Simulation-based Performance Analysis of Split-Connection with Different TCP Variants over Wireless Link

Yashar Komijany
University of Tehran
y.komijany@ece.ut.ac.ir

Arash Hooshmand
University of Tehran
arash@cad.ece.ut.ac.ir

Abstract- **This paper has several contributions. First, it reviews different causes of TCP performance degradation over wireless links and their solutions. Second, Significance of reaction delay and the need for local retransmission are emphasized both by a huristic discussion and through simulation results. Split-Connection, though it suffers from End-to-End semantic violation of TCP acknowledgements, is shown to be the most effective solution. Third, several TCP variants are compared in order to choose the best TCP protocol over wireless part.**

Keywords- **TCP, Wireless, Split-Connection.**

I. INTRODUCTION

Fast growth of Internet popularity and mobile communication has made it necessary to provide users with Internet connectivity, using wireless technology.

However, there are several problems in providing Internet over wireless networks which have been recently challenged by researchers. TCP performance degradation over wireless links is one of these problems that is well considered during last years. Although several solutions have been proposed to the problem, the results are still far from ideal conditions.

A. Background on TCP

TCP [1] has been used in Internet as a transport protocol to provide a reliable byte-stream service to upper layers by exchange of data and acknowledgement packets (Ack) between sender and receiver. It limits the number of outstanding packets using a sliding window and uses Ack arrivals as a clock to send new data packets to the network.

The first TCP congestion control mechanism developed by Jacobson [2] is Tahoe, the core of which is a variable-size sliding window, which adapts the number of outstanding packets to instant network state using the information carried by Ack packets.

Tahoe has a cyclic window evolution that includes Slow Start and Congestion Avoidance phases. During Slow Start phase, window is rapidly grown up to the estimated capacity of the network and then Congestion Avoidance is employed to probe for further network capacity. This growth is continued until an overflow occurs in one of the intermediate buffers and one of the sent packets is lost. Packet loss is detected by timer expiry, the window size is reduced to one packet and Slow Start is then employed again to fill the connection pipe [2] and start the self-clocking mechanism. Half of maximum window size just before timer expiry is selected as an estimate of network capacity for the next cycle. TCP uses timer expiry to decide when to retransmit packets. The timer has an adaptive timeout that is updated using Ack's interarrival times. Low-frequency average of round-trip times' mean and variance are used to calculate timeout value.

In Reno version [3] developed by Jacobson, Fast Retransmission and Fast Recovery mechanisms were added and resulted in better performance. Fast Retransmission enables sender to detect losses long before coarse-grained TCP timer expiry by inferring the reception of three duplicate Acks as a sign of packet loss. Fast Recovery is used after Fast Retransmission to reduce the window size to half of its value and invoke Congestion Avoidance without returning to Slow Start.

Vegas is another TCP version with a number of modifications, which result in better protocol stability and performance [1]. The most remarkable modification is that interarrival times are used as a congestion signal. Often an interarrival time increase results from buffer length increase and higher congestion probability, which an interarrival time decrease means that intermediate buffers are empty. Vegas uses these signs in addition to packet loss and duplicate Ack to decide on the appropriate window size.

Interested readers are referred to [2, 3, 4] for detailed description of Tahoe and Reno, to [5] for an analytical analysis and to [7] for simulation study. Vegas motivations, details and simulations are frist brought in [10].

B. Problem Definition

The problem of TCP performance degradation over wireless links is studied in this paper by the model sketched in figure 1. In this model, a wireless client Mobile Host (MH) has a TCP connection to a Foreign Host (FH) Internet server through a Base Station (BS).

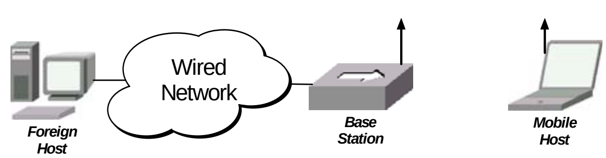

Fig 1. Basic model of a wireless LAN connection.

C. Outline

This paper includes a comparison among different TCP variants over wireless link using both End-to-End and Split-Connection methods. Main causes of TCP performance degradation and solutions to each one are discussed in the following section. Next section has a brief description of Split-Connection approach. Forth section is devoted to a simulation-based comparison of various protocols and our conclusion is brought in section five.

II. Cause Classification And Previous Works

The problem of improving TCP performance over wireless links has been investigated during last few years by research community and many methods are proposed [12]. Based on these works, the main causes of TCP performance degradation over wireless links can be classified in following three groups:

1) Congestion and corruption loss differentiation: TCP infers packet loss as a sign of congestion and reduces its window size. In the presence of wireless links, packets may also be lost due to corruptions caused by fading effects and hand offs. Hence, TCP performance is degraded over these environments because of its inability to distinguish between congestion and corruption losses. When packet loss exceeds a maximum threshold, consecutive window size reduction results in very poor performance [5, 6].

The classic solution to this problem is employing Explicit Loss Notification (ELN) [13] in which receiver explicitly informs sender of packet loss. Knowing that the corruption is the cause of packet loss, sender does not reduce its window and just retransmits the lost packet. ELN Implementation has several practical issues that are not completely solved yet. The most important problem is how receiver can find out whether packet loss is occurred due to congestion or corruption. To solve the just mentioned problem, ELN and Explicit Congestion Notification (ECN) [14] are to be implemented together. ECN is a method of congestion notification in which intermediate nodes (routers) mark the packets instead of dropping them, expecting end-nodes to reduce their rate in response. If a packet is ECN marked, it has experienced congestion but if the packet is lost it has experienced corruptions. Though ECN and ELN combination has many attractions, it has not generally deployed yet, because an unresponsive flow can severely damage fair distribution of network resources [15].

Another pitfall of ELN is that it needs End-to-End TCP modification. Others solutions to congestion and corruption differentiation have been proposed which are mainly based on using RTT interarrival times [16]. TCP Westwood is one of these protocols that uses filtered interarrival time to estimate the instant netword capacity and as previously mentioned this information is used as a threshold between Slow Start and Congestion Avoidance phases. Hence, based on interarrival times Westwood makes a decision on whether the packet loss is resulted from congestion or corruption and in the later case uses appropriate SS and CA thresholds to increase window size to the amount just before packet loss rapidly [17].

2) Multiple packet loss in a window: Another well-known problem of TCP in wireless environments is that TCP cannot recover from multiple packet losses in a window without resorting to a coarse-grained timer expiry [7]. The reason is that TCP uses a cumulative Ack scheme in which one Ack acknowledges reception of all previous data segments. Multiple packet losses are common in wireless links due to burst errors caused by fading and hand offs.

To alleviate this problem, TCP Sack is designed [8] which using sender and receiver modification enables sender to have explicit information about receiver window gaps and the packets that need retransmission. Another sender-side modification to TCP is NewReno [9], which is designed to use partial Acks to decide which packet must be retransmitted.

3) *Reaction Delay:* Even in the best case, if TCP was able to detect corruptions and recover well from multiple packet losses, it had to wait until reception of multiple duplicate Acks and retransmit the lost packet(s). This process takes multiple round-trip times and results in reaction delay that degrades performance. Dependency of TCP throughput to the average round-trip time is approximately given by [11]

$$Th(\tau, q) \approx \frac{1.22L}{\tau\sqrt{q}} \qquad (1)$$

In this equation L is packet length, τ is round-trip time and q is packet loss probability. Eq. 1 shows that as far as local throughput is concerned round-trip time is more influential than packet loss probability. Reducing this reaction time as is done in Link-layer and Split-Connection methods is the solution to the mentioned problem. Local Retransmission of corrupted packets over wireless link not only reduces reaction delay but also decreases effective loss probability as is seen by end-to-end TCP protocol.

Link-layer solutions use FEC/ARQ over wireless link to hide the channel loss from sender view [18, 19]. The channel appears to sender as a high quality but longer channel with a lower capacity. Link-layer solutions suffer from out-of-order delivery which causes duplicate Ack and hence sender window reduction [12]. To cure this problem TCP-aware link-layer solutions are proposed but they need inter-layer communication and difficult implementation [20]. Layer 2 and 4 timers' competition is another problem of link-layer solutions [21].

Split-connection approach is another solution to the reaction delay problem, which splits TCP connection in BS and employs an optimal TCP over wireless part. Therefore, the round-trip time is reduced in the second connection, because sender and receiver are closer to each other and this allows faster reaction to the corruption losses. Split-Connection details are deferred to following sections.

Although a thorough solution to the problem is still far from achievement, we believe and in following sections try to show that whenever connections with long wired and short wireless links are concerned (as it happens in almost every wireless LAN connection to Internet), reaction delay is the utmost cause of throughput degradation. Therefore, local retransmission methods such as Link Layer or Split-

Connection are necessary to improve the performance. Split-Connection would be preferred due to its attractive features such as ability to use a tuned transport protocol over wireless link without need for any end-to-end modification and potential of being combined with Link-Layer solutions.

III. SPLIT-CONNECTION APPROACH

Split-Connection (Indirect-TCP) method [22] is based on breaking TCP connection at the base station into two distinct connections, a standard (Reno) connection from foreign host to the base station and another well-tuned TCP protocol connection over wireless part (figure 2).

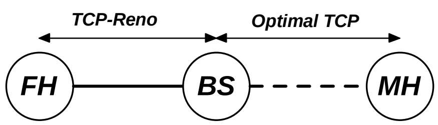

Fig. 2. Split-Connection method.

Packets arrived at BS in the first connection are immediately acknowledged and their payload is then transferred to the second connection. This is called indirect-Ack and is illustrated in figure 3.

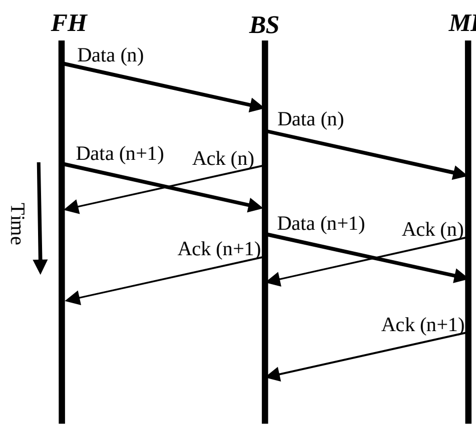

Fig. 3. Indirect-Acknowledgement diagram.

In addition to reaction delay reduction, Split-Connection has several other advantages:

1) The wireless link is fully isolated so that losses due to corruptions and hand offs are hidden from FH TCP sender. Arriving packets in BS are always immediately acknowledged except when the BS buffer is full which causes the arrived packets to be dropped. This helps the wireless link to be treated by End-to-End TCP as a bottleneck link with average capacity of Eq. 1 and queue size of buffer length in BS. Because of flow control and corruption loss recovery separation, losses seen by sender are whether caused by congestion in wired network or BS buffer overflow.

2) Since the connection is broken at the BS, an optimal TCP can be employed over wireless link without any need for End-to-End TCP modification. This makes it easy to combine all of the proposed solutions together, using Split-Connection over End-to-End and optimal TCP and link-layer solutions over the wireless part.

The main problem with Split-Connection is that since two TCP connections are independent, data packets may be acknowledged by base station much before they are actually received by mobile host (Figure 3). The mobile host or base station may go off and acknowledged packets will never be received. Therefore, Split-Connection in its previous form (Indirect-Ack) cannot be used in Internet.

This problem has several solutions [24] and we defer its study to future works [28]. In this paper, we are mainly concerened with first showing the significance of reaction delay and the importance of using a local retransmission scheme and second choosing the best protocol that can be used over wireless part in a Split-Connection method.

IV. SIMULATION STUDY

A. Simulation Setup

The Berkeley NS simulation tool [29] is used for performance evaluation of the Split-Connection method with different TCP protocols over wireless part. Network topology is sketched in figure 6 in which wired network of figure 1 is modeled by a low capacity, long delay link that is typical character of WAN bottleneck links. Bandwidth and delay values are typical values for a connection between a wireless LAN client to an Internet server. The value of 100 msec is chosen as an average round-trip times encountered in Internet [27]. The segments size is assumed 1000 bytes. A buffer of 100 Kbytes is set in the base station. In all Indirect-TCP simulations, TCP Reno is used over the wired link. Wireless link is assumed a memory less channel with a probability of packet loss of PER. Simulation runs of each protocol are performed with wireless packet error rate ranging from zero to 10 percent. For each PER 4000 seconds simulation is performed 3 times. These results are used to calculate mean values, most of which are fitted by quadratic and cubic polynomials.

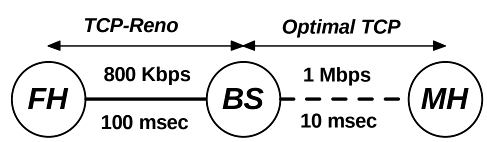

Fig. 4. Simulation network topology

B. Simulation Results

Performance of different End-to-End (E2E) protocols is compared in figure 5. E2E-Reno is the protocol that is currently being used in Internet. It can be seen that E2E-Sack, though it enables TCP to recover from multiple packet loss in a window, does not have any remarkable influence on performane. As priviously mentioned, Westwood cures the congestion and corruptions loss diffrentiation problem. A version of Westwood, known as

WestwoodNR uses NewReno to solve the multiple packet loss problem. The latter performance is shown in figure 5 and it has very considerable improvements.

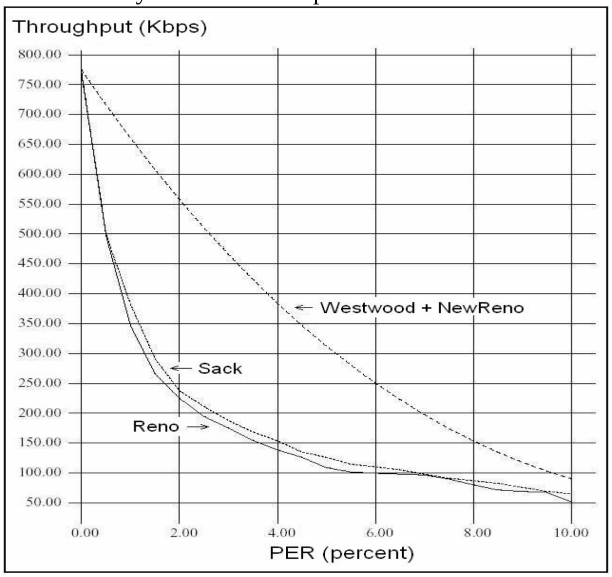

Fig. 5. Various End-to-End TCP performance

The benefits of using Split-Connection method can be seen from figure 6. In this figure, Indirect-Reno, Indirect-Tahoe and Indirect-Vegas are compared to E2E-Reno. Differnece of E2E-Reno and Indirect-Reno is only due to reduction of reaction delay and presence of local retransmission, which results in isolation between wired and wireless links. For losses lower than the 'knee' point, the amount of which depends on the protocol being used over wireless part, local retransmission brought by Split-Connection completely hides losses from End-to-End throughput to be seen.

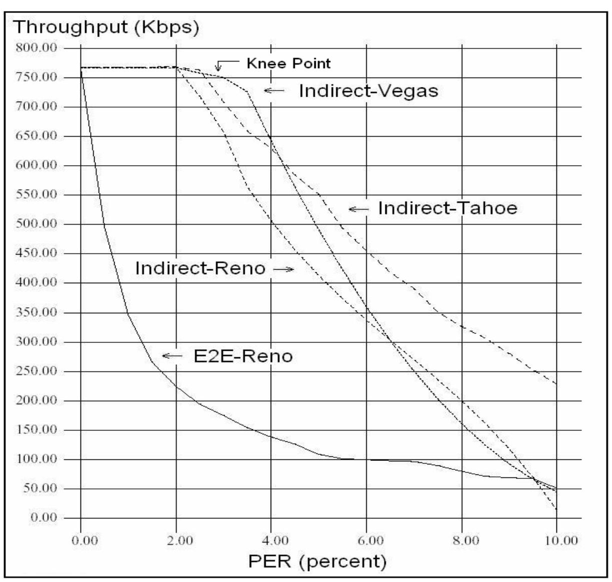

Fig. 6. Performance improvement using Split-Connection

It is also very surprising, though it might be predicted form [7], that Tahoe has a better behavior than Reno, both a better knee point and better overal performance especially at high loss rates. Indirect-Vegas has a better knee point, but its performance degrades more rapidly at high error rates.

Significance of using a TCP with congestion-corruption diffrentiation ability in a local retransmission scheme can be seen from figure 7. In this figure E2E-Reno, Indirect-Westwood and Indirect-Westwood+NR are compared to each other. The difference between E2E-Reno and Indirect-Westwood is due to Westwood's ability to solve the mentioned problem, while the difference between Indirect-Westwood and Indirect-Westwood+NR is due to addition of the NewReno feature. Furthermore, WestwoodNR has a better knee point.

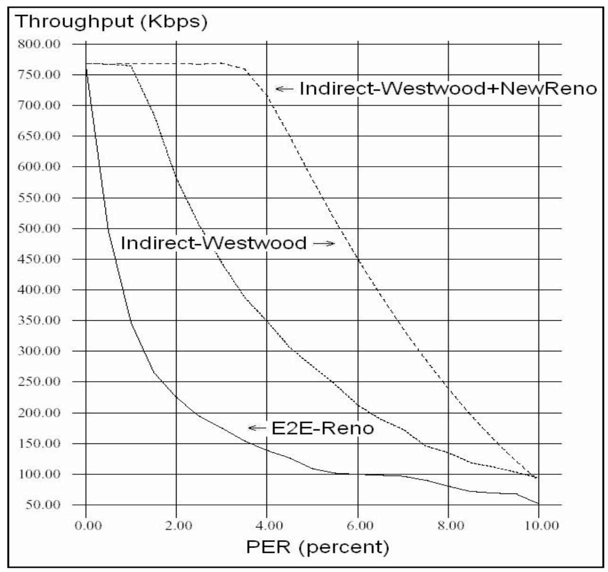

Fig. 7. Performance improvement using Split-Connection

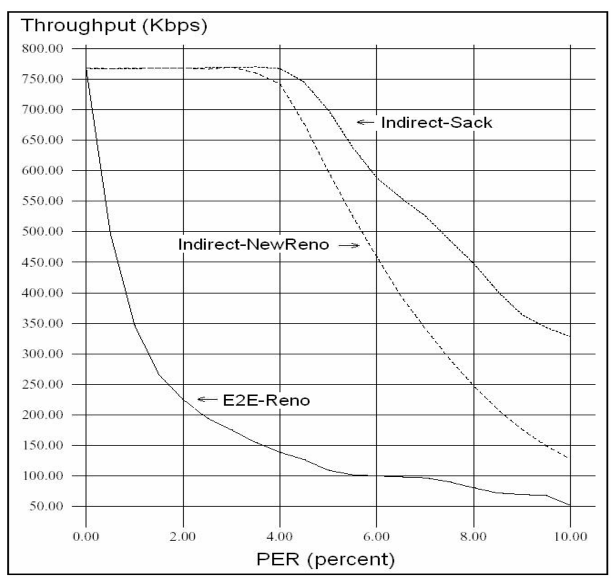

Fig. 8. Performance improvement using Split-Connection

In figure 8, Indirect-Sack and Indirect-Newreno are compared to E2E-Reno. Both of them have a very good knee point, while Sack has better overal performance because of sender explicitly being aware of the packets that need retransmission.

V. CONCLUSION

We conclude that in situations such as the one, which is being considered here, i.e. a long wired and a short wireless path, reaction delay is the most important cause of performance degradation and must be cured by a local retransmission scheme, wheather using Split-Connection method or by a Link-Layer solution. Split-Connectoin is more attractive because it enables us to use aproprate TCP protocol over wireless part without need for any End-to-End modification.Of two remaining causes of performance degradation over wireless link, inability to handle multiple loss in a window is obviously more critical esp. at high loss rates. Tahoe has a more stable behavior than Reno in these situations, in perfect agreement with [7]. Furthermore, TCP Sack is the best wireless protocol among those, which are considered here. The results are briefly shown in figure 9.

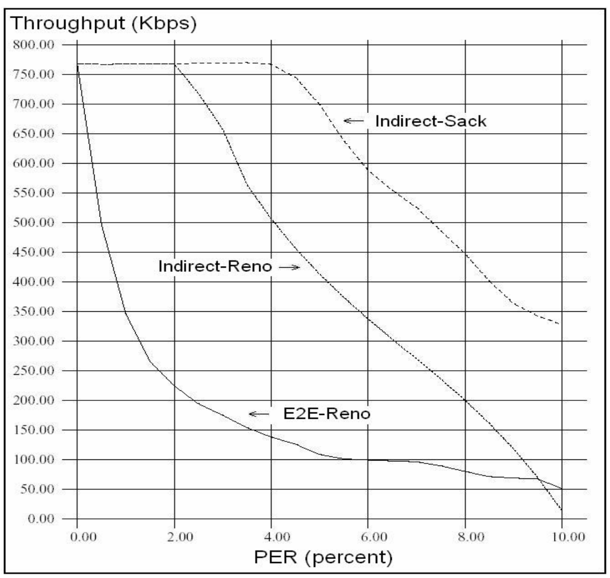

Fig. 9. Split-Connection Performance Improvement

REFRENCES

[1] J. B. Postel. "Transmission Control Protocol". RFC, Information Sciences Institute, September 1981. RFC-793.

[2] V. Jacobson. "Congestion Avoidance and Control". In Proc. ACM Sigcomm 88, August 1998

[3] V. Jacobson, "Modified TCP congestion avoidance algorithm", end2end-interest mailing list, April. 30, 1990

[4] M. Allman, V. Paxson, W. Stevens, *TCP Congestion Control*, RFC, 2001, RFC-2581

[5] T. V. Lakshman, U. Madhow. "The Performance of TCP/IP for Networks with High Bandwidth-Delay Products and Random Loss", IEEE/ACM Trans. on Networking, 5(3):336-350, 1997.

[6] H. M. Chaskar, T. V. Lakshman, U. Madhow, "TCP over Wireless with Link Level Error Control: Analysis and Design Methodology", IEEE/ACM Trans. on Networking, Vol. 7, No. 4, Oct. 1999

[7] K. Fall, S. Floyd, "Simulation-based Comparisons of Tahoe, Reno and Sack TCP". Comp. Comm. Rev., 1996.

[8] Mathis M., Mahdavi J., Floyd S., and Romanow, "A TCP Selective Acknowledgement Options", 1996. RFC-2018.

[9] S. Floyd, T. Henderson, "The NewReno Modification to TCP's Fast Recovery Algorithm", RFC, 1999, RFC-2582.

[10] L. S. Brakmo, S. W. O'Malley, L. L. Peterson, "TCP Vegas: New techniques for congestion detection and avoidance", Proc. ACM Sigcomm, Aug. 1994

[11] M. Mathis, J. Semke, J. Mahdavi, "The Macroscopic behavior of TCP Congestion Avoidance Algorithm", ACM Sigcomm, vol 27, number 3, July 1997.

[12] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, R. H. Katz, "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links". IEEE/ACM Trans. on Networking, 5(6):756-769, 1997.

[13] H. Balakrishnan and R. H. Katz, "Explicit loss notification and wireless web performance," IEEE Global Telecomm. Conference (Globecom'98), Mini Conference, Sydney, Australia, Novermber 1998.

[14] K.K. Ramakrishnan, S. Floyd, A proposal to add Explicit Congestion Notification (ECN) to IP, RFC 2481, Jan. 1999.

[15] S. Floyd, K. Fall, "Promoting the Use of End-to-End Congestion Control in the Internet", IEEE/ACM Trans. on Networking Aug 1999.

[16] Q. Ni, Th. Turletti, W. Fu, "Simulation-based Analysis of TCP Behavior over Hybrid Wireless & Wired Networks", Proc. WWIC 2002, June 2002.

[17] M. Gerla, M. Y. Sanadidi, R. Wang, A. Zanella, C. Casetti, S. Mascolo, "TCP Westwood: Congestion Window Control Using Bandwidth Estimation", proc. of IEEE Globecom 2001, Volume: 3, pp 1698-1702, Nov. 2001.

[18] A-F. Canton, T. Chaed, "End-to-End Reliability in UMTS: TCP over ARQ", proc. of Globecomm 2001

[19] H. Koja, T. Ikenaga, Y. Hori, Y. Oie, "Out-of-Sequence in Packet Arrivals due to Layer 2 ARQ and Its Impact on TCP Performance in W-CDMA Networks". Symposium on Applications and the Internet, Jan. 2003.

[20] H.Balakrishnan, S. Seshan, E. Amir, R. Katz, "Improving TCP/IP Performance over Wireless Networks", Proc. ACM Conf. on Mobile Computing and Networking, Nov 95.

[21] K. Ratnam, I. Mata, "Effect of Local Retransmission at Wireless Access Points on the Round Trip Time Estimation of TCP", 31th Annaual Simulation Symp. April 1998.

[22] Y. Bai, A. T. Ogielski, G. Wu, "Interactions of TCP and Radio Link ARQ Protocol", Proc. of IEEE VTC, pp. 1710-1714. 1999.

[23] A. Bakre and B.R. Badrinath, "I-TCP: Indirect TCP for Mobile Hosts", Proc. of ICDCS, pp. 136–143, 1995.

[24] V. Chirm, V. C. M. Leung, "End-to-End Acknowledgments for Indirect TCP over Wireless Internetworks". Proc. IEEE PacRim'97, pp. 774-777.

[25] K. Brown and S. Singh. "M-TCP: TCP for Mobile Cellular Networks", ACM/SIGCOMM Computer Comm. Review, 27 No.5, pp.19–43, Oct. 1997.

[26] K. Ratnam and I. Matta."WTCP*:* An efficient transmission control protocol for networks with wireless links". Proc. Of 3th IEEE Symp. on Comp. and Comm. (ISCC '98), 1998

[27] V. Paxson, “Measurement and analysis of end-to-end internet dynamics”, Ph.D. dissertation, University of California, Berkeley, 1997.
[28] Y. Komijany, A. Hooshmand, "A New Modified Split-Connection Approach for Improving TCP Performance over Wireless Networks", is going to be submitted to IEE Journal on Communication.
[29] Berkeley NS Simulator, http://www.isi.edu/nsnam/ns