Clustered-loss Retransmission Protocol over Wireless TCP[1]

Fanglei Sun and Victor O.K. Li
Dept. of Electrical and Electronic Engineering
The University of Hong Kong, Pokfulam, Hong Kong, China

Soung C. Liew
Dept. of Information Engineering
The Chinese University of Hong Kong, Hong Kong, China

Abstract **Transmission Control Protocol (TCP) performs well in traditional wired networks where the packet loss rate is low. However, in heterogeneous wired/wireless networks, the high packet loss rate over wireless links may result in excessive invocation of the congestion control algorithm, thus deteriorating the performance of TCP. In this paper, a novel localized link layer retransmission protocol, called Clustered-loss Retransmission Protocol (CLRP), is proposed. CLRP consists of three protocol components, namely, TCP-FH deployed on a fixed host, TCP-MH deployed on a mobile host and CLRP-BS deployed on a base station. CLRP can provide not only explicit distinction between congestion and packet corruption losses, and effective multiple wireless loss information for retransmissions, but also better retransmission control for wireless losses. Thus it is well suited to wireless networks, in which packet loss and bursty packet corruption is a serious problem. Moreover, CLRP does not require any modifications to TCP deployed on fixed hosts.**

I. Introduction

During the past few years, the proliferation of competing technologies and service network models has accelerated the growth of the wireless Internet. The congestion control algorithms embedded in TCP work well in wired networks in preventing congestion collapse. However, in heterogeneous wired/wireless networks, TCP regards both wired and wireless packet losses as an indicator of network congestion, and thus, TCP and its variations, such as TCP Reno, TCP Newreno and TCP SACK, will invoke the congestion control algorithm although the losses may not be caused by congestion. Therefore, how to allow TCP to distinguish between the losses due to congestion and due to packet corruption in a timely fashion has become the crux of the research on wireless TCP. Several approaches to address this problem have been proposed to improve TCP performance over wireless networks [1]. These approaches include end-to-end mechanism like Veno [2], split connections mechanism like M-TCP and localized link layer mechanism like Snoop. This paper focused on the localized link layer solution and the proposed Clustered-loss Retransmission Protocol (CLRP) can perform better than the existing localized link-layer approaches.

We consider the transmission between a fixed host (FH) and a mobile host (MH) relayed through a base station (BS). The Snoop protocol (Snoop) [3,4] installed at the link layer of a BS monitors the packets and ACKs in both MH to FH and FH to MH directions. For a transmission from FH to MH, Snoop stores the packets arriving at the BS and arranges local retransmissions based on the type of ACKs and local timers. From MH to FH, Snoop adds explicit loss notification (ELN) [5], namely, setting the value of one bit in the six reserved bits included in a TCP header, thus allowing MH to distinguish congestion losses from wireless random losses. However, Snoop can only provide single packet loss information within one RTT (round-trip-time). Under high loss rate wireless environment, Snoop does not work well because it mimics the TCP error recovery mechanism, which is not very robust under harsh error conditions. In bursty traffic network, the lack of explicit and accurate information in Snoop degrades the bandwidth utilization sharply. Furthermore, Snoop offers great improvement in wired-cum-wireless networks, i.e. the transmission is from a fixed host to a mobile host. But when used in wireless-cum-wired or wireless-cum-wireless networks, Snoop is regarded as ineffective [6].

Clustered losses result from bursty multiple packet losses. When multiple packets are lost in a TCP window and within one RTT, the congestion window size will be reduced continuously, degrading the throughput nearly to zero. As a result, timeout is used by TCP to recover packet losses. To overcome this defect, a selective acknowledgment (SACK) mechanism is proposed in RFC 2801 [7]. In TCP SACK, several SACK blocks are used to inform the sender about all the segments that have been received successfully, which allows the sender to retransmit only the lost segments. Each SACK block consists of the beginning and the ending sequence number of a consecutive packet block received by the sender, and thus the holes between the SACK blocks are regarded as lost packets. However, TCP SACK will also cause the following problems: 1) SACK blocks piggy-backed in ACKs take up much space left in the TCP option; 2) SACK blocks transmitted between FH and MH decreases the transmission efficiency, particularly for the transmission with small TCP packet size. Furthermore, the mutual interference between TCP SACK and Snoop when processing bursty losses on wireless links is also shown in [8]. Therefore, it is impractical to solve the problem of clustered losses over wireless networks by using a combination of TCP SACK and Snoop. If TCP SACK is used in the BS directly, it may violate the end-to-end semantics of TCP. Recently, much research has been focused on designing a new ACK [9] for wireless TCP. Unfortunately, it encounters the same problems as TCP SACK.

In this paper, for typical heterogeneous wired/wireless networks consisting of three components, namely, fixed host

[1] This research is supported in part by the Areas of Excellence Scheme established by the University Grants Committee of the Hong Kong Special Administrative Region, China, under project No. AoE/E-01/99.

978-3-8007-2909-8/05/$20.00 ©2005 IEEE

(FH), base station (BS) and mobile host (MH), we propose three protocol components deployed on them to overcome the above problems, namely, TCP deployed on FH (TCP-FH), TCP deployed on MH (TCP-MH) and CLRP deployed on BS (CLRP-BS). As mentioned before, no modifications to TCP is required in TCP- FH. The following functions to handle the wireless losses are performed by CLRP-BS and TCP-MH:

- Storing the packets arriving at BS (FH to MH direction) or their sequence information (MH to FH direction)
- Detecting multiple wireless packet losses
- Piggy-backing loss information in ACKs
- Processing the ACKs with loss information
- Retransmitting wireless lost packets.

In addition, in the two data transmission directions from MH to FH and from FH to MH, CLRP-BS and TCP-MH operate similarly, but divide the above functions differently. By performing all the functions, faster recovery and more effective congestion avoidance over wireless links can be provided. Compared with other protocols, the simulation results presented in Section III show that with CLRP and TCP-MH, the transmission delay is reduced and the throughput is greatly improved.

This paper is organized as follows. In Section II, the implementation of CLRP is described. In Section III, the numerical results are given. The conclusion is provided in Section IV.

II. Implementation of CLRP

Besides a minor difference in the transmission in the two opposite directions, the implementation of CLRP and TCP-MH to handle wireless losses is symmetrical. This means that, from MH to FH, CLRP performs the functions of storing sequence information, detecting multiple wireless losses and piggy-backing the loss information, while TCP-MH performs the functions of processing the ACKs with loss information and retransmitting the wireless lost packets. From FH to MH, TCP-MH performs the functions performed by CLRP in the direction from MH to FH, but CLRP performs the function of storing packets arriving at BS and other functions performed by TCP-MH in the direction from MH to FH. The following is the detailed description of the functions of the two protocol components in both directions in heterogeneous wired/wireless networks. The network topologies are illustrated in Fig. 1.

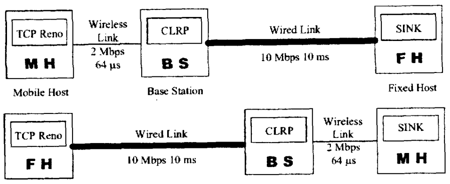

Fig.1. Simulation network topologies with one wireless link

A. *Transmission from MH to FH*

Functions performed by CLRP-BS: Like TCP SACK, the TCP option is used in CLRP-BS. According to the network model and the utilization of other TCP options, such as the timestamp option in RTTM [10], CLRP-BS can flexibly decide the maximum number of lost blocks obtained by an ACK (The value 6 is used in our simulations.). According to the current loss condition, CLRP-BS can also flexibly decide the number of lost blocks to be piggy-backed in an ACK. A lost block stores the sequence numbers of the most recent wireless loss as determined by CLRP-BS. Unlike TCP SACK, multiple packet loss information is conveyed only between BS and MH and only on the ACKs, on which the loss information is piggy-backed. Thus CLRP-BS does not require any modifications to TCP in TCP-FH and it can provide more explicit and accurate loss information for retransmissions with a smaller transmission cost. Moreover, unlike the recovery mechanisms in existing TCP versions and their enhancements in wireless networks, CLRP-BS uses not only duplicate ACKs but also new ACKs to piggy-back loss information. Thus CLRP-BS allows the sender to respond more intelligently to bursty losses than Snoop.

CLRP-BS does not require storing the arriving packets and retransmitting any lost packets because wired networks provide reliable transmission. It only stores the sequence numbers of the received packets so as to determine the sort of losses. For example, a hole between consecutive packets, which persists after several packets have arrived, will be regarded as a wireless loss. Whereas, if the sequence number of a lost packet indicated by some duplicate ACKs (Several duplicate ACKs whose sequence numbers are n indicate that the packet whose sequence number is n has been lost.) is identical with one of sequence numbers stored, it shows that the packet has been transmitted to BS successfully but lost in the later transmission between BS and FH, so the loss will be regarded as a wired congestion loss. However, if the sequence number of a lost packet indicated by duplicate ACKs is not identical with any one of the stored sequence numbers, the loss should be regarded as a wireless loss. When an ACK arrives to the BS, whether a new or a duplicate one, using the above rules, if CLRP-BS detects some wireless losses, then it will piggy-back all their sequence numbers to the ACK as primary explicit multiple wireless loss information. If CLRP-BS does not detect any wireless loss information, the ACK will be transmitted to MH untouched. Therefore, apart from the recovery mechanism to wireless losses provided by the following protocol component TCP-MH, we can still utilize the congestion avoidance mechanism of TCP to handle wired losses, and thus recovery from network losses can be provided.

Functions performed by MH-TCP: Since the packet loss rate over wireless links is high and retransmission control is performed by TCP-MH, minor modifications to TCP-MH are required to take care of the ACKs with explicit multiple wireless loss information and to perform retransmissions of wireless losses. Therefore, a structure list is used in TCP-MH. The structure list consists of many structure cells and each structure cell is composed of the sequence number of a lost packet and the total times that MH has received the sequence information. When an ACK arrives at MH, according to information attached in the ACK, the list is updated, deleting the records of the packets that have been acknowledged, adding the records for the new lost packets or modifying the numbers of times of notifications of lost packets. After updating, if the recorded number of times of any lost packet exceeds the retransmission threshold, the sender will retransmit the lost packet promptly without starting up TCP congestion control. After packet retransmission, the times record will be set to a negative value, say –1. If MH still receives loss information, the times record will remain negative. Generally speaking, according to the type of wireless links, by properly setting the value of the retransmission threshold, aggressive retransmissions can be avoided. In our simulation, the value 2 is chosen as the retransmission threshold for the IEEE 802.11b WLAN. A new ACK and a duplicate ACK are treated differently. The flowchart shown in Fig. 2 summarizes the process.

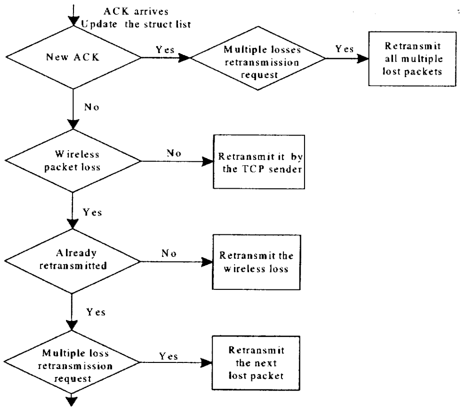
ACK arrives
Update the struct list
New ACK
Yes
Multiple losses retransmission request
Yes
Retransmit all multiple lost packets
No
Wireless packet loss
No
Retransmit it by the TCP sender
Yes
Already retransmitted
No
Retransmit the wireless loss
Yes
Multiple loss retransmission request
Yes
Retransmit the next lost packet

Fig. 2. Flowchart for ACK processing

B. *Transmission from FH to MH*

For data transmission in this direction, CLRP-BS stores all packets received by BS. On the one hand, these packets are used to judge whether a loss is due to network congestion or wireless loss. On the other hand, they support fast local link layer retransmissions of the losses over wireless links. Besides, CLRP-BS performs the functions of processing the ACKs with loss information and retransmitting the wireless loss packets. However, TCP-MH performs the functions of detecting multiple packet losses, piggy-backing the related information on ACKs. Therefore, CLRP-BS and TCP-MH can also work well in the transmission from FH to MH. The implementation details of the two protocol components are the same as mentioned above in this section.

III. Numerical Results

A. *Simulation Topologies*

All simulations in this paper are performed in Network Simulator (NS-2). Fig. 1 and Fig. 3 show the topologies of heterogeneous wired/wireless networks used. In the network with one wireless link, the system consists of a 10 Mbps, 10 ms propagation delay wired channel and a 2 Mbps wireless channel with a negligible propagation delay of 64 μs. The packet size is fixed at 1000 bytes. The maximum congestion window size of the sender is 30 segments. In the network with two wireless links, the parameters are identical with the one wireless link network, except that the propagation delay on the wired channel is 100 ms.

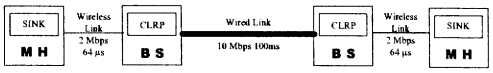

Fig. 3. Simulation network topology with two wireless links

B. *Error Models on Wireless links*

1) *Expo (Exponential) Error Model*

This error model is adopted in the simulations of Snoop [2]. In order to compare with Snoop, we also use it in our simulations under three topologies. The Expo Error Model is a single state error model, which generates errors at a certain rate based on an exponential distribution in the appropriate domain (packet, byte, or time). Actually, since we are dealing with discrete time, the geometric distribution is used to approximate the continuous exponential distribution. In our simulation, we set the unit as packet. However the single state error model cannot reflect the errors occurring in realistic wireless networks. Thus, we use the following error model.

2) *Two-state Markov Error Model*

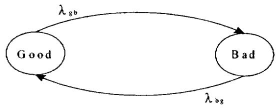

Fig. 4. Two-state Markov Error Model

When considering random noise, multi-path fading and mutual user interference in wireless channels, the Two-state Markov Error Model may be used [11,12]. As shown in Fig. 4, in this model, the wireless link is supposed to be in one of two states: Good or Bad. In the Good state, a geometric packet error model is assumed. A link is assumed to stay in the Good state for a time interval that is geometrically distributed with parameter λ_{gb}. The time spent in the Bad state is also geometrically distributed but with parameter λ_{bg}. Let the average length of the Good state be L_g and the Bad state be L_b. The relation between the average length of the two states and the transitional probabilities can be expressed by the following formulas:

$$L_g = \frac{1}{\lambda_{gb}} \quad \text{and} \quad L_b = \frac{1}{\lambda_{bg}}$$

In our simulations under the Two-state Markov Error Model, we set the average periods of the Good and Bad states as 6s and 0.2s. Since the transmission rate and packet size are fixed at 2 Mbps and 1000 bytes, the corresponding parameters, such as the packet time and the transitional probabilities, can also be derived from the above formulas if needed. Finally we

978-3-8007-2909-8/05/$20.00 ©2005 IEEE

fix the high packet loss rate in Bad state at 50% and vary the packet loss rate in Good state from 0.01% to 10%, and then test the variation of the throughput of the three topologies.

C. *Simulation Results*

In the following, we show our simulation results for different combinations of three types of simulation topologies and two types of error models on wireless links. Due to space limitations, the detailed data analysis is given only for one of the six combinations.

1) *Transmisson from MH to FH*

a) *Expo Error Model*

Fig. 5 shows the throughout performance of TCP Reno, Snoop and CLRP. Unless stated otherwise, the simulation time is set to 300 seconds in our simulations. The performance of the three protocols is close to each other when the packet loss rate (PLR) is less than 0.2%. But when PLR is varied from 0.2% to 1%, the throughput of Snoop is still close to that of TCP Reno, whereas CLRP performs better than the other two protocols. If the PLR is further increased, CLRP has a performance improvement ranging from 6.2% to 48.1%, compared with Snoop, and an enhancement of 39.0% to 147.1%, compared with TCP Reno.

b) *Two-state Markov Error Model*

As shown in Fig. 6, when the PLR in Good state is varied from 0.01% to 1%, CLRP has a comparable performance with Snoop, and they both have certain improvements over TCP Reno. When PLR is above 1%, however, CLRP has distinct improvements over Snoop, from 14.8% to 77.5%. Compared with TCP Reno, the improvement of CLRP is quite impressive, achieving a performance gain of 49.5% to 286.9%. The results demonstrate that CLRP is more robust than the other protocols over wireless networks with bursty losses.

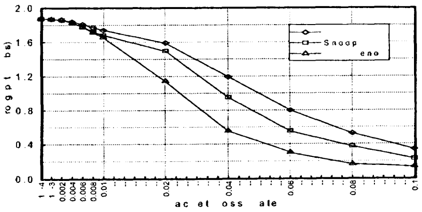

Fig. 5. Throughput versus PLR in Expo Error Model

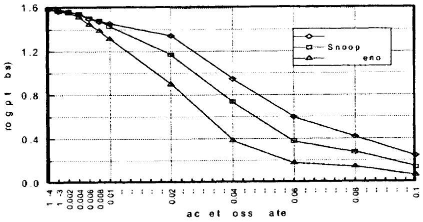

Fig. 6. Throughput versus PLR in Good state

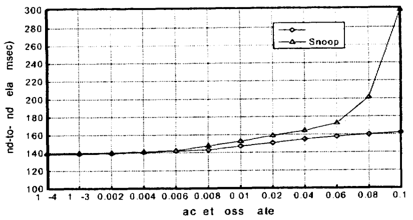

Fig. 7. End-to-End delay versus PLR in Good state

Fig. 7 compares the mean end-to-end delay of CLRP and Snoop, in a simulation time span of 60 seconds. If the packet is successfully transmitted from MH to FH, the end-to-end delay is mainly determined by the propagation, transmission, and queuing delays. However, if the packet is lost due to either wireless loss or network congestion, TCP retransmits the lost packet by performing the related recovery algorithms or appealing to timeout to recover the lost packet. As a result, the end-to-end delay is significantly prolonged. From Fig. 7, when PLR is varied from 0.01% to 10%, the mean end-to-end delay of CLRP is maintained at a constant level, fluctuating between 0.14s to 0.16s, while in Snoop, this value increases sharply from 0.14s to 0.30s. The reason for the poor performance of Snoop at the high bursty loss rate is that it cannot recover from packet losses until the related duplicate ACKs arrive and it cannot deal with multiple losses in one window and in one RTT in a timely fashion.

Fig. 8 and Fig. 9 show the variation of the congestion window size in 100 seconds for CLRP and Snoop when the PLR in Good state is 10%. It is observed that Snoop always keeps the congestion window size less than 12 and frequently reduces it to one for timeouts, while CLRP usually increases the congestion window size to 16 and sends packets under a bigger window. After getting the exact reason of the loss and enough recent wireless loss information, the timely retransmissions of CLRP avoid excessive idle time and further lower the probability of the occurrence of timeouts.

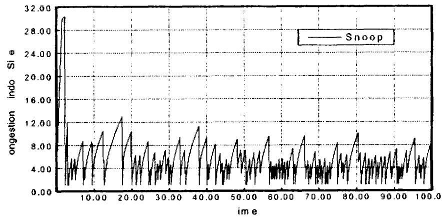

Fig. 8. Snoop congestion window size

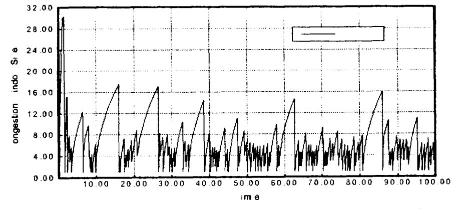

Fig. 9. CLRP congestion window size

2) *Transmission from FH to MH*

a) *Expo Error Model*

In Fig. 10, we see that, compared with TCP Reno, the throughput of CLRP and Snoop are much improved even if PLR is low. The improvement is more pronounced when PLR is above 1%. When PLR is around 10%, CLRP has a performance enhancement of 27.5% over Snoop, and about 705.2% over TCP Reno.

b) Two-state Markov Error Model

As shown in Fig. 11, CLRP is robust to high burst loss rate. Deserving special attention is the fact that the performance of CLRP has steady improvements over Snoop. Even when the PLR is 0.01%, CLRP has an enhancement of 6.3%. At a PLR of 10%, the enhancement is 41.3% over Snoop and 1334.2% over TCP Reno.

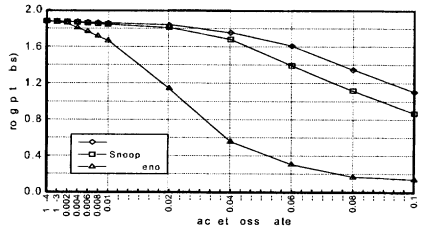

Fig. 10. Throughput versus PER in Expo Error Model

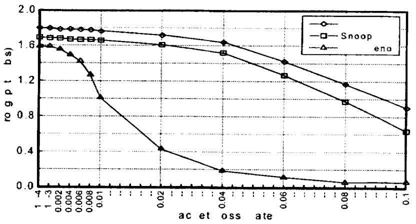

Fig. 11. Throughput versus PLR in Good state

3) Transmission from MH to MH

a) Expo Error Model

As presented in Fig. 12, CLRP has small and steady improvements of throughput over Snoop at all PLRs. When the PLR is higher than 6%, the throughput of Snoop is close to that of TCP Reno, while CLRP has a distinct improvement over the other two. This result shows that when packets are transmitted over wireless links with high PLR, the interference between the two wireless links leads to a sharp decrease in the bandwidth utilization, while CLRP has stronger adaptability to this wireless environment.

b) Two-state Markov Error Model

From Fig. 13, CLRP has a prominent and steady improvement over Snoop, ranging from 5.0% to 169.2%. This shows that CLRP is superior to Snoop in combatting bursty losses. Compared with the simulation results in the above error model, however, the throughput of the three protocols is greatly reduced, so the adverse effects of bursty losses on the networks from MH to MH cannot be neglected.

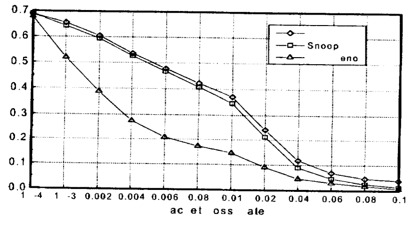

Fig. 12. Throughput versus PLR in Expo Error Model

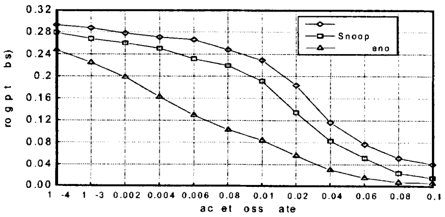

Fig. 13. Throughput versus PLR in Good state

IV. CONCLUSION

Performance improvement of TCP over wireless networks is an important problem in the wireless Internet. In this paper, we propose a novel localized link layer protocol called Clustered-loss Retransmission Protocol (CLRP) to enhance the performance of TCP over heterogeneous wired/wireless networks. CLRP consists of three protocol components, namely, TCP-FH deployed on a fixed host, TCP-MH deployed on a mobile host and CLRP-BS deployed on base station. Analyses and simulation results show that CLRP can provide not only explicit and effective wireless loss information, but also better retransmission control for wireless losses.

REFERENCES

[1] H. Balakrishnan, V. Padmanabhan, S. Seshan, and R. H. Katz, "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links," *IEEE/ACM Transactions on Networking*, December 1997.

[2] C. P. Fu and S. C. Liew, "TCP Veno: TCP Enhancement for Transmission over Wireless Access Networks," IEEE J. on Selected Areas in Commu, Vol. 21, No. 2, pp. 216-228, Feb. 2003.

[3] H. Balakrishnan, S. Seshan, and R. H. Katz, "Improving Reliable Transport and Handoff Performance over Cellular Wireless Networks," *ACM Wireless Networks* ,Vol. 1, No. 4, December 1995.

[4] H. Balakrishnan, "Challenges to Reliable Data Transport over Heterogeneous Wireless Networks," Ph.D. thesis, UC Berkeley, May 1998.

[5] H. Balakrishnan and R. H. Katz, "Explicit Loss Notification and Wireless Web Performance," *Proc. IEEE LOBLECOM*, Sydney, Australia, November 1998.

[6] G. Xylomenos and G. C. Polyzos, "Quality of Service Issues in Multi-service Wireless Internet Links," *Proc. of the International Workshop on QoS in Multi-service IP Networks (QoS-IP) 2001*, pp. 347–365

[7] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, "TCP Selective Acknowledgment and Option," RFC 2801, IETF, October 1996.

[8] S. Vangala and M. Labrador, "The TCP SACK-Aware-Snoop Protocol for TCP over Wireless Networks," *Proc. IEEE VTC*, Orlando, October 2003.

[9] W. Ding and A. Jamalipour, "A New Explicit Loss Notification with Acknowledgment for Wireless TCP," *Proc. IEEE International Symposium on Personal, Indoor and Mobile Radio Communication (PIMRC2001)*, San Diego, CA, September 2001.

[10] V. Jacobson, R. Braden, and D. Borman, "TCP Extensions for High Performance," RFC1323, May 1992.

[11] A.A. Abouzeid, S. Roy and M. Azizoglu, "Stochastic Modeling of TCP over Lossy Link," *Proc. IEEE INFOCOM 2000*, Tel Aviv, Israel, March 2000.

[12] M. Gerla, M. Sanadidi, R. Wang, A. Zanella, C. Casetti, and S. Masco, "TCP Westwood: Window Control Using Bandwidth Estimation," *Proc. IEEE LOBECOM*, San Antonio, Texas, USA, November 2001.