Wireless TCP Performance with Link Layer FEC/ARQ*

A. Chockalingam[†], **Michele Zorzi**[‡] and **Velio Tralli**[‡]

[†]Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore-560012, INDIA
fax: +91-80-334-7991 – e-mail: achockal@ece.iisc.ernet.in

[‡]Dipartimento di Ingegneria, Università di Ferrara, via Saragat 1, 44100 Ferrara, ITALY
fax: +39-0532-768602 – e-mail: zorzi@ing.unife.it, vtralli@deis.unibo.it

Abstract— **In this paper, we investigate the performance of TCP Tahoe and NewReno on wireless links with a FEC/ARQ protocol at the link layer. The effect of link layer (LL) parameters like the LL packet size, number of LL transmission attempts, and FEC code rate on the TCP throughput is evaluated. It is shown that the retransmissions at the link layer must persist long enough to outlast the average bad state duration of the Rayleigh fading process in order to achieve performance better than TCP without an ARQ at the link layer. It is further shown that with a suitable link layer in place, TCP Tahoe offers as high a throughput as offered by other enhanced versions like NewReno in the considered wireless environment.**

I. INTRODUCTION

Wireless data applications like e-mail/web browsing, mobile computing, etc., are gaining increased attention due to rapid advances in the areas of wireless communications and the Internet. Transmission Control Protocol (TCP) is an end-to-end transport protocol in the Internet Protocol (IP) suite, which is widely used in popular applications like telnet, ftp and http [1]. TCP guarantees reliable, in-sequence delivery of packets. TCP performance gets severely affected when used on channels which are typically characterized by high error rates (e.g., wireless channels) [2].

Tahoe and Reno versions of TCP are commonly used implementations. [1]. Several other enhancements to TCP, including New Reno, SACK, and Vegas, have been proposed to improve TCP performance [3], [4]. Such enhanced versions have been shown to provide better performance than Tahoe in wireline networks [4]. However, in [5], it has been shown that TCP enhancements like NewReno and Vegas do not offer significant improvement compared to Tahoe on slowly fading, low bandwidth-delay product wireless channels. There it was suggested that a clever design of the lower layers where memory in the packet error process (naturally present on wireless links because of the fading behavior) is preserved could be more effective than the development/use of more complex TCP versions.

TCP performance with a link layer (LL) automatic repeat request (ARQ) protocol has been investigated through analysis and experimentation, respectively, in [6] and [7]. However, the above studies did not consider forward error correction (FEC) and TCP versions other than Tahoe. In direct sequence code division multiple access (DS-CDMA) cellular data services specification (IS-99 standard), a radio link protocol with a negative acknowledgement based selective repeat scheme is used to reduce the frame error rate seen by the TCP layer [8]. In this paper, we investigate the performance of TCP Tahoe and NewReno on wireless links with a FEC/ARQ protocol at the link layer (LL), considering error correction codes of different code rates. The effect of link layer (LL) parameters like the LL packet size, number of LL transmission attempts, and FEC code rate on the TCP throughput is evaluated.

The rest of the paper is organized as follows. In Section 2, the TCP/LL protocol stack is described. The system model and the channel model considered are presented in Section 3. Section 4 provides the simulation results comparing the performance of TCP Tahoe and NewReno at different parameter settings. Conclusions and areas of future research are provided in Section 5.

II. TCP/LL PROTOCOL STACK

TCP works well on low error rate links where congestion is the primary cause of packet loss. On wireless links, the error rate seen by the TCP layer can be improved by the use of suitable FEC and ARQ at the link layer. The LL ARQ error recovery machanism is introduced below the TCP layer and above the physical layer. Figure 1 shows a typical TCP/LL protocol stack between the base station interworking function (IWF) and the mobile terminal. In between the TCP layer and the link layer, there can be an IP (internet protocol) layer and a PPP (point-to-point protocol) layer. The base station IWF assigns the mobile terminal a temporary IP address upon call establishment. This IP address is unique and valid for the duration of the data call. The PPP layer is used for initial call establishment and to negotiate initial optional link capabilities. Since the bulk throughput performance of the stack during data transfer phase mainly depends on the dynamics of window adaptation and error recovery procedures at the TCP and link layers, we focus mainly on TCP and LL.

The versions of TCP that are considered are TCP Tahoe and TCP NewReno [3],[5]. TCP Tahoe is a commonly used implementation. It uses a *fast retransmit* mechanism to recover lost packets. TCP NewReno version, on the other hand, is a proposal to improve the performance of TCP when multiple packet losses are encountered by making use of a *fast recovery* mechanism. Detailed description of these two and other proposed versions of TCP can be found in [3]. The TCP

*This work was partially supported by the Center for Wireless Communications at UC San Diego. M. Zorzi was with CWC/UCSD. V. Tralli was with CNR/CSITE, Bologna, Italy.

0-7803-5284-X/99/$10.00 © 1999 IEEE.

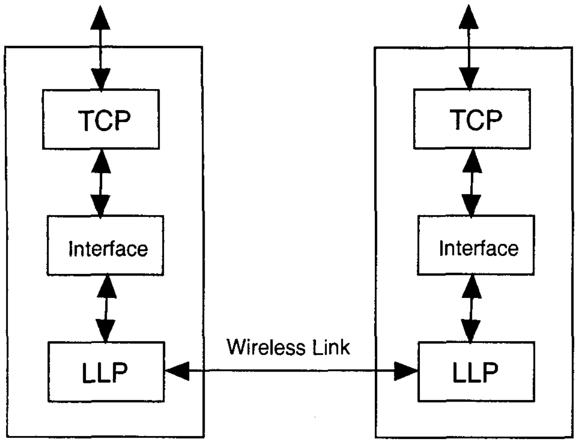

Fig. 1. TCP/LL Protocol Stack

layer receives data from the application and assembles them into TCP packets and passes them to the LL. The maximum TCP packet size is typically in the range of 500 to 2000 bytes.

The link layer segments the TCP packet in to several LL packets. At the LL, an ARQ mechanism performs partial error recovery through a limited number of LL retransmissions in case of LL packet error. The LL ARQ is typically a NAK (negative acknowledgement) based selective repeat scheme. That is, the receiver does not acknowledge correct LL packets; it only requests the retransmission of LL packets that were not received correctly. In the event of LL failure due to excessive LL packet errors, control is passed on to the TCP layer. Unrecovered errors at the LL will be recovered by the TCP layer. The ARQ mechanisms at the TCP and LL layers thus interact with each other and affect the overall system performance depending on channel conditions.

Forwrad error correction (FEC) coding can be applied by adding partity bits to the data bits of the LL packets which can significantly improve the successful reception of LL packets. We consider the use of (255,131,18) and (255,87,26) BCH codes for error correction. Note that (255,131,18) BCH code has 255 bits in a codeword comprising of 131 information bits and 124 parity bits (approximately a rate 1/2 code), and can correct up to 18 errors in a codeword of 255 bits. Likewise, the (255,87,26) BCH code is approximately a rate 1/3 code.

III. System Model

In this section, we describe the system model and the fading channel model we consider to evaluate the throughput performance of the TCP/LL stack. Data exchange using TCP involves a connection setup phase, a data transfer phase, and a call tear down phase. In this paper, we are primarily interested in the bulk throughput performance of TCP, for example, when a large data file is transferred from the base station to a mobile terminal. Consequently, we model only the data transfer phase of the protocol which dominates in the resulting performance. We consider a wireless data link operating at 1.5 Mbps transmission rate. The TCP packet is N_T data bytes long and the LL packet is N_L data bytes long. Thus each TCP packet is segmented into $\lceil \frac{N_T}{N_L} \rceil$ LL packets. The number of retransmission attempts allowed at the LL for recovery of lost LL packets is n_l. The bandwidth-delay product of the link is assumed to be negligibly small. In other words, the LL NAKs and TCP ACKs from the mobile receiver arrive instantaneously at the base station IWF. Also the NAK/ACK packets are assumed to arrive error-free. These assumptions are reasonable because in wireless local environments the propagation delays are small, and the NAK/ACK packets can be relatively smaller in size than data packets (40 bytes of ACK vs 500-1500 bytes of Data in TCP). We assume a TCP packet size of 1024 bytes. At 1.5 Mbps rate, this corresponds to a packet transmission time of about 5.5 ms. A minimum timeout (MTO) value of 100 packets (i.e., 550 ms) is used. This value corresponds to about one tick of timeout granularity in a BSD implementation of TCP where the timeout granularity is 500 ms [5]. We do not consider the issue of mobility in the present study. Finally, we model the correlation in the multipath Rayleigh fading process using a first-order Markov model as follows.

A. Fading Channel Model

In [10], the packet success/failure process on a flat Rayleigh fading channel was modeled as the outcome of a comparison of the instantaneous signal-to-noise ratio to a threshold value, SNR_t: if the received SNR is above the threshold, the packet is successfully decoded with probability 1; otherwise, it is lost with probability 1. The model of [10], while allowing analytical expressions for the Markov parameters, does not fully capture two features of the physical layer: (i) the fact that the fading envelope is sometimes not constant throughout the packet duration, and (ii) the effect of coding and packet length. In order to properly study these aspects, a more elaborate model needs to be developed, in which the bit-by-bit evoluation of the fading process is tracked. The resulting complexity makes the problem analytically intractable, and therefore we resorted to simulation, following the Rayleigh fading model proposed by Jakes [9].

The simulated evolution of the fading envelope is sampled once per bit, and a bit error is generated with the correct probability according to the value of the noise power and of the fading envelope itself. In this manner, the fading evolution is mapped into a bit error process. The latter is then mapped into a *packet* error process by appropriately segmenting the error stream into blocks and by accounting for possible use of error correction codes [11].

From the simulation, the average packet error rate and the conditional probability of a packet success given that the previous packet was in error are computed. These two quantities are then used to fully characterize a two-state Markov approximation for the packet error process. This model, proposed in [11], may be seen as a reasonable compromise between accuracy and simplicity: while it may lead to some modeling errors in some cases, it does capture the memory inherently present in the wireless channel. This provides a much better approximation than the memoryless models often used in the literature.

Based on the above discussion, in the following we will assume that the packet error model evolves as a two-state Markov chain. That is, the transition probability matrix of the channel is given by

$$M_c = \begin{pmatrix} p & 1-p \\ 1-q & q \end{pmatrix}, \tag{1}$$

where p and $1-q$ are the probabilities that the j^{th} packet transmission is successful, given that the $(j-1)^{st}$ packet

FEC Code	f_d (Hz)	SNR	P_E	$1-q$	$(1-q)^{-1}$
No FEC	1	10 dB	3.13e-01	1.30e-01	7.68
		20 dB	3.70e-02	1.46e-01	6.83
		30 dB	3.87e-03	1.61e-01	6.21
	10	10 dB	3.17e-01	1.28e-01	7.81
		20 dB	3.69e-02	1.54e-01	6.50
		30 dB	3.77e-03	1.67e-01	6.00
	100	10 dB	3.18e-01	1.39e-01	7.19
		20 dB	3.87e-02	2.50e-01	3.99
		30 dB	4.74e-03	5.16e-01	1.94
(255,87,26) BCH Code	1	10 dB	8.00e-02	6.51e-02	15.35
		20 dB	8.88e-03	6.72e-02	14.89
		30 dB	7.26e-04	8.35e-02	11.98
	10	10 dB	8.15e-02	7.71e-02	12.96
		20 dB	8.61e-03	1.47e-01	6.82
		30 dB	9.76e-04	3.56e-01	2.81
	100	10 dB	9.58e-02	3.48e-01	2.87
		20 dB	1.42e-02	7.07e-01	1.41
		30 dB	2.03e-03	9.40e-01	1.06

Table 1. Markov parameters for $N_L = 32$ data bytes obtained from fading channel simulation

transmission was successful or unsuccessful, respectively. Note that $(1-q)^{-1}$ represents the average length of a burst of packet errors.

For the Rayleigh fading channel, the Markov parameters are computed off-line via the simulation approach just described. Table 1 shows the Markov parameters at different SNRs, with and without FEC, for $N_L = 32$ data bytes. Similarly Markov parameter values for other N_L values and (255,131,18) BCH code were obtained from compuer simulations.

IV. Results

The performance of TCP Tahoe and NewReno are evaluated through simulations. The TCP simulation code is written in C. The Markov parameters p and q for the channel model as a function of average SNR at different Doppler frequencies, LL packet sizes, and FEC code rates are obtained through independent simulations as described in Section 3.A. These channel parameters are then used as inputs to the TCP simulations. The TCP parameters used in this study are: TCP packet size $N_T = 512, 1024$ data bytes, maximum advertized window width $W_{max} = 24$ TCP packets, minimum timeout $MTO = 100$ TCP packets, and fast retransmit threshold $K = 3$. The Doppler frequencies considered are f_d = 1, 10, 100 Hz. Low values of Doppler like 1 Hz and 10 Hz are typical of slow moving users; e.g., pedestrian users. As seen in Table 1, packet errors in such scenarios tend to be bursty due to high correlation in the fading process. Large values of Doppler like 100 Hz are typical of fast moving users like vehicular users. Packet errors are less bursty at high Doppler frequencies. The various LL packet sizes considered are 32, 64, and 128 data bytes. When FEC is employed, additional parity bytes are added to these data bytes. The FEC codes considered here are (255,131,18) and (255,87,26) BCH codes. Since the code rate of these codes are approximately 1/2 and 1/3, respectively, we will refer to them as rate 1/2 and rate 1/3 BCH codes. The number of LL retransmit attempts for LL ARQ considered are $n_l = 1, 8, 16, 128$.

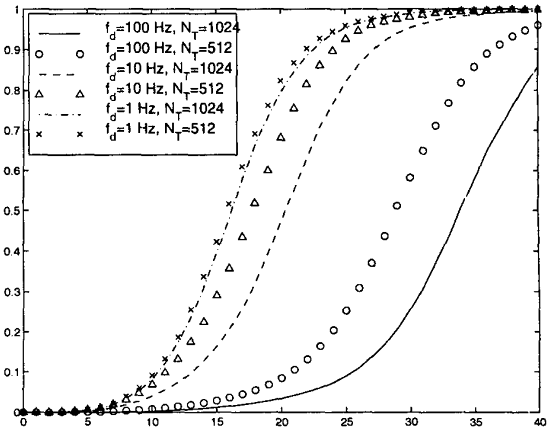

Fig. 2. Effect of TCP packet size and Doppler bandwidth on the performance of TCP Tahoe without LL FEC/ARQ. N_T = 1024, 512 bytes. f_d = 1, 10, 100 Hz.

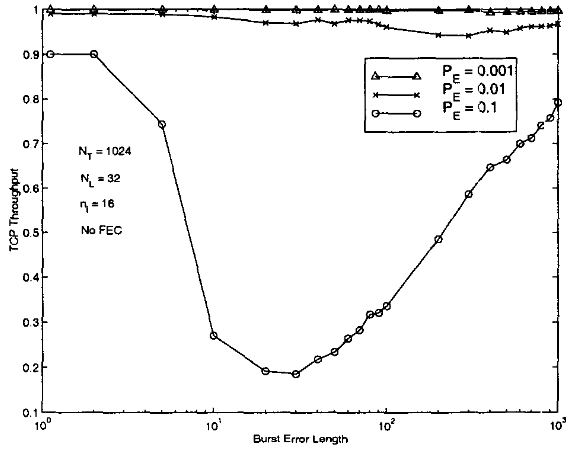

Fig. 3. Performance of TCP Tahoe with LL ARQ as a function of burst error length for different P_E values at n_l = 16. N_T = 1024 bytes. N_L = 32 bytes.

Figure 2 shows the effect of TCP packet size on the throughput performance of TCP Tahoe without LL FEC/ARQ as a function of average SNR in dB [1] at different Doppler frequencies. It can be seen that correlation in packet errors benefit TCP performance. That is, for a given TCP packet size, throughput achieved at $f_d = 1$ Hz is significantly better than at $f_d = 100$ Hz, which is consistent with the observation made in [5]. This is because for a given average error rate, the "error events" are more in the case of $f_d = 100$ Hz than $f_d = 1$ Hz, and this makes the TCP to often shrink its window to 1 and lose in performance. Such performance loss can be minimized if shorter TCP packets are used instead of large TCP packets. This can be observed from Figure 2 where for a given average SNR the performance of TCP is better with short TCP packets (512 data bytes) than large TCP packets (1024 data bytes) on channels with low correlation ($f_d = 100$ Hz). However, on highly correlated channels ($f_d = 1$ Hz) the

[1] Note that in Figures 2, 5, 6, 7 and 8 x-axis represents average SNR in dB and y-axis represents TCP throughput.

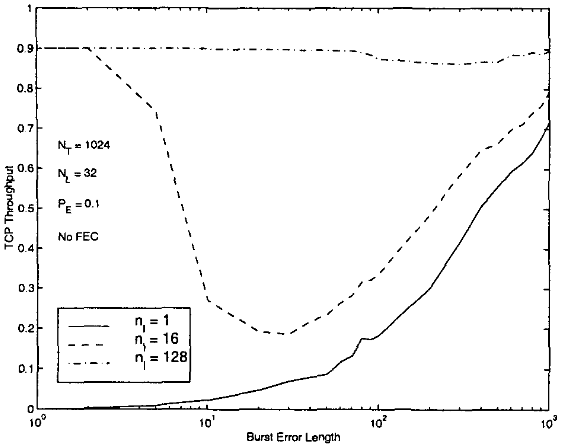

Fig. 4. Performance of TCP Tahoe with LL ARQ as a function of burst error lenth for different n_l values at $n_l = 16$. $N_T = 1024$ bytes. $N_L = 32$ bytes.

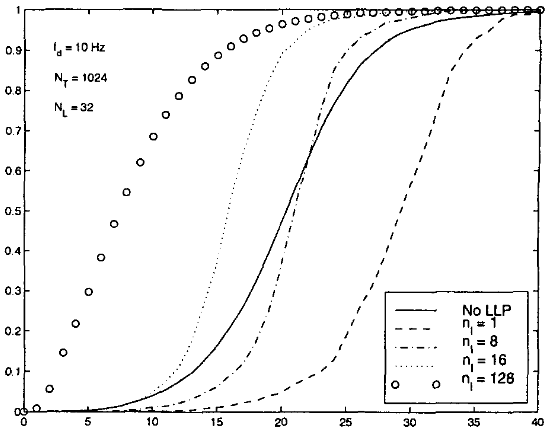

Fig. 5. Comparison of TCP Tahoe performance as a function of average SNR with and without LL ARQ for different n_l values at $f_d = 10$ Hz. $N_T = 1024$ bytes. $N_L = 32$ bytes.

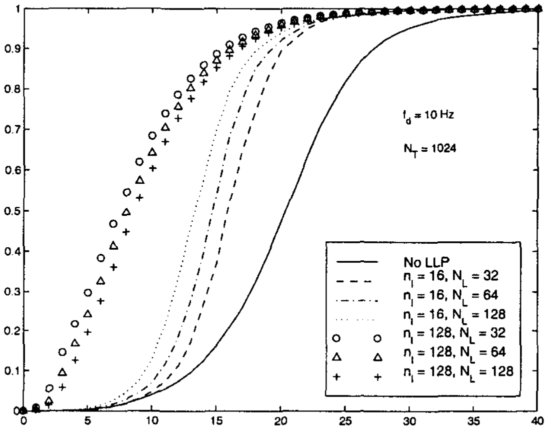

Fig. 6. Effect of LL packet size (N_L) and number of LL transmission attempts (n_l) on TCP Tahoe performance at $f_d = 10$ Hz. $N_T = 1024$ bytes. $N_L = 32, 64, 128$ bytes.

effect of TCP packet size on the throughput performance is negligible.

Figure 3 shows the effect of the LL ARQ without FEC on the TCP Tahoe throughput performance as a function of the LL burst error length when $N_T = 1024$ data bytes. The LL packet size is $N_L = 32$ data bytes and the number of LL retransmits $n_l = 16$. The curves are parameterized by different average packet error rates, $P_E = 0.001, 0.01, 0.1$. It is seen than the throughput performance remains almost independent of the LL burst error length at very low error rates (e.g., $P_E = 0.001, 0.01$). This is expected because the error rate itself is very low and the performance is dominated primarily by the large amount of packets successfully delivered during the good state of the channel. However, the burst error length significantly influences the throughput performance when the error rates are relatively high ($P_E = 0.1$). When the burst error length is small ($<< n_l$), the LL ARQ with $n_l = 16$ retransmission attempts succeeds in LL error recovery with a high probability resulting in good TCP throughput. However, as the error burst length is increased, the performance degrades due to the inability of the LL ARQ to recover within n_l retransmission attempts and the TCP will exhaust its window of data sooner. This explains why the performance is worst when burst error length is about 20. Burst error lengths in excess of 20 result in increasingly better performance. This performance reversal occurs because packet "error events" seen at the TCP layer will be less for large LL error bursts (as explained in Figure 2 for $f_d = 1$ Hz) which is beneficial. TCP throughput can be improved by increasing the number of retransmission attempts n_l. This is evident from Figure 4 where the TCP throughput is plotted for $n_l = 1$, 16 and 128 when $P_E = 0.1$. It is seen that the throughput performance is very poor for $n_l = 1$ whereas for $n_l = 128$ throughputs as high as 0.9 are consistently achieved. This illustrates that the LL ARQ must persist long enough to outlast the bad state of the channel for achieving good throughputs at the TCP layer.

In Figure 5 the performance of TCP Tahoe with and without LL ARQ is compared for $N_T = 1024$ data bytes, $N_T = 32$ data bytes, and $f_d = 10$ Hz. It is seen that when $n_l = 1$, the performance of TCP with ARQ is worse than TCP without ARQ. This is because, congestion window permitting, TCP will send the next TCP packet as soon as the LL"s failure to recover a LL packet is intimated to the TCP layer. This exhausts the usable window too soon resulting in many fast retransmits and timeouts. Increasing n_l results in increasingly better throughput. For the chosen Doppler frequency and packet sizes, it is seen that a n_l value of 16 is required for TCP with LL ARQ to perform better than TCP without LL ARQ, and $n_l = 128$ results in very good performance. At an average SNR value of 10 dB, $n_l = 128$ results in a throughput of about 0.7 whereas $n_l \leq 16$ result in throughputs much lower than 0.1. Figure 6 shows the effect of LL packet size ($N_L = 32, 64, 128$ data bytes) on the TCP throughput when $n_l = 16$ and 128 for $N_T = 1024$ data bytes and $f_d = 10$ Hz. When $n_l = 16$, increasing the LL packet size (from 32 to 64 and 128 data bytes) improves the throughput. However with n_l values as high as 128, there is very little difference in throughput when different LL packet sizes are used.

The effect of LL ARQ with FEC at different Doppler fre-

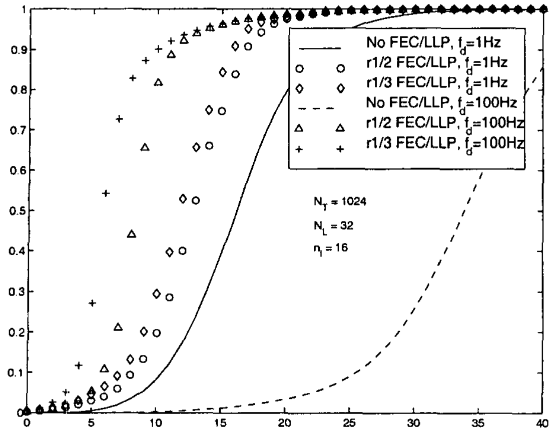

Fig. 7. TCP Tahoe performance with LL FEC/ARQ at different Doppler and FEC code rate. $N_T = 1024$ bytes. $N_L = 32$ bytes. $n_l = 16$. Rate 1/2 and rate 1/3 BCH code. $f_d = 1, 100$ Hz.

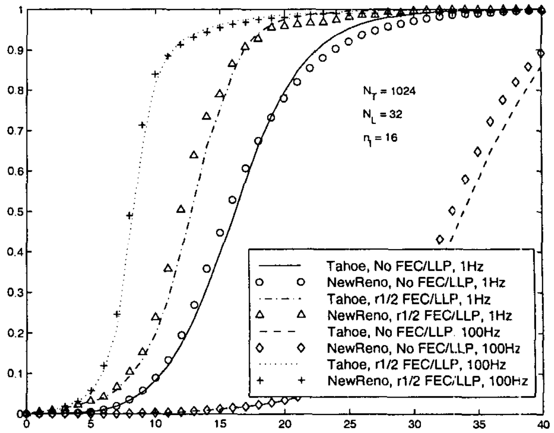

Fig. 8. Comparison of TCP Tahoe and NewReno performance with Rate 1/2 BCH FEC/ARQ LL at different Doppler. $N_T = 1024$ bytes. $N_L = 32$ bytes. $n_l = 16$. $f_d = 1, 100$ Hz.

quencies ($f_d = 1, 100$ Hz) is illustrated for $N_T = 1024$ data bytes, $N_L = 32$ data bytes, and $n_l = 16$. Performances with Rate 1/2 and 1/3 BCH codes are plotted and compared. FEC is seen to improve the throughput as expected. However, the performance improvement with rate 1/3 FEC relative to rate 1/2 FEC is not very much. This is partly because most of the throughput improvement comes from LL ARQ than FEC. This observation was made by comparing the achieved throughputs with only LL ARQ and with FEC/ARQ. However, the energy efficiency achieved in both cases need to be looked into because, as we have shown earlier in the case of access protocols [12], maximizing the throughput alone does not guarantee good energy efficiency which influences the battery life in portable communication devices. Preference of FEC or ARQ and LL parameter choice from a energy efficiency stand point can be an useful extension to this wireless TCP study.

Finally, a comparison between the performance of TCP Tahoe and TCP NewReno with LL FEC/ARQ is made in Figure 8 (rate 1/2 FEC) for $N_T = 1024$ data bytes, $N_L = 32$ bytes, $n_l = 16$, and $f_d = 1, 100$ Hz. These plots indicate that with FEC/ARQ at the link layer, the performance of Tahoe is as good as that of NewReno. In other words, in the considered wireless environment with a suitable LL FEC/ARQ scheme, TCP Tahoe is a good choice and there seems to be no compelling need to go in for complex TCP variants like TCP NewReno.

V. Conclusions

We studied the throughput performance of TCP Tahoe and TCP NewReno on wireless fading links with a FEC/ARQ protocol at the link layer. The effect of LL parameters like LL packet size, number of LL transmission attempts, and FEC code rates at different Doppler frequencies were evaluated through computer simulations. It was shown that retransmissions at the LL must persist long enough to outlast the bad state of the channel in order to achieve performance better than TCP without LL ARQ. Also, between LL ARQ and FEC, it was the LL ARQ that gave the most improvement in throughput. In this context, preference of FEC or ARQ and LL parameter choice from an energy efficiency stand point can be an useful extension to this wireless TCP study. Finally in the considered wireless environment with a suitable LL FEC/ARQ scheme, TCP Tahoe is a good choice and there seems to be no compelling need to go in for complex TCP variants like TCP NewReno.

References

[1] W. R. Stevens, *TCP/IP Illustrated, Volume 1*, Addison Wesley, 1994.

[2] T. V. Lakshman and U. Madhow, "The performance of TCP/IP for networks with high bandwidth-delay products and random loss," *IEEE/ACM Trans. on Networking*, pp. 336–350, June 1997.

[3] K. Fall and S. Floyd, "Comparisons of Tahoe, Reno, and Sack TCP," manuscript, ftp://ftp.ee.lbl.gov, March 1996.

[4] O. Ait-Hellal and E. Altman, "Analysis of TCP Vegas and TCP Reno," *Proc. IEEE ICC'97*, 1997.

[5] A. Chockalingam, M. Zorzi, and R. R. Rao, "Performance of TCP on Wireless Fading Links with Memory," *Proc. IEEE ICC'98*, vol. 2 , pp. 595-600, Atlanta, June 1998.

[6] A. Kumar and J. Holtzman, "Comparative performance analysis of versions of TCP in a local network with a lossy link, Part II: Rayleigh fading mobile radio link," *Tech. Rep. WINLAB-TR-133*, Nov. 1996.

[7] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R. H. Katz, "A comparison of mechanisms for improving TCP performance over wireless links," *ACM/IEEE Trans. on Networking*, December 1997.

[8] A. Chockalingam and G. Bao, "Performance of TCP/RLP Protocol Stack on Correlated Fading DS-CDMA Wireless Links," Proc. IEEE VTC'98, Ottawa, May 1998

[9] W. C. Jakes, Jr., *Microwave mobile communications*, New York: John Wiley & Sons, 1974.

[10] M. Zorzi, R. R. Rao, and L. B. Milstein, "On the accuracy of a first-order Markov Model for data transmission on fading channels," *Proc. IEEE ICUPC'95*, pp. 211-215, November 1995.

[11] M. Zorzi, R. R. Rao, and L. B. Milstein, "Error statistics in data transmission over fading channels," *IEEE Trans. Comm.*, vol. 46, Nov. 1998.

[12] A. Chockalingam and M. Zorzi, "Energy Efficiency of Media Access Protocols for Mobile Data Networks," *IEEE Trans. Comm.*, vol. 46, pp. 1418–21, Nov. 1998.