TCP Optimization through FEC, ARQ and Transmission Power Tradeoffs*

Dhiman Barman[1], Ibrahim Matta[1], Eitan Altman[2], and Rachid El Azouzi[3]

[1] {dhiman,matta}@cs.bu.edu, 111 Cummington St., Dept. of Computer Science, Boston University, Boston, MA 02215, USA
[2] {altman@sophia.inria.fr}, INRIA, B.P. 93, 2004 Route des Lucioles, 06902, Sophia-Antipolis Cedex, France
[3] {Rachid.Elazouzi@lia.univ-avignon.fr}, LIA/CERI-Université d'avignon Agroparc 339, chemin des Meinajaries BP 1228, 84911 Avignon, France

Abstract. TCP performance degrades when end-to-end connections extend over wireless connections — links which are characterized by high bit error rate and intermittent connectivity. Such link characteristics can significantly degrade TCP performance as the TCP sender assumes wireless losses to be congestion losses resulting in unnecessary congestion control actions. Link errors can be reduced by increasing transmission power, code redundancy (FEC) or number of retransmissions (ARQ). But increasing power costs resources, increasing code redundancy reduces available channel bandwidth and increasing persistency increases end-to-end delay. The paper proposes a TCP optimization through proper tuning of power management, FEC and ARQ in wireless environments (WLAN and WWAN). In particular, we conduct analytical and numerical analysis taking into account the three aforementioned factors, and evaluate TCP (and "wireless-aware" TCP) performance under different settings. Our results show that increasing power, redundancy and/or retransmission levels always improves TCP performance by reducing link-layer losses. However, such improvements are often associated with cost and arbitrary improvement cannot be realized without paying a lot in return. It is therefore important to consider some kind of net utility function that should be optimized, thus maximizing throughput at the least possible cost.

1 Introduction

When TCP connections extend over wireless links, many factors such as interference, multipath fading, user mobility and atmospheric conditions may cause errors resulting in frame losses over the wireless links. It's not fair to expect TCP to perform well over such links because it was designed to perform mainly over wired networks where there is almost no channel or random packet losses. Nor can we replace TCP overnight because 90% of Internet applications exchange traffic through this protocol.

* This work was supported in part by NSF grants ANI-0095988, ANI-9986397, EIA-0202067 and ITR ANI-0205294, and by grants from Sprint Labs and Motorola Labs.

Various strategies have been proposed to combat this problem which can be classified along the following strategies: split-connection, proxy-based, link-layer error control, and end-to-end.

In the split-connection approach such as [3], the wireless link is hidden from the TCP sender by terminating the connection at the base station. A more reliable connection is established over the wireless link. But this approach violates the end-to-end semantics of TCP and there exists overhead of maintaining state information at the base station.

In proxy-based approaches such as Snoop [4], a base station monitoring a wireless link tries to do local recovery for wireless losses. In ELN-based versions, a TCP-aware base station monitoring the packets going over the wireless link marks the acknowledgment packets with ELN (Explicit Loss Notification) bit if packet loss is due to wireless loss and not congestion. TCP performance can thus be improved at the cost of extra state maintenance at the base station.

Link-layer approches such as forward error correction (FEC) and automatic repeat request (ARQ) attempt to hide channel losses from the TCP sender by cleaning wireless links of the wireless losses. Link-layer solutions are appealing as they do not incur the overhead associated with TCP-awareness. Introduction of FEC consumes wireless resources but at the same time reduces the link loss rate. On the other hand, link loss rate can be reduced by increasing the transmission power but this means higher transmission costs. Link losses can also be alleviated by using retransmission mechanisms such as ARQ schemes but they increase end-to-end delay thus reducing end-to-end throughput. In [5], the authors analyzed the tradeoff between the bandwidth utilized by FEC and goodput gained by a TCP session. They proposed an algorithm to compute the optimal code that maximizes TCP goodput. In our model, wireless channels are more vulnerable to bit error rates than erasures (i.e., frames getting lost in the wireless channel).

In the end-to-end approach, attempts are made to improve TCP performance at end hosts without any aid from the network (e.g. from base stations). Such proposals include TCP-SACK and TCP-Westwood [7].

Two main conclusions can be drawn from the previous schemes, either we should hide the wireless link from the TCP sender or we should make the TCP sender aware of the occurrence of wireless loss. In this paper, we study the joint effects of FEC, ARQ-SR (Selective Repeat) and power management on TCP performance with and without ELN-type feedback. We evaluate the effects of wireless link losses and the corrective measures on the end-to-end behavior of TCP using different models and settings. In a recent work [12], the authors studied a combined effect of power and FEC on TCP performance and they provided analytical expressions for optimal values of power and redundancy.

Our contribution in this paper is four fold. Firstly, we consider all the significant parameters (transmission power, FEC and ARQ) for evaluating TCP performance over wired/wireless links. Secondly, we consider two different models of link-layer mechanisms and show the effects on the end-to-end measured RTT of TCP. Thirdly, we consider end-to-end TCP performance under settings in which there are both congestion and wireless losses. Finally, we apply the analysis on an informed model of TCP (which does not take congestion control actions on wireless losses) and show the combined effects of the aforementioned parameters on TCP. Our analytical and numerical evaluations show that increasing power, redundancy and/or retransmission levels always improves TCP performance by reducing link-layer losses. However, such improvements are often associated with cost and arbitrary improvement cannot be realized without paying a lot in return. It is therefore important to consider some kind of net utility function that should be optimized, thus maximizing throughput at the least possible cost.

The remainder of the paper is organized as follows. In Section 2, we define the model of TCP in a hybrid wired/wireless environment. We describe the model for FEC (Section 2.1), ARQ (Section 2.2), and power management (Section 2.3). In Section 3 we derive TCP throughput in the context of our model. We also show expected RTT computation under different link-layer models (Section 3.2). In Section 4, we consider an end-to-end model for informed TCP throughput in which the TCP sender knows about wireless losses. We present our numerical evaluation in Section 5. Finally we conclude and describe future avenues for research in Section 6.

2 Model

2.1 Forward Error Correction

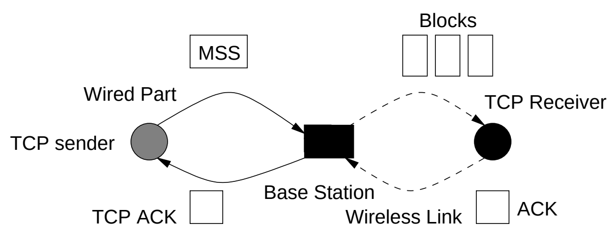

Fig. 1. A model for a hybrid wired/wireless network. A TCP connection extends over a wireless link through a base station. MSS is the packet size on the wired part and a packet is chunked into smaller blocks on the wireless link

In our model (as represented in Figure 1), we assume that the TCP packets of size MSS traverse both wired and wireless links. We consider that the base station has large enough buffer. This was found to prevent unfairness among TCP flows caused by buffer size smaller than advertised windows [15]. Even if buffer overflow occurs at the base station, our model effectively treats such losses as congestion. It is assumed that packets are acknowledged and acknowledgments traverse through the same base station [9].

We consider a wireless link where data are transmitted as blocks of length K each. FEC encodes each one of these blocks into a codeword of length N, with $N > K$. The *redundancy ratio,* x , is defined as the ratio of the amount of redundancy due to FEC, $(N - K)$, to the block length, K, i.e.,

$$x = (N - K)/K \tag{1}$$

We denote by B the bandwidth of the wireless channel, and by D its one-way propagation delay. Each MSS sized segment is divided into $X = \frac{MSS}{K}$ many K-sized blocks. A TCP segment is decoded properly if all X blocks are received uncorrupted and decoded by the receiver. The link layer implements ARQ-SR/FEC error control scheme. We assume strong CRC code so that the probability of not detecting a corrupted block is practically zero. In contrast, only a subset of bit errors can be corrected by FEC [13]. We also assume that the feedback (ACK/NACK) messages are well protected or the probability of acknowledgement losses is negligible so that no retransmissions are needed for these messages.

The underlying hybrid ARQ-SR/FEC mechanism is characterized by (N, K, δ, e_c, e_d), where N is the number of bits in a code block, K is the number of information bits in a code block, δ is the maximum number of allowable retransmissions, e_c is the maximum number of correctable bits in a code block, and e_d is the maximum number of corrupted bits which can be detected. In here, a block corresponds to a fixed-size link-layer packet. Note that N depends on K, e_c and e_d. In general, an (N, K) code with minimum distance d_{min} can detect e_d errors and correct e_c errors, where $e_d + e_c \leq d_{min} - 1$ and $e_c \leq e_d$ [13]. The coding gain is given by $G_{coding} = \frac{K}{N} \cdot d_{min}$. For example, if Reed-Solomon encoding is used, then $G_{coding} = \frac{1}{1+x} \cdot (N - K + 1) = \frac{K \cdot x+1}{1+x}$ using Eq. (1).

2.2 Automatic Repeat Request (Selective Request)

If FEC does not succeed to decode one block, the link-level error mechanism turns to ARQ-SR for the retransmission of the block. The retransmission will be attempted a maximum number of times, referred to as the persistency of ARQ-SR and denoted by $\delta, \delta = 0, 1, 2, ...$ $\delta = 0$ means that there are no retransmissions and that ARQ-SR is disabled. If after δ trials the frame does not get through the wireless link, ARQ-SR assumes that the frame cannot be locally recovered, and leaves for TCP the correction of the frame on an end-to-end basis. This is important because if the link layer keeps trying indefinitely, it would increase the end-to-end delay and RTT and may lead to TCP timeout (unless the base station informs the TCP sender about the local recovery process.)

The ARQ-SR receiver at the output of the wireless link acknowledges each block with a positive ACK or a NACK. When a NACK is received at the input of the wireless link, the corresponding block is directly retransmitted, and given priority over all other blocks that have not been transmitted. We have considered two different models of ARQ-SR in computing average end-to-end RTT which we describe in detail in Section 3.2.

2.3 Power Management

Link reliability can be improved by increasing the transmission power. In fact, the bit error probability, p_e, decreases when the ratio (E_b/N_0) increases, where E_b is the received energy per bit and N_0 is the noise power spectral density. Note that the relationship between the bit error probability, and the ratio (E_b/N_0) is a function of the modulation technique. Let y represent the transmission power. The received energy per bit, E_b is equal to $A \cdot y/B$ where A is the attenuation. Increasing the transmission power improves TCP performance but causes greater energy consumption and aggravates the interference with other neighboring communications.

3 The Analytical Framework

3.1 TCP Throughput Evaluation

Different analytical formulas for TCP throughput have been proposed in the literature. The general form can be written as follows:

$$\lambda(x, y, \delta) = \frac{1}{1+x} \cdot f(RTT, P_{Loss}) \tag{2}$$

where $f(\cdot, \cdot)$ is the TCP throughput function which depends on RTT and average packet loss probability. The most commonly used formula for TCP throughput is given by [14]:

$$f(RTT, p) = \frac{1}{RTT} \min\left\{W_{\max}, \frac{1}{\sqrt{\frac{2bp}{3}} + T_0 \min(1, 3\sqrt{\frac{3bp}{8}})p(1+32p^2)}\right\} \tag{3}$$

where $W_{\max}$ is the maximum congestion window size of the TCP sender, b represents the effect of delayed acknowledgements, and T_0 is the TCP retransmission timeout value. RTT is the round trip time which depends on the persistency level of ARQ-SR, δ, the amount of redundancy, x, and packet loss probability, P_{Loss}. P_{Loss} is the probability that a TCP segment is discarded because of link errors in the wireless channel. The probability P_{Loss} can be evaluated as:

$$P_{Loss} = [1 - (1 - P_{Block})^X] \tag{4}$$

where P_{Block} is the block error probability at the output of the decoder. P_{Block} depends on x, δ, and y. We consider Reed-Solomon coding and approximate $P_{Block} \approx [1-(1-p_e)^K]^{\delta+1}$. p_e depends on power and the particular modulation scheme as tabulated in Table 1. In our numerical evaluation we consider three modulation schemes, namely, Gaussian M-ary Shift Keying (GMSK), Differentiated Binary Phase Shift Keying (DBPSK) and Gaussian Frequency Shift Keying (GFSK).[1]

[1] It is to be noted that there is an effective method for dealing with correlated error channels by using interleaved coded data such that the bursty channel is transformed into a channel having independent errors [13].

Table 1. p_e for different modulation schemes, $\epsilon \leq (2^K - 1)[4p_e(1-p_e)]^{d_{min}/2}$, probability of detecting errors in a block is greater than $1-\epsilon$, α is a constant, A is the attenuation, N_0 is the noise spectral density, ΔF is the size of the frequency band, and the factor $\frac{1+Kx}{1+x}$ is the coding gain for Reed-Solomon coding.

Modulation	GMSK	DBPSK	GFSK
p_e	$\frac{1}{2}\mathrm{erfc}\left(\sqrt{\frac{\alpha A y}{N_0 \Delta F}\frac{Kx+1}{1+x}}\right)$	$\frac{1}{2}\exp\left(-\sqrt{\frac{\alpha A y}{N_0 \Delta F}\frac{Kx+1}{1+x}}\right)$	$\frac{1}{2}\exp\left(-\frac{1}{2}\sqrt{\frac{\alpha A y}{N_0 \Delta F}\frac{Kx+1}{1+x}}\right)$

3.2 RTT computation

In this section, we compute the average RTT of a connection under two models of link-layer transmissions. We assume that data blocks are quickly acknowledged by ARQ-SR and sizes of acknowledgements are of negligible size as compared to data blocks. Thus the transmission of a block and reception of its acknowledgement takes $\tau = 2D + \frac{N}{B}$. Let $\delta_i \in \{0, 1, \ldots, \delta\}, i = 1, 2, \ldots, X$ be the number of times we retransmit block i of a TCP packet (recall a TCP packet of size MSS is divided into X blocks each of size K).

Model I: In the first model of expected RTT computation, we assume that the next block is transmitted after the sender receives acknowledgement of the previous block (i.e., stop-and-wait). Then, the round-trip time of a TCP packet can be written as:

$$RTT = T + 2D + \frac{XN}{B} + \sum_{i=0}^{X} \delta_i \tau \tag{5}$$

where T is the round trip time of the wired part of the TCP connection. RTT is a random variable and the randomness comes from random variables δ_i, which are i.i.d. and geometric. In order to capture the effect of the wireless part, we are assuming that delay variability on the wired part is less [8]. We have the following:

$$P\{\delta_i = k\} = \begin{cases} \frac{P_{Block}^k (1-P_{Block})}{1-P_{Block}^{\delta+1}} & 0 \leq k \leq \delta, \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

The expected RTT, $E[RTT]$ is derived in the full version of the paper [6] and is given by $E[RTT] = T + 2D + \frac{XN}{B} + X\tau P_{Block}\left[\frac{1}{1-P_{Block}} - \frac{(\delta+1)P_{Block}^{\delta}}{1-P_{Block}^{\delta+1}}\right]$. We assume that retransmission is done with probability 1.

Model II: In this model, we assume that all the blocks are transmitted back-to-back, then it will lead to a different value of $E[RTT]$ (derived in the full version) given by, $E[RTT] = T + 2D + \frac{XN}{B} + \tau\delta - \frac{1}{(1-P_{Block}^{\delta+1})^X} \int_0^{\frac{XN}{B}+\tau\delta} \prod_{i=1}^{X} \left(1 - P_{Block}^{\lfloor (z - \frac{iN}{B})/\tau \rfloor + 1}\right) dz$. The expression for $E[RTT]$ is difficult to solve analytically, and therefore, we resort to evaluating it numerically.

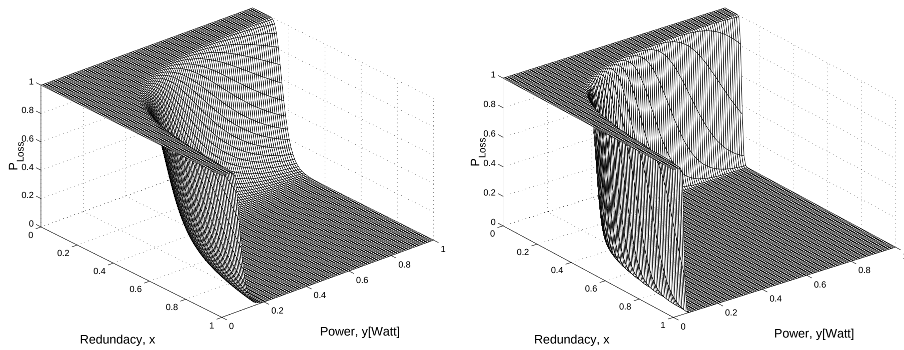

Fig. 2. Behavior of the segment loss probability, P_{Loss}, vs. the redundancy, x, the transmission power, y and the persistency level, $\delta = 0$ (left) and $\delta = 10$ (right)

3.3 Cost Evaluation

Consider a need to transfer S TCP segments, each MSS size segment will be transmitted in a total X of $K(1+x)$ sized codewords over the wireless link, where each codeword could be retransmitted $0 \leq \delta_i \leq \delta$ times. Thus the cost of transfer depends on the transmission power, the amount of redundancy introduced and the level of persistency. We consider two cost terms: a term which takes energy consumption into account, and a term which considers the amount of wireless resources employed. If the redundancy introduced by FEC is x, then $[S \cdot MSS \cdot (1+x)](1+E[\delta_i])$ bits must be transmitted in order to deliver S segments of MSS bits each. Accordingly, the cost of the resources required to complete the transfer is given by $c_{resources} = k_r \cdot S \cdot MSS \cdot (1+x)(1+E[\delta_i])$ where k_r (expressed in bit^{-1}) is a constant which represents the cost of the bandwidth required to transfer a bit. Given that the energy transmitted per bit is given by $\frac{y}{B}$, the energy consumption required to complete the transfer is $[S \cdot MSS \cdot (1+x)](1+E[\delta_i])y/B$. As a result, if k_e (expressed in Joule^{-1}) represents the cost of a unit of energy, then the total cost is $c_{energy} = k_e \cdot S \cdot MSS \cdot (1+x)(1+E[\delta_i]) \cdot y/B$. Accordingly, when the redundancy is x, the energy transmitted per bit is y and the persistency is bounded by δ, the cost of the transfer can be evaluated as:

$$c(x, y, \delta) = c_{energy} + c_{resources} = S \cdot MSS \cdot (1+x)(1+E[\delta_i])(\frac{k_e y}{B} + k_r) \quad (7)$$

The constants k_r and k_e depend on many factors such as user preferences, the type of terminal and the costs of power and communication. There are other cost functions that could be realized based on the same metrics.

3.4 Objective Function Maximization

Let us define the objective function $\gamma(x, y, \delta)$ as follows:

$$\gamma(x, y, \delta) = \lambda(x, y, \delta)/c(x, y, \delta) \quad (8)$$

Now we want to evaluate the values of x, y and δ which maximize the function $\gamma(x, y, \delta)$ given by:

$$\gamma(x, y, \delta) = \frac{f(E[RTT], P_{Loss})}{(1+x)^2 S \cdot MSS \cdot (1 + E[\delta_i])(k_e y/B + k_r)} \tag{9}$$

where Eq. (9) is obtained by substituting Eq. (2) and (7) in Eq. (8).

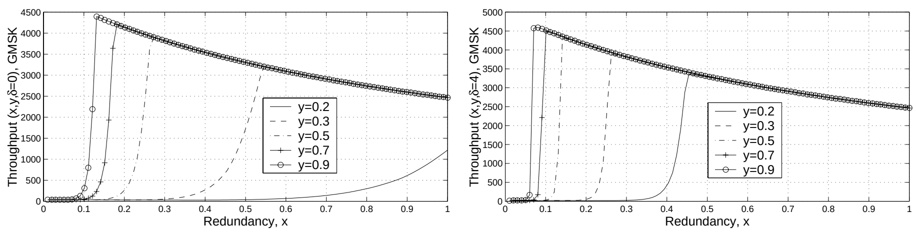

Fig. 3. Behavior of the throughput vs. the redundancy ratio for different values of the transmission power, y and persistency level, $\delta = 0$ (left) and $\delta = 4$ (right) for RTT Model I

4 End-to-end TCP Performance

Using Eq. (3) we may not know the ideal desired behavior of TCP which discriminates congestion losses from channel losses. In this section, we consider a channel error informed TCP throughput model which shows the performance of informed TCP under different conditions. This througput model should reflect the desired TCP behavior over lossy links, i.e., i) the TCP source should not reduce its sending rate in case of packet loss due to wireless link error (in fact it should keep probing for available bandwidth), and ii) the source should follow normal TCP control rules otherwise. We follow the analysis shown in [11] to model an Internet path having wired and wireless parts. Although the work modeled an end-to-end path using a four-state Markov model, in the analysis the authors only considered average case losses without accounting for bursty losses due to correlated channel conditions. It is part of our future work to derive throughput formula for *informed* TCP using a more detailed model that explicitly account for bursty channel losses similar to the existing work studying standard TCP behavior [1, 2, 10]. Using the analytical model of the desired behavior derived in [11], we have:

$$\lambda(x, y, \delta) = \frac{1}{4E[RTT]}\left(3 + \sqrt{25 + 24\theta}\right), \quad \theta = \frac{1 - P(S_1)}{P_{e2e} - P(S_1)} \tag{10}$$

Note that $\theta \approx \frac{1-P_{Loss}}{P(I_1)}$ where $P(S_1)$ is the packet loss probability on the wireless

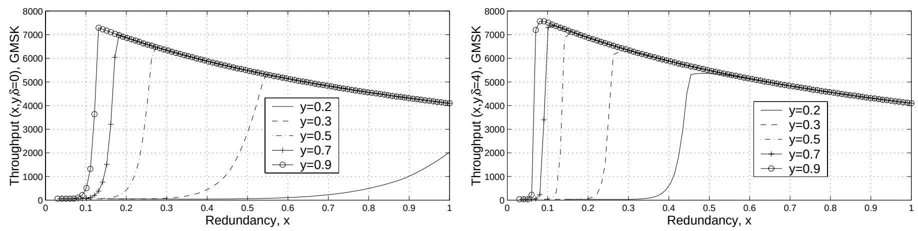

Fig. 4. Behavior of the throughput vs. the redundancy ratio for different values of the transmission power, y and persistency level, $\delta = 0$ (left) and $\delta = 4$ (right) for RTT Model II

link given by Eq. (4), and $P(I_1)$ is average congestion loss probability (we set it to 1% in our experiments). We obtain a utility function by replacing Eq. (10) as $f(\cdot, \cdot)$ in Eq. (9) and then maximize it. Note that Eq. (10) reduces approximately to the throughput equation (without timeouts) given in [14] when $P(S_1) = 0$.

5 Numerical Analysis

In this section we apply the analytical framework developed in the previous sections to the Gaussian M-ary Shift Keying (GMSK) modulation technique, which is used in the General Packet Radio Service. The behavior of Differentiated Binary Phase Shift Keying (DBPSK), which is used in IEEE 802.11 is found similar to GMSK. We present the plots corresponding to DBPSK in the full version.

Table 2. Parameters used in numerical examples. A (attenuation) is given by $\left(\frac{c}{4\pi d f_c}\right)^2$. W_{max} is set to > 64 packets. $c = 3\times10^8$ m/s, $\alpha = 0.8, b = 1, \epsilon \leq (2^K - 1)[4p_e(1 - p_e)]^{d_{min}/2}$

	K	MSS	ΔF	$N_0(e^{-20})$	$k_e/k_r, k_r$	d(m)	T_0	T(ms)	D	f_c
GMSK	260 bits	128 bytes	25MHz	1.379	100,1	150	4s	100	10ms	500MHz

In our experiments we have used a set of values for the parameters which we tabulate in Table 2. In Figure 2, we observe the variation of TCP packet loss probability, P_{Loss} for $\delta = 0$ and $\delta = 10$. The value of P_{Loss} decreases with increasing redundancy, x and increasing power level, y (cf. Eq.(4)). More redundancy increases the error correcting capability of the codeword and more power increases the signal-to-noise ratio reducing the bit error probability. In Figure 3, we observe the variation of TCP end-to-end throughput using Model I of the expected RTT. We can see that throughput increases with increase in power level. For a given value of power and δ, with increase in x, throughput first increases to a maximum value and then reduces. The throughput reduces because

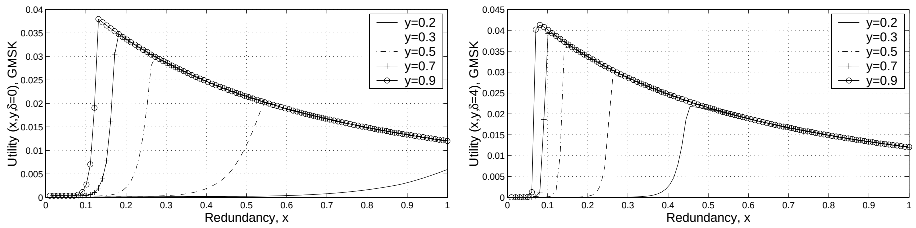

Fig. 5. Behavior of the utility vs. the redundancy ratio for different values of the transmission power, y and persistency level, $\delta = 0$ (left) and $\delta = 4$ (right) for RTT Model I

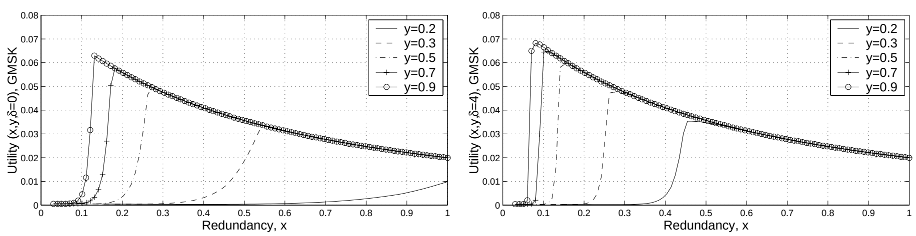

Fig. 6. Behavior of the utility vs. the redundancy ratio for different values of the transmission power, y and persistency level, $\delta = 0$ (left) and $\delta = 4$ (right) (RTT Model II)

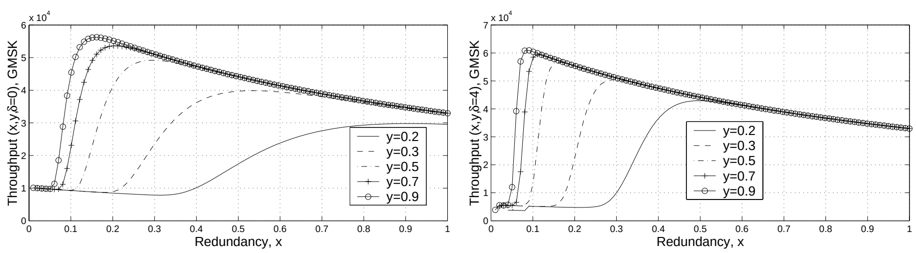

Fig. 7. Behavior of the throughput vs. the redundancy ratio for different values of the transmission power, y and persistency level, $\delta = 1$ (left) and $\delta = 4$ (right)(wireless-loss informed TCP model)

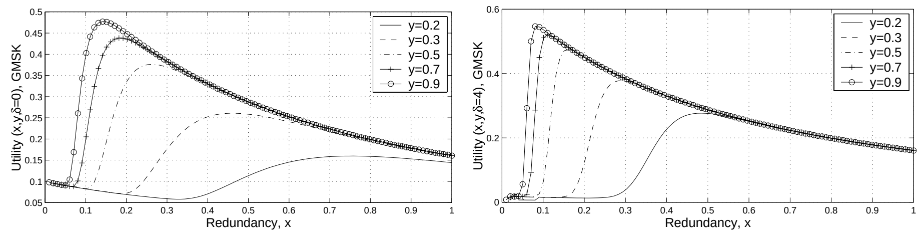

Fig. 8. Behavior of the utility vs. the redundancy ratio for different values of the transmission power, y and persistency level, $\delta = 1$ (left) and $\delta = 4$ (right) (wireless-loss informed TCP model)

more redundancy reduces the effective bandwidth. We can see that increase in δ for a given power level and x, improves the throughput. We observe similar behavior in Figure 4 with RTT model II. In Figure 5, we observe the interesting variation of utility function depending on power, redundancy and persistency using RTT model I. For a given value of power and persistency level, the utility function attains a maximum value at a certain redundancy level. Beyond that, utility reduces after a point because the gain in throughput is outweighted by the increased cost of redundancy. We observe the same interesting pattern of utility function variation in Figure 6 as we observe in Figure 5.

In Figure 7, we plot the throughput using the wireless-loss informed TCP throughput model found in Eq. (10). We kept the end-to-end congestion loss rate at 1% and maximum window size is set to > 64 packets. We observe the improved TCP performance as a result of TCP's awareness of the reason of packet loss. Numerically, we find the optimum values of the parameters, x_{opt}, y_{opt} and δ_{opt} to be 0.0201, 0.9265 and 3, respectively. Figures 2-8 represent the modulation scheme GMSK. We repeated the same experiments for the modulation scheme DBPSK, for which we show representative results in the full version.

6 Conclusion and Future Work

New kinds of losses, additional constraints and new infrastructure demand new solutions to reliable data delivery over wireless channels. In this paper, we have tried to observe a combined effect of all palpable metrics which affect TCP performance in this new environment. We have seen that increasing power, redundancy and retransmission levels always improves the performance by reducing link-layer losses. But such improvements are often associated with cost and arbitrary improvement cannot be realized without paying a lot in return. It is therefore important to consider some kind of net utility function that should be optimized, thus maximizing throughput at the least possible cost. As a part of future work, we would like to obtain some closed-form analytical expressions so that we gain better insights into the interactions among different parameters. Such expressions can be used by the base station to dynamically adjust control parameters according to channel conditions. We intend to validate our analysis using ns-2 simulations.

References

1. Eitan Altman, Kostya Avrachenkov, and Chadi Barakat. TCP in Presence of Bursty Losses. In *Performance Evaluation*, volume 42, pages 129–147, October 2000.
2. F. Anjum and L. Tassiulas. On the Behavior of Different TCP Algorithms over a Wireless channel with Correlated Packet Losses. In *Proceedings of ACM SIGMETRICS*, March 1999.
3. Ajay Bakre and B.R. Badrinath. I-TCP: Indirect TCP for Mobile Hosts. In *Proceedings of the 15th International Conference on Distributed Computing Systems*, 1995.
4. H. Balakrishnan, S. Seshan, and R. Kartz. Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks. In *ACM Wireless Networks, 1(4)*, Dec 1995.
5. Chadi Barakat and Eitan Altman. Bandwidth Tradeoff between TCP and Link-level FEC. In *Proceedings of IEEE International Conf. on Networking, France*, July 2001.
6. Dhiman Barman, Ibrahim Matta, Eitan Altman, and Rachid El Azouzi. TCP Optimization through FEC, ARQ and Transmission Power Tradeoffs. Technical report, Boston University, Computer Science Department, Boston, MA 02215, 2003.
7. Antonio Capone and Fabio Martignon. Bandwidth Estimates in the TCP Congestion Control Scheme. In *Proceedings of IWDC'01*, Italy, 2001.
8. Marcelo M. Carvalho and J.J.Garcia-Luna-Aceves. Delay Analysis of IEEE 802.11 in Single-Hop Networks. In *Proceedings of IEEE ICNP*, Atlanta, Georgia, 2003.
9. Mun Choon Chan and Ramachandran Ramjee. TCP/IP Performance over 3G Wireless Links with Rate and Delay Variation. In *Mobicom'02*, 2002.
10. A. Chockalingam, M. Zorzi, and R.R. Rao. Performance of TCP on Wireless Fading Links with Memory. In *Proceedings of IEEE ICC*, June 1998.
11. Özgür B. Akan and Ian F. Akyildiz. ARC: The Analytical Rate Control Scheme for Real-Time Traffic in Wireless Networks. 2003.
12. Laura Galluccio, Giacomo Morabito, and Sergio Palazzo. An Analytical Study of a Tradeoff Between Transmission Power and FEC for TCP Optimization in Wireless Networks. In *Proceedings IEEE INFOCOM'03*, 2003.
13. J.G.Proakis. *Communication Systems Engineering*. Prentice Hall International Editions, 1994.
14. J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modeling TCP Throughput: A Simple Model and Its Empirical Validation. In *Proceedings of ACM/SIGCOMM '98*, Vancouver, Canada, October 1998.
15. Saar Pilosof, Ramachandran Ramjee, Danny Raz, Yuval Shavitt, and Prasun Sinha. Understanding TCP Fairness over Wireless LAN. In *Proceedings of IEEE INFOCOM'03*, 2003.