Optimal Design of Hybrid FEC/ARQ Schemes for TCP over Wireless Links with Rayleigh Fading

Francesco Vacirca, *Student Member*, *IEEE*, Andrea De Vendictis, *Student Member*, *IEEE*, and Andrea Baiocchi, *Senior Member*, *IEEE*

Abstract—In this paper, we investigate interactions between TCP and wireless hybrid FEC/ARQ schemes. The aim is to understand what is the best configuration of the wireless link protocol in order to guarantee TCP performance and channel efficiency. Interactions between TCP and different link layer mechanisms are evaluated by means of an analytic model that reproduces: 1) a Rayleigh fading channel with FEC coding, 2) a generic selective repeat ARQ Protocol, and 3) the TCP behavior in a wired-cum-wireless network scenario. The analytic model is validated by means of *ns*-based simulations. The analysis represents a contribution to the optimal design of link layer parameters of wireless networks crossed by TCP/IP traffic. The main findings can be summarized as follows: 1) fully reliable ARQ protocols are the best choice for both TCP performance and wireless link efficiency and 2) optimal values of FEC redundancy degree from the point of view of energy efficiency maximizes TCP performance as well.

Index Terms—Wireless communication, TCP modeling, hybrid ARQ.

1 Introduction

In recent years, the issues regarding the behavior of TCP over wireless links have been extensively addressed in the networking research community, both because TCP represents the most widespread transport protocol in the current Internet (about 90 percent of the overall IP traffic is carried by TCP according to recent experimental measurements [1]) and because Internet access through wireless technologies is taking on rapidly. Recent and thorough reviews of general problems and possible solutions regarding the specific "TCP over wireless" topic can be found, for example, in [2] and [3].

In current and future wireless technologies, hybrid FEC/ARQ strategies are frequently used (e.g., UMTS, GPRS, and 802.11a) and, since both FEC and ARQ strongly change the way a wireless link is "seen" by TCP (e.g., end-to-end delays, packet loss probabilities and available capacity), studying their interaction with TCP behavior is a key element for the design of wireless links supporting mobile computing applications.

The aim of the present work is to investigate the impact of these protocols on TCP performance through both analytic methods and simulations. In particular, we are interested in studying how the persistence degree of the radio link layer Automatic Repeat reQuest (ARQ) mechanism and the redundancy degree of the Forward Error Correction (FEC) affect both end-to-end TCP performance and wireless link efficiency.

The analysis is carried out by defining a detailed analytic model that reproduces the link layer packet error process in a Rayleigh fading channel with convolutional FEC coding and selective repeat ARQ protocol. This model is then integrated in a Markov-based TCP model to derive end-to-end TCP performance. All the models are validated and corroborated by means of *ns* simulations.

The large number of related works that appeared in recent literature testifies to the great interest for the topic addressed in the present paper. A part of these studies only considers the presence of FEC in the wireless link, neglecting the use of ARQ mechanisms (e.g., [4], [5], and [6]): [4] and [5] find the value of FEC redundancy degree that guarantees TCP performance without impairing channel efficiency, by assuming a residual packet loss probability in the wireless channel after FEC decoding. In [6], FEC redundancy degree is computed that strikes an optimal tradeoff between energy consumption and TCP throughput.

Another part of the literature only takes into account ARQ mechanisms without considering the impact of FEC strategies. With respect to the three commonly considered solutions—semireliable ARQ (i.e., with a fixed limit on the number of transmissions), persistent and fully reliable ARQ (i.e., unlimited number of transmission attempts), and ARQ with a dynamical adaptive limit—there is no agreement on which of them is the best in order to maximize TCP performance.

Most of the works on TCP over ARQ wireless link implicitly assume that wireless links are not fully shielded from "random" errors caused by the nonideality of the radio channel (i.e., they assume an unreliable or semireliable link

• *The authors are with the Department of Information and Communication, University of Roma, La Sapienza, Via Eudossiana 18, 00184, Roma, Italy. E-mail: {vacirca, devendictis}@infocom.uniroma1.it, andrea.baiocchi@uniroma1.it.*

Manuscript received 26 Feb. 2004; revised 31 Aug. 2004; accepted 4 Oct. 2004; published online 15 Feb. 2006.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-0036-0204.

layer); according to a number of works (e.g., [7] and [8]), the advantage of a not fully persistent ARQ is that it would reduce spurious TCP retransmission timeouts and packet reordering with a straightforward benefit for TCP.

Another "school of thought" (e.g., [9], [10], [11], [12], and [13]) asserts that the wireless link layer should be fully reliable to preserve TCP performance. This choice guarantees that every loss detected by TCP is due to network congestion and that in this way the TCP syntax is respected.

As for the adaptive solutions, in [14], the authors propose a link layer algorithm that adapts the maximum number of link layer transmission attempts according to a target loss rate, that is used as a parameter to describe a desired QoS for a TCP connection. They show that TCP performance can be improved by adapting the maximum number of transmissions to the desired QoS rather than leaving it constant.

Interactions between TCP and hybrid FEC/ARQ mechanisms have been already studied in some previous works ([15], [16], and [17]). In [15], only simulation results are provided, whereas, in [16] and [17], mixed analytic/simulation studies (similar to the one presented in this work) are carried out.

With respect to these works, a big effort has been made to model and analyze the TCP behavior with a hybrid FEC/ARQ link layer protocol in a *correlated* Rayleigh fading radio channel (Jakes' model). Also, compared to earlier works that use a two state Markov process to model the radio channel (e.g., [9], [15]), we use fitting methods to derive the Markov channel from radio channel related parameters that bear a clear physical meaning (e.g., signal to noise ratio, doppler drift, and FEC coding redundancy); in fact, an accurate mapping between radio channel related parameters and Markov parameters is a key element to model different FEC codes by using a Markov chain.

The major original contribution presented in this work is the definition of a complete analytic and simulative framework that integrates and accounts for:

1. a correlated error radio channel with parameters directly derived by physical channel characteristics,
2. the presence of both a convolutional FEC coding and a selective repeat ARQ in the wireless link,
3. the queueing delay at the wireless link layer buffer, and
4. the end-to-end TCP dynamics interacting with the underlying network impairments.

The novelty is the detailed modeling we set up by *jointly* considering all the above points in a single work. Each of them separately (e.g., the Markov model for the radio channel) is not innovative by itself, but no work, as far as we know, puts all of these strongly interacting pieces together.

As a further contribution, the analytic framework allows us to highlight some original results on the effectiveness of high FEC coding degrees on TCP performance and on wireless link energy efficiency when TCP is used as transport protocol. The main findings of our study can be summarized as follows: 1) A fully reliable ARQ protocol is the best choice from either the TCP and wireless channel efficiency perspective. 2) The FEC redundancy degree, optimal from the point of view of wireless efficiency, is the one that maximizes TCP performance as well.

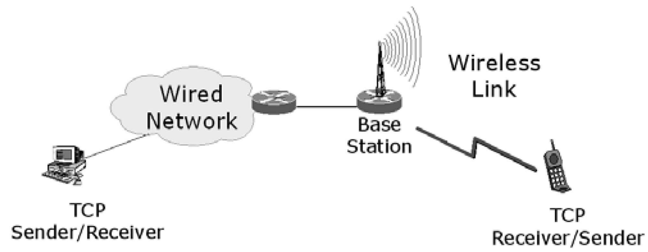

Fig. 1. Network scenario.

The rest of the paper is structured as follows: In Section 2, we depict the network reference scenario. The channel model, the ARQ link layer model, and the TCP model are described in Section 3. In Section 4, we discuss analytic and simulation results. Finally, we give the conclusions in Section 5.

2 Reference Network Scenario

The reference network scenario considered in this work is depicted in Fig. 1. A TCP sender (receiver) placed on a mobile node is connected to a TCP receiver (sender) placed within the wired section of the Internet. In the (simulation and analytic) analysis presented in this paper, we consider Reno version as TCP reference implementation. We also carried out simulations with TCP NewReno since today it is one of the most popular versions of TCP. Details of TCP end-to-end congestion control can be found in [18].

A number of earlier experimental works (see, for example, [19] and [20]) has shown that the Internet is far from being ideal: Congestion packet loss events are quite frequent (even 1 to 10 percent). In order to reproduce this nonideality, that highly affects end-to-end TCP performance, TCP packets are dropped in the wired portion of our network scenario with probability p_f according to a random process assumed independent of TCP connection status.[1] The wired network also introduces a fixed round-trip delay T_f.

As for the radio interface, we assume a centralized radio access handled by a Base Station (BS). The Mobile Station (MS) communicates with the wired node through the BS, connected to the wired network with a dedicated link. In the wireless interface, time is divided into frames of duration τ and every frame provides capacity for both uplink and downlink paths. The BS assigns the capacity dynamically: When the transmission buffer is not empty, the MAC layer notifies a capacity request and the BS assigns the capacity. The maximum TCP capacity assignable to a single mobile user in a direction (uplink or downlink) is C.

The Link Layer (LL) implements a selective repeat ARQ mechanism for error recovery. LL Service Data Units (SDUs) are accepted from upper layers[2] and segmented

1. This hypothesis is reasonable when the traffic generated by the target TCP connections is a negligible fraction of the overall traffic crossing the TCP connection path in the considered wired network.

2. A link layer SDU corresponds to an IP packet carrying a TCP segment.

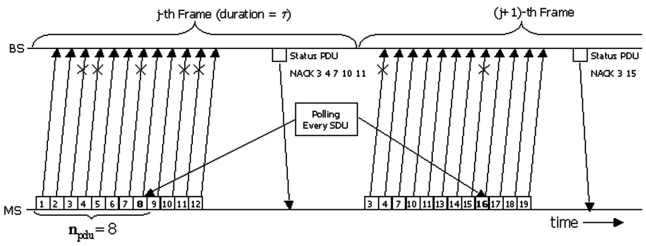

Fig. 2. Selective repeat ARQ protocol: example of time evolution.

into n_{pdu} blocks of l bits. l_b bits are added to every block for the error detection; we assume that the error detection code is able to detect every error. The block of $l + l_b$ bits is encoded with a convolutional encoder by adding l_c redundancy bits; therefore, the rate of the code is $R_c = \frac{l+l_b}{l+l_b+l_c}$. Afterward, each encoded block is referred to as a Protocol Data Unit (PDU). The SDU and PDU lengths are denoted with L_{sdu} and L_{pdu}, respectively. The size $L_{pdu} = l + l_b + l_c$ is fixed so that l has to be adjusted accordingly as R_c varies; therefore, $n_{pdu} = \lceil \frac{L_{sdu}}{L_{pdu} \cdot R_c - l_b} \rceil$. Every PDU is placed in the LL transmission buffer, assumed infinite, and transmitted when the MAC layer indicates that there are available transmission resources. The transmission power is set according to a power control loop with a target signal to noise ratio γ_{target}.

On reception, every PDU is passed to a Viterbi convolutional decoder with soft decision; then, error detection is performed. When all the n_{pdu} PDUs belonging to a SDU are received correctly, the SDU is reassembled and delivered to the upper layer preserving the original SDU sequence.

After sending a whole SDU, the LL transmitter entity polls the receiver for an acknowledgment (status report). According to the selective repeat ARQ, the receiver sends a PDU containing the status report (Status PDU) indicating the PDUs received correctly and the ones to be retransmitted. When the Status PDU is received, PDUs buffered in the retransmission buffer of the sender entity are deleted or retransmitted according to the status report. Every PDU can be transmitted at most M_t times. When this number is reached, the transmitter entity discards that PDU and all PDUs belonging to the same SDU. Given the limited value of M_t, a positive SDU loss probability p_w results in the wireless link. An example of the ARQ mechanism when the number of PDUs in a SDU is 8 is depicted in Fig. 2.

3 Analytic Models

In this section, we describe the model of the error process on a LL PDU in case of Rayleigh fading with convolutional FEC coding and soft-decision Viterbi decoding (Section 3.1), the model of the selective repeat ARQ mechanism (Section 3.2) and the one that reproduces the end-to-end TCP behavior (Section 3.3). These models are validated against simulations in Section 4.

3.1 Radio Channel Model

The error process on the LL PDUs is modeled with a two-state discrete Markov chain (Gilbert-Elliot channel [21])

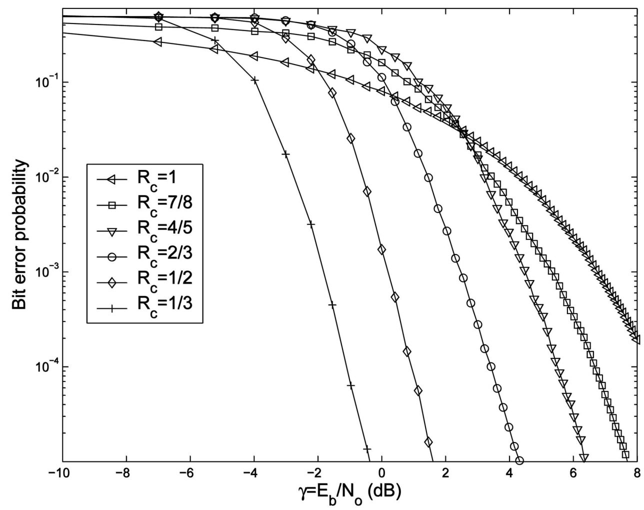

Fig. 3. Residual bit error probability after soft decoding for different code rate R_c varying $\gamma = \mathcal{E}_b/\mathcal{N}_0$.

which is widely used in the related literature to characterize wireless transmission process in a Rayleigh fading channel (see, for example, [22]). One state (GOOD) has very low PDU error probability p_G, whereas the other one (BAD) has high PDU error probability p_B, corresponding to deep fade periods.

The discrete Markov chain with time unit τ (a radio frame time) is characterized by the transition probabilities g and b, that represent the conditional probabilities that the channel state remains, respectively, in the GOOD and in the BAD state during the sampling time τ, provided it was in the same state in the previous sampling interval.

The problem addressed in this section is the definition of a procedure that matches the values of Markov process parameters (b, g, p_B, and p_G) to physical parameters (i.e., signal to noise ratio, Doppler drift, and FEC code redundancy degree).

To this aim, we assume that the channel is in the GOOD state when the signal to noise ratio $\gamma = \mathcal{E}_b/\mathcal{N}_0$ is higher than a threshold (γ_{th}) and in the BAD state when γ is less or equal to γ_{th}.

Assuming an ideal interleaving process, the mean LL PDU error probability in the GOOD and in the BAD state are, respectively:

$$p_G = E\Big[1 - (1 - f(\gamma))^{l+l_b} | \gamma > \gamma_{th}\Big],$$

$$p_B = E\Big[1 - (1 - f(\gamma))^{l+l_b} | \gamma \le \gamma_{th}\Big],$$

where $f(\gamma)$ is the residual bit error probability after FEC decoding according to the modulation format and the coding scheme. Given the coding and modulation scheme, $f(\gamma)$ can be computed numerically. Fig. 3 depicts the residual bit error probability after soft decoding as a function of γ for different values of the FEC convolutional code rate R_c. These values have been obtained by means of matlab simulations with a QPSK modulation scheme and the following codes and generators described in [24]:

$$R_c = 1 \text{ (no FEC code)},$$
$$R_c = \frac{7}{8} \; (K = 8, [3\;4\;11\;20\;41\;100\;201\;377]),$$
$$R_c = \frac{4}{5} \; (K = 2, [237\;274\;156\;255\;337]),$$
$$R_c = \frac{2}{3} \; (K = 4, [236\;655\;337]),$$
$$R_c = \frac{1}{2} \; (K = 9, [753\;561]),$$
$$R_c = \frac{1}{3} \; (K = 9, [711\;663\;557]).$$

We assume no limitation on transmission power and that the power control loop manages to keep γ to stick on γ_{target} on average, i.e., $\gamma = \gamma_{target} \cdot a/\langle a\rangle$, where a is the square of the fading envelope and $\langle\cdot\rangle$ indicates the mean value of the argument. The average LL PDU error probability in the GOOD and in the BAD state can be computed as follows:

$$p_G = \frac{1}{Pr\{a > a_{th}\}} \int_{a_{th}}^{\infty} p_A(a) \left(1 - \left[1 - f\left(\gamma_{target} \cdot \frac{a}{\langle a\rangle}\right)\right]^{l+l_b}\right) da, \quad (1)$$

$$p_B = \frac{1}{Pr\{a \le a_{th}\}} \int_{0}^{a_{th}} p_A(a) \left(1 - \left[1 - f\left(\gamma_{target} \cdot \frac{a}{\langle a\rangle}\right)\right]^{l+l_b}\right) da, \quad (2)$$

where $p_A(a)$ is the probability density function (pdf) of a and a_{th} is the value of a so that $\gamma = \gamma_{th}$, i.e., $a_{th} = \langle a\rangle \cdot \gamma_{th}/\gamma_{target}$.

In case of a correlated Rayleigh fading process (Jakes' model [23]) with unitary mean power ($\langle a\rangle = 1$), the pdf of a is e^{-a} when $a \ge 0$ and 0 otherwise; thus:

$$Pr\{a > a_{th}\} = \int_{a_{th}}^{\infty} p_A(a) da = e^{-a_{th}},$$

$$Pr\{a \le a_{th}\} = \int_{0}^{a_{th}} p_A(a) da = 1 - e^{-a_{th}}.$$

According to [23], the expected rate,[3] at which the square of the fading envelope a crosses a specified level a_{th} in the positive (or negative) direction $N_{a_{th}}$ is equal to $\sqrt{2\pi a_{th}} f_d e^{-a_{th}}$, where f_d is the maximum doppler drift, obtained as the ratio between the mobile user speed v and the wavelength of the carrier λ.

Considering that in a period of duration t' the expected time in which a is less or equal to a_{th} is $Pr(a \le a_{th}) \cdot t'$ and that during the same period a crosses the a_{th} level $N_{a_{th}} \cdot t'$ times, the average duration of periods in which a is less or equal to a_{th} is given by the ratio between $Pr(a \le a_{th}) \cdot t'$ and $N_{a_{th}} \cdot t'$:

$$\frac{Pr(a \le a_{th})}{N_{a_{th}}} = \frac{e^{a_{th}} - 1}{\sqrt{2\pi a_{th}} f_d}, \quad (3)$$

and the average duration of periods in which a is greater than a_{th} is:

$$\frac{Pr(a > a_{th})}{N_{a_{th}}} = \frac{1}{\sqrt{2\pi a_{th}} f_d}. \quad (4)$$

Setting (3) and (4) equal to the mean sojourn time in the BAD state ($\tau/(1-b)$) and in the GOOD state ($\tau/(1-g)$) of the Markov chain, respectively, it is possible to compute the transition probabilities g and b.

To determine the Markov channel parameters (b, g, p_B, and p_G), it remains to compute a_{th}. As suggested in [25], a_{th} is computed numerically by minimizing the mean square error between the autocorrelation function of the Rayleigh fading process of the Jakes' model and the autocorrelation function of a "synthetic" process generated by the Markov chain. To this aim, we define r_G to be the square root of the expectation of a conditional on $a > a_{th}$ and r_B the square root of the expectation of a conditional on $a \le a_{th}$. It can be found that $r_G = \sqrt{1 + a_{th}}$ and $r_B = \sqrt{1 - a_{th}/(e^{a_{th}} - 1)}$.

Then, we generate a "synthetic" fading process by associating r_G and r_B to the GOOD and the BAD state of the Markov chain, respectively. The autocorrelation function at kth time lag ρ_k generated by this process is given by:

$$\rho_k = \begin{pmatrix} r_G & r_B \end{pmatrix} \begin{pmatrix} g & 1-b \\ 1-g & b \end{pmatrix}^k \begin{pmatrix} r_G\psi_G \\ r_B\psi_B \end{pmatrix}, \quad k \ge 0, \quad (5)$$

where ψ_G and ψ_B are the steady state probabilities of the Markov chain: $\psi_G = (1-b)/(2-b-g)$, $\psi_B = 1 - \psi_G$.

a_{th} is then derived by minimizing the error between the autocorrelation of the fading process and the "synthetic" one:

$$\min_{a_{th}} \left\{ \sum_k \left(J_0(2\pi f_d k\tau) - \rho_k \right)^2 \right\}, \quad (6)$$

where $J_0(\cdot)$ is the Bessel function of the first kind and order zero ([23]). To compute a_{th} numerically, the minimum square error function is truncated. It is worth noticing that the value of a_{th} depends only on $f_d \cdot \tau$.

To sum up, the Markov chain parameters can be computed as follows:

$$p_G = e^{a_{th}} \int_{a_{th}}^{\infty} e^{-a} \left(1 - \left[1 - f(\gamma_{target} \cdot a)\right]^{R_c \cdot L_{pdu}}\right) da,$$

$$p_B = \frac{1}{1 - e^{-a_{th}}} \int_{0}^{a_{th}} e^{-a} \left(1 - \left[1 - f(\gamma_{target} \cdot a)\right]^{R_c \cdot L_{pdu}}\right) da,$$

$$b = 1 - \frac{\sqrt{2\pi a_{th}} f_d \tau}{e^{a_{th}} - 1},$$

$$g = 1 - \sqrt{2\pi a_{th}} f_d \tau,$$

with a_{th} computed from (6).

3. The level crossing rate at which a stochastic process $a(t)$ crosses a given level A from up to down (or down to up) is $N_A = \int_0^\infty \dot{a} p(A, \dot{a}) d\dot{a}$, where $\dot{a}$ indicates the time derivative of a and $p(A, \dot{a})$ is the joint density function of a and $\dot{a}$ for a equal to A.

TABLE 1
Markov Channel Parameters for $L_{pdu} = 800$ Bits and $f_d \cdot \tau = 0.1$ (Corresponding to $b = 0.8064, g = 0.7825$)

	$\gamma_{target} = 3$		$\gamma_{target} = 2$		$\gamma_{target} = 1.25$	
R_c	p_B	p_G	p_B	p_G	p_B	p_G
1/1	0.333	0.193	0.5	0.402	0.8	0.759
7/8	0.333	0.106	0.5	0.304	0.8	0.682
4/5	0.333	0.055	0.5	0.237	0.8	0.616
2/3	0.276	$7.5 \cdot 10^{-4}$	0.499	0.055	0.8	0.374
1/2	0.18	$2.6 \cdot 10^{-9}$	0.374	$3.4 \cdot 10^{-5}$	0.794	0.046
1/3	0.116	$2 \cdot 10^{-16}$	0.248	$6 \cdot 10^{-10}$	0.583	$4.7 \cdot 10^{-5}$

In Table 1, the packet error probabilities in the BAD and in the GOOD state are shown for some values of γ_{target} and R_c, when $f_d \cdot \tau = 0.1$ and $L_{pdu} = 800$ bits; the corresponding value of b and g are, respectively, 0.8064 and 0.7825.

3.2 Selective Repeat ARQ Model

In this section, we describe the discrete Markov chain modeling the wireless link selective repeat ARQ introduced in Section 2. A preliminary version of this model has been already presented in [29].

The selective repeat ARQ protocol is modeled with a discrete time Markov chain which reproduces the transmission of SDUs segmented in n_{pdu} PDUs of equal size on a Gilbert-Elliot channel under the following assumptions:

1. A single and isolated SDU is considered.
2. The n_{pdu} PDUs belonging to the SDU start their transmission attempts all together and up to n_{pdu} PDUs can be transmitted in a single time interval of duration τ.
3. A selective ack/nack is delivered without errors by the end of the time interval τ.
4. Retransmissions of erroneous PDUs take place in the next time interval.
5. The error events of different PDUs conditional on the state of the channel are independent of one another.

The hypotheses 1) and 2) greatly simplify the behavior of the ARQ protocol described in Section 2 and used to validate the model with simulations. Validation results in Section 4.1 point out that the ARQ model yields quite accurate results also in case of *non* isolated SDUs.

A Markov chain state is defined by the vector $\overline{s} = (J, X)$, where:

- $J \in \{\mathrm{GOOD}, \mathrm{BAD}\}$ is the Gilbert-Elliot channel state; in the following, the GOOD and the BAD state are referred as G and B, respectively.
- $X \in [0, n_{pdu}]$ is the number of PDUs, belonging to the considered SDU, not successfully transmitted yet.

In Fig. 4, the Markov chain of the ARQ retransmission process is shown. A transition from the state (J, k) to the state (J', k') represents the event that the channel state changes from J to J' and $k - k'$ PDUs are transmitted successfully and k' PDUs are received with error; only transitions from k to k' with $k' \leq k$ are allowed. Given that the PDU error events are independent of one the other conditional on J, the probability of failing k' transmissions out of k in the G and in the B state is, respectively:

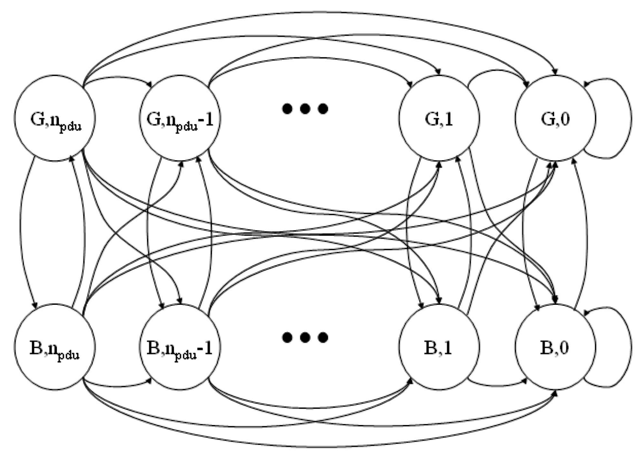

Fig. 4. Markov chain describing ARQ protocol.

$$b_G(k, k') = \binom{k}{k'} p_G^{k'} \cdot (1 - p_G)^{k-k'} \qquad k' = 0, \ldots, k$$

and

$$b_B(k, k') = \binom{k}{k'} p_B^{k'} \cdot (1 - p_B)^{k-k'} \qquad k' = 0, \ldots, k,$$

where p_G and p_B are the error probabilities in the state G and B of the Markov channel as derived in the previous section.

According to our hypotheses, all the PDUs composing a SDU are transmitted starting at the same frame (of duration τ). So, the one step transition probability matrix of the process is:

$$\mathbf{P} = \begin{bmatrix} g \cdot \mathbf{T}_G & (1-g) \cdot \mathbf{T}_G \\ (1-b) \cdot \mathbf{T}_B & b \cdot \mathbf{T}_B \end{bmatrix},$$

where g and b are the Markov channel transition probabilities, computed in the earlier section, and $\mathbf{T}_G$ and $\mathbf{T}_B$ are defined as:

$$\mathbf{T}_G = \begin{bmatrix} 1 & 0 & \ldots & 0 \\ b_G(1,0) & b_G(1,1) & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ b_G(n_{pdu},0) & b_G(n_{pdu},1) & \ldots & b_G(n_{pdu},n_{pdu}) \end{bmatrix},$$

$$\mathbf{T}_B = \begin{bmatrix} 1 & 0 & \ldots & 0 \\ b_B(1,0) & b_B(1,1) & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ b_B(n_{pdu},0) & b_B(n_{pdu},1) & \ldots & b_B(n_{pdu},n_{pdu}) \end{bmatrix}.$$

Let $\mathbf{T}_G^*$ ($\mathbf{T}_B^*$) be a matrix obtained from $\mathbf{T}_G$ ($\mathbf{T}_B$) by removing the first row and the first column, the matrix $\mathbf{Q}$ is defined as:

$$\mathbf{Q} = \begin{bmatrix} g \cdot \mathbf{T}_G^* & (1-g) \cdot \mathbf{T}_G^* \\ (1-b) \cdot \mathbf{T}_B^* & b \cdot \mathbf{T}_B^* \end{bmatrix}. \qquad (7)$$

$\mathbf{Q}$ is the one step transition probability matrix of a transient Markov chain, whose absorption time η gives the number of frames needed to complete the delivery of the PDUs making up an SDU. It is well known that

$$F(n) = Pr\{\eta > n\} = \boldsymbol{\pi}\mathbf{Q}^n\mathbf{e} \text{ for } n \geq 0, \tag{8}$$

where $\mathbf{e} = [1 \dots 1]^T$ and $\boldsymbol{\pi}$ is the $2n_{pdu}$ components row vector of the initial probabilities. It is $\boldsymbol{\pi} = [0, 0, \dots \alpha_G, 0, 0, \dots \alpha_B]$; α_G and α_B are computed as the probability distribution at the absorption for a single PDU ($n_{pdu} = 1$). Let us define:

$$\mathbf{A} = \begin{bmatrix} g(1-p_G) & (1-g)(1-p_G) \\ (1-b)(1-p_B) & b(1-p_B) \end{bmatrix},$$
$$\mathbf{B} = \begin{bmatrix} gp_G & (1-g)p_G \\ (1-b)p_B & bp_B \end{bmatrix}.$$

Then, $\boldsymbol{\alpha} = [\alpha_G, \alpha_B]$ is the solution of $\boldsymbol{\alpha} = \boldsymbol{\alpha}(\mathbf{I} - \mathbf{B})^{-1}\mathbf{A}$ with the condition $\alpha_G + \alpha_B = 1$. This choice of the initial probability distribution is used to fix hypothesis 1). In fact, in the actual running of the ARQ protocol in saturation, PDUs from different SDUs are transmitted together and a new SDU starts as soon as at least one PDU worth of capacity is available. This scenario is realistic with a greedy TCP source unless it experiences such a high packet loss rate so that it can not fill the transmission pipe.

The probability of failure of SDU delivery is given by $F(M_t)$, with $F(\cdot)$ as in (8), provided that M_t is the maximum number of transmission attempts per PDU.

The expected value of η is the mean number of frames required to deal with an SDU and can be evaluated as follows:

$$\begin{aligned} E[\eta] &= \sum_{n=0}^{\infty} nPr\{\eta = n\} \\ &= \sum_{n=0}^{\infty} F(n) = \boldsymbol{\pi}(\mathbf{I} - \mathbf{Q})^{-1}\mathbf{e}. \end{aligned} \tag{9}$$

The mean number of frames required per SDU when the number of transmissions per PDU is limited to M_t is

$$\begin{aligned} E[\eta | \eta \leq M_t] &= \sum_{n=0}^{M_t - 1} F(n) \\ &= \boldsymbol{\pi}(\mathbf{I} - \mathbf{Q}^{M_t})(\mathbf{I} - \mathbf{Q})^{-1}\mathbf{e}. \end{aligned} \tag{10}$$

If we set $n_{pdu} = 1$ in the above model, i.e., we consider the transmission process of a single PDU,[4] the model can be used to compute the mean number of transmissions per PDU, $\mathcal{N}_\infty$ and $\mathcal{N}_{M_t}$, respectively, in the case of unlimited transmissions and in the case of a limited number of transmissions. In this case, the model reduces to a four-state Markov chain, representing a simple retransmission mechanism under a Gilbert-Elliot channel.

$F(M_t)$, $E[\eta | \eta \leq M_t]$, and $\mathcal{N}_{M_t}$ will be used in the next section as input parameters of the TCP model.

3.3 Analytic Model for TCP Behavior in Wired-Cum-Wireless Environments

In this section, we define the TCP analytic model able to reproduce the stationary behavior of a TCP Reno source. As also pointed out in the rest of this section, a part of the input parameters to the TCP model (the ones related to the wireless link) are obtained by applying the selective repeat ARQ model depicted in Section 3.2.

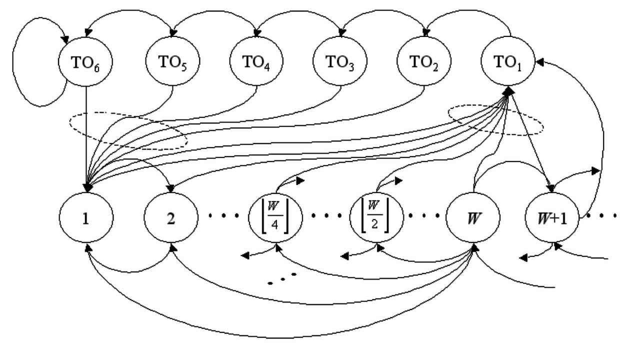

Fig. 5. Markov chain describing TCP behavior.

In order to define the model, the following hypotheses are assumed:

1. TCP source is "greedy," always ready to send new data (persistent TCP connection model). The rapid development of "peer-to-peer" applications and high-speed access links that allow the exchange of a large amount of data among users makes this hypothesis realistic. Besides, already now, according to [1], persistent streams carry a high proportion (50 to 60 percent) of the total amount of traffic on the Internet.
2. TCP Slow Start is neglected.
3. TCP packet losses are assumed to be independent.
4. The TCP receiver does not limit the value of the TCP sender congestion window according to Allman et al. [26].
5. The TCP ACK packets are always delivered to TCP sender.
6. The LL buffer is infinite.

The model is based on a discrete Markov chain. Although the Markov chain approach has been already used in earlier works (e.g., [27], [28]), with respect to them, we also include the multiple Fast Retransmit events due to multiple loss within a single TCP transmission window[5] and the effects of queueing delay within the LL buffer.

The discrete Markov chain describing the stationary dynamics of TCP is depicted in Fig. 5. States are divided into two groups: The first one represents the TCP congestion window behavior during congestion avoidance algorithm, the state index represents the congestion window size, and the second group (from TO_1 to TO_6) represents timeout dynamics. To simplify the notation, in the following, the states $TO_1, TO_2, \dots, TO_6$ are referred to, respectively, as $0, -1, \dots, -5$.

The transition probability q_{ij} from the state i to the state j is:

4. Given the model hypotheses, the transmission process of a PDU is equivalent to the one of a SDU made up of a single PDU.

5. From simulations, we observed this is a quite significant occurrence when TCP Reno is used.

$$q_{ij} = \begin{cases} (1-\nu)^i & \text{if } j = i+1, i > 0 \\ \binom{i}{c}\nu^c(1-\nu)^{i-c} & \text{if } j = \lfloor \frac{i}{2^c} \rfloor, j > 1, i > 3, \\ & \text{for } c \in N, 1 \le c < \log_2(i) \\ 1 - \sum_{k \neq 1} q_{ik} & \text{if } j = 1, i > 3 \\ \sum_{k=0}^{2} \binom{i}{i-k}\nu^{i-k}(1-\nu)^k & \text{if } j = 0, i > 3 \\ \sum_{c=1}^{i} \binom{i}{c}\nu^c(1-\nu)^{i-c} & \text{if } j = 0, i = 1,2,3 \\ \nu & \text{if } j = \max\{-5, i-1\}, \\ & -5 \le i \le 0 \\ 1-\nu & \text{if } j = 1, -5 \le i \le 0 \\ 0 & \text{otherwise,} \end{cases} \quad (11)$$

where $\nu = 1-(1-p_f)(1-p_w)$ is the TCP packet loss probability due to wireless impairments and/or to wired network congestion and these two sources of packet loss are assumed to be independent of each other.

The packet loss probability in the wireless link $p_w = F(M_t)$ is derived from (8) by applying the selective repeat ARQ model described in Section 3.2 and depends on the radio channel Markov model parameters (p_B, p_G, b, and g), the SDU size (n_{pdu}) and the ARQ persistence degree (M_t).

The first item of (11) represents the probability of having no packet losses when the window size is i: In this case, TCP increases the congestion window by one packet and we have a transition from the state i to the state $i+1$. The second item represents the probability of having multiple Fast Retransmit during congestion avoidance if the congestion window is larger than three packets: If there are c packet losses within a time unit (i.e., within a single window of packets), we assume TCP performs c times the Fast Retransmit algorithm so to halve the window c times (transition from the state i to the state $\lfloor \frac{i}{2^c} \rfloor$). We make the rough simplification that the multiple Fast Retransmit is performed in a single time unit.[6] This simplification is required to keep the state space size feasible and it is supported by the satisfactory agreement with simulations as shown in Section 4. The third item in (11) is derived from the condition that $\sum_j q_{ij} = 1, \forall i$. The fourth and the fifth items represent the probability of having a timeout, respectively, when the congestion window is larger and smaller or equal to three packets. When the congestion window is not larger than three packets, a single or multiple packet loss always induces a timeout in the TCP sender. If the congestion window is larger than three packets (say W packets), timeout is induced when less than three packets out of W packets survive to a multiple loss event. Conversely, we assume that if the lost packets are less or equal to $W-3$, they are always revealed by means of the Fast Retransmit algorithm (see the second item in (11)) because there are at least three duplicate ACKs within the network able to trigger the Fast Retransmit. Furthermore, after a timeout, the model neglects the presence of Slow Start. In our opinion, this is a valid approximation for two reasons: On the one hand, the time in which TCP performs Slow Start is generally negligible with respect to the congestion avoidance period. On the other hand, timeout events are performed when the packet loss rate is very high; in these cases, the average congestion window is small as well as the Slow Start threshold and the Slow Start phase is short as well. The sixth item in (11) represents the probability that, during a timeout, a packet is lost. The seventh item represents the probability that, during a timeout, a packet is received correctly.

The time t_i spent in the state i is state dependent and represents the time necessary to transmit all the packets belonging to a window of size $i = 1, 2, 3, \ldots$ or to perform the timeout algorithm. We have:

$$t_i = \begin{cases} \frac{i}{C_d} & \text{if } i \ge C_d \cdot E[T] \text{ and } i > 0 \\ E[T] & \text{if } i < C_d \cdot E[T] \text{ and } i > 0 \\ RTO \cdot 2^{-i} & \text{if } i \le 0, \end{cases} \quad (12)$$

where C_d is the capacity available for the TCP connection at link layer and $E[T]$ is the average end-to-end round-trip delay computed excluding queueing delay experienced within the LL buffer. In fact, we are considering a variation of a "single bottleneck lossy link" TCP model: The role of the bottleneck with its buffer is played by the wireless link layer and the LL buffer; $E[T]$ is equivalent to the base TCP round trip time and TCP packets are lost with probability ν independently of TCP dynamics.

In (12), according to a fluid approximation used in a number of TCP related works, the first item represents the time necessary to transmit a window of i packets when the capacity available is saturated; the second item represents the case in which the packets within a window of size i are transmitted without suffering queueing delays since there is enough available capacity. The third item takes into account the duration of each state $TO_j, j = 1, \ldots, 6$: According to TCP mechanisms (exponential back off), the duration of the jth consecutive timeout is 2^{j-1} times the current Retransmission TimeOut (RTO); from the sixth consecutive timeout on, the duration is 2^5 times the RTO. As generally assumed in literature (e.g., [27]), RTO is chosen equal to five times the average round-trip time:

$$RTO = 5 \cdot \frac{\sum_{i>0} t_i \cdot \phi_i}{\sum_{i>0} \phi_i}, \quad (13)$$

where ϕ_i are the steady state probabilities of the Markov chain depicted in Fig. 5: In order to compute the average round-trip time in (13), we only take into account the TCP congestion avoidance states.

Assuming the bottleneck is the wireless link, the available capacity C_d in (12) is:

$$C_d = C \cdot \frac{1}{\mathcal{N}_{M_t}} \cdot \frac{L_{sdu}}{n_{pdu} \cdot L_{pdu}}, \quad (14)$$

where C is the maximum capacity assigned to the TCP connection and $\mathcal{N}_{M_t}$ is the mean number of transmission attempts per PDU given the maximum number of transmissions at LL is M_t (see Section 3.2).

$E[T]$ in (12) is the sum of two terms: the average round-trip time in the wired network $E[T_f]$ and the TCP packet delay spent to cross the radio interface $E[T_w]$. $E[T_w]$ depends on the packet retransmissions occurred at LL:

6. Actually, multiple packet losses can produce either multiple Fast Retransmit or timeout events. Moreover, the events caused by a multiple loss last in general more than a single time unit.

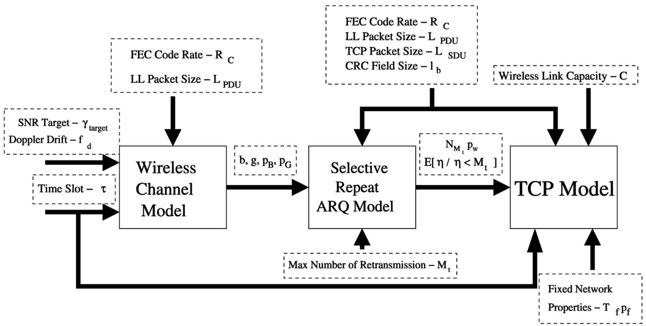

Fig. 6. Summary of the relationships among the analytic models.

$$E[T] = E[T_f] + E[T_w] = T_f + \tau \cdot E[\eta | \eta \leq M_t] + \tau, \qquad (15)$$

where T_f is the round-trip time of the wired network, τ is the duration of radio frame and $E[\eta|\eta \leq M_t]$ is the average number of radio frames necessary to deliver a TCP packet across the wireless interface as derived from (10). The last term τ takes into account the transmission delay of the TCP ACK on the wireless link, assuming that it is sent without errors.

According to the Little's law, the TCP throughput is the ratio between the average window and the average value of the window holding time t_i in (12):

$$\Lambda = \frac{E[W]}{E[t_i]} = \frac{\sum_i \max(1, i) \cdot \phi_i}{\sum_i t_i \cdot \phi_i}. \qquad (16)$$

This model holds for $\nu > 0$. Since we assume an infinite backlog buffer, in case of no lossy links ($\nu = 0$), the model degenerates to an unstable Markov chain; however, in that case, TCP throughput is simply C_d given in (14), i.e., the available bottleneck capacity.

Summing up, the TCP analytic model depends on the following two groups of input parameters: A first group of parameters that reproduces the wired part of the network crossed by the TCP flow (packet loss probability in the wired network p_f and average round-trip time in the wired network T_f); a second group of parameters that reproduces the wireless link properties, also derived from the selective ARQ model (available capacity C_d on the wireless link, packet loss probability p_w, the time necessary to a packet to cross the wireless link $E[T_w]$).

For the sake of clarity, a scheme that summarizes the relationship among input and output parameters of the models is depicted in Fig. 6.

4 Results and Discussion

In this section, we discuss the results provided by either the analytic models and the simulations and examine the interactions between hybrid FEC/ARQ protocol and TCP.

All the simulations have been carried out with an UMTS-TDD module for *ns* [30], [31] that simulates the TD-CDMA radio interface with the link layer configured in the Acknowledged Mode (see [32] for a detailed description of the UMTS RLC layer). This interface reflects the characteristics of the generic system described in Section 2.

In every simulation, the TCP source is placed on the mobile node. The system parameters are listed in Table 2: The value of those parameters that do not vary in the ensuing analysis are reported in the table. As for the FEC codes, we use the convolutional codes introduced in Section 3.1. The wireless link is assumed to be the bottleneck with a maximum capacity C of 100 kbyte/s assignable to the user. So, with no overhead at all, the maximum TCP/IP throughput would be 100 pkts/s.

4.1 Selective Repeat ARQ Model Validation

In Fig. 7, we compare simulations with ARQ model results, specifically, the quantities $\mathcal{N}_{M_t}$ and p_w. These are the most critical ARQ parameters as for their impact on TCP performance.

The SDU arrival process at the LL layer is driven by the TCP source. This is done to challenge the assumption of isolated SDUs in the analysis carried out in Section 3.2. We also verified that the model yields no observable errors with respect to simulations in case of isolated SDUs; those results are not reported here for the sake of space.

The agreement between model and simulations is very good for $\gamma_{target} = 2$ and more than satisfactory for $\gamma_{target} = 1.25$. The accuracy of the model improves as γ_{target} grows up: For $\gamma_{target} \geq 3$, there are virtually no errors.

According to the simulation setting, hypothesis 2) of the ARQ model (Section 3.2) does not hold. As a matter of fact, only 10 PDUs per frame can be transmitted over the

TABLE 2
System Parameter Values Used in the Performance Evaluation

Symbol	Value	Description
γ_{target}	variable	Target SNR at the wireless receiver
R_c	variable	FEC code rate
M_t	variable	Max number of LL PDU tx
L_{pdu}	100 byte	LL PDU size
L_{sdu}	1000 byte	LL SDU size
l_b	16 bit	CRC field size
f_d	10 Hz	Doppler drift
τ	10 ms	LL frame time
C	100000 byte/s	Wireless link capacity
T_f	variable	TCP RTT in the wired part of the network
p_f	variable	Packet loss probability in the wired part of the network

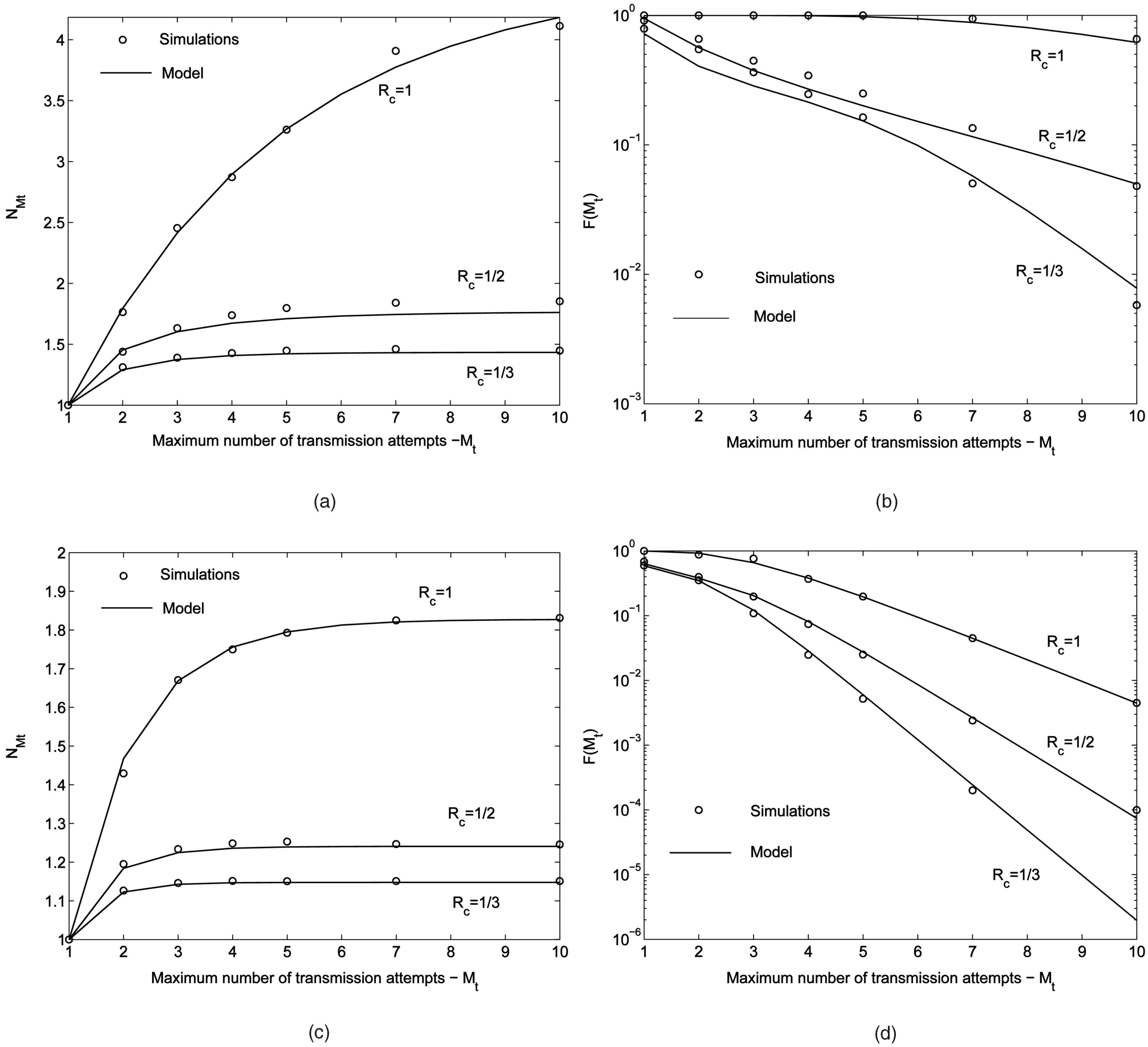

Fig. 7. Selective repeat ARQ model validation. (a) $\mathcal{N}_{M_t}$ ($\gamma_{target} = 1.25$). (b) p_w ($\gamma_{target} = 1.25$). (c) $\mathcal{N}_{M_t}$ ($\gamma_{target} = 2$). (d) p_w ($\gamma_{target} = 2$).

wireless link while $n_{pdu} > 10$ for all considered values of R_c. In spite of this, we get quite a good accuracy even with relatively simple model of Section 3.2.

4.2 TCP Performance and Wireless Efficiency

The ARQ persistence degree that maximizes end-to-end TCP performance is the first issue addressed in this section.

In Fig. 8, TCP throughput is shown as a function of M_t, for different values of the FEC code rate R_c (1, 2/3, 1/2, 1/3). In all figures but Fig. 8b, $p_f = 0$, so that no packet losses take place in the wired network; in Fig. 8b, p_f is set to 0.01. The first two figures are obtained with $T_f = 0.4$ s, the other two figures by setting $T_f = 0$ s, in order to consider a network scenario without the fixed part of the Internet and to focus only on the wireless link. In this last scenario ($p_f = T_f = 0$), we consider two different values of γ_{target}, i.e., $\gamma_{target} = 1.25$ and $\gamma_{target} = 2$. The chosen Doppler drift $f_d = 10$ Hz corresponds to a mobile user speed of 2.7 km/h at a carrier frequency of 2 GHz. Similar results have been obtained with different values of f_d.

The qualitative behavior of TCP performance is captured faithfully by the model. In particular, in every situation, model and simulation results agree on the optimal value of M_t; moreover, the steep increase of TCP throughput with M_t is predicted correctly by the model. Exploiting the results about the ARQ model validation (Section 4.1), we trace back the different error sources. The TCP model input parameters are affected by some error: For instance, the maximum relative error on $\mathcal{N}_{M_t}$ is 5.5 percent for $\gamma_{target} = 1.25$ and 2.6 percent for $\gamma_{target} = 2$, the relative error on p_w is 35 percent (maximum), 12.4 percent (average) for $\gamma_{target} = 1.25$ and 13.7 percent (maximum), 6.7 percent (average) for $\gamma_{target} = 2$, by considering only values of $M_t \geq 3$ where simulations are more reliable. These errors affect TCP expecially for small values of M_t, i.e., when the residual packet loss is not negligible.

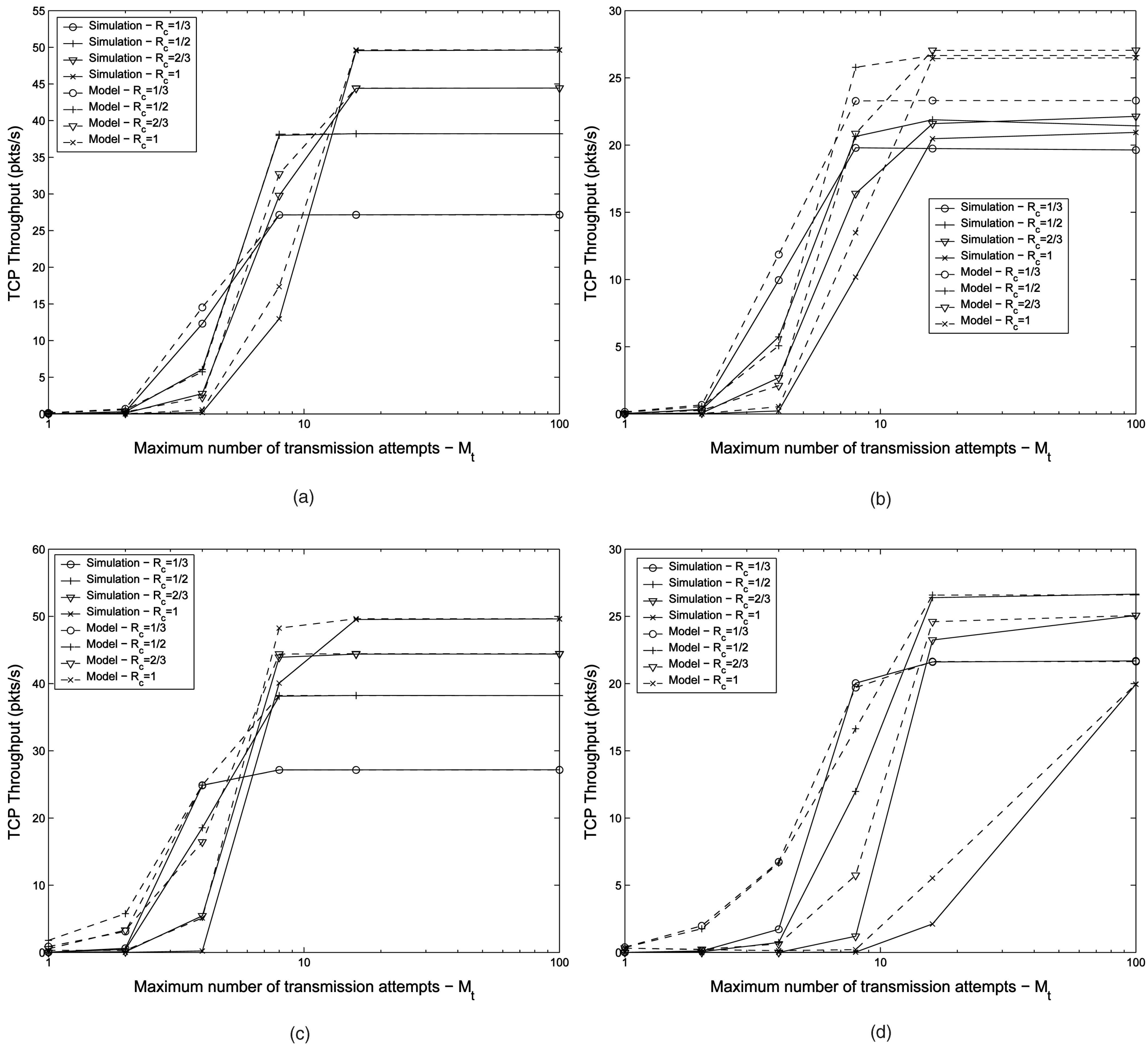

Fig. 8. Impact of the maximum number of retransmissions on TCP throughput. (a) $p_f = 0, T_f = 0.4$ s, $\gamma_{target} = 2$. (b) $p_f = 0.01, T_f = 0.4$ s, $\gamma_{target} = 2$. (c) $p_f = 0, T_f = 0$ s, $\gamma_{target} = 2$. (d) $p_f = 0, T_f = 0$ s, $\gamma_{target} = 1.25$.

The TCP model appears to bring about appreciable errors for $p_f = 0.01$. The TCP model error magnitude is well on the order of other TCP models defined for lossy links (e.g., see the popular references [9] and [33]).

In general, the TCP model is the more accurate the smaller the packet loss is and the larger T_f is. Anyway, it gives useful values for every considered scenario and it can be used reliably to dimension ARQ and FEC parameters.

Simulation results of Fig. 8 have been carried out with TCP Reno as transport protocol. The same simulations have been performed with the NewReno version of TCP; Fig. 9 shows the comparison between Reno and NewReno performance. Although TCP NewReno slightly improves TCP Reno performance, thanks to a more effective Fast Recovery algorithm that avoids multiple Fast Retransmits, its behavior is not far from the one of TCP Reno; for this reason, TCP NewReno simulation results are not shown in the following discussion.

Given a code rate R_c, TCP throughput is a nondecreasing function of M_t; therefore, not limiting the maximum number of ARQ retransmission attempts appears to be the optimal choice as for the TCP performance.

As already pointed out in the introduction, some works argue that a persistent ARQ degree may cause spurious timeouts in the TCP sender with a consequent TCP throughput reduction. Although in our model we do not take into account possible spurious timeout events, there is no evidence of spurious timeout in our extensive (and accurate in our opinion) simulations.

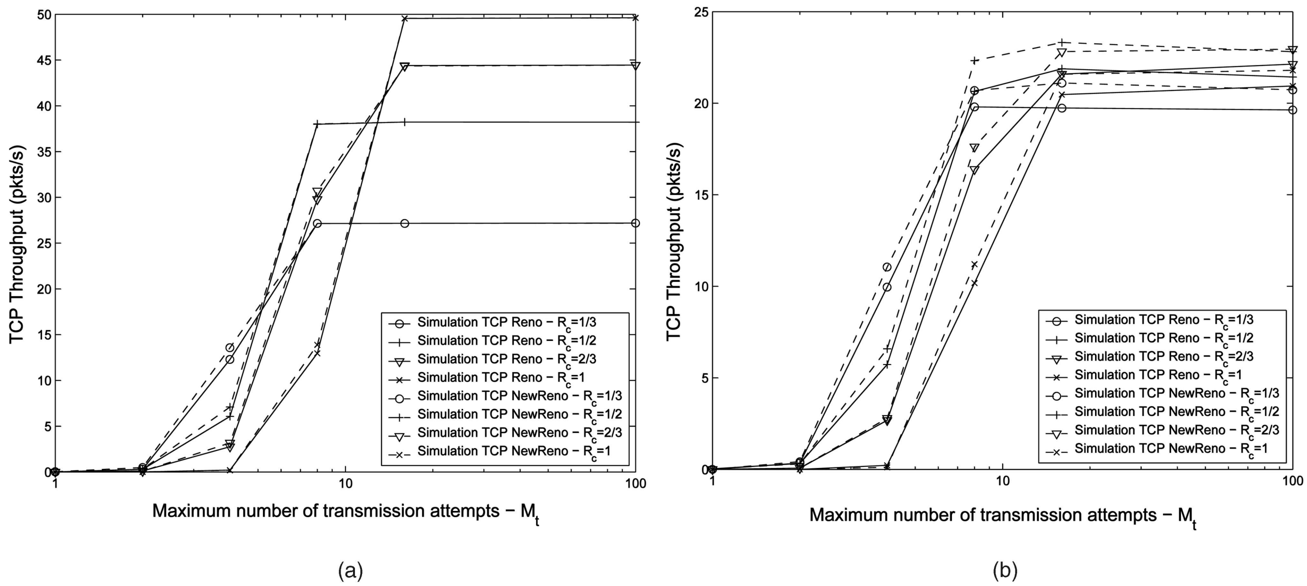

Fig. 9. Comparison between TCP Reno and TCP NewReno throughput ($\gamma_{target} = 2$, $T_f = 0.4$ s). (a) $p_f = 0$. (b) $p_f = 0.01$.

However, also assuming that spurious timeout events are significant, avoiding them (to TCP packet loss benefit) does not improve TCP performance. Suppose that a LL PDU belonging to the X packet reaches the maximum number of transmission attempts because of the persistent bad channel condition, and it is discarded by the wireless link layer; this fact induces the loss of the whole X packet (even PDUs received correctly are discarded). After X has been discarded, LL attempts to transmit PDUs belonging to the next packet $X+1$, if available. It is important to notice that there are no reasons that let the PDUs of $X+1$ arrive to destination before the instant in which the PDUs of X would have arrived if they would not be discarded. At this point, the TCP sender entity reacts to the loss either with a "necessary" fast retransmit or with a "necessary" (i.e., not spurious) timeout: If RTO is far to expire, the loss of the X packet would trigger a fast retransmit event that could be avoided. If the RTO expires, X is retransmitted either if X has been discarded ("necessary" timeout) and if X has not been discarded (spurious timeout). Moreover, in this second case, the "necessary" timeout induces the TCP sender to exit from the recovery procedure 1) later than a spurious timeout because the ACK generated by the X packet arriving after the spurious timeout acknowledges new data before the ACK generated by the TCP sender retransmission and 2) slower because the duplicate ACKs triggered by the packets received before X are not considered by the sender.

To sum up, we verified by means of a lot of simulations that spurious timeout events are negligible in the considered network scenario. Even if present, rare spurious timeout would become "necessary" timeout events if a limited number of retransmission attempts were implemented: So, no advantage is obtained by reducing ARQ persistence degree. Conversely, to implement fully persistent ARQ protocol gives a significant advantage since it avoids wireless packet losses that highly affect TCP performance.

The second issue we want to address is the effect of the ARQ persistence degree on the energy efficiency of the link layer with a hybrid ARQ protocol. For this purpose, we define the ratio between the number of useful bits and the number of bits actually transmitted per SDU:[7]

$$E_{ARQ}(M_t) = \frac{(l + l_b) \cdot n_{pdu}}{(l + l_b)/R_c \cdot n_{pdu} \cdot \mathcal{N}_{M_t}} = \frac{R_c}{\mathcal{N}_{M_t}}.$$

E_{ARQ} gives a direct indication of the energy efficiency obtained in the wireless link by limiting the number of transmission attempts to packet loss probability's disadvantage.

However, when TCP as transport layer is used, SDUs that are discarded by the wireless link layer are retransmitted by the end-to-end TCP error recovery mechanisms (Fast Retransmit or timeout). This effect is taken into account by defining another energy efficiency metric:

$$E_{TCP}(M_t) = E_{ARQ}(M_t) \cdot (1 - p_w(M_t)).$$

In Fig. 10, E_{ARQ} and E_{TCP} are shown for γ_{target} equal to 3 and 1.25 as a function of M_t for different values of R_c. As it can be noted by comparing the plots on the left with the corresponding ones on the right, where TCP retransmissions are accounted for, the higher efficiency is always obtained with an unlimited number of transmissions at the link layer, i.e., when the channel impairments are hidden to TCP. This can be explained considering that, since every packet dropped by the wireless link is replicated by TCP, limiting the maximum number of transmissions can reduce the average number of transmissions per PDU, but it increases the number of packets retransmitted by TCP. The global effect of this phenomenon is a loss in energy efficiency caused by a limited number of transmission attempts. If M_t is upper bounded by protocol limitations (i.e., when the ARQ protocol settings does not permit a large value of M_t), the maximum efficiency is obtained by

7. We are considering the l_b bits used for error detection as useful bits because necessary to implement ARQ protocol.

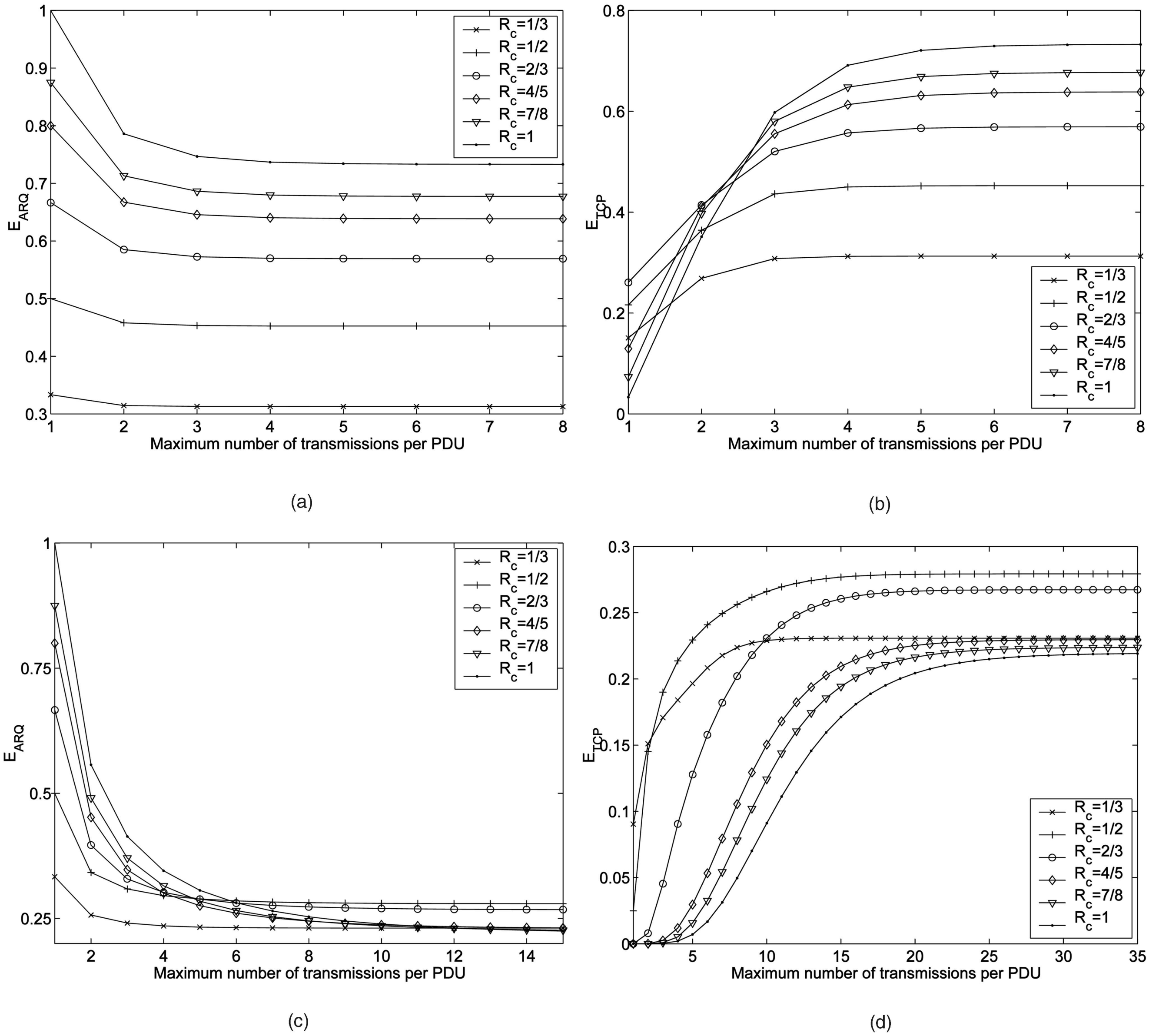

Fig. 10. Energy efficiency as function of M_t for $f_d = 10$ Hz. (a) $\gamma_{target} = 3$. (b) $\gamma_{target} = 3$. (c) $\gamma_{target} = 1.25$. (d) $\gamma_{target} = 1.25$.

using the maximum allowed value of M_t. The value of the optimal code rate is given by the envelope of the curves shown in Fig. 10b and Fig. 10d that depends on the value of γ_{target}.

It is worth noting that the TCP point of view reverses the indication coming from LL layer as to the best choice of M_t that optimizes the wireless link efficiency. This example shows the importance of a cross-layer analysis aiming at capturing the interplay of the procedures belonging to different layers interacting in the wireless link.

Since we verified that the best choice for the ARQ algorithm is to use a fully persistence degree, from now on, we only consider an unlimited number of ARQ retransmission attempts, i.e., $M_t = \infty$.

Finally, given a fully persistent ARQ protocol in the wireless link ($M_t = \infty$), we want to determine the optimal FEC code rate degree that guarantees TCP performance and channel efficiency.

In Fig. 11, the impact of FEC code rate R_c on TCP performance is shown for different values of γ_{target}. The maximum TCP throughput is obtained for values of the FEC code rate that depend on the particular value of γ_{target}: For large values of γ_{target} (e.g., 2 and 5), we obtain the maximum TCP throughput by not using any code (i.e., $R_c = 1$); conversely, for small values of γ_{target} (e.g., 1.25), the best TCP performance is obtained with a code rate about equal to $1/2$.

It is a well-known result that there exists a FEC code rate that maximizes the link efficiency, provided a fully persistent ARQ protocol for a given channel status. In fact, if, on the one hand, the presence of FEC code redundancy implies a waste of capacity utilized to protect information data, on the other hand, it reduces the average number of ARQ transmissions: There exist a value of FEC code rate that balances these two effects. This is a general result, independent of the used transport protocol (i.e., with UDP we would have same results). We find that, with $M_t = \infty$,

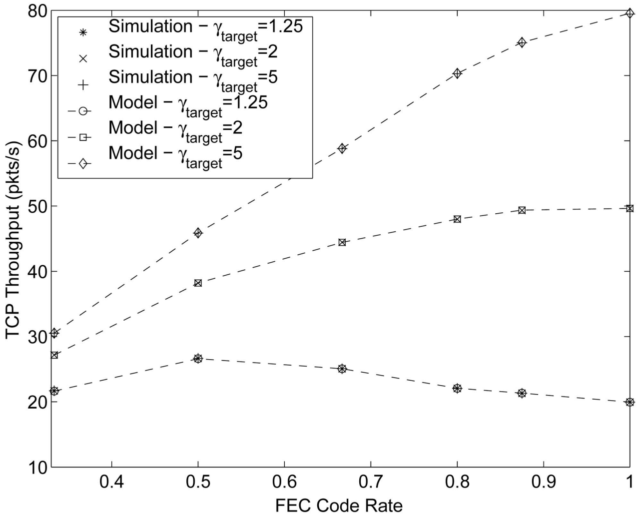

Fig. 11. Impact of the FEC code rate on TCP throughput ($p_f = 0$, $M_t = \infty$, $T_f = 0.4$ s).

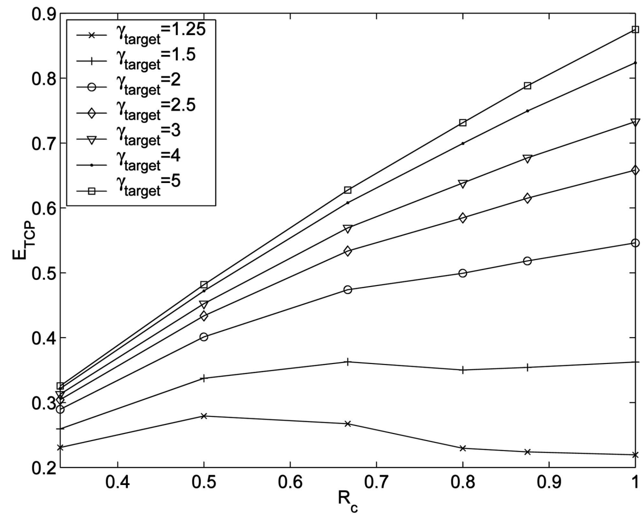

Fig. 12. Channel efficiency as function of FEC code rate ($M_t = \infty$).

the optimal value of FEC code rate that maximizes the efficiency E_{TCP} also yields the maximum TCP throughput. This is not true when M_t is limited; in fact, maximizing the efficiency $E_{TCP}(M_t)$ for a given M_t is equivalent to maximizing the available capacity C_d, i.e., the TCP bottleneck capacity; however, since $p_w > 0$ in this case (lossy link), TCP throughput could benefit from a reduction of the bottleneck capacity (due to a stronger FEC code), provided that this substantially reduces the packet loss probability p_w.

In Fig. 12, we show the efficiency E_{TCP}[8] as a function of the FEC code rate R_c. Different values of γ_{target}, are considered (from 1 to 5). Comparing Fig. 12 and Fig. 11 for the same values of γ_{target}, it turns out that, given a fully persistent ARQ protocol, the maximum TCP throughput is obtained when channel efficiency is maximum as well.

The degree of FEC code that maximizes both TCP throughput and channel efficiency depends on the channel quality. This is confirmed by the Fig. 13, where TCP throughput is shown as function of γ_{target}, for different values of the FEC code rate. Fixing the other parameters, the best FEC code rate depends on the particular values of γ_{target}: As γ_{target} increases, the code rate required to obtain the best TCP performance also increases.

Since the proposed analytic model is able to numerically compute the efficiency, by exploiting it, it is possible to choose dynamically the optimal FEC code rate according to some measurements of the actual state of the physical channel (e.g., γ_{target} and $f_d \cdot \tau$ could be estimated and then the models in Sections 3.1 and 3.2 could be applied). However, this is beyond the aim of the present work and, therefore, it is not further investigated here.

5 Conclusions

In this work, we investigate the interactions between hybrid FEC/ARQ mechanisms and TCP procedures. The main contribution is to give some precise indications on the design of the link layer protocol for mobile computing applications. We have shown that there are no practically useful alternatives to the adoption of a fully persistent ARQ protocol on the wireless link to guarantee TCP performance. Furthermore, we verified that, in case of a fully persistent ARQ, choosing the FEC code values that optimizes channel efficiency is also the best choice from the point of view of TCP performance.

To achieve such conclusions we developed: 1) a mapping of physical radio channel parameters and of modulation and coding parameters to a two-state Markov model parameters, 2) an analytic model for a generic selective repeat ARQ protocol, and 3) a TCP model in a wired-cum-wireless network scenario. In spite of quite rough approximations to define relatively simple models, extensive and accurate simulations point out satisfactory agreement with model results. The model appears to capture the key phenomena that impact end-to-end TCP performance; we think that the complexity explosion that would incur with a more detailed model is hardly paid back by a reduction of model errors.

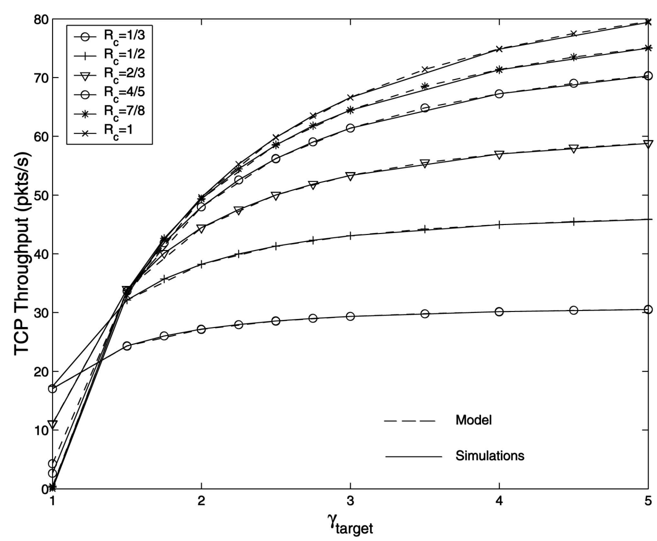

Fig. 13. Impact of γ_{target} on TCP throughput ($p_f = 0$, $M_t = \infty$, $T_f = 0.4$ s).

8. Note that, if the ARQ protocol is fully persistent, E_{ARQ} and E_{TCP} are equivalent since $p_w = 0$.

ACKNOWLEDGMENTS

This work was partially supported by the Italian FIRB Project PRIMO.

REFERENCES

[1] N. Brownlee and K.C. Claffy, "Understanding Internet Traffic Streams: Dragonflies and Tortoises," *IEEE Comm. Magazine,* Oct. 2002.
[2] G. Xylomenos, G.C. Polyzos, and P. Mahonen, "TCP Performance Issues over Wireless Links," *IEEE Comm. Magazine,* Apr. 2001.
[3] C. Barakat, E. Altman, and W. Dabbous, "On TCP Performance in a Heterogeneous Network: A Survey," *IEEE Comm. Magazine,* Jan. 2000.
[4] C. Barakat and E. Altman, "Bandwidth Tradeoff between TCP and Link-Level FEC," *Computer Networks,* June 2002.
[5] B. Liu, D.L. Goeckel, and D. Towsley, "TCP-Cognizant Adaptive Forward Error Correction in Wireless Networks," technical report, Computer Science Dept., Univ. of Massachusettes, Aug. 2001, http://www-net.cs.umass.edu/papers/papers.html.
[6] L. Galluccio, G. Morabito, and S. Palazzo, "An Analytical Study of a Tradeoff between Transmission Power and FEC for TCP Optimization in Wireless Networks," *Proc. IEEE Infocom Conf.,* 2003.
[7] H. Balakrishnan, S. Seshan, and R.H. Katz, "Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks," *Wireless Networks,* Dec. 1995.
[8] A. DeSimone, M.C. Chuah, and O.C. Yue, "Throughput Performance of Transport-Layer Protocols over Wireless LANs," *Proc. IEEE Globecom Conf.,* 1993.
[9] H.M. Chaskar, T.V. Lakshman, and U. Madhow, "TCP over Wireless with Link Level Error Control: Analysis and Design Methodology," *IEEE/ACM Trans. Networking,* vol. 7, no. 5, Oct. 1999.
[10] R. Ludwig, "A Case for Flow-Adaptive Wireless Links," Technical Report UCB/CSD-99-1053, Univ. of California at Berkeley, May 1999.
[11] R. Ludwig and R.H. Katz, "The Eifel Algorithm: Making TCP Robust against Spurious Retransmissions," *ACM Computer Comm. Rev.,* vol. 30, no. 1, Jan. 2000.
[12] M. Methfessel, K. Dombrowski, P. Langendörfer, H. Frankenfeldt, I. Babanskaja, I. Matthaei, and R. Kraemer, "Vertical Optimization of Data Transmission for Mobile Wireless Terminals," *IEEE Wireless Comm. Magazine,* Dec. 2002.
[13] F. Vacirca, A. DeVendictis, and A. Baiocchi, "Investigating Interactions between ARQ Mechanisms and TCP over Wireless Links," *Proc. European Wireless Conf.,* Feb. 2004.
[14] C.F. Chiasserini and M. Meo, "A Reconfigurable Protocol Setting to Improve TCP over Wireless," *IEEE Trans. Vehicular Technology,* 2002.
[15] A. Chockalingam, M. Zorzi, V. Tralli, "Wireless TCP Performance with Link Layer FEC/ARQ," *Proc. Int'l Conf. Comm.,* 1999.
[16] C. Barakat and A. Al Fawal, "Analysis of Link-Level Hybrid FEC/ARQ-SR for Wireless Links and Long-Lived TCP traffic," INRIA technical report, Feb. 2003.
[17] D. Barman, I. Matta, E. Altman, and R. ElAzouzi, "TCP Optimization through FEC, ARQ and Transmission Power Tradeoffs," *Proc. Conf. Wired/Wireless Internet Comm.,* Feb. 2004.
[18] R.W. Stevens, *TCP/IP Illustrated, volume I The Protocols.* Addison-Wesley, 1994.
[19] V. Paxson, "End-to-End Internet Packet Dynamics," *IEEE/ACM Trans. Networking,* vol. 7, no. 3, June 1999.
[20] A. DeVendictis and A. Baiocchi, "Wavelet Based Synthetic Generation of Internet Packet Delays," *Proc. Internet Technical Committee Conf.,* Dec. 2001.
[21] E.N. Gilbert, "Capacity of Burst Noise Channels," *The Bell System Technical J.,* vol. 39, pp. 1253-1256, 1960.
[22] 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 Universal Personal Comm.,* Nov. 1995.
[23] W.C. Jakes, *Microwave Mobile Communications.* John Wiley and Sons, 1974.
[24] J.G. Proakis, *Digital Communications,* fourth ed. McGraw-Hill, 2001.
[25] M. Patzold, U. Killat, and F. Laue, "A Deterministic Digital Simulation Model for Suzuki Processes with Application to a Shadowed Rayleigh Land Mobile Radio Channel," *IEEE Trans. Vehicular Technology,* vol. 45, no. 2, May 1996.
[26] M. Allman and A. Falk, "On the Effective Evaluation of TCP," *ACM Computer Comm. Rev.,* vol. 29, no. 5, Oct. 1999.
[27] C. Casetti and M. Meo, "A New Approach to Model the Stationary Behavior of TCP Connections," *Proc. IEEE Infocom Conf.,* Mar. 2000.
[28] J. Padhye, V. Firoiu, and D. Towsley, "A Stochastic Model of TCP Reno Congestion Avoidance and Control," CMPSCI Technical Report, Jan. 1999.
[29] F. Vacirca, A. DeVendictis, A. Todini, and A. Baiocchi, "On the Effects of ARQ Mechanisms on TCP Performance in Wireless Environments," *Proc. IEEE Globecom 2003 Conf.,* Dec. 2003.
[30] ns-LBL network simulator ns-2. 1b9a, documentation and software available via http://www.isi.edu/nsnam/ns/index.html, 2006.
[31] A. Todini and F. Vacirca, "UMTS Module for ns," http://net.infocom.uniroma1.it, 2006.
[32] TS25.322, *Radio Link Control (RLC) Protocol Specification,* Release 5 v5.1.0, June 2002.
[33] J. Padhye et. al, "Modeling TCP Reno Throughput: A Simple Model and Its Empirical Validation," *IEEE/ACM Trans. Networking,* vol. 8, no. 2, Apr. 2000.

Francesco Vacirca received the Laurea degree in telecommunications engineering in 2001 from University of Rome "La Sapienza," Italy. From September 2001 to December 2002, he participated in the Ramon project, working on reconfigurable wireless systems. He is currently a PhD student in the Infocom Department at the University of Rome "La Sapienza" and he is a visiting researcher at the Telecommunications Research Center Vienna. His research focuses on wireless network optimization for mobile computing applications. He is a student member of the IEEE.

Andrea De Vendictis received the Laurea degree in telecommunications engineering in 1999 and the Dottorato di Ricerca (PhD degree) in information and communications engineering in 2004, both from the University of Roma "La Sapienza." From April 2000 to March 2001, he worked in the telecommunications research consortium CoRiTeL, where he took part in the research activity of the IPTM (IP Traffic Modeling) project. Currently, he is in the INFOCOM Department at the University of Roma "La Sapienza" where he is involved in the italian FIRB PRIMO project. His main research interests regard TCP performance evaluation and Internet congestion control. He is a student member of the IEEE.

Andrea Baiocchi received the Laurea degree in electronics engineering in 1987 and the Dottorato di Ricerca (PhD degree) in information and communications engineering in 1992, both from the University of Roma "La Sapienza." Since 1998, he has been an associate professor in communications in the School of Engineering at the University "La Sapienza." His main scientific contributions are on traffic control in ATM networks, queuing theory, and radio resource management in cellular mobile networks. His current research interests are focused on traffic models and dimensioning algorithms for IP networks and on mobile computing, specifically TCP adaptation and packet scheduling over the wireless access interface. These activities have been carried out also in the framework of national (CNR, MIUR) and international (European Union, ESA) projects. He has published more than 60 papers on international journals and conference proceedings and he has participated to the Technical Program Committees of several international conferences. He is a senior member of the IEEE.