Enhancing TCP Performance in Mobile Ad hoc Networks

Ameer Ahmed, M.H.Zaidi, Kashif Sharif
{52ameer, drzaidi, kashif.sharif}@niit.edu.pk
NUST Institute of Information Technology (NIIT), Rawalpindi, Pakistan

Abstract: *Transmission Control Protocol (TCP) is widely used end-to-end transport layer communication protocol. It has self-generating error and flow control schemes. It performs well in wired networks, where packet losses are primarily due to congestion. Packet loss in wireless networks could be due to bit error rate (BER) or disconnections due to mobility. If standard TCP is applied on wireless networks, it leads to performance degradation. In this paper, performance of Newreno, Selective Acknowledgement (SACK), Explicit Congestion Notification (ECN) and Westwood have been analyzed through simulations and then new solution Enhanced Newreno (E-Newreno) has been proposed and tested through simulations. Results for E-Newreno show more stable and consistent results. Its link utilization mechanism is much efficient than other protocols.*

Keywords: *TCP, Bit error rate, Timeout, Disconnections*

1. INTRODUCTION

Due to the rapid developments in the area of wireless communication and the popularity of the Internet packet data services for applications like e-mail, file transfer like ftp, web surfing like http, interactive remote terminal applications like telnet, mobile computing etc., over wireless are gaining importance. Most of these applications make use of reliable end-to-end transport protocol such as TCP [8], [9], 13]. The transmission control protocol (TCP) is the most commonly used transport layer protocol for current Internet applications. It has been proved to perform reliably in traditional stationary wired networks where losses are mostly caused by network congestion. However, when TCP is applied to wireless networks, it results in degraded and sub-optimal performance. TCP misinterprets the packet losses due to transmission related losses as due to congestion and hence invokes congestion control mechanism resulting in reduction of the link utilization and consequently, a significant degradation in performance in the form of poor throughput. This is due to the medium characteristics of the wireless networks. Medium characteristics of wireless networks are low bandwidth, high bit error rates and disconnections due to mobility. These high error rates and disconnections cause timeouts and as a result TCP has to initiate slow start phase. Many solutions have been proposed to improve the performance of TCP in wireless ad hoc networks. These proposed enhancements fall into three categories: Link Layer, End-to-End, and Split-Connection. A survey on these solutions can be found in [6], [12], [13].

In this paper, we have compared and analyzed the performance of four End-to-End solutions: Newreno, SACK, ECN and Westwood over mobile ad hoc networks using Network Simulator (ns2) [11]. After evaluating the performance of these protocols, an enhancement "E-Newreno" has been proposed and tested on ns2 .The rest of paper is organized in the following manner. Sections 2 and 3 give a brief overview of simulated protocols and enhancement proposed over Newreno. Section 4 describes the experimental setup. Section 5, describes different scenarios for performance measurement and there is a conclusion and future work in sections 6 and 7.

2. SIMULATED PROTOCOLS

2.1 New Reno

The TCP Newreno protocol improves the performance of TCP Reno after multiple packet losses by remaining in the fast recovery mode if the first new acknowledgement received after a fast retransmission is partial [10]. Such partial acknowledgements indicate multiple packet losses within the single window of data. From the three duplicate acknowledgements, the sender infers a packet loss and retransmits the indicated packet. Remaining packets are retransmitted on arrival of single duplicate acknowledgement. In Newreno, there is a variable called recover that keeps the record of highest sequence number transmitted before the start of fast retransmission. By remaining in fast recovery mode, TCP recovers from losses at the rate of one segment per round-trip time, rather than waiting until a timeout to occur as TCP Reno does often. However, the sender still assumes that losses are a result of congestion and invokes congestion control procedures by shrinking its congestion window. TCP Newreno applies only for those TCP connections that are unable to use the TCP SACK [7].

2.2 Explicit Congestion Notification (ECN)

TCP Congestion Control is mainly governed by the detection of lost or dropped segments. When a sender detects segment losses, it assumes that this is due to congestion and it should therefore lower its sending rate. TCP with Explicit Congestion Notification (ECN) [1], [4], [5], [11] requires more functionality from the routers (all the routing hosts should be ECN capable). For ECN to work, a bit ECN Capable Transport (ECT) in Type of Service field in IP header (bit 6) is marked to tell about the ECN capability of sender. Another bit Congestion Experienced (CE) in Type of Service field in IP header (bit 7) is marked by the router if it experiences congestion which is used by the receiver to tell the sender about congestion by marking ECN Echo flag (bit 9) in TCP header's option field. Routers inform the TCP sender of incipient congestion by signaling that sending rate should be lowered. In response to ECN Echo flag, TCP sender marks Congest Window Reduced flag (CWR) bit 8 in TCP header's option field to inform the receiver about lowering the sending rate. In this way, the sender is informed on time when congestion starts building up and segment drops can be avoided. If the entire internetworking infrastructure is made capable of conveying ECN notifications, a mobile host would reasonably infer that drops in the absence of ECN signaling are due to random losses and hence that congestion control algorithms should not be applied and current sending rate should be sustained. ECN can lead to better overall TCP performance by avoiding retransmissions and timeouts. ECN can be made more efficient in wireless environment if ICMP Source Quench message (ISQ) is used from the intermediate routers for congestion notification to the sender. Through ISQ sending host will be notified earlier about packet loss or congestion occurrence.

2.3 SACK

SACK is a type of selective acknowledgements for the TCP to provide the sender with sufficient information to recover quickly from multiple packet losses within a single transmission window [1], [7]. Each acknowledgement contains information about up to three noncontiguous blocks of data that have been received successfully by the receiver. Each block of data is described by its starting and ending sequence numbers describing the left and right edges of blocks of received data. The congestion control actions are performed at the sender whenever losses occur. SACK uses Reno's Fast Recovery algorithm and each packet loss leads to congestion avoidance as compared to Newreno (once for all in single window). TCP with SACK option performs better than standard TCP in situations where there are multiple packet losses within a window of outstanding data. However this scheme is not good when the sender's window size is small [7].

The receiver awaits the receipt of data by means of retransmissions to fill the gaps between received blocks. When missing segments are received, data receiver acknowledges the data by advancing the left window edge in the Acknowledgement Number Field (ANF) of the TCP header. Sender receiving ACK with SACK option builds a mask to keep track of all the packets sent, packets lost and packets retransmitted.

2.4 Westwood

TCP Westwood (TCPW) is a recently proposed sender side modification of TCP Newreno and follows the fundamental end-to-end design principle of TCP without requiring any intervention from intermediate hosts or other layers [3]. In contrast with TCP Reno, which simply halves the congestion window after three duplicate ACKs, TCPW takes more informed decision with the help of bandwidth estimate and enhances the window congestion control and back-off process. It selects a slow start threshold (*ssthresh*) and a congestion window (*cwnd*) that is consistent with the effective connection rate at the time congestion is experienced. This mechanism is called faster recovery. Whenever a sender perceives a packet loss (via 3 duplicate ACKs), the sender uses a Bandwidth Estimate (BE) to properly set the congestion window (*cwnd*) and the slow start threshold (*ssthresh*).

Bandwidth estimate is based on acknowledgements rate coming from the receiver. By backing off to *cwnd* and *ssthresh* values that are based on the estimated bandwidth (rather than simply halving the current values as Newreno does), TCPW avoids overly conservative reductions of *cwnd* and *ssthresh* and thus gives a faster recovery. Most importantly, TCPW is very effective in handling wireless losses in static wireless networks. This is because TCPW uses the current estimated rate as reference for resetting the congestion window. TCPW provides significantly higher utilization than Reno but it cannot work over mobile ad hoc networks because of constantly changing hosts position and performs better with RED as a queuing mechanism. TCPW overestimates or underestimates the bandwidth due to mobility of wireless hosts and results in more losses, less throughput and high delays. This is the reason for bad performance in wireless ad hoc networks.

3. ENHANCED NEWRENO (E-NEWRENO)

Selection of Newreno for enhancement out of four protocols was due to its better performance. It has more stable and consistent results as compared to other protocols. The major problem that Newreno faces is the frequent timeout occurrence because of packet loss either due to BER or due to disconnections resulting from mobility. Due to these frequent timeouts, TCP has to go to slow start phase more often and it results in poor link utilization. We have proposed an enhancement in Newreno called Enhanced Newreno or "E-Newreno" for handling such frequent timeouts due to link loss (BER) or disconnection due to mobility and frequent reduction in congestion window and initiation of slow start phase. E-Newreno is an enhanced version of Newreno.

In this algorithm, whenever a timeout occurs, sender TCP sends a probe packet to the receiver. When receiver receives the probe packet, it generates an acknowledgement with the last acknowledgement number that the sender is expecting. When sender receives this acknowledgement, it remains in congestion avoidance phase (as Newreno does on arrival of 3rd duplicate) [10] and starts sending data without going to slow start phase. This is shown in (1) and (2). By using the new levels of *ssthresh* and *cwnd* after timeout, we can achieve better throughput.

$$\text{ssthresh} = \max\left(\frac{\text{Flight Size}}{2}, 2 * \text{SMSS}\right) \quad (1)$$

$$cwnd = ssthresh + 3 * SMSS \quad (2)$$

Flight Size is the amount of data that has been sent but not yet acknowledged. SMSS is the maximum segment size that a widow allows TCP to send.

E-Newreno algorithm not only saves TCP from data packet loss (only probe packets are lost if timeouts are occurred after disconnection or BER) but it also saves from frequent decrease in window size by going to slow start phase on each timeout. Timeouts are experienced either by the drop of transmitted packet or by the drop of acknowledgement of that packet. Simulations results show that E-Newreno not only saves TCP from packet losses but it also helps to achieve better throughput and link utilization and this difference in better performance can be seen in the next section.

4. SIMULATION ENVIRONMENT

To analyze the performance of these four protocols and our proposed solution over mobile ad hoc networks, we did a comprehensive simulation study using the Network Simulator (ns2) version 2.26 on the basis of throughput, packet loss ratio and average delay under two different scenarios: mobility (variable speed) and both mobility and link loss using uniform error model. In simulation study, there were 20 mobile hosts and there were 20 FTP connections among them. Hosts were arranged in a wireless grid of 1200x800 (meters) with the help of random way point model and same model was used for all the simulations. We ran the simulations for 500 seconds with AODV as the routing protocol on 802.11b physical medium access control protocol and Droptail as queuing mechanism.

5. SIMULATION SCENARIOS

5.1 Mobility

In this scenario, we change the speed of mobile hosts and then performance of all protocols has been measured.

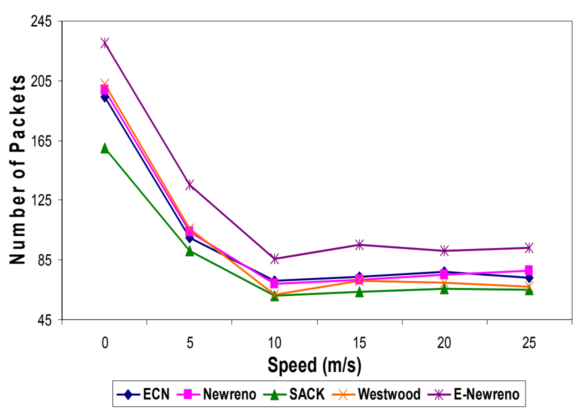

Figure 1- Average throughput (in packets) on variable speeds.

When all the hosts are stationary, Westwood shows better throughput as shown in figure 1. When mobility is introduced (from 5 m/s), its performance degrades due to wrong bandwidth estimation causing congestion and packet losses. All four protocols suffer from timeouts. Average throughput of ECN is better when it is getting timely congestion notification. But its performance is declined when sender receives delayed response about congestion from the receiver and more hops are involved between sender and receiver. Throughput of Newreno is dependent on congestion occurrence. SACK contains cumulative ACKs and drop in acknowledgements causes it to drop the throughput. E-Newreno performs better at all the mobility levels and shows better throughput. Its mechanism of remaining in congestion avoidance phase after timeout and resumption of transmission at congestion avoidance helps to achieve better throughput and more efficient link utilization.

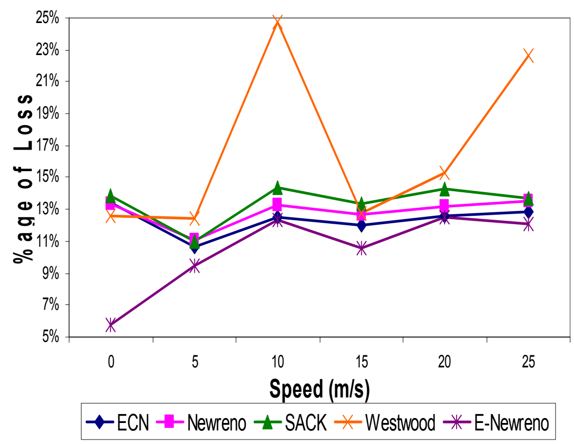

Figure 2- Percentage of lost packets at variable speeds.

At the beginning, E-Newreno has the minimum packet loss of all the remaining four protocols. When the hosts start moving (at 5 m/s), this loss ratio starts increasing but remains minimum from all other protocols. Only probe packets are dropped in E-Newreno. Percentage of lost packets in ECN is minimum than all other protocols (as shown in figure 2). It is due to better congestion handling. Percentage of lost packets of Westwood is inconsistent and sometimes greater as compared to others. This is due to constantly changing bandwidth (hosts are mobile) and wrong estimation of bandwidth. This is the reason for lower throughput for this protocol. Drop of cumulative ACKs of SACK causes more retransmissions and more packets losses. Newreno suffers from packet losses due to timeouts.

Average end-to-end delay of E-Newreno is least from all other protocols except at 10 m/s & 20 m/s where congestion causes some delay and packets loss (as shown in figure 2 and 3). This delay and packet loss is due to high transmission rate after timeouts and not going to slow start phase. Delay of ECN is smaller than other protocols (as shown in figure 3). End to end delay depends on the congestion on the routing hosts. That's why, ECN performs better. But again Westwood shows more delay at 10 m/s, 20 m/s and above. Overestimation of bandwidth causes more delay, more packet loss and hence less throughput. SACK shows more delay due to cumulative ACKs drop and retransmission of those packets after timeouts. Newreno has no mechanism to sniff congestion, which causes congestion, packets loss and timeouts.

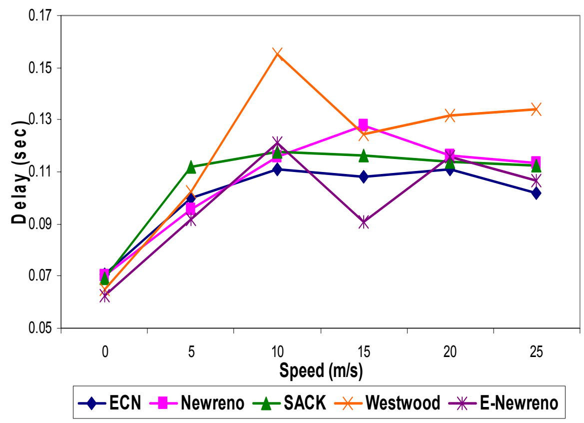

Figure 3- Average end-to-end delay on variable speeds.

In this scenario, the overall performance of E-Newreno in the presence of mobility factor is better as compared to other protocols. It achieves highest throughput and efficient link utilization still at some points in the presence of high delay and packets loss (as shown in figure 2 and 3). An average gain of 21% is achieved by E-Newreno in throughput if we compare Newreno and E-Newreno. ECN shows better performance and gives slightly better throughput, minimum delay and packet loss. This is due to congestion buildup notifications when the route changes because route lookup takes time and causes some delay at the network layer but none of the four protocols compete E-Newreno.

5.2 Mobility and Link Loss

In this scenario, we change the speed in the presence of wireless Link Loss (BER) and then the performance of all the protocols has been measured with the help of defined performance metrics.

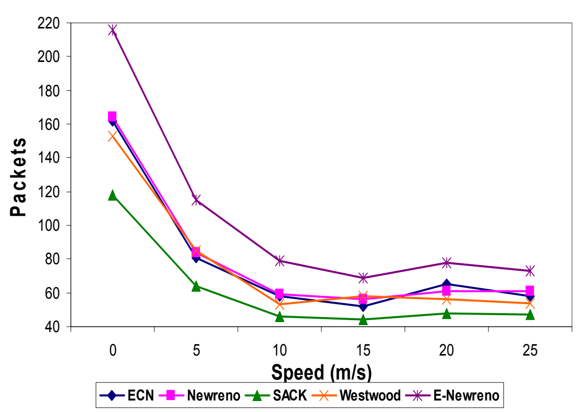

Figure 4- Average throughput on variable speeds and link loss rate = 1%.

Now both mobility and link loss are affecting the throughput of the network. Without link loss, average throughput of E-Newreno was nearly 230 packets (as shown in figure 1) but now it has been reduced to 215 packets at 0 m/s. In this scenario, throughput of E-Newreno is again much better than all other protocols. Since Newreno doesn't have to wait for any explicit response from the network or from the receiver as in the case of ECN and Westwood so its throughput is better and consistent (as shown in figure 4).

When ECN gets timely notifications, it performance improves as it is at 20 m/s speed. At 15 m/s, Westwood estimates the bandwidth of directly connected hosts efficiently so its throughput is better like ECN at 20 m/s. But whenever wrong estimation occurs, Westwood drops the throughput. SACK contains cumulative ACKs, so ACKs drop causes it to drop throughput. Furthermore, SACK uses TCP Reno's Fast Recovery algorithm so it drops its congestion window on each packet drop. But Newreno, ECN, Westwood and E-Newreno drop their congestion window once for all the packets in a single window.

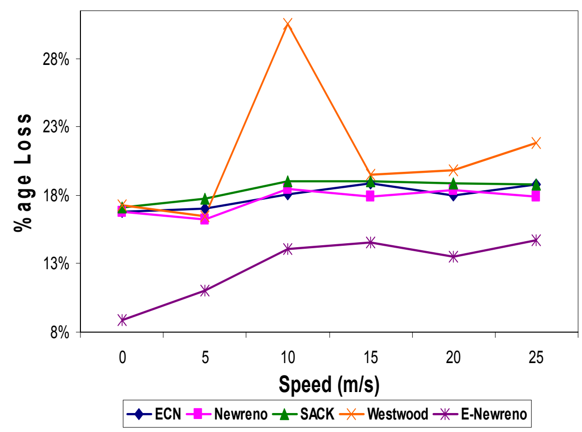

Figure 5- Packet loss percentage on variable speed and link loss rate = 1%.

Without link loss, packet loss of E-Newreno was nearly 5% (as shown in figure 2) but now it has been increased to 9% packets at 0 m/s. This trend of packet loss goes on increasing as speed increases. As E-Newreno has greater throughput therefore it has less loss percentage. Only probe packets are dropped after timeouts occurrence. The variation in loss for E-Newreno is due to variation in delay and hence variation in throughput.

Percentage of packet loss of Newreno is less whenever there is no congestion but when congestion occurs, its performance drops and loss percentage is increased (as shown in figure 5). Loss percentage of ECN increases when it gets delayed congestion building notifications or due to drop of congestion notifications. Loss ratio of Westwood is drastically increased at 10 m/s. This is due of overestimation of bandwidth and packets start dropping at the queues. Drop of cumulative ACKs of SACK cause more retransmissions and more packets loss are experienced at the link and at the receiving end.

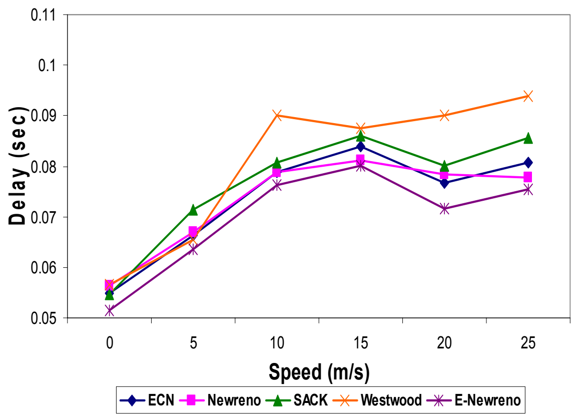

Figure 6 - Average delay on variable speeds and link loss (BER) = 1%.

When the hosts are static, delay for E-Newreno is less than all other four protocols (as shown in figure 6). Delay for E-Newreno is least from its competitors. The variation in delay for E-Newreno is due to variation in transmission rate after resumption of transmission after timeout and this results in variation in throughput also. Since it has high transmission rate after timeouts by remaining in the congestion avoidance phase, delay is varying due to this reason.

At 5 m/s speed, ECN, Newreno and Westwood have experienced less delay and have more throughput and less packet loss. Newreno experiences less delay except at 20 m/s speed where ECN shows better performance due to better congestion handling. It receives congestion notifications timely and sets the transmission window accordingly and efficiently. It saves ECN from packet loss and delay. Westwood experiences much delay due to over estimation of bandwidth causing congestion, more losses and delay at 10 m/s speed and above. Since SACK's ACKs drop causes sender to retransmit packets, it causes more congestion and more delay.

In this scenario, E-Newreno still out performs other protocols in the presence of both mobility and link loss factors. E-Newreno gains 23 % in throughput as compared to its competitor Newreno. Out of the remaining four protocols, performance of Newreno is better. It has consistent results. It doesn't have to wait for any response from the network as with ECN and Westwood. Performance of ECN is dependent on congestion notification. If this notification is dropped then ECN is unable to handle the congestion wisely. That's why, its performance is not good in this scenario because congestion marked packets could be lost either due to mobility or due to link loss. Westwood is dependent on rate estimation. If this rate is estimated wrongly then bad results are experienced.

6. CONCLUSION

One of the most challenging developments in computer networks today is the merger of mobile communications with communicating devices like laptops, PDAs etc. With the increasing importance of host mobility and the popularity of TCP on fixed networks, there is a greater desire for a reliable version of TCP protocol which could be used in mobile ad hoc networks. TCP should be optimized to deal with the problems caused by mobility and link loss and it should provide end-to-end semantics.

From simulation results of four already proposed protocols (ECN, Newreno, Westwood and SACK), it is clear that each solution has its own limitations and dependencies. ECN performs well if it is notified prior to congestion buildup because congestion marked packets can be lost or delayed, otherwise Newreno has more stable and consistent results as compared to other protocols. Westwood suffers from wrong estimation and it works fine only with RED [3]. Hosts mobility results in wrong rate estimation. Cumulative ACKs drop in SACK forces it to retransmit already received packets after timeouts and they are again dropped at the receiver.

After the comparison and simulation study of these four protocols, we have presented a new mechanism "E-Newreno" that deals with timeouts more efficiently. The idea of probe packets saves from packet loss after timeouts. It not only helps in saving TCP from packet loss after timeouts but it also helps to achieve better throughput and link utilization. Since more packets are transmitted after resumption of communication, therefore this increased transmission causes some where delays and packets loss due to congestion at the link but it can be permissible to get a high throughput.

7. FUTURE WORK

From simulation results, it is clear that two factors badly affect the performance of TCP in wireless environment. These factors are link loss (BER) and mobility. TCP is directly dependent on routing and the link layers. In this paper, we have proposed a new mechanism "E-Newreno" and compared it with other solutions under same routing protocol (AODV) and link layer queue (Droptail). Therefore it is strongly suggested that performance of TCP should be analyzed with different routing protocols by varying different queues on link layer. This way the performance of TCP can be analyzed in a broader perspective and new solutions can be proposed.

If any loss segregation mechanism is implemented on TCP which could differentiate the causes of drop either due to congestion or due to bit error rate then we can further improve E-Newreno performance. On bit error loss, we can avoid of going to congestion avoidance or slow start phase after timeout. This way it will definitely improve the TCP performance on lossy wireless links.

ACKNOWLEDGEMENTS

I am thankful to Mr. Nadeem Ahmed, Mr. Ejaz Ahmed and my friends for providing me guidance and technical support in conducting this research work. I am also thankful to Network Simulator developers for providing such a smart platform for carrying this research work.

REFERENCES

[1] W. Ding and A. Jamalipour, "A new explicit loss notification with acknowledgment for wireless TCP", The 12th IEEE International Symposium on Personal, Indoor and Mobile Radio Communication, CA, Sep 30-Oct 3, 2001.

[2] G. Xylomenos, G. C. Polyzos, P. Metri, and M. Saaranen. "TCP Performance Issues over Wireless Links," IEEE Communications Magazine, vol. 39, no. 4, pp. 52-58, April 2001.

[3] C. Casetti, M. Gerla, S. Mascolo, M. Y. Sanadidi, R. Wang. "TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links", in Proceedings of Mobicom 2002, Italy. July 2001.

[4] K. Ramakrishnan, S. Floyd, D. Black. "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, September 2001.

[5] Kostas Pentikousis. "TCP in Wired-cum-Wireless Environment." IEEE Communications Surveys, Fourth Quarter 2000.

[6] Floyd, S., Mahdavi, J., Mathis, M. and M. Podolsky. "An Extension to Selective Acknowledgement (SACK) Option for TCP", RFC 2883, July 2000.

[7] Richard Stevens. TCP/IP Illustrated, Volume 1: The Protocols, Addison-Wesley, 1994.

[8] R. Ludwig, R., Gurtov, A. and F. Khafizov. "TCP over Second (2.5G) and Third (3G) Generation Wireless Networks" RFC 3488, February 2003.

[9] Floyd, S. and T. Henderson. "Newreno Modification to TCP's Fast Recovery Algorithm", RFC 3782, April 2004.

[10] Peng, Fei. and Jian Ma , "A proposal to apply ECN into Wireless and Mobile Networks", draft-fpeng-ecn-04.txt, May, 2001.

[11] Network Simulator 2 (ns2), http://www.isi.edu/nsnam/ns/.

[12] J. Postel, "Transmission Control Protocol," RFC 793, September 1981.

[13] Ashish Natani, Jagannadha Jakilinki, Mansoor Mohsin and Vijay Sharma, "TCP for Wireless Networks", November 12, 2001.

[14] Kanchan Chaudhary and Phu Truong, "TCP Enhancements for Mobile Internet", October 22, 2002.