is detected. Also, such a combination improves TCP's ability to differentiate random wireless packet losses from losses caused by congestion [1].

A. Timestamp-based Available Rate Estimation

TCP New Jersey employs timestamp based available rate estimator at the sender side to probe the end-to-end path bandwidth available to the connection. The sender therefore adjusts the congestion window proactively to better utilize the network resource and avoid the congestion caused by flows contending for the limited bandwidth. It operates based on the rate of the data packets on the forward link. The gap between the consecutive packets conveys the information of the network utilization. The faster the packets arrive, the more the network resource can be utilized. It estimates the available rate according to Eq. (1):

$$R_n = \frac{RTT \times R_{n-1} + Ln}{(t_n - t_{n-1}) + RTT} \quad (1)$$

where Rn is the estimated bandwidth when the nth packet arrives at time tn, tn-1 is the previous packet arrival time, Ln is the size of data of nth packet, and RTT is the TCP’s estimation of the end-to-end round trip time delay at time tn. tn – tn-1 is known as delta.

The sender then interprets this rate as the Optimal congestion Window (ownd) in unit of segment by Eq. (2):

$$\mathrm{owdn}_n = \frac{RTT \times R_n}{\text{Segment Size}} \quad (2)$$

where Segment Size is the fixed segment size. The result from the estimator reflects the desirable sending rate for the forward channel. The reliability of the available bandwidth estimation comes from the strong resemblance between the forward traffic pattern and the reverse ACK pattern. After knowing the loss due to congestion it estimates the available rate and accordingly sets the cwnd so that it results into better throughput. The timestamps, which are delivered by ACKs and received at the sender side, closely reveal the arrival traffic pattern at the receiver side.

Due to the ability of TCP New Jersey to accurately estimate the available network rate on the fly, ssthresh can therefore be adjusted in a more effective and dynamic way. Basically what the estimation result implies is the amount of rate that can be utilized without causing network congestion. In New Jersey, upon receiving an ACK that acknowledges the delivery of a new packet, the sender sets ssthresh to ownd computed by Eq. (2). Therefore, ssthresh closely follows the dynamics of the available rate of the path, by which the process enters and leaves the slow start and congestion avoidance phases proactively, and is adaptive to changing network conditions. By doing this, the sending rate would have a fast response to every changing network condition and converges to the optimal rate faster and more accurately without self-induced congestion.

B. Congestion Warning

The ECN scheme marks packets probabilistically, while the average queue length lies between minth and maxth. The router thereby not only informs the sender of the congestion, but also influences on which TCP connection the congestion window size would be adjusted due to its randomness in packet marking. Echoing back ECN information from the receiver to the sender takes time, whereas network situation is constantly changing. Although ECN provides valuable congestion information, this information may not be timely enough for the sender to make the right decision suitable for the current network status under all circumstances. Moreover, ECN is sensitive to parameter settings [2]. Improper parameter settings may lead to unsatisfactory TCP performance [3]. Therefore a simpler congestion notification scheme, namely congestion warning (CW), with fewer parameter settings, yet still provides essential and accurate congestion information to the sender. In CW the router shall mark all the packets when the average queue length exceeds a threshold (thresh) and leave the TCP sender who receives marks to decide its window adjustment strategy. CW inherits the same information bits used in the original ECN implementation, i.e., the CE bit in the IP header and the ECE and CWR bits in the TCP header to convey the congestion warning information.The calculation of the average queue length in CW also differs from that of the original ECN. In the original ECN, the average queue length is largely dependent on a long-term averaging value of the instantaneous queue length [4]. However, in CW, a larger queue weight is preferred since we expect the average queue length to closely track the instantaneous queue length and at the same time smooth out small spikes in the instantaneous queue length. A close tracking of the instantaneous queue length provides the sender with more accurate buffer information of the router. The original queue weight suggested in [4] is rather small, e.g., 0.002. Experiments show that a queue weight of 0.2 would be good enough to track and smooth the instantaneous queue length.

C. Loss Recovery Mechanism in TCP New Jersey

Figure 1 shows the flow chart of the basic functionality and loss recovery mechanism of TCP New Jersey. When timeout occurs due to the packet loss, it revises the value of cwnd and ssthresh. Generally cwnd value is set to one and it starts from slow start phase. When a new ACK comes then it first checks whether it is with CW mark or not. If it is with CW mark then it assumes that there is congestion in the network and it triggers the rate control procedure. After reducing sending rate it checks whether the outstanding packets are less than cwnd. If it is so, then it sends packet otherwise it waits for the ACK. But if there is an ACK with no CW mark then it works as TCP Newreno.

When 3 DUPACKs come then first it checks whether 3rd DUPACK is with CW mark or not. If it is so then assumes that the network is congested and uses rate control procedure to set the value of cwnd and ssthresh and then retransmit the lost packet. If the 3rd DUPACK is not with CW mark then it assumes that the loss is due to the wireless errors and therefore it inflats the value of cwnd and ssthresh and just retransmits the lost packet without doing rate control.

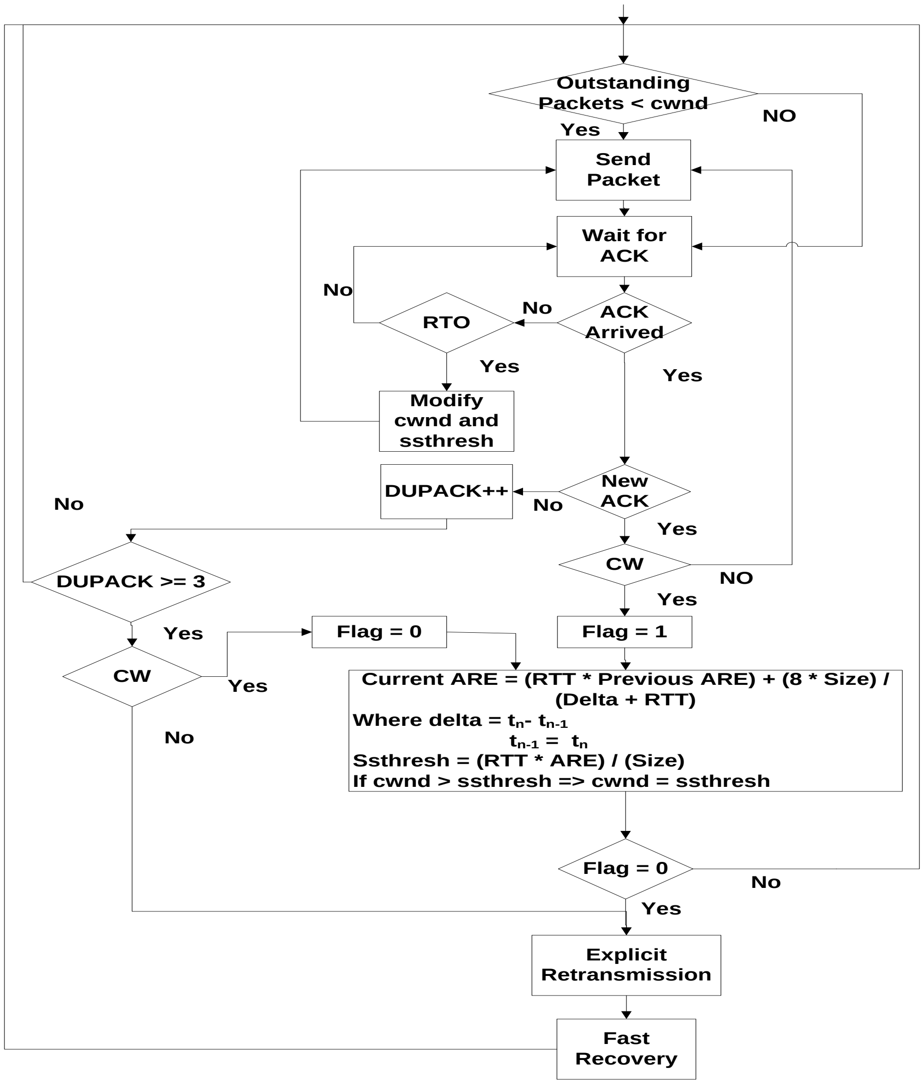

Fig. 1: Functionality of TCP New Jersey

III. SIMULATION RESULTS

In this section validation of loss recovery mechanism of TCP New Jersey through simulations has been done. Five simultaneous TCP flows have been taken to introduce congestion and to introduce wireless losses well-known two-state markov model has been used. The simulation topology and simulation parameters are shown below.

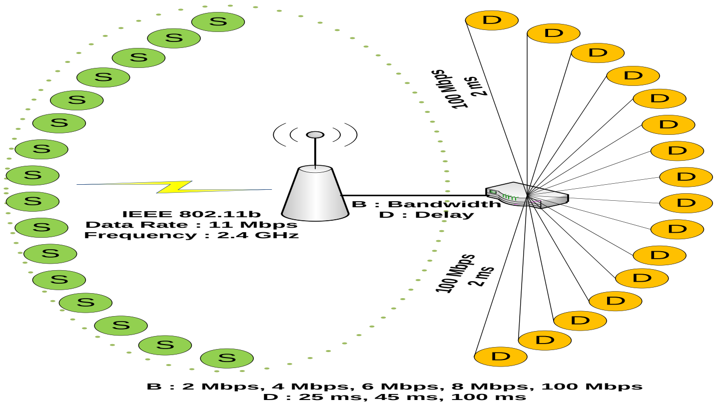

Fig. 2: Simulation Topology

Topology	Fixed
Number of wireless nodes	16
Number of Base-stations	1
Number of wired nodes	17
Antenna	Omnidirectional
MAC protocol	IEEE 802.11g
SIFS Time	10 µs
DIFS Time	28 µs
Preamble Length	96 bits
PLCP Header Length	40 bits
PLCP Data Rate	6 Mbps
Propagation Model	Two Ray Ground
BDP	4Mbps * 25ms
Packet Size	960 Bytes
Maximum cwnd	25 as per [(2*B*D)/(8*Packet Size)]
Error Model	Two-State Markov Model
Simulation Time	100 sec (100 sec to 200 sec)
TCP Variant	TCP New Jersey

Table 1: Selection of Parameters

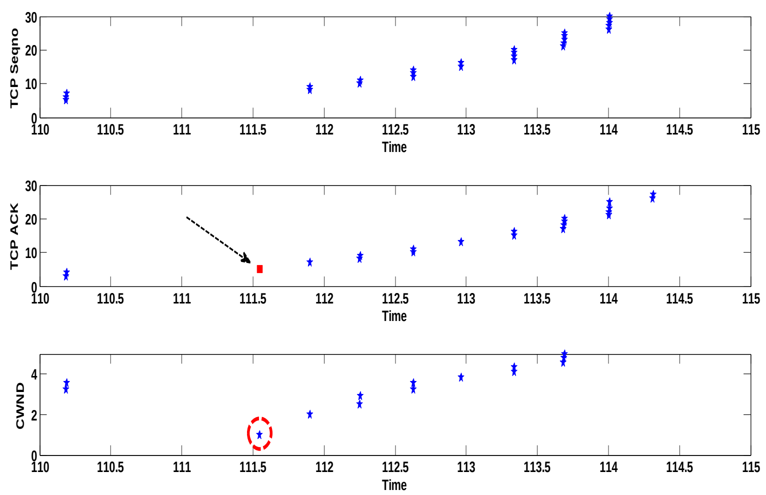

Fig. 3: TCP New Jersey Sender's Response to Timeout

From Figure 3, it can be seen that when timeout occurs then TCP New Jersey makes cwnd to 1 blindly and starts from slow start phase. Timeout has been indicated as a red mark and the value of cwnd at timeout is highlighted with red mark.

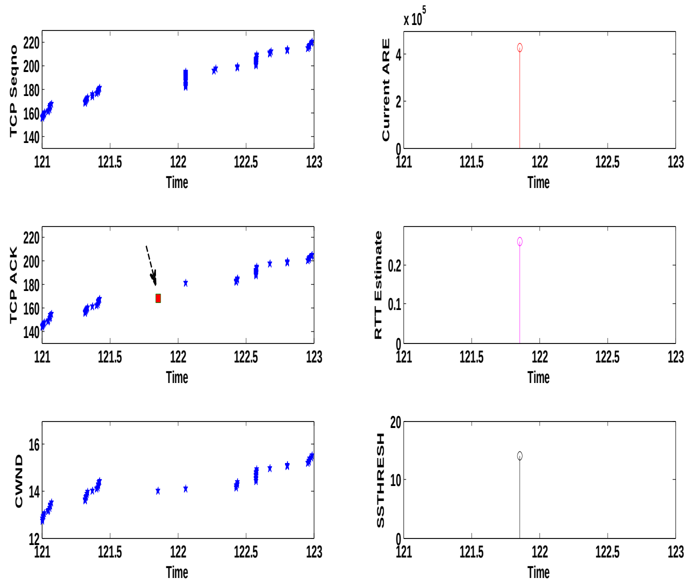

Fig. 4: TCP New Jersey Senders Response to 3 DUPACKs with CW mark

As shown in Figure 4 when three DUPACKs come to the TCP sender with CW mark then it assumes that there is congestion in the network and hence it triggers the rate control procedure. It calculates the value of ssthresh with Eq. (2). Then if the value of cwnd is higher than ssthresh then it makes the cwnd equal to ssthresh and then starts from congestion avoidance phase.

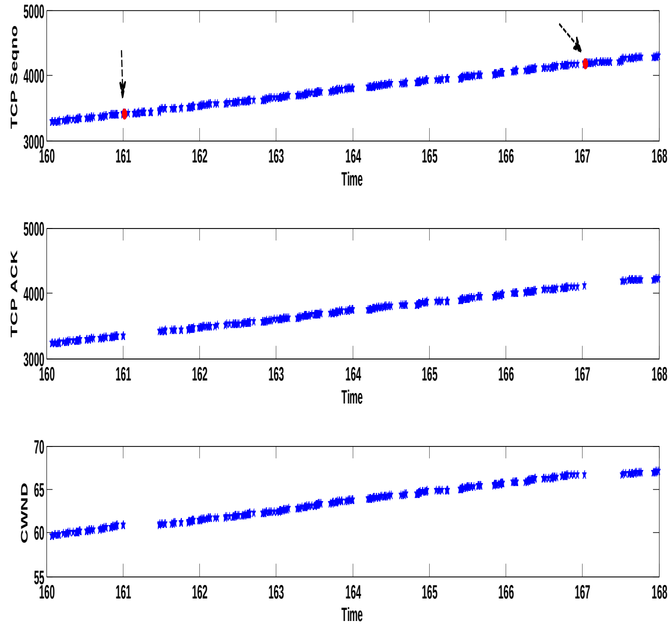

Fig. 5: TCP New Jersey Sender's Response to 3 DUPACKs without CW mark

Figure 5 indicates that when three DUPACKs without CW mark come to the sender then TCP New Jersey explicitly retransmit the lost packet and retains the value of cwnd because it assumes that the loss is due to the wireless errors and there is no need to reduce the value of cwnd. Then it enters into the fast recovery phase and then starts from congestion avoidance phase.

A. Case where ARE is inapproprioate

As per the equation of ARE, it can be said that it depends upon the value of inter-arrival gap of the packets at receiver (delta), RTT and previous value of ARE. The investigation of the problem faced by TCP New Jersey due to the inappropriate estimation of ARE has been done by taking 0.1% FER.

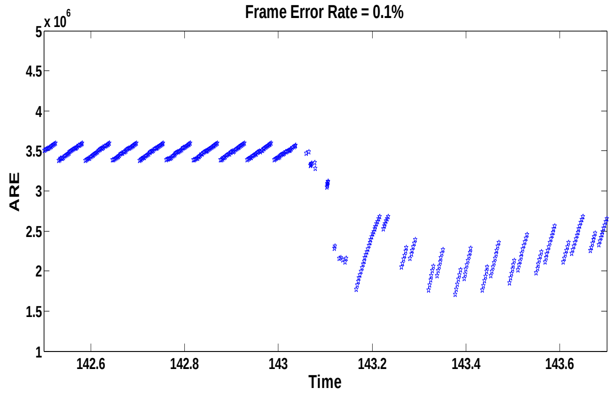

Fig. 6: ARE Vs. Time (FER = 0.1%)

To do the analysis some portion has been taken in which ARE value is decreased as shown in Figure 6. Now from Eq. (1) it can be said that as the value of RTT and delta is increased, the value of ARE should be decreased. Here the value of ARE is decreased so that it is necessary for us to analyze the value of delta and RTT at the same time. In the ARE equation RTT means smoothed RTT plus RTT variance so both values and their addition is also considered.

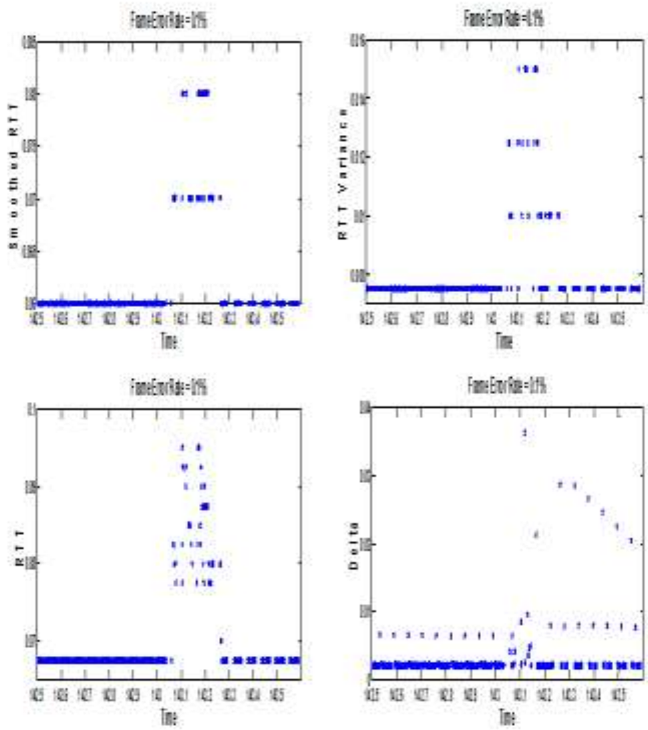

Fig. 7: ARE Parameters Vs. Time (FER = 0.1%)

From Figure 6 and Figure 7, it can be seen that when the value of RTT increases ARE starts decreasing. Observing the value of delta it can be seen that its value is also increased at that particular time.

It is known that the reason behind the increment in delta value is that packets take much time to reach at the receiver and it results into the increment in RTT. Now the values of the RTT and delta are attributed to the value of instantaneous queue so analysis of the queue length has been done.

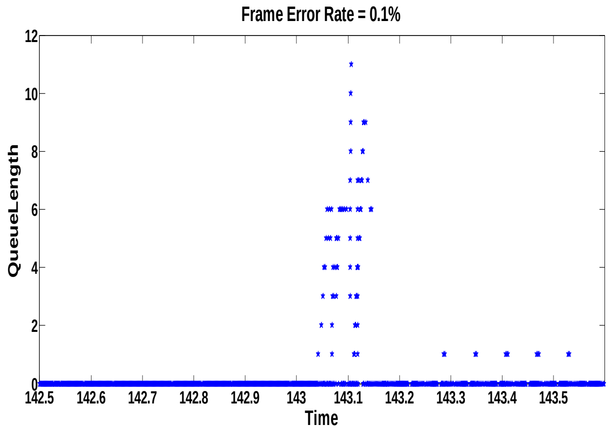

Fig. 8: Queue Length Vs. Time (FER = 0.1%)

From Figure 8, it can be seen that the value of queue length is also increased at that particular time. That means queue is getting full and ultimately it effects on the value of ARE.

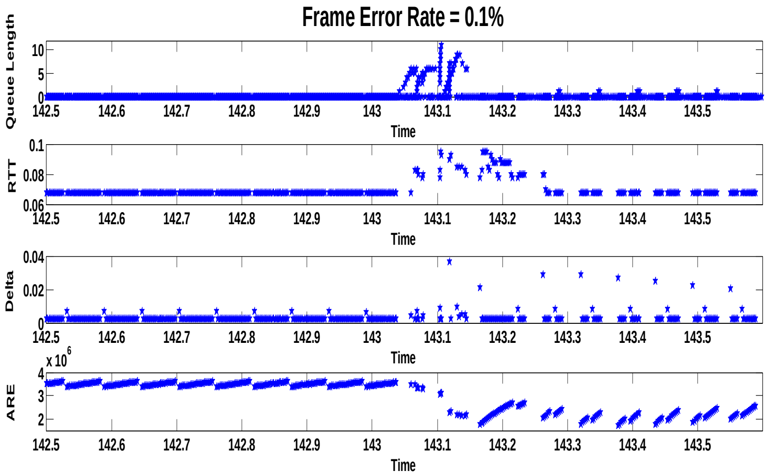

Fig. 9: Parameters affecting ARE Vs. Time (FER = 0.1%)

Combining all the analysis in Figure 9, it can be said that as queue length is increased the values of RTT and delta are also increased which results into the degradation in the value of ARE.

In this paper behaviour of TCP New Jersey is discussed theoretically as well as by simulations. Also the problem with ARE has been investigated. The mechanism of TCP New Jersey is inappropriate in some cases, where despite of sufficient rate at the network, it can't be used and so that reduction in the sending rate of TCP occurs.

IV. CONCLUSION & FUTURE SCOPE

TCP New Jersey utilises rate based congestion control mechanism for revising TCP state variables. It advocates for retaining of cwnd using loss discrimination particularly after 3 DUPACKs. Any kind of over estimation is prevented by utilising ECN notification. It modifies TCP response in case of non-congestion RTO and hence results into quick network utilisation. However, the additional increase in RTT reduces ARE. ARE is dependent on previous values and hence may not be raised to a value consistent with the network capacity. This in turn, compromise with network utilisation due to inferior value for instantaneous sending rate.

In future, it is desired to provide a mechanism for preventing undue reduction in ARE using a cross layer architecture.

ACKNOWLEDGMENT

I would like to thank my guide Prof. Purvang D. Dalal, Dharmsinh Desai University, Nadiad for his suggestions and guidance. He has remained a constant source of inspiration, guidance and encouragement to me. It is a matter of pride for me to express my deep gratitude and indebtedness to Dr. Nikhil J. Kothari, Head, Department of Electronics & Comm. Engg., Dharmsinh Desai University, Nadiad, for his encouragement and invaluable support. My thanks and appreciations also go to my colleague in developing the project and people who have willingly helped me out with their abilities. I strive to perform better and reach un-scalable heights so as to glorify the Institute and all those who associate with it or are its mentors. Last but not least, I am cordially thankful to my parents & my wife for their painstaking patience and co-operation while completing this work.

REFERENCES

[1]. Kai Xu Ye Tian and N.Ansari. Improving tcp performance in integrated wireless communications networks. Elsevier, Computer Networks 47 (2005), pages 219-237, August 2004.

[2]. T. Bonald, M. May, and J. C. Bolot, “Analytic evaluation of RED performance,” in INFOCOM 2000, vol. 3, Mar. 2000, pp. 1415–1424.

[3]. M. May, J. Bolot, C. Diot, and B. Lyles, “Reasons not to deploy RED,” in Proc. Int. Workshop Quality-of-Service (IWQoS), 1999, pp. 260–262.

[4]. S. Floyd and V. Jacobson, “Random early detection gateways for congestion avoidance,” IEEE/ACM Trans. Networking, vol. 1, pp. 397–413, Aug. 1993.