Analysis and Heuristics for the Characterization of Selective Repeat ARQ Delay Statistics Over Wireless Channels

Michele Rossi, *Student Member, IEEE,* and Michele Zorzi, *Senior Member, IEEE*

***Abstract*—In this paper, we consider a point-to-point wireless transmission where link layer ARQ is used to counteract channel impairments. In particular, we refer, as an example, to a 3G cellular system, where a dedicated channel is used between a mobile terminal and its serving base station. Our aim is to find accurate and fast heuristics for the characterization of link layer and higher level (e.g., application level) packet delay. Existing methods to obtain such statistics are often based on recursive computations or large-sized matrix manipulations. For these reasons, they are too complex to be successfully applied in a mobile terminal due to memory, delay, and energy constraints. In this paper, we first present an analytical framework to compute link-layer packet delivery delay statistics as a function of the packet error rate; then we extend the model in order to find the statistics related to higher level packets (i.e., to aggregates of link layer packets). Both in-order and out-of-order delivery of link-layer packets to higher levels are considered. The goodness of the channel model considered in the analysis is proved by means of accurate channel simulation results. The obtained statistics are then characterized by highlighting their properties as a function of the round-trip time and the error rate at the link layer. Finally, fast and accurate heuristics are derived directly from the analysis. These heuristics are very simple (piecewise linear functions), so they can be effectively used in a mobile terminal to obtain accurate delay statistics estimates with little computational effort.**

***Index Terms*—ARQ delay statistics, delay analysis, heuristics, higher layer QoS, selective repeat ARQ protocols.**

I. Introduction

In THIS paper, we focus on a third-generation (3G) cellular system where wide-band code-division multiple access (W-CDMA) is used on the air interface as the channel access technique. 3G systems are intended to provide global mobility support, with wide range of services including telephony, paging, messaging, Internet, high-quality video, and broadband data. To realize a certain network quality of service (QoS), a bearer service with clearly defined characteristics and functionalities has to be set up from the source to the destination. The characteristics of this bearer service depend on the kind of traffic in which the user is interested. One of the challenges in such complex systems is to counteract the errors due to the wireless medium (propagation phenomena) and to the interference. With this aim at the physical layer, both channel coding and interleaving are employed. In addition, ARQ techniques may be used at the link layer to reduce the residual error probability at the output of the physical layer. ARQ solutions use queueing and retransmission of lost packets to combat channel errors. The effect of link-layer retransmissions is twofold. On the one hand, they are able to reduce the residual packet error probability. On the other hand, they introduce a random delay on the delivery of link-layer packets that translates into a variable delay for the application layer flow. To better understand the importance and the implications derived from these additional delays, let us refer to a video/audio streaming data transfer. In that case, packets arriving too late at the receiver should be discarded, as they do not respect the timing constraints imposed by the streaming application running at higher layers. That is, due to the ARQ retransmissions process, the application streaming buffer at the receiver becomes empty *(buffer starvation)*, which results in gaps in the playout process. This results in a sudden degradation of the user perceived performance. Therefore, when ARQ is employed, the degree of satisfaction of the user is directly affected by the error/delay statistics at the link-layer output. In other words, the correct understanding of the delays involved in the ARQ retransmission process and their effect on higher layers performance is a pivotal point to provide and maintain the user desired QoS.

The focus of this paper is on the characterization of the delay statistics when ARQ is utilized at the link layer. The fundamental ARQ schemes can be classified into Stop-and-Wait (SW), Go-back-N (GBN) and Selective Repeat (SR) ARQ [1], [2]. SR-ARQ is widely used due to its superior throughput performance. In ARQ schemes, the transmitter sends packets (PDUs) consisting of payload and error detection codes (a cyclic redundancy check field is inserted into each packet). At the receiver side, based on the result of the error detection procedure, acknowledgment messages are sent back to the transmitter (ACK or NACK, according to the result of error detection). The sender schedules packet retransmissions based on such messages.

In the presence of an ARQ protocol, we can subdivide the overall PDU delay into three contributions [3]. The first contribution is the *queueing delay in the source buffer,* i.e., the time between the PDU release by higher layers and the instant of its first transmission over the channel. This term depends on the channel behavior, the ARQ technique, and the PDU arrival process. The second contribution is commonly referred to as *transmission delay* and is the time elapsed between the first transmission

Manuscript received January 29, 2003; revised April 16, 2003. This work was supported by ERICSSON Research. This work was presented in part at IEEE VTC-Spring 2003, Jeju, Korea.

The authors are with the Department of Engineering, University of Ferrara, 44100 Ferrara, Italy (e-mail: mrossi@ing.unife.it; mzorzi@ing.unife.it).

Digital Object Identifier 10.1109/TVT.2003.815980

and the correct reception of the PDU. This term depends on the channel behavior and on the ARQ retransmission technique. The last contribution, referred to as *resequencing delay*, is the time spent in the receiver resequencing buffer. In more detail, even if the sender transmits packets in order, due to random errors and consequent retransmissions, they can be received out of sequence. In that case, PDUs with higher identifiers must wait in the receiver resequencing buffer until all the PDUs with lower identifiers have been correctly received. By considering a tagged PDU, this last term depends on errors experienced by all PDUs sent in the same round-trip in which the tagged one has been transmitted for the first time. The *resequencing delay* is nonzero when *in-order delivery* is considered. Another possibility is to configure the link layer in the *out-of-order* delivery mode. In that case PDUs are passed to higher layers without waiting for the correct reception of out of sequence packets and the resequencing delay at the link layer is equal to zero. We refer as *delivery delay* to the sum of *transmission* and *resequencing* delay.

Several studies have been performed on the delay performance of the SR-ARQ protocol over a wireless channel [3]–[12]. In [5], Konheim derived the exact single link-layer PDU delay distribution considering a finite round-trip delay and an independent (i.i.d.) error process. The independent channel has been considered by several other authors [6]–[8], [10]. Rosberg and Shacham in [7] derive the distribution of the buffer occupancy and of the resequencing delay at the receiver under a heavy traffic assumption[1] to give the network designer some guidelines on the choice of the buffer capacity at the receiver. Rosberg and Sidi in [8] analyzed the joint distribution of transmitter and receiver buffer occupancies. In [13], the authors investigate the effect of forward/backward channel memory on ARQ error strategies using flowgraph analysis; GBN- and SR-ARQ are compared in terms of their throughput efficiency. In [9], Zorzi and Rao studied the adequateness of two- and three-state Markov channel models for predicting delay of ARQ/queueing systems in correlated fading channels. In that work, they show that a three-state Markov model, in many cases, can provide very accurate delay predictions, whereas a simple two-state model can lead to good throughput estimates but is inadequate for predicting delays. In [4], Fantacci investigates the SR-ARQ performance over a time-varying channel [14]–[17] deriving the mean packet delay and the mean queue length for both the zero and the finite round-trip delay case. Some interesting results on how different error statistics affect ARQ performance are reported by Lu and Chang in [11]. In particular, they considered both the kth-order Markovian channel error model and the gap error model. In [3], Kim and Krunz account for a time-varying channel, a finite round-trip delay, and a Markovian traffic source. Here, a mean analysis is developed for all the ARQ delay contributions. The delivery delay distribution in the out-of-order delivery case has been studied in [12], where a general framework is presented to obtain redundancy check failure, throughput efficiency, and single PDU delivery delay performance.

[1]A new PDU to be sent over the channel is always available at the transmitter ARQ buffer.

The main drawback of the analytical approaches presented so far in the literature is their computational complexity that makes their usage very difficult in a mobile terminal equipment (ME) due to energy, memory, and time constraints. These techniques are time-consuming due either to the manipulation of matrices whose size quickly increases with the number of link layer PDUs sent in a full link layer round-trip time m or to the presence of recursive formulations that are memory and time-consuming as m increases. For instance, the complexity of the algorithm proposed in [5] increases exponentially with the round-trip time value. Hence, a real-time fast and accurate estimate of the delay statistics experienced both by link-layer PDUs and higher layer packets (seen as an aggregate of link layer PDUs) still remains an open issue. These estimates can be useful to obtain *delay-aware* cost functions to be used, for example, when delay-sensitive flows are transmitted over the channel to adapt link and/or physical layer settings to the delay requirements imposed by those flows. In real-time services, packets arriving too late are useless; as they are passed to the application layer, they are discarded, and the system resources consumed for their transmission (and possible retransmissions if a link layer is considered) are wasted. Hence, the control of the maximum delay perceived by application layer packets (that translates in the control of the maximum number of allowed retransmissions at the link layer) is a pivotal point in order to save system resources while achieving a better quality (saved resources can be exploited to send new data instead of useless retransmissions of old packets).

The simplest measure for the quality perceived by the final user is the residual packet[2] error probability at the output of the link layer $(P_{\rm pkt})$. However, when delay-constrained flows are considered, this metric is no longer sufficient, because packets whose time deadline has expired (even if correctly received) are discarded and are a source of error as well as corrupted packets. The presence of a link layer is able to reduce the residual error rate $P_{\rm pkt}$, but the delay experienced by the packets becomes stochastic due to the random nature of the errors affecting the wireless link. In this case, a more accurate estimate of the packet drop rate is needed. Such an estimate can be obtained accounting for the delay constraints as follows: $P_{\rm drop} = P_{\rm pkt} + (1 - P_{\rm pkt}){\rm Prob}\{{\rm delay} > d_{\max}\}$, i.e., a packet is considered erroneous either if it is corrupted or if the delay that it experiences during its transmission over the wireless link exceeds some delay constraint $(d_{\max})$, where $d_{\max}$ is intended as the maximum delivery delay that (for instance) an IP packet can tolerate when it is transmitted over the wireless link.

The chief aim of this paper is to derive analysis to characterize the delivery delay statistics Prob{delay$>$ $d_{\max}$} for both the *in-order* and the *out-of-order delivery* case. Moreover, from the manipulation of the analytical results, we directly derive simple and accurate heuristics able to describe such statistics with minimal computational effort. The merit of this paper is to go further into the characterization of ARQ delivery delay statistics, investigating also the delivery delay regarding higher layers packets [link layer service data units (SDUs)], that in our analysis are

[2]Here with the term *packet* we mean the data packet unit assembled at the output of the link layer that, in general, is composed by an aggregate of several link layer PDUs.

considered composed of an integer number (K) of link-layer packets. Simple, fast, and accurate heuristics for the estimation of SDU statistics are reported considering both in-order and out-of-order delivery of link-layer SDUs. These heuristics could be effectively used, for example, in channel adaptive algorithms as an estimate of the delay perceived by packets belonging to the application layer data flow *(delay utility functions).*

The remainder of this paper is organized as follows. In Section II, referring (as an example) to a Universal Mobile Telecommunications System (UMTS), we report some graphs obtained by simulation to characterize the link-layer PDU error process accounting for both interleaving and channel coding. Based on the reported measurements, we can affirm that the channel model that we use in the rest of this paper to develop analysis and heuristics is accurate, i.e., it gives satisfactory results for typical physical layer settings. In Section III, we present analysis for the evaluation of the delivery delay statistics for a single PDU and for an aggregate of link-layer PDUs considering in-order delivery of link-layer PDUs (Section III-A, C, and D). Some properties of such statistics are highlighted in Section III-B and C for the delivery and transmission statistics, respectively. In Section IV, we approximate the SDU delivery delay statistics obtained in Section III-D by means of simple and accurate heuristics. The out-of-order delivery delay statistics are investigated in Section V, where an approximate but very accurate analysis is reported. Finally, in Section VI, some conclusions are given.

II. Discussion on the Channel Model

In this section, as a sample scenario, we refer to a UMTS cellular system where W-CDMA is used as the radio interface ([18]–[20]) and we evaluate by simulation how the PDU error process at the UMTS link layer (RLC [21]) is characterized in terms of burst length (b) and average PDU error rate (p). Both channel coding (convolutional with rate 1/2) and interleaving are considered. The simulated scenario is composed of nine hexagonal cells, where a base station (Node B) is placed at the center of each cell and a given number of users are considered to be able of moving inside the coverage area (in the results reported here we consider a population of 100 users with speed uniformly distributed in {0, 7, 49} Km/h). The whole cell structure is wrapped around in order to avoid border effects. As an example scenario, we consider that each user is using a downlink dedicated channel, where the traffic source is continuous, as can occur when either streaming flows or FTP file transfers are considered. The cell radius is 200 m, and a downlink dedicated channel (DCH) is allocated for each user where the minimum and maximum powers are $P_{\min}^{\text{downlink}} = -15$ dBW and $P_{\max}^{\text{downlink}} = -5$ dBW, respectively. Path loss, shadowing, and multipath fading phenomena are considered as well. In Figs. 1 and 2, the average PDU error burst length (b) is reported as a function of the mean PDU error rate (p). Vertical error bars are reported to indicate the confidence interval (95%) of mean burst length measurements, whereas plus and cross symbols are used to plot the eightieth percentile of the burst length b. In the first figure, we consider a link layer with a logical[3] bit rate of 30 Kbps and a large interleaving depth (80 ms), whereas in Fig. 2 the RLC bit rate is higher (120 Kbps) and a smaller (40 ms) time transmission interval (TTI, i.e., the interleaving depth) is considered. As a first observation, one can note that at low RLC bit rates, the error process tends to be independent (the i.i.d. case is reported for comparison as a solid bold line in both figures). Hence, the independent PDU error assumption in this case seems to be a good approximation. Instead, as the RLC logical bit rate increases (Fig. 2), more PDUs can be sent in each radio frame and the PDU error process is likely affected by longer bursts. It is worth noting that a larger TTI has a beneficial effect on system performance. The effect of a longer TTI is twofold: on the one side the RLC round-trip delay increases (negative effect). On the other side, the interleaving is more effective in breaking the error bursts at the bit level (positive effect). Moreover, in the case where TTI$= 80$ ms, a typical value for the RLC round-trip delay is 220 ms. Hence, considering a typical RLC PDU size of 360 bits, we have that $m \approx 75$ PDUs can be sent during a full RLC round-trip delay. Referring again to Fig. 2, we can note that $b \ll m \approx 75$ and in that case (see Fig. 3, where we report the delivery delay distribution obtained

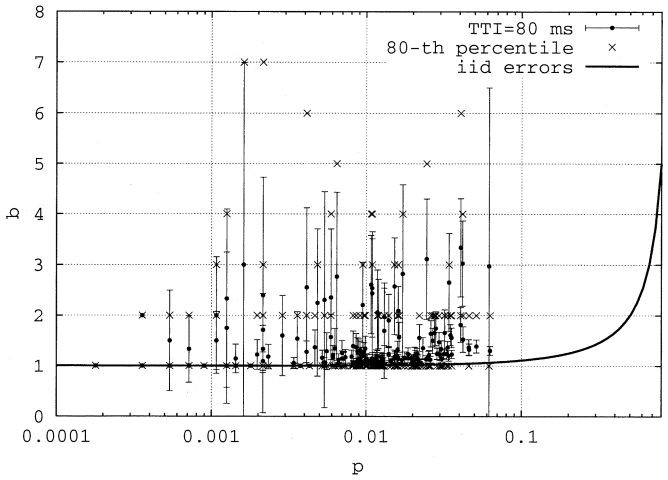

Fig. 1. DCH spreading factor $= 128$ (RLC bit rate $= 30$ Kbps), code rate $= 1/2$, TTI $= 80$ ms, RLC PDU length $= 360$ bits.

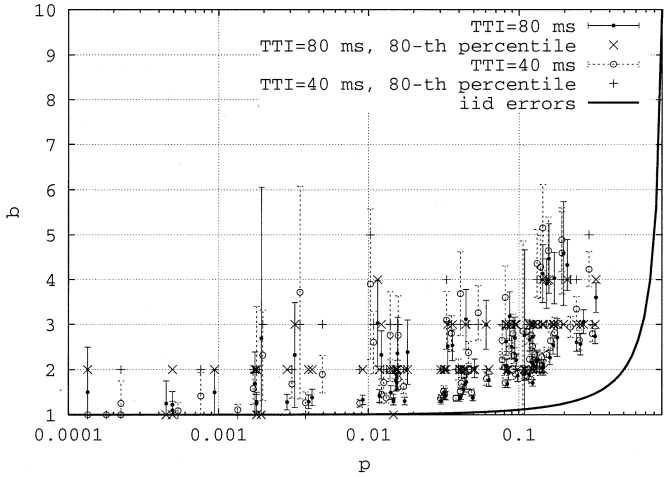

Fig. 2. DCH spreading factor $= 32$ (RLC bit rate $= 120$ Kbps), code rate $=$ 1/2, TTI $\in$ {40, 80} ms, RLC PDU length $= 360$ bits.

[3]The available bit rate before coding, i.e., the useful data bit rate.

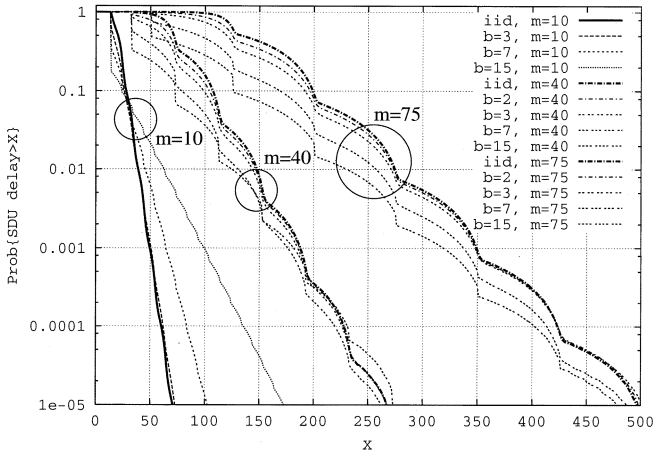

Fig. 3. Link-layer SDU complementary cumulative delivery delay distribution as a function of the burst length b considering $m \in \{10, 40, 75\}$, $p = 0.1$, and $K = 14$.

for an aggregate of K RLC PDUs (RLC SDU) considering a two-state channel error model [14]–[17] for a fixed PDU error rate $p = 0.1$), the delivery delay statistics of a RLC SDU, after a few round-trip times, is quite close to the one achieved for the independent case (the burst length observed in the simulations is lower than three for $p = 0.1$). In conclusion, the i.i.d. channel model for a UMTS system can be a good approximation for the link-layer error process. This is mainly due to the large interleaving value used at the physical layer and (at high bit rates) to the large value of the round-trip delay ($b \ll m$). Moreover, one can note from Fig. 3 that an approximation for the bursty case can be achieved by a rigid rotation of the i.i.d. curve. This rough approximation, in many cases, should suffice as a heuristic to drive delay adaptive algorithms. For these reasons, in this paper we consider an independent link-layer packet error process. This model, despite its simplicity, in many cases gives a good approximation for the actual error process as seen at the link layer of a UMTS system. The extension to the correlated case is being studied.

III. Analysis

In general, the link layer of a third-generation cellular system (see, for instance, [21] for the UMTS case) can operate either in the unacknowledged mode (UM) or in the acknowledged mode (AM). Moreover in AM, higher layer packets (SDUs) can either be released *in-order* or *out-of-order* to upper layers. In the *in-order delivery* case, a given SDU, say, SDU number i, is released to higher levels only after all SDUs with lower identifier have been correctly received or discarded. An SDU is said to be discarded when one or more PDUs composing it reached the maximum number of retransmission attempts without being correctly received. In this last case, the SDU may be passed to the higher layer and is marked as erroneous. In the *in-order-delivery* case, we refer to the PDU delivery delay as the time elapsed between the instant where the PDU is transmitted for the first time over the channel and the instant where it can be released in-order to upper levels, i.e., no outstanding PDUs with lower identifier are present. The SDU delivery delay is defined as the time elapsed between the instant where the first PDU composing the SDU is transmitted for the first time over the channel and the instant where all the PDUs composing it have been received correctly in-order. Instead, when the RLC is configured to support the *out-of-order delivery* of SDUs, a given SDU, say, SDU i, may be passed to higher layers when all the PDUs composing it have been correctly received (or discarded), regardless of the transmission status (correctly received, delivered, in flight) of other SDUs. In this case the SDU delivery delay is equal to the SDU transmission delay and corresponds to the time elapsed between the instant in which the first PDU composing a SDU is transmitted for the first time over the channel and the instant where all the PDUs composing the SDU have been correctly received. In this case, the SDU can be passed to upper levels regardless of the status of previously transmitted SDUs. Among other settings, another important parameter is the maximum number of retransmission attempts permitted for each link layer PDU (referred to as L in this paper). In Section IV, we first investigate the delivery delay statistics considering in-order-delivery of link-layer SDUs and limited retransmission attempts, whereas in Section V the out-of-order delivery of link layer SDUs is considered.

In the remainder of this paper, the following quantities are computed.

1) In Section III-A, we compute the single PDU delivery delay distribution in the in-order delivery case $P_d[t]$.
2) The transmission delay statistics regarding a link-layer SDU $P_{tx}[K,t]$ is derived in Section III-C. K is the (integer) number of PDUs composing a link layer SDU.
3) In Section III-D, the link-layer SDU delivery delay statistics in the in-order delivery case $P_d[K,t]$ is reported. From $P_d[K,t]$, the SDU complementary cumulative delivery delay distribution (ccdf$[K,t]$) is obtained.
4) Based on the analysis, an accurate approximation for the complementary cumulative distribution in the in-order delivery case ccdf$[K,t]$ is derived in Section IV.
5) Finally, in Section V, a very accurate and simple approximation for the SDU delivery delay in the out-of-order delivery case is reported.

A. Single PDU Delivery Delay Statistics for In-Order Delivery of RLC SDUs

The problem to be solved is to find the delivery delay distribution of an aggregate of K link layer PDUs.[4] Following the procedure discussed (and justified) later, in Section III-D, this statistics can be computed by means of a discrete-time convolution between two statistics: the single PDU delivery delay distribution ($P_d[t]$) and the K PDU's transmission time distribution ($P_{tx}[K,t]$). $P_d[t]$ gives the probability that a given single PDU is delivered in-order in exactly t slots,[5] whereas $P_{tx}[K,t]$ is used to track the probability that K new PDUs are transmitted over the channel in t slots. Both $P_d[t]$ and $P_{tx}[K,t]$ depend on L, RTT and p, i.e., the maximum number of retransmissions allowed for each PDU, the round-trip time and the PDU error probability, respectively.

In this section, we compute the delivery delay regarding a single PDU, i.e., the time needed by a PDU transmitted at time t to be correctly delivered at the receiver side. We consider in-order

[4]Throughout this paper, we consider that an SDU is composed by an integer number (K) of link layer PDUs.

[5]The slot duration corresponds to the packet transmission time.

delivery, i.e., the PDU is delivered in-order only if all PDUs with lower identifier transmitted before it have been correctly received or have been transmitted $L+1$ times without any success. A given PDU can be released to higher layers only when all PDUs with lower identifier have been released. In the following analysis, we say that a PDU is *resolved* in the slot where it is either correctly transmitted or discarded (due to the reaching of the maximum number of allowed transmissions, i.e., transmitted $L+1$ times without any success). We label a PDU as *unresolved* if it has been transmitted fewer than $L+1$ times without success.

We consider a pair of nodes that are exchanging packets through a noisy wireless link where the link layer is configured in the AM mode. The time is slotted, and the slot time corresponds to the single PDU transmission time. Moreover, PDUs are transmitted continuously, i.e., there are no empty slots (*heavy traffic assumption;* see [7] and [3]). This assumption can be justified in the case where an FTP file transfer or a video/audio streaming flow is considered. In these cases, in fact data are likely sent back-to-back without idle times in order to respect the timing imposed by the video/audio data flow and to avoid underutilization of the assigned resources. These are very important classes of service for which we expect such kind of behavior. Nevertheless, in the cases where this is not true, the heavy traffic assumption, at least for the in-order delivery delay, is surely a worst case delay analysis. This can be explained as follows: in the in-order delivery case, a tagged packet has to wait for the correct reception of all (and only) the other packets sent in the same round in which it has been transmitted for the first time (see later in this section for a justification of this fact). In the heavy traffic case, we always have that the number of packets sent in a round is at its maximum value (i.e., the number of packets that can be transmitted in one round trip). Hence, we also have the highest probability that at least one packet is corrupted in that round, i.e., that the tagged packet will have to wait for out-of-order packets.

Moreover, let m be the (integer) number of PDUs transmitted in a full RLC RTT. We consider that an acknowledgment message indicating the delivery status of a given PDU transmitted in the generic slot t is available at the sender at the beginning of slot $t+m$. Now consider a tagged PDU transmitted in slot t. Moreover, consider all PDUs transmitted in slot $\{t-m+1, t-m+2, \ldots, t-1\}$, i.e., the last $m-1$ PDUs transmitted before the tagged one (we say that these PDUs are the packets transmitted in the same window in which the tagged PDU has been transmitted for the first time; in the analysis we refer to these PDUs as *blocking PDUs*). Then, we refer as *starting window* the set of the tagged PDU plus these $m-1$ packets. Note that these $m-1$ PDUs are the only ones that the tagged PDU must eventually wait for after its correct delivery, i.e., the only PDUs with lower identifier. In fact, a PDU sent for the first time in window position i is resent in the same window position until success or until $L+1$ transmission failures. In any case, only from this point on can a new PDU be transmitted over the channel using the same window position occupied by such PDU. Given this transmission/retransmission behavior and that PDU identifiers are incremented sequentially in increasing order, it is clear that each PDU must eventually wait for the resolution (intended either as correct reception or the $(L+1)$st transmission failure) only of the PDUs contained in its starting window. This simple but effective model to characterize ARQ has been widely used in the literature so far (e.g., [3] is the most recent work using it). Here, we use it to characterize the delay performance of a 3G link-layer scheme. Note that 3G retransmission schemes have been enriched by new timers and features with respect to traditional ARQ solutions. These additional features have been necessary to ensure a fault-tolerant fully programmable and deadlock free scheme that, however, basically remains a selective repeat algorithm. Moreover, our idealized scheme can be seen as a best case for the delay performance achieved using a 3G compliant link layer because in our model, packets are always acknowledged in the shortest possible time ($m-1$ slots), so the time spent between their previous transmission and the following retransmission is also minimized. As a consequence, the delivery delay for a given link-layer packet is the shortest as well. The only configuration where this is not true is when the link layer is programmed to transmit multiple copies of the same packet in the same round-trip. However, this is an energy-inefficient, resource-inefficient, and particular case designed to decrease delivery delay at the cost of a degraded throughput performance. Nevertheless, our results are still valid as a best case estimate that is useful to obtain delay evaluations (that is the aim of this paper). Extensions of the model to better describe a 3G link layer behavior are being studied. However, it is worth noting that the insights derived in this paper are general and still hold in a more realistic scenario.

In the following, we compute the delivery delay statistics for the tagged PDU. With the term PDU delivery delay, we indicate the number of slots elapsed between the instant in which the tagged packet is transmitted for the first time over the channel and the slot in which this PDU is finally released in-order to higher layers, i.e., the slot where both the tagged PDU and all the $m-1$ blocking PDUs have been resolved. Moreover, we compute such statistics at the sender side, i.e., we say that the tagged PDU can be released in-order in the slot where the last unresolved PDU (comprising the tagged packet itself) is transmitted and resolved at the sender side. Note that this statistics differs with respect to the statistics at the receiver side by a constant term (the sum of the single path propagation delay and physical layer processing D_s). In order to proceed with the analysis, let us subdivide the time in rounds of m slots. In particular, we refer to round 0 as the one including the PDUs transmitted in slots $\{t-m+1, t-m+2, \ldots, t\}$, where t is the slot in which the tagged PDU has been transmitted for the first time over the channel. Without loss of generality, in the following we will assume that the tagged packet is transmitted in position m of round 0, i.e., $t = m$.

To perform an exact analysis of the PDU delay statistics, it is necessary to keep track of the number of retransmission attempts performed for each PDU transmitted in the starting window and of its delivery status (resolved or unresolved). Hence, to describe the probabilistic evolution of the tagged PDU delivery process, we would build a chain characterized by at least $(2 \times (L+1))^m$ states. However, for large values of m, this would lead to such a large state space as to make the problem very difficult to solve. To avoid this, in the following analysis, we limit the amount of information to be tracked. In particular, at any time, we only track the full state (transmission attempts and delivery status) of the tagged PDU and the number of unresolved blocking packets. The resulting analysis is not exact but will be shown to be very accurate.

Moreover, we use the probability $P[\text{error}] = p$ to decide whether a PDU in a given slot is erroneous or not. Note that, given that a PDU is transmitted in error in a slot, the probability that it has reached its maximum number of transmission attempts is equal to the probability that the PDU has been already transmitted for L times without any success before the considered slot ($P[L\text{prev. errors}] = p^L$). So, the probability that a PDU is discarded in a slot, i.e., the PDU has been transmitted in error for $L+1$ consecutive times, is equal to $P[\text{discard}] = P[\text{error}]P[L\text{prev. errors}] = p^{L+1}$. Hence, a PDU in a given slot is transmitted successfully with probability $q = 1 - p$, is discarded with probability $p_d = p^{L+1}$, and remains unresolved without being discarded with probability $1-q-p_d = p(1-p^L)$.

Let $u \in \{0,1\}$ be a Boolean variable indicating the tagged PDU resolving state, i.e., whenever the tagged packet has been resolved or not. In particular, we use the notation $u = 0$ and $u = 1$ for resolution and no-resolution, respectively. Using this notation, we define $\psi[n,i,u]$ as the joint probability that, at the end of round i, $i \geq 0$ there are n unresolved PDUs among positions $(i-1)m+1$ through $(i-1)m+m-1$ and the state of the tagged PDU[6] is u. In particular, the probability to have, at the end of round 0, n blocking PDUs and the tagged packet state equal to u is given by

$$\psi[n,0,u] = \binom{m-1}{m-n-1} \times \sum_{k=0}^{m-n-1} \left[\binom{m-n-1}{k} \times q^k p_d^{m-n-1-k} \times (1-q-p_d)^n p^u q^{1-u} \right] \quad (1)$$

where $n \in \{0,1,\ldots,m-1\}$, $u \in \{0,1\}$, $q = 1-p$, $p_d = p^{L+1}$, and p is the PDU error probability. In the equation above, we compute the probability that n out of the $m-1$ PDUs transmitted (in round 0) are unsuccessful, but they have been transmitted less than $L+1$ times (term $(1-q-p_d)^n$, in this case they must be retransmitted in the next round and in the same position), that the remaining $m-1-n$ PDUs are resolved either due to their correct transmission (term q^k) or to their discard (term $p_d^{m-n-1-k}$), and that the tagged PDU is in state u (where $u = 0$ means correctly transmitted).

The function $\psi[n,i,u]$, for $i > 0$, is evaluated recursively in the following way:

$$\begin{aligned} \psi[n,i,0] &= \sum_{k=n}^{m-1} \left(\psi[k,i-1,1]q + \psi[k,i-1,0]\right) \varphi(k,n) \\ \psi[n,i,1] &= \sum_{k=n}^{m-1} \psi[k,i-1,1] p \varphi(k,n) \end{aligned} \quad (2)$$

where $\varphi(k,n)$ is the probability to resolve (correctly receive or discard) $k-n$ PDUs over k in any order and is computed as follows:

$$\varphi(k,n) = \sum_{r=0}^{k-n} \binom{k}{n} \binom{k-n}{r} q^r p_d^{k-n-r} (1-q-p_d)^n. \quad (3)$$

[6]Following our definition of round, this PDU is always transmitted in the last slot of each round, i.e., in this case in position im.

Moreover, the probability of having, at the end of round i, n unresolved packets among positions $(i-1)m+1$ through $(i-1)m+m-1$ and the tagged PDU in state 0 ($\psi[n,i,0]$) is computed as the probability of having $k \in \{n,\ldots,m-1\}$ unresolved blocking PDUs (term $\psi[k,i-1,1]q + \psi[k,i-1,0]$) in the previous round $(i-1)$ and that exactly $k-n$ of these PDUs are resolved in round i (term $\varphi(k,n)$). The term $\psi[k,i-1,1]$ is multiplied by q to account for the case where the tagged packet is resolved at the end of round i. A similar reasoning is made in the computation of $\psi[n,i,1]$. In this case the term $\psi[k,i-1,0]$ is absent because a resolved PDU can never be marked as unresolved. In this case the term $\psi[k,i-1,1]$ is multiplied by p to reflect that the tagged packet is still unresolved at the end of round i.

The delivery delay statistics, indicated as $P_d[t = \xi m+u], \xi \geq 0, u \in \{0,1,\ldots,m-1\}$, i.e., the probability to have a delivery delay for the single PDU equal to t slots is approximated by means of the ψ function

$$P_d[t = \xi m+\eta] = \begin{cases} \psi[0,0,0], & t = 0 \\ \sum_{k=0}^{m-1} \psi[k,\xi-1,1] q \mathcal{C}(k), & \eta = 0, t \in \mathcal{D} \\ \sum_{k=1}^{\eta} \psi[k,\xi,0] \mathcal{C}(k) P_\eta, & \eta \neq 0, t \in \mathcal{D} \\ 0, & \text{elsewhere} \end{cases} \quad (4)$$

where $\mathcal{D} = \{1,2,\ldots,Lm\}$, $\eta \in \{0,1,\ldots,m-1\}$, and $\xi \geq 0$. $\mathcal{C}(k)$ is the probability to resolve k PDUs in any order and is given by

$$\mathcal{C}(k) = \sum_{i=0}^{k} \binom{k}{i} q^i p_d^{k-i} \quad (5)$$

where P_η is the probability that the last of the k PDUs in error is transmitted in position η of round ξ, where $\eta \in \{1,2,\ldots,m-1\}$. P_η is computed as follows:

$$P_\eta = \frac{\binom{\eta-1}{k-1}}{\binom{m-1}{k}}. \quad (6)$$

In (4), the way $P_d[t]$ is computed depends on the value of η. In more detail, if the tagged packet is released in-order at a given time $\xi m + (\eta = 0)$, this means that it is released in the slot in which it resolved.[7] Given that, all blocking packets are necessarily resolved up to and including the end of round ξ. When $\eta \neq 0$, instead, at the time in which the tagged packet is resolved, there is at least one blocking packet to be resolved. In this case, $P_d[t]$ is evaluated summing over $1 \leq k \leq \eta$ the probability ($\psi[k,\xi,0]$) that k blocking packets are unresolved at the end of round ξ and that these packets are all resolved (term $\mathcal{C}(k)$) in the current round (round $\xi+1$). Moreover, given that PDU errors are described by means of an i.i.d. process, the position of the last resolved blocking packet is distributed in a uniform manner [7]. We then evaluate the probability of resolving the last blocking PDU in position η using (6). $P_d[t]$ is reported in Fig. 4 and compared against the one computed by simulation for $m = 40$, $p = 0.1$, and $L = 3$.

[7]Remember that the tagged PDU is always sent at the end of each round.

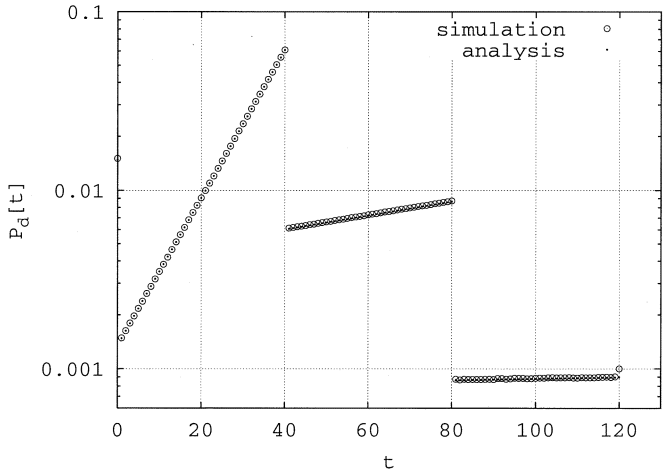

Fig. 4. Single PDU delay statistics $P_d[t]$. Comparison between analysis and simulation for $m = 40$, $p = 0.1$, and $L = 3$.

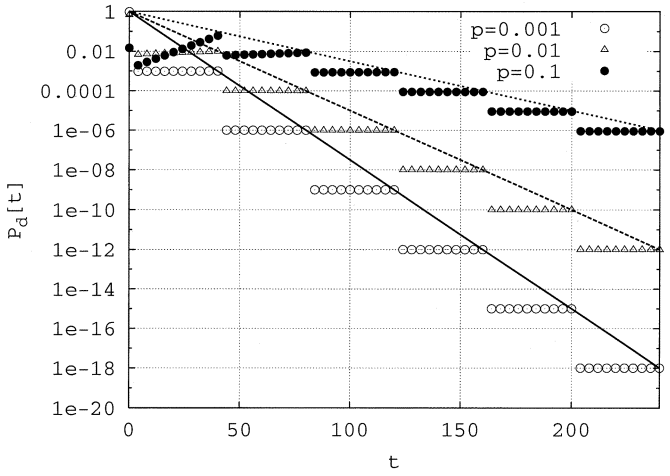

Fig. 5. Asymptotic behavior of $P_d[t]$ obtained by analysis; $m = 40, p = 0.1$ and $L = 7$.

B. Statistical Properties of $P_d[t]$

In this section, we investigate the main characteristics of the function $P_d[t]$. In what follows, for the sake of simplicity, we refer to the case where L is unlimited. The obtained results hold also in the limited case.

As above, let round 0 be the one where the tagged packet is transmitted for the first time. Moreover, express the delivery delay as $d = \xi m + \eta, \xi \geq 0, \eta \in \{0, 1, \ldots, m-1\}$. The probability that an erroneous PDU in round 0 is transmitted correctly in a round up to and including round r is given by $P_c[r] = 1 - p^r$. The probability that the tagged packet is resolved in position m of round s, $s > 0$, i.e., at time $d_1 = sm$, is given by

$$P_d[d_1 = sm] = \sum_{k=0}^{m-1} \psi[k, 0, 1] P_c[s]^k p^{s-1}(1-p) \quad (7)$$

where we compute the probability to have k blocking packets, that all of them are resolved before slot d_1 (term $P_c[s]^k$), and that the tagged packet is resolved exactly in slot d_1 (term $p^{s-1}(1-p)$). Now, let us compute the probability to resolve the tagged PDU at time $d_2 = sm + 1$, i.e., in the first slot of the following round:

$$P_d[d_2 = sm + 1] = \sum_{k=0}^{m-1} \psi[k, 0, 1] P_c[s]^k p^s(1-p). \quad (8)$$

In this case $\psi[k, 0, u]$ must be viewed as the probability that the PDU occupying position 1 in round 0 was in error $(u = 1)$ and that exactly k packets out of the remaining $m-1$ are erroneous, $P_c[s]^k$ is the probability that PDU in slot $\{2, 3, \ldots, m\}$ is transmitted correctly up to and including round s, and $p^s(1-p)$ represents the probability that the PDU transmitted in the first position of round 0 is resolved in slot d_2. Observing (7) and (8), it is clear that the following relation holds:

$$P_d[sm + 1] = P_d[sm]p \quad s \geq 0. \quad (9)$$

This is a general result that will be very useful in order to find very fast and accurate approximations of such statistics. Moreover, using again these equations, we can find the difference between the probability to resolve the tagged packet in the first slot and at the end of a given round $(s > 0)$

$$P_d[(s-1)m + 1] - P_d[sm] = \sum_{k=0}^{m-1} \psi[k, 0, 1] p^{s-1}(1-p)\Delta[s, k] \quad (10)$$

where $\Delta[s, k] = P_c[s]^k - P_c[s-1]^k$. At this point, by rewriting $P_c[s]^k$ as $(1 - p^s)^k = \sum_{i=0}^{k} \binom{k}{i} (-p^s)^i$, it is easy to verify that $\Delta[s, k]$ tends to zero as s increases. In practice, this convergence is very fast and, as a consequence[8], $P_d[t]$, after a short transient phase, becomes almost constant inside each round (see Figs. 4 and 5). These considerations allow us to assert that the asymptotic behavior of the delivery delay statistics is described by the following equation:

$$y[t] = p^{\left(\frac{t - t_0}{m}\right)}. \quad (11)$$

In more detail, (9) and the fact that, after a first transient phase, all points in a round have the same value allow us to conclude that, in the logarithmic domain, and after a transient phase, the values of the delivery delay probability in the first (last) slot of each round are placed over a straight line. This observation allows us to find, after a short transient phase, the exact behavior of $P_d[t]$ simply using (11). Given m and p, the only parameter that needs to be computed is the value of t_0 that can be found by solving equation $P_d[sm] = y[sm]$ for s sufficiently large.[9]

In Fig. 5, we report $P_d[t]$ and the straight lines indicating the asymptotic behavior by varying p considering $m = 40$ and $L = 7$.

C. Link-Layer SDU Transmission Delay Statistics

In this section, we compute the transmission delay statistics of an aggregate of K PDUs (link-layer SDU), i.e., the statistics of the time elapsed between the instant in which the first

[8]This is ensured by the fact that $P_d[t]$ is nondecreasing inside each round, i.e., $P_d[sm + 1] \leq P_d[sm + 2] \leq \cdots \leq P_d[sm + m]$, $s \geq 0$. This can be verified from (4).

[9]Note that for p values up to and including 0.1, $s = 1$ suffices to derive a very accurate approximation for t_0.

PDU composing an SDU is transmitted over the channel and the slot in which the Kth PDU (the last composing the SDU) is transmitted for the first time over the channel. First of all, note that a new PDU is sent over the channel in the generic slot t if and only if a PDU had been resolved m slots earlier (in slot $t-m$). In fact, each PDU at the time of its first transmission occupies a given position in its starting window. Such PDU is then transmitted every m slots in the same window position until success or until $L+1$ transmission failures have occurred. In other words, a given window position is occupied for transmission by a single PDU up to and including the slot where it is finally resolved. Only from this point, can the slot be assigned to a new PDU.[10] Hence, the number of PDUs resolved in a generic time interval, say, $[t, t+N]$, is exactly equal to the number of new PDUs transmitted in the time interval $[t+m, t+m+N]$. In particular, the *new transmission process* is simply the *enabling process* deterministically right shifted by m slots.

In the following analysis, we adopt the same assumptions made in Section III-B, i.e., we do not track the exact state of each PDU but we decide whenever a PDU is successfully transmitted, still unresolved, or discarded using the probabilities q, $1-q-p_d$, and p_d, respectively.

In this section, we refer to the *SDU transmission time* as the number of slots elapsed between the time in which the first PDU composing the SDU is transmitted for the first time and the slot where the last (the Kth) PDU is transmitted for the first time over the channel. Hence, the probability that a full SDU is transmitted in exactly t slots, say, in the time interval $[i, i+t]$, $i \geq 0$, corresponds to the probability that $K-1$ new PDUs are taken from the input queue and transmitted for the first time over the channel in slot $i+1$ through $i+t$ given[11] that the first PDU composing the SDU packet is transmitted for the first time in slot i. We refer to this probability as $P_{tx}[K,t]$. Moreover, recall that a packet resolution in a given slot implies the transmission of a new packet m slots apart; this probability is the same as that of resolving $K-1$ packets in $t-1$ slots, say, $[i-m+1, i+t-m]$, where the last packet must be resolved in the $(t-1)$st slot (slot $i+t-m$) given that we have the first resolution in slot $i-m$.

Hence, a good approximation of the transmission delay statistics $P_{tx}[K,t]$ can be obtained using (12) at the bottom of the page, where $t_{\max} = m(L+1)+K-1$. In (12), we compute the probability to have $K-2$ PDU resolutions over $t-2$ slots in any order (terms p_d^r and q^{K-2-r}) and that the PDU transmitted in the last (the tth) slot is also resolved (term $q+p_d$). Note that the statistics above is conditioned on having the resolution of the first PDU (of K) in the first slot. Moreover, $P_{tx}[K,t]$ is zero for $t < K$ and $t > t_{\max}$. The first case is trivial, whereas the second case is justified as follows. In the worst case, the first PDU is transmitted for the first time in a window where all the remaining $(m-1)$ PDUs are also transmitted for the first time. Moreover, in the very worst case all these PDUs are retransmitted repeatedly for L times in each of the following L rounds. Only from this point on can the remaining $K-1$ PDUs be transmitted in position 1 through $K-1$. Note that this reasoning is valid as long as $K-1 \leq m$, but it can easily be extended to the more general case in which $K-1 > m$. In this paper, we focus on the case where $K-1 \leq m$ because it is a very common situation under typical logical channel settings in UMTS. For instance, considering a link-layer logical bit rate of $B = 120$ Kbps, a typical PDU size of 360 bits, and a link-layer round-trip time of RTT$= 220$ ms, we have that about 75 PDUs are transmitted in a full round-trip $(m \approx 75)$. In addition, a SDU of 1 Kbyte is transmitted in 23 PDUs. Moreover, for a fixed RTT, m increases with B. However, our analysis, with minor changes (it is sufficient to change $t_{\max}$), still holds also in the case where $K-1 > m$.

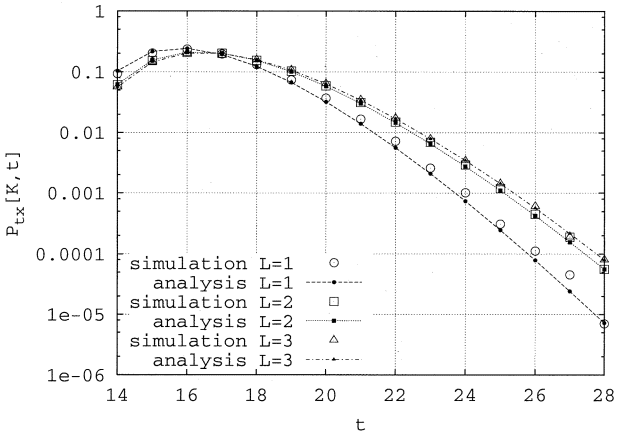

Fig. 6. Transmission delay statistics $P_{tx}[K,t]$; comparison between analysis and simulation for $m = 40$, $K = 14$, and $p = 0.1$.

In Fig. 6, we compare the results obtained from (12) with those achieved by simulation. By observing this figure, we can conclude that the statistics derived using (12) is reasonably accurate for small L values, and it tends very quickly to the exact statistics as L increases.

Next, we compute the asymptotic behavior of $P_{tx}[K,t]$. In particular, we are interested in the computation of the ratio $P_{tx}[K,t+1]/P_{tx}[K,t]$ as a function of t. Observing that $\binom{i+1}{j} = (\frac{i+1}{i+1-j})\binom{i}{j}$, $i > j$, we can rewrite $P_{tx}[K,t+1]$ as

$$P_{tx}[K,t+1] = (1-q-p_d)\frac{t-1}{t+1-K}P_{tx}[K,t] \qquad (13)$$

that holds for any t such that $K \leq t < t_{\max}$. From (13), it is straightforward to note that $P_{tx}[K,t+1]/P_{tx}[K,t]$ for large

$$P_{tx}[K,t] = \begin{cases} (q+p_d)\sum_{r=0}^{K-2}\binom{t-2}{r}\binom{t-2-r}{K-2-r}p_d^r q^{K-2-r}(1-q-p_d)^{t-K}, & K \leq t \leq t_{\max} \\ 0, & \text{elsewhere} \end{cases} \qquad (12)$$

[10]We say, in this case, that the resolution of a given PDU *enables* the transmission of a new PDU over the channel m slots apart, i.e., in the same window position of the next round. In the sequel, this process will be referred to as *enabling process*.

[11]Where the last (Kth) PDU must be transmitted for the first time in slot $i+t$, whereas the remaining $K-2$ PDUs (excluding the first and the Kth) can be transmitted in any order in slots $i+1$ through $i+t-1$.

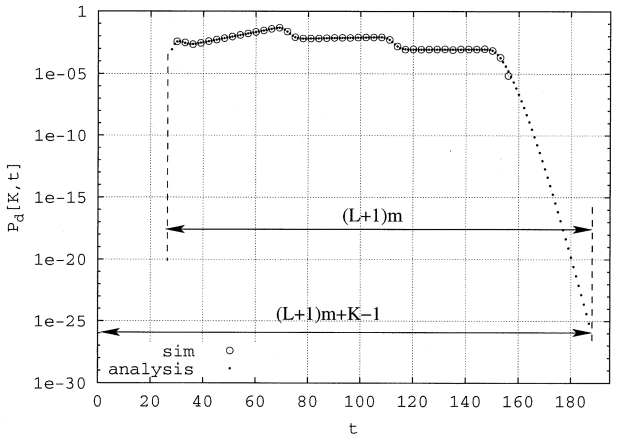

Fig. 7. $P_d[K,t]$ for $m = 40$, $K = 28$, $L = 3$, and $p = 0.1$.

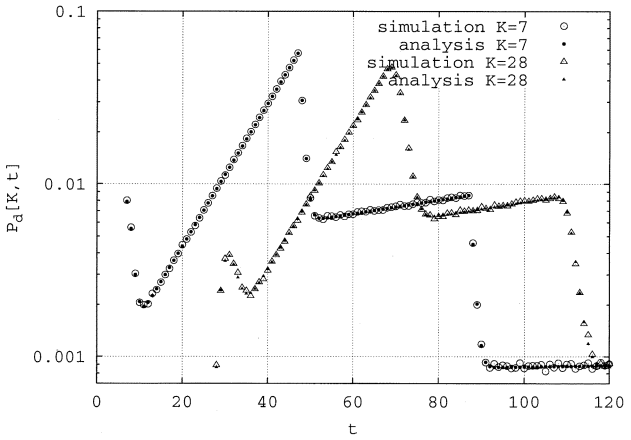

Fig. 8. $P_d[K,t]$; comparison between $K = 7$ and $K = 28$ for $m = 40$, $L = 3$, and $p = 0.1$.

values of t tending to the constant value $(1-q-p_d)$. In this case $P_{tx}[K,t]$ can be very well approximated by a straight line (in the logarithmic domain). These results will be used in Section IV to derive fast and accurate approximations of SDU delay statistics.

D. SDU Delivery Delay Statistics for In-Order Delivery Case

Here we derive the delivery delay statistics of a link-layer SDU, i.e., the number of slots elapsed between the instant where the first PDU composing that SDU is transmitted for the first time over the channel and the instant in which the full SDU is received and can be passed in-order to higher layers. To find such statistics, we subdivide the SDU delivery time in two contributions, where the first one is given by the time elapsed between the first transmission of the first PDU composing the SDU and the slot where the last (Kth) PDU is transmitted over the channel. To track this delay, we use the transmission statistics computed in (12). From this point on, we use the single PDU delivery delay statistics in (4) to track the time needed for PDU K to be delivered in-order. Note that the slot where this last PDU can be delivered in-order to higher layers is the same slot where the whole SDU can be delivered in-order. This is justified by the fact that the PDUs that eventually block the delivery of the Kth packet at the instant of its first transmission are only those with an identifier lower than the one assigned to that PDU. In conclusion, by using a discrete-time convolution product of these two contributions, we are able to obtain the delay statistics $P_d[K,t]$ of a full SDU

$$P_d[K,t] = \begin{cases} 0, & t < K \\ \sum_{r=K}^{t} P_{tx}[K,r] P_d[t-r], & t \geq K. \end{cases} \quad (14)$$

In Fig. 7 we compare the SDU delivery delay statistics obtained by simulation against the one obtained analytically using (14). Here, we note that simulation points can be estimated only until error probabilities of the order of $p \approx 10^{-5}$. For lower values of p, the statistics cannot be obtained due to the rare occurrences of the corresponding events. In Fig. 8, instead, we report simulation and analytical results by varying the number of PDUs composing one SDU (K). As can be observed from these figures, analysis and simulation points are in very good agreement. Note also that $P_d[K,t]$ is zero for $t > (L+1)m + K - 1$, i.e., where $P_{tx}[K,t]$ is zero.

The most useful quantity is the ccdf of $P_d[K,t]$, i.e., the probability that the SDU delivery time exceeds a given number of slots, formally

$$ccdf[K,t] = 1 - \sum_{x=0}^{t} P_d[K,x]. \quad (15)$$

Moreover, recall that our analysis is not inclusive of the single path propagation delay (D_s) regarding the delivery of the Kth PDU (Section III-A). The complementary delivery delay statistics comprehensive of D_s is computed as follows:

$$\text{Prob}\{\text{SDU delay} > t\} = \begin{cases} 0, & t < D_s \\ ccdf[K, t - D_s], & t \geq D_s \end{cases}. \quad (16)$$

IV. Accurate Approximation of the SDU Complementary Cumulative Distribution Function ccdf$[K,t]$

We believe that the ccdf statistics would be very useful if available at any user terminal. It can be used, for example, to predict in advance performance metrics, or used directly in some kind of cost function.

The main drawback of the analysis presented in Section III is that it is computationally too complex to be effectively used in a mobile terminal, since the involved computations would be time and energy consuming. Moreover, unless the time spent for the computation of the statistics is small with respect to the period of time in which the channel behavior (p) remains constant, then it results to be useless. For this reason, in addition to energy requirements, we have computation delay requirements dictated by the stationarity period characterizing the PDU error process. For these reasons, here, after investigating the major properties of the complementary cumulative distribution, we will propose a very low-complexity and fast heuristic able to accurately approximate such statistics.

In Fig. 9, the cumulative complementary distribution ccdf$[K,t]$ is plotted by varying L for $m = 40$ and $p = 0.1$. First of all, we note that this statistics presents a cyclic behavior, where the cycle length is equal to m. Moreover,

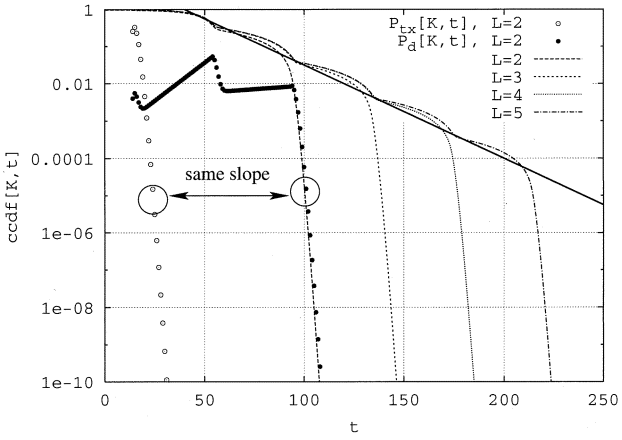

Fig. 9. ccdf$[K,t]$ by varying L and p for $K = 14$, $m = 40$.

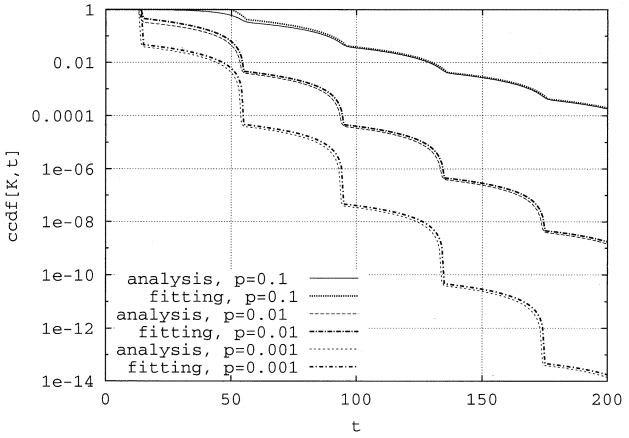

Fig. 10. ccdf$[K,t]$; comparison between analysis and fitting for $K = 14$, $m = 40$, L unlimited.

the property expressed by (11) still holds, i.e., in the logarithmic scale the points at the beginning of each round (slots $K + sm + 1$, $s \geq 0$) are aligned on a straight line (solid bold line in Fig. 9), described again by (11) (appropriately setting the parameter t_0). Moreover, for a given value of L, the statistics follows the behavior of the one where L is unlimited approximately until $t = K + Lm$; whereas from this point on (in the last round where the ccdf is greater than zero, i.e., in $t \in (K + Lm, K + (L + 1)m - 2]$[12]) ccdf$[K,t]$ starts decreasing very quickly. It is very interesting to note that the slope concerning this last part appears to be the same as the one characterizing the SDU delivery delay statistics $(P_d[K,t])$ and the SDU transmission delay statistics $(P_{tx}[K,t])$, both reported for comparison in Fig. 9. Therefore, in this last part, the statistics follows a straight line (in the logarithmic domain) whose behavior at time t is characterized by (13). To obtain a good approximation of this last part, we can simply use (13) by taking the limit slope, i.e., the one reached when t grows to infinity. In conclusion, for $t \geq K + Lm$, the complementary cumulative distribution can be well approximated by

$$y_f[t] = (1 - q - p_d)^{t - t_0'} \tag{17}$$

Now, in order to find a heuristic able to entirely describe such statistics, we need to generate the first part, i.e., the one observed for $0 \leq t \leq K + Lm$. This is quite simple using (11) to find the ccdf values at the beginning of each round and by noting that, in the linear domain, the points inside each round are aligned on a straight line. In conclusion, the behavior of ccdf$[K,t]$, for $0 \leq t \leq K + Lm$, can be well approximated by a piecewise linear function, whereas we can use (17) to approximate the tail of the distribution $(K + Lm < t \leq K + (L + 1)m - 2)$.

In the next, we first report the function used to fit the ccdf statistics when L is unlimited. Let t_i be the first slot of round i, $i \geq 1$, then $t_i = K + (i - 1)m + 1$. As discussed above, we can use the function $y[t]$ to find the probability corresponding to the points at the beginning of each round t_i. Moreover, we can approximate the behavior of ccdf between these points by means of a line whose extremes are $y[t_i]$ and $y[t_{i+1}]$. In practice, ccdf can be well approximated by the following function:

$$f'[K,t] = \begin{cases} 1, & t \in [0, K-1] \\ 1 - \psi[0,0,0]q^{K-1}, & t = K \\ y[t_i] + \frac{(y[t_{i+1}] - y[t_i])(t - t_i)}{m}, & t \in [t_i, t_{i+1}), \\ & \forall\, i \text{ such that } i \geq 1. \end{cases} \tag{18}$$

In Fig. 10, we compare the cumulative distribution obtained analytically against the one derived using (18). As can be observed from that figure, the approximation is very accurate for any p.

Now, we investigate how to approximate ccdf$[K,t]$ when L is limited. For this purpose, we consider the approximation given by (18) for $t \in [0, K + Lm]$, whereas, to fit the tail of the ccdf $(t \in (K + Lm, K + (L + 1)m - 2])$, we use the function $y_f[t]$. Formally

$$f[K,L,t] = \begin{cases} f'[K,t], & t \in [0, K + Lm] \\ y_f[t], & t \in (K + Lm, K + (L+1)m - 2] \\ 0, & t > K + (L+1)m - 2 \end{cases} \tag{19}$$

the parameter t_0' in $y_f[t]$ can be computed by requiring that $y[K + Lm + 1] = y_f[K + Lm + 1]$, i.e., that the first point of $y_f[t]$ match with $y[t]$ in the last round. Hence, t_0' is found as

$$t_0' = t_{L+1} - \frac{\log(y[t_{L+1}])}{\log(1 - q - p_d)}. \tag{20}$$

At this point, to obtain the complete statistics from (19), the only parameter that needs to be specified is t_0 [(11)]. This parameter, for a given K, can be accurately fitted and stored in a lookup table as a function of p. This can be achieved obtaining the analytic curves and fitting $y[t_i]$ with ccdf$[K,t_i]$ for a sufficiently large i (in order to capture the asymptotic behavior of the statistics in the logarithmic domain). Moreover, the number of points to be stored in such a table could be limited by exploiting some properties of t_0 as a function of p, i.e., observing that the behavior of t_0 as a function of p is linear for $p \geq 0.004$ and that, for $p \leq 0.001$, it is linear in the $\log p$ domain. Fig. 10 has been obtained using this first method.

[12] $P_d[K,t]$ is zero for $t > (L + 1)m + K - 1$, so ccdf is zero for $t > (L + 1)m + K - 2$.

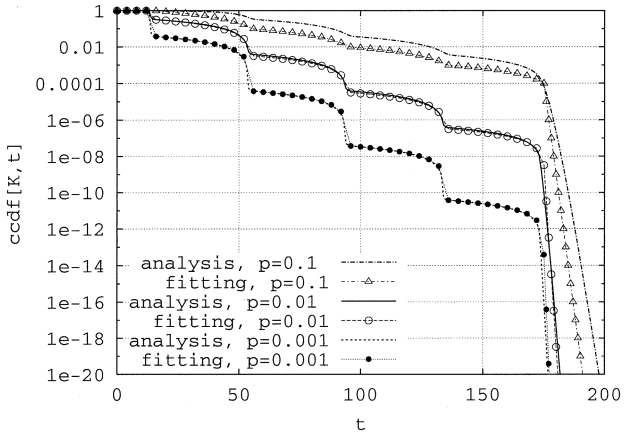

Fig. 11. Comparison between ccdf$[K,t]$ and $f[K,L,t]$ for $K=14$, $m=40$, $L=4$.

Another very effective way to estimate t_0 is to consider the probability $ccdf[K,K]=1-\psi[0,0,0]q^{K-1}$, i.e., the probability to deliver the SDU in more than K slots. Now, supposing that ccdf$[K,t_1=K+1]$ is well approximated by[13] $y[t_1]$, and taking ccdf$[K,K]$ as an estimate of this quantity, we can find t_0 as follows:

$$t_0=K+1-m\times\frac{\log(1-\psi[0,0,0]q^{K-1})}{\log p}. \qquad (21)$$

In Fig. 11, we compare the distribution obtained analytically against the one derived from (19), and using this last method to evaluate t_0. At low p values, t_0 is approximated very well and the statistics is fitted accurately. However, as p increases ($p=0.1$ in that figure), the assumptions made in the computation of t_0 are less accurate (the fitting, in this case, is less accurate too).

The heuristic expressions given in (17)–(21) make it possible to accurately approximate the delay statistics of aggregates of K link-layer packets by means of a piecewise linear function. The parameters needed for this computation are the round-trip time (expressed in number of packets transmitted, i.e., normalized and rounded up to the packet transmission time), and an estimate of the link-layer packet error probability p. From these values, it is easy to compute $\psi[0,0,0](q(q+p_d)^{m-1})$ and to find $f'[K,t]$ and $f[K,L,t]$. These functions are then used to estimate the delay ccdf from (11) and (17) based on the estimate of the parameter t_0 as given in (21). As the number of parameters to be estimated and stored is small and the approximate analytical expressions are simple, this approximate technique can be easily implemented on terminals with constrained resources, so that delay-driven algorithms can be implemented on handsets.

In Section V, we find the statistics of such an aggregate of K link-layer PDUs in the out-of-order delivery case.

[13]Note that (11), after a transient phase, gives the exact value of ccdf$[K,t_i]$, $i\geq 1$. Moreover, the length of this phase for small p values tends to zero and the following approximation holds: ccdf$[K,t_1]\approx y[t_1]$.

V. SDU Delivery Delay Statistics in the Out-of-Order Delivery Case

In the out-of-order delivery case, each link-layer SDU can be passed to higher layers when all the PDUs composing it have been correctly received regardless of the delivery status of other SDUs.

In the following, we compute the delivery delay distribution regarding a single SDU. As in Sections II–IV, we proceed starting from the slot where the first PDU composing that SDU is transmitted for the first time over the channel. Without loss of generality, we assume that such PDU is transmitted in position 1 in its transmission round (referred to as round 1). We refer to $t=1$ as the slot where this transmission occurs. Note that, for the reasons discussed in Sections II–IV, we are sure that in slot $1-m$, a successful transmission occurred. Hence, the whole statistics is conditioned on the fact that a correct transmission occurred in slot $1-m$. Once again, we assume K to be the (integer) number of link-layer PDUs composing one SDU.

Next, we present an approximate approach that is able to fit very accurately the SDU delivery statistics when $K\ll m$ or $K\leq m$ and $p\ll 1$. This analysis is valid as long as $K\leq m$. However, note that in UMTS, due to the large link-layer round-trip time (up to 220 ms), this condition is likely verified. In the computation of the out-of-order delivery delay statistics we relax the hypothesis of having a finite L, i.e., we assume an infinite number of retransmission attempts.

Our analysis is based on the consideration that, when $K\ll m$ and/or $p\ll 1$, with almost probability 1, all the K PDUs composing the tagged SDU are transmitted for the first time in round 1. Where this hypothesis holds, the delay statistics, i.e., the probability that a tagged SDU is released in slot η, $\eta\in\{1,2,\ldots,m\}$, of round ξ, $\xi\geq 0$, given that the first PDU composing it is transmitted for the first time in slot 1, can be found with great accuracy using (22) at the bottom of the page, where $\xi\geq 0$, $\eta\in\{1,2,\ldots,m\}$. The functions f_1, f_2 and $\mathcal{P}$ are

$$f_1(x,y)=\binom{x-2}{y-2}p^{x-y}q^{y-1} \qquad (23)$$

$$f_2(x,y)=\begin{cases}1 & x=1\\ f_1(x,y) & x>1\end{cases} \qquad (24)$$

$$\mathcal{P}(K,k_1,\xi)=(1-p^{\xi})^{K-k_1}(1-p^{\xi+1})^{k_1-1}p^{\xi}q. \qquad (25)$$

$$P_d[\xi m+\eta]=\begin{cases}0, & \xi m+\eta<K\\ f_1(\eta,K)q^K, & \xi=0,\ \eta\geq K\\ \sum_{i=K}^{m}\sum_{k_1=1}^{\eta}\left[f_1(i-\eta+1,K-k_1+1)f_2(\eta,k_1)\mathcal{P}(K,k_1,\xi)\right], & \xi>0,\ \eta<K\\ \sum_{k_1=1}^{K}f_1(\eta,k_1)\mathcal{P}(K,k_1,\xi), & \xi>0,\ \eta\geq K\end{cases} \qquad (22)$$

In particular, in (22), for $\xi m + \eta \in \{K, \ldots, m\}$, i.e., $\xi = 0$ and $\eta \in \{1, 2, \ldots, m\}$, we compute the probability that, in round 0, there are K successes[14] in slot 1 through η and that the last (the Kth) success is in position η (term $f_1(\eta, K)$), i.e., all the SDU is transmitted during the first round and the last PDU composing it (the Kth) is transmitted in position η. Then, we multiply this probability by the probability that all these K PDUs are transmitted successfully in round 1 (term q^K).

For $\xi m + \eta > m$, we subdivide the case where $\eta < K$ from the case where $\eta \geq K$. In the first case, we compute the probability (term $f_2(\eta, k_1)$) of transmitting k_1 PDUs in round 1 (or equivalently of having k_1 successes in round 0) in slot 1 through η, where the last successful PDU (the k_1th) is transmitted in slot η. Moreover, with the term $f_1(i - \eta + 1, K - k_1 + 1)$, we account for the remaining $K - k_1$ transmissions (of the PDUs composing the tagged SDU) to be placed in slots $\eta+1$ through m of the first round (where i is the number of slots needed to transmit the K PDUs, i.e., the position where the Kth PDU is transmitted). After that, we use the $\mathcal{P}$ function to compute the probability that the k_1-1 PDUs transmitted in slot j_1, $j_1 < \eta$, are correctly received up to round $\xi+1$ (term $(1 - p^{\xi+1})^{k_1-1}$), that the PDU in position η is transmitted correctly in slot $\xi m+\eta$ (term $p^{\xi}q$) and that the $K - k_1$ PDUs in positions j_2, $j_2 > \eta$ are successfully transmitted up to and including slot ξm (term $(1 - p^{\xi})^{K-k_1}$).

When $\eta \geq K$, instead, we first compute the probability that k_1 PDUs are transmitted in positions {1,2,…, η} (in the first round) and that the last (the k_1th) of these PDUs is transmitted exactly in position η (term $f_1(\eta, k_1)$). Then, using the function $\mathcal{P}$, we account for the probability that the PDUs in slots {1,2,…, $\eta-1$} are correctly transmitted up to and including round $\xi+1$, that the k_1th PDU is transmitted correctly in position η of round $\xi+1$, and that all the remaining PDUs ($K - k_1$, that are transmitted in position $j \in \{\eta + 1, \ldots, m\}$) are successfully transmitted up to and including round ξ.

In Fig. 12, we report the comparison of the ccdf between the *in-order* and *out-of-order* delivery cases. As expected, the out-of-order case is characterized by the lowest delivery delay; for some values of t, in this case, the cumulative delivery delay probability is reduced by a factor of three with respect to the *in-order* delivery case. This could be very useful in the presence of delay constrained flows. It is worth noting that also in this case, the delivery delay statistics are characterized by a cyclic behavior that can be captured using (11) by appropriately tuning the t_0 parameter. In this case, the statistics can be accurately fitted noting that the first part of each round is characterized by a constant value, whereas after this phase the distribution starts decreasing until the beginning of the following round. Hence, by using twice (11), i.e., to track the starting point of each round and the ending point of the corresponding constant phase, and using again straight lines to approximate the decreasing behavior after constant periods, it is straightforward to obtain accurate and fast heuristics also for the out-of-order delivery case. These heuristics are not shown here due to space constraints but can be obtained based on what was explained in Section III.

[14]As noted above, these successes will enable K new PDU transmissions in the following round.

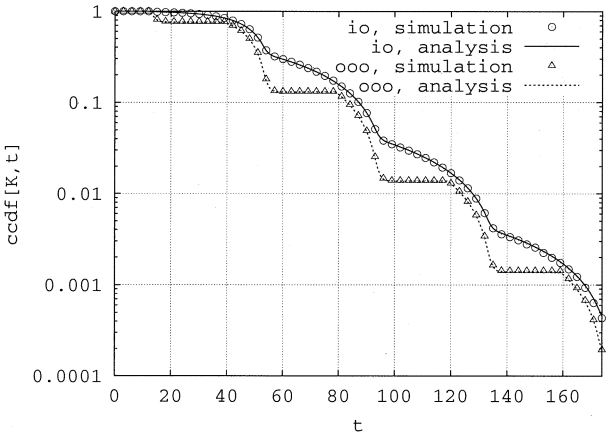

Fig. 12. ccdf$[K, t]$: comparison between in-order (io) and out-of-order (ooo) delivery cases considering $K = 14$, $p = 0.1$, and $m = 40$.

VI. Conclusion

In this paper, we considered a point-to-point wireless link where selective repeat ARQ is used to counteract channel impairments. An analytical framework is developed first to find accurate statistics of both link-layer and an aggregate of link-layer packets. Then, based on the properties of the statistics obtained by analysis, we derived heuristics for its approximation. Both the in-order and the out-of-order delivery of link-layer packets is considered. The merit of the paper is to give accurate, simple, and computationally fast heuristics for the characterization of delivery delay statistics of higher layer packets when ARQ retransmission techniques are used at the link layer. These heuristics could be effectively used, for example, in channel adaptive algorithms as an estimate of the delay perceived by packets belonging to the application layer data flow *(delay utility functions).*

References

[1] S. Lin, D. Costello, and M. Miller, "Automatic-repeat-request error control schemes," *IEEE Commun. Mag.*, vol. 22, pp. 5–17, Dec. 1984.

[2] D. Bertsekas and R. Gallager, *Data Network*, 2 ed. Englewood Cliffs, NJ: Prentice-Hall, 1992.

[3] J. G. Kim and M. M. Krunz, "Delay analysis of selective repeat ARQ for a markovian source over a wireless channel," *IEEE Trans. Veh. Technol.*, vol. 49, pp. 1968–1981, Sept. 2000.

[4] R. Fantacci, "Queueing analysis of the selective repeat automatic repeat request protocol for wireless packet networks," *IEEE Trans. Veh. Technol.*, vol. 45, pp. 258–264, May 1996.

[5] A. G. Konheim, "A queueing analysis of two ARQ protocols," *IEEE Trans. Commun.*, vol. COM-28, pp. 1004–1014, July 1980.

[6] M. Anagnostou and E. Protonotarios, "Performance analysis of the selective repeat ARQ protocol," *IEEE Trans. Commun.*, vol. COM-34, pp. 127–135, Feb. 1986.

[7] Z. Rosberg and N. Shacham, "Resequencing delay and buffer occupancy under the selective repeat ARQ," *IEEE Trans. Inform. Theory*, vol. 35, pp. 166–173, Jan. 1989.

[8] Z. Rosberg and M. Sidi, "Selective-Repeat ARQ: The joint distribution of the transmitter and the receiver resequencing buffer occupancies," *IEEE Trans. Commun.*, vol. 38, pp. 1430–1438, Sept. 1990.

[9] M. Zorzi and R. Rao, "On channel modeling for delay analysis of packet communications over wireless links," presented at the 36th Annu. Allerton Conf. Communications, Control And Computing, Allerton House, Monticello, IL, Sept. 1998.

[10] N. Guo and D. Morgera, "Frequency-Hopped ARQ for wireless network data services," *IEEE J. Select. Areas Commun.*, vol. 12, pp. 1324–1337, Oct. 1994.

[11] D. Lu and J. Chang, "Performance of ARQ protocols in nonindependent channel errors," *IEEE Trans. Commun.*, vol. 41, pp. 721–730, May 1993.

[12] M. Airy and J. M. Harris, "Analytical model for radio link protocol for 15–95 CDMA systems," in *Proc. IEEE VTC 2000-Spring*, vol. 3, Tokyo, Japan, May 2000, pp. 2434–2438.
[13] Y. J. Cho and C. K. Un, "Performance analysis of ARQ error controls under markovian block error pattern," *IEEE Trans. Commun.*, vol. 42, pp. 2051–2061, Feb./Mar./Apr. 1994.
[14] M. Zorzi, R. Rao, and L. Milslein, "Error statistics in data transmission over fading channels," *IEEE Trans. Commun.*, vol. 46, pp. 1468–1476, Nov. 1998.
[15] ——, "On the accuracy of a first-order markov model for data block transmission on fading channels," in *Proc. IEEE ICUPC*, Tokyo, Japan, Nov. 1995, pp. 211–215.
[16] M. Zorzi and R. Rao, "On the statistics of block errors in bursty channels," *IEEE Trans. Commun.*, vol. 45, pp. 660–667, June 1997.
[17] ——, "Lateness probability of a retransmission scheme for error control on a two-state markov channel," *IEEE Trans. Commun.*, vol. 47, pp. 1537–1548, Oct. 1999.
[18] 3GPP Third Generation Partnership Project. [Online]. Available: http://www.3gpp.org
[19] J. Laiho, A. Wacker, and T. Novosad, *Radio Network Planning and Optimization for UMTS*. New York: Wiley, 2001.
[20] H. Holma and A. Toskala, *WCDMA for UMTS: Radio Access for Third Generation Mobile Communications*, Rev. ed. New York: Wiley, 2001.
[21] "Third Generation Partnership Project,", Tech. Spec. Group Radio Access Network; Radio Link Control Spec. (Rel. 5) G. T. 25.322.

Michele Rossi (S'02) was born in Ferrara, Italy, on October 30, 1974. He received the Laurea degree *(summa cum laude)* in electrical engineering from the University of Ferrara in 2000, where he currently is pursuing the Ph.D. degree.

During academic year 2000/2001, he was a Research Fellow at the Department of Engineering, University of Ferrara. His research interests are in TCP/IP protocols on wireless networks, TCP/IP header compression, performance analysis of selective repeat link-layer retransmission techniques, efficient multicast data delivery, and mobility in 3G cellular networks.

Michele Zorzi (S'89–M'95–SM'98) was born in Venice, Italy, in 1966. He received the Laurea and Ph.D. degrees in electrical engineering from the University of Padova, Italy, in 1990 and 1994, respectively.

During academic year 1992–1993, he was on leave at the University of California, San Diego (UCSD), attending graduate courses and doing research on multiple access in mobile radio networks. In 1993, he joined the Faculty of the Dipartimento di Elettronica e Informazione, Politecnico di Milano, Italy. After spending three years with the Center for Wireless Communications at UCSD, in 1998 he joined the School of Engineering, University of Ferrara, Italy, where he is currently a Professor. His present research interests include performance evaluation in mobile communications systems, random access in mobile radio networks, ad hoc and sensor networks, and energy-constrained communications protocols. He is a member of the Editorial Board of *Wiley Journal of Wireless Communications and Mobile Computing* and *ACM/URSI/Kluwer Journal of Wireless Networks.*

Dr. Zorzi is Editor-In-Chief of the IEEE WIRELESS COMMUNICATIONS MAGAZINE. He is a member of the Editorial Board of the IEEE TRANSACTIONS ON COMMUNICATIONS, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, and IEEE TRANSACTIONS ON MOBILE COMPUTING. He was Guest Editor for Special Issues of IEEE PERSONAL COMMUNICATIONS MAGAZINE (Energy Management in Personal Communications Systems) and IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS (Multimedia Network Radios).