Computer Networks 55 (2011) 2873–2885

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

An algorithm for controlling packet size in IEEE 802.16e networks

Giovanni Ciccarese, Mario De Blasi, Pierluigi Marra, Cosimo Palazzo *, Luigi Patrono

Department of Innovation Engineering, University of Salento, Via Monteroni, 73100 Lecce, Italy

ARTICLE INFO

Article history:
Received 9 September 2009
Received in revised form 21 December 2010
Accepted 10 June 2011
Available online 22 June 2011

Keywords:
IEEE 802.16e
ARQ efficiency
Adaptive packet size

ABSTRACT

This paper proposes an algorithm to be used in IEEE 802.16e networks for adapting MAC PDU size to wireless channel behavior when ARQ is adopted at MAC layer. The algorithm is based on an analytical approach for dynamically evaluating the optimal packet size. The latter is derived from an expression of the ARQ protocol efficiency, obtained by exploiting a finite-state Markov error model which also takes into account Adaptive Modulation/Coding. The effectiveness of the designed algorithm in improving TCP performance has been evaluated.

© 2011 Elsevier B.V. All rights reserved.

1. Introduction

IEEE 802.16e standard [1] for Wireless Metropolitan Area Networks (WMAN) is being to play an important role in providing broadband mobile access. In order to make communications more robust to fading and spectrally efficient, Adaptive Modulation/Coding (AMC) and an optional Automatic Repeat reQuest (ARQ) protocol at MAC layer have been specified. However, aiming at a greater flexibility, a number of points have been left open. In particular, the standard gives the possibility to format MAC Protocol Data Units (PDUs) of variable length in order to adapt it to wireless channel behavior. In fact, when bit error rate is low, packet size should be high to minimize overhead, whereas in bad channel conditions a small size is preferable to reduce packet error probability.

Unfortunately, due to the time-variant characteristics of the channel, the choice of the best packet size is very critical when dealing with mobile networks. In [2,3] this issue has been addressed without taking into account the exploitation of AMC systems and adopting the traditional two-state Gilbert–Elliot error model. In [4] an analytical formulation of the problem has been developed considering an Adaptive Modulation System (AMS), but only a suboptimal solution, found statically for each modulation scheme, has been given. In [5] an IEEE 802.16e network with ARQ-enabled connections is considered and, assuming a memoryless channel, the optimal packet size has been evaluated as a function of the FEC (Forward Error Correction) block error rate.

In this paper[1] we also consider an IEEE 802.16e network with ARQ-enabled connections and, taking into account both the wireless channel behavior and the AMC, we propose an algorithm which exploits an analytical expression of the ARQ protocol efficiency. The latter is derived by adopting, as an error model, a Discrete Time Markov Chain (DTMC) with a number of "good" states, each one corresponding to a different burst profile (modulation format/code rate pair) [6].

After the validation of the ARQ protocol efficiency model, the effectiveness of the proposed solution in improving performance of upper layer protocols has been shown. In particular, TCP Goodput achievable when a Mobile Subscriber Station (MSS) [1] accesses an Internet server through an IEEE 802.16e link has been evaluated.

* Corresponding author. Tel.: +39 0832 297222.
E-mail addresses: gianni.ciccarese@unisalento.it (G. Ciccarese), mario.deblasi@unisalento.it (M. De Blasi), pierluigi.marra@unisalento.it (P. Marra), cosimo.palazzo@unisalento.it (C. Palazzo), luigi.patrono@unisalento.it (L. Patrono).

[1] This paper is an extended version of [12].

1389-1286/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2011.06.010

Nomenclature

α	parameter of Exponential Weighted Moving Average (EWMA)
γ_0	average signal-to-noise ratio (SNR)
Γ_i	threshold value used for partitioning SNR, with $1 \leqslant i \leqslant N$
η	ARQ protocol efficiency
φ_i	burst profile associated to state C_i of chain S_n, with $1 \leqslant i \leqslant N$
π_0	steady-state probability associated with state **B** of the chain $X(n)$
π_b	steady-state probability associated with state **b** of the chain $Y(n)$
π_g	steady-state probability associated with state **g** of the chain $Y(n)$
π_k	steady-state probability associated with state $\mathbf{G}_k$ of the chain $X(n)$, with $1 \leqslant k \leqslant L$
$\pi_k^{(a)}$	probability that $X(n) = \mathbf{G}_k$ when the transmission of a MAC PDU starts, with $1 \leqslant k \leqslant L$
C_i	value of state of the chain S_n, with $1 \leqslant i \leqslant N$
$F(\gamma, k)$	FER as a function of the received SNR γ and of the burst profile k
F_i	average FER for each state C_i of S_n, with $1 \leqslant i \leqslant N$
f_m	maximum Doppler frequency
G_P	mean number of MAC PDUs transmitted in a slot
L	number of states of the chain $X(n)$ used as an error model
N	number of states of the chain S_n used as a channel model
$N(\Gamma_i)$	crossing rate of the level Γ_i, with $1 \leqslant i \leqslant N$
l	amount (in bytes) of user data in a MAC PDU
l_{opt}	value of l which maximizes ARQ protocol efficiency
h	MAC PDU overhead (in bytes)
p_i	steady-state probability associated with state C_i of the chain S_n, with $1 \leqslant i \leqslant N$
p_{BB}	transition probability of the chain $X(n)$ from state **B** to state **B**
p_{bb}	transition probability of the chain $Y(n)$ from state **b** to state **b**
p_{BG_k}	transition probability of the chain $X(n)$ from state **B** to state $\mathbf{G}_k$
p_{G_kB}	transition probability of the chain $X(n)$ from state $\mathbf{G}_k$ to state **B**
$p_{G_kG_j}$	transition probability of the chain $X(n)$ from state $\mathbf{G}_k$ to state $\mathbf{G}_j$
$p_{G_kG_j}^{(m)}$	m-step transition probability of the chain $X(n)$ from state $\mathbf{G}_k$ to state $\mathbf{G}_j$
p_{gg}	transition probability of the chain $Y(n)$ from state **g** to state **g**
P_S	probability that the transmission of a MAC PDU is successfully
$p(s\|k)$	probability that the transmission of s consecutive bytes is successful under the condition that it starts in a slot n such that $X(n) = \mathbf{G}_k$
$p(s\|k,z)$	probability that the transmission of s consecutive bytes is successful under the condition that it starts in a slot n such that $X(n) = \mathbf{G}_k$ and that the slot n is the zth slot available in a frame, with $1 \leqslant z \leqslant U$
R	mean number of bytes transmitted in a slot
R_0	number of bytes transmitted in a slot n such that $X(n) = \mathbf{B}$
R_k	number of bytes transmitted in a slot n such that $X(n) = \mathbf{G}_k$, with $1 \leqslant k \leqslant L$
R_g	number of bytes transmitted in a slot n such that $Y(n) = \mathbf{g}$
S	ARQ protocol throughput
S_B	mean number of information bytes delivered to the upper layer in a slot
S_n	DTMC used as a channel model
S_P	mean number of MAC PDUs transmitted successfully in a slot
$S(t)$	instantaneous signal-to-noise ratio at time t
$t_{i,j}$	transition probability of the chain S_n
T_S	sampling period of $S(t)$, duration of a slot, FEC block transmission time
U	number of consecutive slots available in a frame
V	number of slots in a frame
$X(n)$	L-state DTMC used as an error model
$Y(n)$	2-state DTMC used as an error model

The rest of the paper is organized as follows. Section 2 is a review of IEEE 802.16e and, in particular, of the related ARQ. Section 3 outlines the error model adopted in this work. The analytical formulation of ARQ protocol efficiency is given in Section 4. Section 5 specifies the designed packet size control algorithm. Section 6 reports the simulation model. In Section 7, simulation results are discussed. Finally, conclusions and areas of future research are provided in Section 8.

2. IEEE 802.16e standard and the ARQ mechanism

IEEE 802.16e standard [1] specifies Physical and MAC layers for WMANs operating as broadband mobile access systems. It provides for operating at 2–6 GHz with data rates of 2–70 Mb/s. An IEEE 802.16e WMAN can offer Quality of Service (QoS) to traffic flows: the MAC layer defines QoS signaling mechanisms and functionalities which make it possible to control data transmissions and to manage dynamic bandwidth allocation procedures.

There are two types of stations in an IEEE 802.16e network: the Mobile Subscriber Station (MSS) and the Base Station (BS), which regulates all the transmissions in the WMAN according to a point-to-multipoint communication scenario. Either Frequency Division Duplexing (FDD) or Time Division Duplexing (TDD) can be used for separating uplink and downlink transmissions. In TDD mode, the durations of uplink and downlink subframes in which a frame is divided are dynamically determined by the BS on the basis of current traffic and adopted QoS policy. TDM and TDMA are exploited, respectively, for the down-

Table 1
IEEE 802.16e burst profiles.

k	Modulation	Code rate	Uncoded block size (bytes)	Coded block size (bytes)
1	BPSK	1/2	12	24
2	QPSK	1/2	24	48
3	QPSK	3/4	36	48
4	16-QAM	1/2	48	96
5	16-QAM	3/4	72	96
6	64-QAM	2/3	96	144
7	64-QAM	3/4	108	144

link and the uplink. In each time slot (of duration T_S) one OFDM symbol, corresponding to one FEC block, can be transmitted. It carries an amount of bytes determined by the code rate and the modulation scheme. In this regard, seven different modulation scheme/code rate pairs, called *burst profiles*, are provided by the standard (Table 1). The burst profile which allows to obtain the highest spectral efficiency for a target FEC Error Rate (*target FER*) can be dynamically selected by an AMC system. The maximum adaptation rate is once a frame.

The BS uplink scheduling module processes the bandwidth requests sent by MSSs and determines the time slots in which each MSS will be allowed to transmit.

IEEE 802.16e MAC layer provides a connection-oriented service. A connection is identified by a unique Connection Identifier (CID) and is unidirectional, so two connections have to be established for a bidirectional data transfer between the BS and a MSS. At MSS initialization, for each direction (uplink/downlink), two management connections, named Basic and Primary, are established between the MSS and the BS and a third management connection, called Secondary, may be optionally generated. The three types of connections reflect the fact that there are inherently three different levels of QoS for the management traffic between an MSS and the BS.

An ARQ mechanism can be enabled on a per-connection basis.

A MAC Protocol Data Unit (PDU) begins with a fixed-length header. The header may be followed by the payload of the MAC PDU. If present, the payload consists of zero or more subheaders and zero or more MAC Service Data Units (SDUs) and/or fragments thereof. A MAC PDU may contain an error detection field (CRC), which is mandatory only if ARQ is enabled. Among the header fields are the CID, the packet length (LEN) and a field (Type) specifying basically which subheaders are employed. Subheaders allow to exploit two optional functions, *fragmentation* and *packing*, aiming at more efficiently using the available bandwidth.

At first let us consider the case ARQ is not enabled. When fragmentation is exploited, a MAC SDU is divided into one or more MAC PDUs. The fragment carried by a PDU is tagged with its position in the SDU (*first fragment*, *continuing fragment* or *last fragment*) and assigned a sequence number. The tag and the sequence number, which are used for reassembly, are inserted in a Fragmentation SubHeader (FSH), the former in a field called Fragmentation Control (FC), the latter in a field called Fragmentation Sequence Number (FSN). The standard allows to add a FSH even if the MAC PDU carries an unfragmented SDU. In this case, the FC field is set to *unfragmented*.

Packing provides for several SDUs to be encapsulated in a MAC PDU. A Packing SubHeader (PSH) is normally inserted before each SDU and a Length field in the PSH specifies the sum of the PSH size plus the SDU size. However, SDUs can be packed back-to-back without any separator if fixed-size SDUs are carried through the connection; the recipient is able to extract them from the PDU since it gets to know their size during the connection setup.

Packing and fragmentation could also be combined: a PDU could carry a number of unfragmented SDUs together with one or more fragments of other SDUs as well as it could pack two or more fragments derived from the same SDU or from different SDUs. Since PSHs are used, the PSH format also contains FC and FSN fields.

Now let us consider that ARQ is enabled on a connection. In this case, each MAC SDU is fragmented into blocks (called *ARQ blocks*), which are assigned Block Sequence Numbers (BSNs) available within the sending window.

Whenever a MAC PDU has to be created, the sender grabs and encapsulates into the PDU a number of ARQ blocks. If they derive from the same SDU and are contiguous, a FSH subheader is used; otherwise, packing is used and a PSH subheader is inserted before each sequence of blocks which are contiguous and related to a same SDU. In both cases, the FSN field of a subheader contains the BSN assigned to the first block in the sequence.

The number of ARQ blocks encapsulated into a PDU can vary from packet to packet. A PDU could carry blocks transmitted for the first time as well as blocks to be retransmitted. If needed, blocks could also be rearranged.

The acknowledgment information, named ARQ Feedback (ARQF) in the standard, consists of a list of ARQ Feedback Information Elements (IEs). Each IE has a header and, depending on the IE type, can contain a number of maps (up to four). Four IE types have been defined (Fig. 1): (a) *Cumulative* IE, which is employed to cumulatively acknowledge blocks (in this case, no map is included); (b) *Selective* IE, which uses the maps as bitmaps with each bit indicating whether a given block has been correctly received or not; (c) *Cumulative and Selective* IE, which is a combination of previous two types; (d) *Cumulative and Block Sequence* IE, which, in addition to cumulatively acknowledge blocks, exploits each map to specify two or three sequences, each one representing a sequence of contiguous blocks either correctly received or not correctly received.

An ARQ Feedback could be sent as a standalone MAC management message on the Basic management connection as well as piggybacked as a packed payload within a packed MAC PDU (Fig. 2). The presence of acknowledgment information within a PDU is indicated by a bit in the Type field. The standard specifies neither when an ARQ Feedback should be sent nor the number and the type of IEs to be inserted in the ARQF.

A number of ARQ parameters are defined. For most of them the standard specifies a set of allowed values and provides for the assumed value to be established during the connection setup.

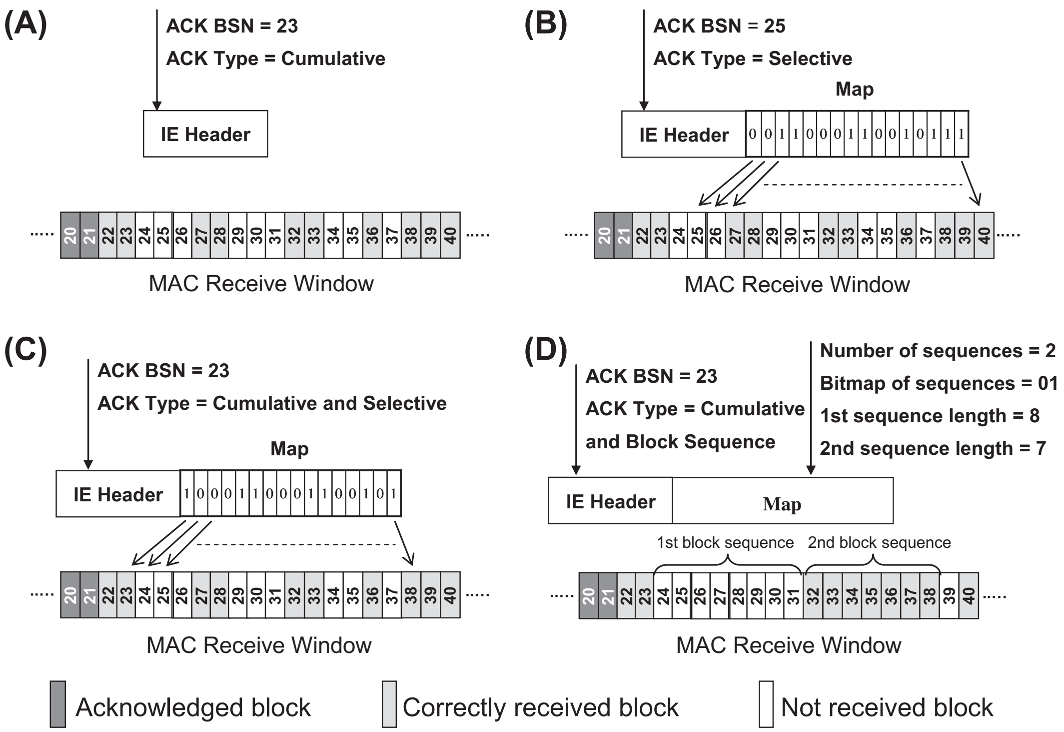

Fig. 1. ARQ Feedback IE structure.

Header	PSH	ARQ Feedback	PSH	Block Sequence #1	…	PSH	Block Sequence #*n*	CRC

Fig. 2. An ARQ Feedback is piggybacked into a data MAC PDU.

3. Markov channel and error models

The purpose of this Section is to describe the Channel Model and the Error Model we have adopted in order to define an analytical efficiency model of the ARQ protocol contemplated.

3.1. Channel model

In this work an urban scenario has been considered, so that only Rayleigh multipath fading has been taken into account in modelling the wireless channel. Lognormal shadowing has been neglected assuming that the average signal-to-noise ratio (SNR) γ_0 at the receiver can be kept constant by using power control at the transmitter.

The above assumptions have enabled us to adopt a channel model similar to that proposed in [7]. It consists of a finite-state DTMC S_n (Fig. 3), which is built by sampling the received instantaneous SNR $S(t)$ with a period T_S, i.e. the slot duration within an IEEE 802.16e frame (see Section 2), and by quantizing it to N levels with respect to a set of thresholds $\Gamma_0 = 0, \Gamma_1, \Gamma_2, \ldots, \Gamma_N = +\infty$. The state of the process S_n is C_i ($i = 1,2,\ldots,N$) if the sampled SNR, $S(nT_S)$, is in the range $[\Gamma_{i-1}, \Gamma_i)$. Several methods have been proposed to find the thresholds Γ_i (for $i = 1,2,\ldots,N-1$). In our work

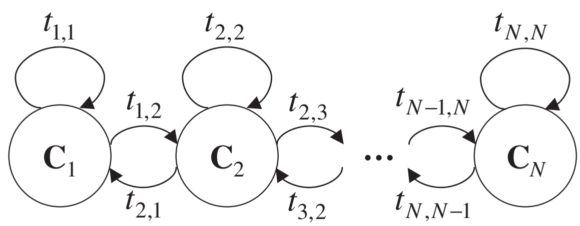

Fig. 3. Markov channel model.

we determine them by using the method proposed in [7], which partitions SNR levels making the fade state duration equally likely for every range $[\Gamma_{i-1}, \Gamma_i)$.

Now, let $N(\Gamma_i)$, $i = 1,2,\ldots,N$, denote the frequency with which $S(t)$ crosses the signal level Γ_i in the positive (or negative) direction. $N(\Gamma_i)$ can be obtained as in [7]:

$$N(\Gamma_i) = \sqrt{\frac{2\pi\Gamma_i}{\gamma_0}} f_m \exp\left(-\frac{\Gamma_i}{\gamma_0}\right), \tag{1}$$

where f_m is the maximum Doppler frequency.

The average duration over which $S(t)$ stays in $[\Gamma_{i-1}, \Gamma_i)$ can be expressed as

$$\tau_i = \frac{p_i}{N(\Gamma_{i-1}) + N(\Gamma_i)}, \tag{2}$$

where $\{p_i,\ i = 1,2,\ldots,N\}$ is the steady-state distribution of the chain S_n. In fact, τ_i can be evaluated as the ratio of the total time that $S(t)$ remains in $[\Gamma_{i-1}, \Gamma_i)$ in a interval T and the mean number of such signal segments during this interval. The former is given by p_iT, since p_i represents the long term proportion of time the process S_n spends in state C_i [8], the latter is equal to $[N(\Gamma_{i-1}) + N(\Gamma_i)]T$, since $[N(\Gamma_{i-1}) + N(\Gamma_i)]$ is the mean number of times per time unit that $S(t)$ leaves the range $[\Gamma_{i-1},\ \Gamma_i)$.

Instantaneous SNR $S(t)$ in a Rayleigh-fading channel is exponentially distributed, so that the probability density function $f_{S(t)}(\gamma)$ is

$$f_{S(t)}(\gamma) = \frac{1}{\gamma_0}\exp\left(-\frac{\gamma}{\gamma_0}\right), \quad \gamma \geqslant 0 \tag{3}$$

and the probability p_i, for $i = 1,2,\ldots,N$, can be obtained as

$$\begin{aligned} p_i &= \int_{\Gamma_{i-1}}^{\Gamma_i} f_{S(t)}(\gamma)d\gamma = \int_{\Gamma_{i-1}}^{\Gamma_i} \frac{1}{\gamma_0}\exp\left(-\frac{\gamma}{\gamma_0}\right)d\gamma \\ &= \exp\left(-\frac{\Gamma_{i-1}}{\gamma_0}\right) - \exp\left(-\frac{\Gamma_i}{\gamma_0}\right). \end{aligned} \tag{4}$$

In order to make the average fade duration of each state C_i large enough to cover the sampling period T_S, we can set τ_i equal to cT_S with $c > 1$. The constant c should be chosen between 3 and 8, as recommended in [7].

Finally, to complete the model, let us consider the transition probabilities $t_{i,j} = P(S_{n+1} = j|S_n = i)$ of S_n. Assuming that the transitions of S_n between non-adjacent states occur with a very low probability, the quantities $t_{i,j}$ can be evaluated as:

$$\begin{aligned} &t_{i,i+1} \approx \frac{N(\Gamma_i)T_S}{\pi_i}, \quad i = 1,2,\ldots,N-1, \\ &t_{i,i-1} \approx \frac{N(\Gamma_{i-1})T_S}{\pi_i}, \quad i = 2,3,\ldots,N, \\ &t_{i,j} \approx 0, \quad |i-j| \geqslant 2, \\ &t_{i,i} = 1 - \sum_{j=1, j\neq i}^{N} t_{i,j}, \quad i = 1,2,\ldots,N. \end{aligned} \tag{5}$$

The above formulae derive from the following considerations: (a) the product $\pi_i t_{i,j}$ represents the frequency of transitions (in transitions/slot) from state C_i to state C_j of S_n [8]; (b) the product $N(\Gamma_i)T_S$ represents the frequency of transitions (in transitions/slot) from state C_i to state any C_j such that $j > i$, but, if the transitions of S_n between non-adjacent states occur with a very low probability ($t_{i,j} \approx 0$ if $|i-j| \geqslant 2$), the product $N(\Gamma_i)T_S$ is a good approximation of the mean number of transitions per slot from state C_i to state C_{i+1} (i.e. $\pi_i t_{i,i+1}$) and, analogously, $N(\Gamma_{i-1})T_S$ is a good approximation of the mean number of transitions per slot from state C_i to state C_{i-1} (i.e. $\pi_i t_{i,i-1}$).

3.2. Error model

An error model (Fig. 4) built on the chain S_n described above has been proposed in [6]. We have adopted the same model, but this has required the assumption that the burst profile can be adapted once a time slot (see Section 2) and, moreover, an extension in order to take into account AMC systems [9]. In fact, the model in [6] considers AMS techniques only.

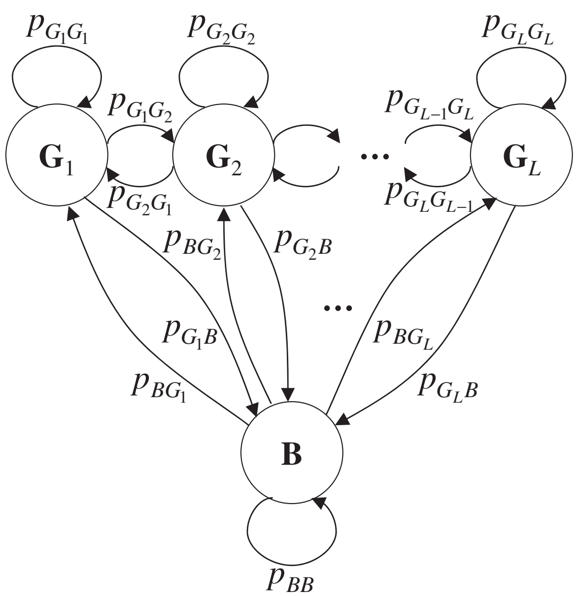

Fig. 4. Markov error model.

The error model consists of a DTMC, $X(n)$, which is built as in [6] by considering $L + 1$ receiver SNR threshold values that determine the burst profile to be used. In particular, each state C_i of S_n is associated with the burst profile φ_i which leads to the highest spectral efficiency and also satisfies a given *target FER*. Note that the above thresholds are different compared to those used for quantizing the process $S(t)$. The chain $X(n)$ has a $L + 1$ state error structure with one "bad" state and L "good" states corresponding to the L burst profiles. The state of $X(n)$ is $\mathbf{G}_k$ (the kth "good" state) if a FEC block transmitted at time n can be correctly decoded and carries R_k bytes, with $1 \leqslant k \leqslant L$; the state of $X(n)$ is $\mathbf{B}$ (the "bad" state) if the FEC block transmitted at time n is received in error. We have supposed the FEC block transmitted during a slot n such that $X(n) = \mathbf{B}$ codes R_m bytes if $\mathbf{G}_m$ was the last "good" state visited before entering state $\mathbf{B}$. This assumption results accurate enough if each burst profile is associated with many states of the process S_n. In fact, under this condition it is rather unlikely that, during the sojourn of $X(n)$ in state $\mathbf{B}$, SNR variations are so large as to determine a change of the burst profile. As an alternative approach, we could have adopted an error model consisting of L "good" states and L "bad" states [4], with each "bad" state associated with a different burst profile. Anyway, in most practical cases the above condition holds, so that the lower number of states of the chain $X(n)$ has driven us to adopt the model proposed in [6].

The transition probabilities of the chain $X(n)$ can be calculated as described in [6] and in [10]. In particular, they can be expressed as a function of the transition probabilities of the chain S_n and of the average FEC block Error Rate F_i for each state C_i of S_n, $i = 1,2,\ldots,N$.

As in [6] the transition probability p_{BB} from state **B** to state **B** is given by:

$$\begin{aligned} p_{BB} &= P\{\text{error at } t = n+1|\ \text{error at } t = n\} \\ &= \frac{P\{\text{error at } t = n+1,\ \text{error at } t = n\}}{P\{\text{error at } t = n\}} \\ &= \frac{p_1F_1t_{1,1}F_1 + p_1F_1t_{1,2}F_2 + p_2F_2t_{2,1}F_1 + \ldots}{\sum_{i=1}^{N} p_iF_i} \\ &= \frac{\sum_{i=1}^{N}\sum_{j=1}^{N} p_iF_it_{i,j}F_j}{\sum_{i=1}^{N} p_iF_i} \end{aligned} \tag{6}$$

In the same manner, we obtain the other transition probabilities:

$$p_{BG_k} = \frac{\sum_{i=1}^{N}\sum_{\varphi_j=k} p_iF_it_{i,j}(1-F_j)}{\sum_{i=1}^{N} p_iF_i},\ k = 1,2,\ldots,L, \tag{7}$$

$$p_{G_kB} = \frac{\sum_{\varphi_i=k}\sum_{j=1}^{N} p_i(1-F_i)t_{i,j}F_j}{\sum_{\varphi_i=k} p_i(1-F_i)},\ k = 1,2,\ldots,L, \tag{8}$$

$$p_{G_kG_z} = \frac{\sum_{\varphi_i=k}\sum_{\varphi_j=z} p_i(1-F_i)t_{i,j}(1-F_j)}{\sum_{\varphi_i=k} p_i(1-F_i)},\ k,\ z = 1,2,\ldots,L. \tag{9}$$

The average FER F_i for the state C_i can be derived by the expression of FER $F(\gamma,k)$ as a function of the received SNR γ and of the burst profile k. Since instantaneous SNR is exponentially distributed, F_i can be evaluated by

$$F_i = \frac{1}{p_i}\int_{\Gamma_{i-1}}^{\Gamma_i} F(\gamma,\varphi_i)\frac{1}{\gamma_0}\exp\left(-\frac{\gamma}{\gamma_0}\right)d\gamma. \tag{10}$$

Based on work in [9], the following analytical approximation can be considered for FER:

$$F(\gamma,k) = \begin{cases} 1 & \text{if } 0 < \gamma < \gamma_{pk}, \\ a_k\exp(-g_k\gamma) & \text{if } \gamma \geqslant \gamma_{pk}, \end{cases} \tag{11}$$

where a_k, g_k and γ_{pk} can be evaluated by a least-squares fitting of the exact FER curves obtained by simulation.

4. Efficiency of ARQ protocol

In this paper we consider an acknowledgment policy of the ARQ protocol specified in IEEE 802.16e and set the values of some related parameters (see Section 6) so that the following assumptions, on which the analytical model we derive for the ARQ protocol efficiency is based, are acceptable:

(a) the source is greedy;
(b) MAC PDUs can be sent back-to-back;
(c) useless retransmissions are avoided.

Then, let h (in bytes) denote the PDU overhead (header, subheaders and CRC) and l denote the amount (in bytes) of carried user data, so that the total PDU size is equal to $l + h$. Since the overhead h does not vary much with l, h is assumed to be a constant. Then, let π_k, for $0 \leqslant k \leqslant L$ ($k = 0$ refers to state **B**), be the steady-state distribution of the chain $X(n)$.

Under the hypothesis that the FEC block transmitted during a slot n such that $X(n) = \mathbf{B}$ codes R_m bytes if $\mathbf{G}_m$ was the last "good" state visited before entering state **B** (see Section 3), the mean number R_0 of bytes which are coded into a FEC block transmitted during a slot n such that $X(n) = \mathbf{B}$ is

$$R_0 = \sum_{k=1}^{L} R_kP(E_k), \tag{12}$$

where $P(E_k)$ is the probability of the event $\{\mathbf{G}_k$ is the last "good" state visited before entering state $\mathbf{B}\}$. The probability $P(E_k)$ can be calculated as the ratio between the flow from the state $\mathbf{G}_k$ to the state **B** and the flow from any "good" state to the "bad" state. The former is equal to $p_{G_kB}\pi_k$, the latter can be expressed as $(1 - p_{BB})\pi_0$. In fact, by using the global balance equation for the state **B**,

$$\sum_{k=1}^{L} p_{G_kB}\pi_k = \sum_{k=1}^{L} p_{BG_k}\pi_0 = \pi_0\sum_{k=1}^{L} p_{BG_k} = (1-p_{BB})\pi_0. \tag{13}$$

Hence,

$$R_0 = \sum_{k=1}^{L} R_k\frac{p_{G_kB}\pi_k}{(1-p_{BB})\pi_0} = \frac{1}{\pi_0}\sum_{k=1}^{L} R_k\pi_k\frac{p_{G_kB}}{1-p_{BB}}. \tag{14}$$

Let us evaluate the efficiency η of the ARQ protocol as the ratio between the mean number S_B of information bytes delivered to the upper layer in a slot and the maximum number R_L of bytes a FEC block can carry:

$$\eta = \frac{S_B}{R_L}. \tag{15}$$

Since useless retransmissions are avoided, S_B is equal to l multiplied for the mean number S_P of PDUs transmitted successfully in a slot. In turn, S_P is equal to the mean number G_P of packets transmitted per slot multiplied by the probability P_S that a packet transmission is successful. So,

$$\eta = \frac{lS_P}{R_L} = \frac{lG_PP_S}{R_L}. \tag{16}$$

Since G_P is equal to the ratio of the mean number R of bytes which are transmitted in a slot and the packet size $(l + h)$, we obtain

$$\eta = \frac{l}{l+h}\frac{R}{R_L}P_S. \tag{17}$$

Using (14), the value of R can be expressed as

$$\begin{aligned} R &= \sum_{k=0}^{L} R_k\pi_k = R_0\pi_0 + \sum_{k=1}^{L} R_k\pi_k \\ &= \sum_{k=1}^{L} R_k\pi_k\frac{p_{G_kB}}{(1-p_{BB})} + \sum_{k=1}^{L} R_k\pi_k \\ &= \sum_{k=1}^{L} R_k\pi_k\left(\frac{p_{G_kB}}{1-p_{BB}} + 1\right). \end{aligned} \tag{18}$$

By assuming that the source can transmit during U consecutive slots of every frame and that a frame consists of V slots (with $U \leqslant V$), the throughput S, in bytes per time unit, is given by:

$$S = \frac{S_B}{T_S}\frac{U}{V} = \eta\frac{R_L}{T_S}\frac{U}{V}. \tag{19}$$

In order to calculate P_S, let $p(s|k)$ denote the probability that the transmission of s consecutive bytes is successful under the condition that it starts in a slot n such that $X(n) = \mathbf{G}_k$. Moreover, let $\pi_k^{(a)}$ be the steady-state probability that $X(n) = \mathbf{G}_k$ upon arrival of a PDU, i.e. at time when PDU transmission starts. From the theorem of total probability, P_S can be written as:

$$P_S = \sum_{k=1}^{L} p(l+h|k)\pi_k^{(a)}. \tag{20}$$

If the source can use all the slots in a frame for data transmission ($U = V$), $p(s|k)$, for $1 \leqslant s \leqslant l + h$ and for $1 \leqslant k \leqslant L$, is equal to:

$$p(s|k) = \begin{cases} 1 & \text{if } s \leqslant R_k, \\ \sum_{j=1}^{L} p_{G_kG_j}p(s-R_k|j) & \text{if } s > R_k. \end{cases} \tag{21}$$

Note that a FEC block can carry bytes belonging to different PDUs. Then, under the condition that PDUs are transmitted back-to-back, for $1 \leqslant k \leqslant L$:

$$\pi_k^{(a)} = \frac{R_k\pi_k}{R}. \tag{22}$$

In fact, the probability $\pi_k^{(a)}$ represents[2] the proportion of arrivals which find the chain $X(n)$ in state $\mathbf{G}_k$. Within a sufficiently large time window of δ slots, in average there are $\delta R/(l + h)$ arrivals, a part of which, $\delta R_k\pi_k/(l + h)$, finds the chain $X(n)$ in state $\mathbf{G}_k$.

Let us assume now that $U < V$. Let, then, $p(s|k,z)$ denote the probability that the transmission of s consecutive bytes is successful under the conditions that it starts in a slot n such that $X(n) = \mathbf{G}_k$ and that the slot n is the zth slot available in a frame. Observing that, for $1 \leqslant s \leqslant l + h$, $1 \leqslant k \leqslant L$ and $1 \leqslant z \leqslant U$,

$$p(s|k,z) = \begin{cases} 1 & \text{if } s \leqslant R_k, \\ \sum_{j=1}^{L} p_{G_kG_j}p(s-R_k|j,z+1) & \text{if } s > R_k,\ z < U, \\ \sum_{j=1}^{L} p_{G_kG_j}^{(V-U+1)}p(s-R_k|j,1) & \text{if } s > R_k,\ z = U, \end{cases} \tag{23}$$

where $p_{G_kG_j}^{(m)}$ is the m-step transition probability from state $\mathbf{G}_k$ to state $\mathbf{G}_j$ of $X(n)$, the success probability can be written:

$$P_S = \frac{1}{U}\sum_{k=1}^{L}\sum_{z=1}^{U} p(l+h|k,z)\pi_k^{(a)}. \tag{24}$$

[2] Note that PASTA property does not hold, since the PDU arrival process is not a Poisson process.

5. A MAC PDU size control algorithm

5.1. Evaluation of the optimal packet size

The efficiency model defined in previous Section is very accurate as clearly shown by the curves illustrated in Section 7. We could differentiate η in (17) with respect to l in order to determine the packet size l_{opt} which maximizes the efficiency of ARQ protocol. This results, however, quite difficult, so that we have derived an approximated model, $Y(n)$, by which the optimal packet size can be evaluated in a simpler way. The process $Y(n)$ is a two-state DTMC built on the error model $X(n)$: $Y(n)$ is "good" ($\mathbf{g}$) if $X(n) \neq \mathbf{B}$, $Y(n)$ is "bad" ($\mathbf{b}$) if $X(n) = \mathbf{B}$. Then, the steady-state probabilities of $Y(n)$, π_g for state $\mathbf{g}$ and π_b for state $\mathbf{b}$, are:

$$\begin{aligned} \pi_g &= \sum_{k=1}^{L}\pi_k, \\ \pi_b &= \pi_0. \end{aligned} \tag{25}$$

The mean sojourn time of $Y(n)$ in state $\mathbf{b}$ is equal to the mean sojourn time of $X(n)$ in state $\mathbf{B}$, so that the transition probability p_{bb} is equal to p_{BB}. Furthermore, the transition probability p_{gg} can be derived from the following balance equation [8]

$$(1-p_{gg})\pi_g = (1-p_{bb})\pi_b. \tag{26}$$

Now, let R_g denote the mean number of bytes which are transmitted in a slot n such that $Y(n) = \mathbf{g}$. Then,

$$\begin{aligned} R_g &= \sum_{k=1}^{L} R_k P(X(n) = \mathbf{G}_k | X(n) \neq \mathbf{B}) \\ &= \sum_{k=1}^{L} R_k \frac{P(X(n) = \mathbf{G}_k, X(n) \neq \mathbf{B})}{P(X(n) \neq \mathbf{B})} \\ &= \sum_{k=1}^{L} R_k \frac{P(X(n) = \mathbf{G}_k)}{P(X(n) \neq \mathbf{B})} = \sum_{k=1}^{L} R_k \frac{\pi_k}{1-\pi_0}. \end{aligned} \tag{27}$$

Moreover, let us consider the random variable τ representing the difference between the arrival instant of a PDU (i.e., the instant at which the transmission of a PDU starts) within a given slot and the beginning of that slot (see Fig. 5). Assuming that τ is uniformly distributed in $[0,T_S)$, so that its mean value $E[\tau]$ is equal to $T_S/2$, and considering that the transmission of a packet starts immediately after the end of the transmission of the previous one (see Section 4), the mean transmission time (in slots) of a packet is an integer number.

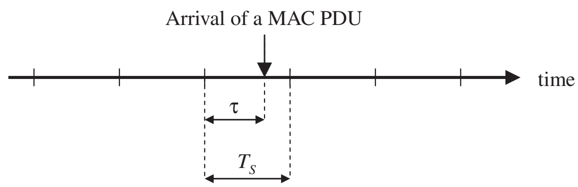

Fig. 5. Random variable τ represents the difference between the arrival instant of a packet and the beginning of the arrival slot.

Based on the values in Table 1, we can suppose that $l + h \geqslant R_g$ and, hence, the success probability P_S of a PDU can be calculated as

$$P_S = P\left\{Y(n) = \mathbf{g} \text{ upon arrival of the PDU}, Y(n+1) = \mathbf{g}, \ldots, Y\left(n + \frac{l+h}{R_g} - 1\right) = \mathbf{g}\right\}. \tag{28}$$

If $\pi_g^{(a)}$ denotes the $P\{Y(n) = \mathbf{g}$ upon arrival of the PDU$\}$, we obtain

$$\begin{aligned}P_S &= P\left\{Y(n+1) = \mathbf{g}, \ldots, Y\left(n + \frac{l+h}{R_g} - 1\right) = \mathbf{g} | Y(n) = \mathbf{g} \text{ upon arrival of the PDU}\right\} \pi_g^{(a)} \\ &= P\left\{Y(n+2) = \mathbf{g}, \ldots, Y\left(n + \frac{l+h}{R_g} - 1\right) = \mathbf{g} | Y(n) = \mathbf{g} \text{ upon arrival of the PDU}, Y(n+1) = \mathbf{g}\right\} \cdot \\ &\quad P\left\{Y(n+1) = \mathbf{g} | Y(n) = \mathbf{g} \text{ upon arrival of the PDU}\right\} \pi_g^{(a)} \\ &= P\left\{Y(n+2) = \mathbf{g}, \ldots, Y\left(n + \frac{l+h}{R_g} - 1\right) = \mathbf{g} | Y(n+1) = \mathbf{g}\right\} p_{gg} \pi_g^{(a)} \\ &= P\left\{Y(n+3) = \mathbf{g}, \ldots, Y\left(n + \frac{l+h}{R_g} - 1\right) = \mathbf{g} | Y(n+2) = \mathbf{g}\right\} p_{gg}^2 \pi_g^{(a)} \\ &= \cdots = (p_{gg})^{\frac{l+h}{R_g} - 1} \pi_g^{(a)}.\end{aligned} \tag{29}$$

Using the same approach adopted in deriving formula (22), we obtain

$$P_S = \frac{R_g \pi_g}{R} (p_{gg})^{\frac{l+h}{R_g} - 1} = \frac{R_g}{R} (1 - \pi_0)(p_{gg})^{\frac{l+h}{R_g} - 1}. \tag{30}$$

From (17) and (30), the efficiency η can be written as:

$$\eta = \frac{l}{l+h} \frac{R}{R_L} \frac{R_g}{R} (1 - \pi_0)(p_{gg})^{\frac{l+h}{R_g} - 1} = \frac{l}{l+h} \frac{R_g}{R_L} (1 - \pi_0)(p_{gg})^{\frac{l+h}{R_g} - 1}. \tag{31}$$

Taking the derivative of (31) with respect to l and imposing it equal to 0, we obtain, after some simplifications, the following equation:

$$l(l+h)\ln(p_{gg}) + R_g h = 0. \tag{32}$$

Solving (32) for the unknown l, the value l_{opt} which maximizes the efficiency η is given by

$$l_{opt} = \sqrt{\frac{h^2}{4} - \frac{R_g h}{\ln(p_{gg})}} - \frac{h}{2}. \tag{33}$$

If FER is not too high, we can assume p_{gg} very close to one. Then, under the condition $(1 - p_{gg}) \ll 1$,

$$\ln(p_{gg}) = \ln[1 - (1 - p_{gg})] \approx -(1 - p_{gg}) \tag{34}$$

and

$$l_{opt} \approx \sqrt{\frac{h^2}{4} + \frac{R_g h}{1 - p_{gg}}} - \frac{h}{2} = \sqrt{\frac{h^2}{4} + Mh} - \frac{h}{2} = \frac{h}{2}\left(\sqrt{1 + \frac{4M}{h}} - 1\right), \tag{35}$$

where $M = R_g/(1 - p_{gg})$, i.e. the product of R_g and the mean sojourn time (in slots) of $Y(n)$ in state "good".

So, the optimal packet size can be evaluated as a function of the overhead h and the mean number M of bytes which are transmitted during a sojourn of $Y(n)$ in state "good".

Note that normally $h \ll M$, so that the following approximate solution can be also considered:

$$l_{opt} \approx \frac{h}{2}\sqrt{\frac{4M}{h}} = \sqrt{Mh}. \tag{36}$$

The value of l_{opt} in (35) has been derived assuming that a source can use all the slots in a frame for data transmission and that the burst profile can vary from slot to slot. However, its accuracy has been also proved when a source can transmit only during U consecutive slots of every frame and the burst profile can vary once a frame, that is the maximum adaptation rate specified in IEEE 802.16e standard.

5.2. Algorithm description

The algorithm proposed in this paper (see Table 2) just exploits the model given in (36). The value l_{opt} is dynamically evaluated by the sender of an IEEE 802.16e connection and used to determine the number of ARQ blocks which each PDU should carry.

To better capture the dynamics of wireless channel, the algorithm provides for M to be iteratively estimated by an Exponential Weighted Moving Average (EWMA): the algorithm evaluates the EWMA W of values assumed by the variable A representing the number of bytes transmitted during a sojourn time of $Y(n)$ in "good" state.

At the beginning of each sojourn, A is set to zero, as well as the variables N and H, which denote, respectively, the number of PDUs transmitted during the sojourn and the total control information carried by them. Afterwards, the values of A, N and H are updated on the basis of the ARQ Feedbacks received.

The sender manages a list which contains some information about transmitted PDUs: each element consists of the identifier id of a PDU, the size s_{id} of the PDU and the size of its control information h_{id}. When a PDU is sent, the sender associates the identifier id with every ARQ block carried by it.

Upon arrival of an acknowledgment for an ARQ block, the sender checks if in the list there is an element for the identifier id associated with the block. In that case, if the acknowledgment is positive, s_{id} and h_{id} are used to increase, respectively, A and H; otherwise, if the acknowledgment is negative and $N > 0$, the following three actions are taken: (a) A is increased by the value (A/N), which represents an estimation of the mean packet size s, based on the assumption that the random variables x_1 and x_2 shown in Fig. 6 are uniformly distributed in $[0,s)$; (b) W is evaluated; (c) the value of l_{opt} is updated by using (36) and adopting the arithmetic mean (H/N) of control information size per packet as an estimation of h. The element for the identifier id is then removed from the list.

Note that a negative acknowledgment should be transmitted soon after the end of a sojourn of $Y(n)$ in "bad" state. Hence, upon arrival of this control information, if $N > 0$ the values of A, N and H related to the last sojourn in "good" state are used to update l_{opt}. Actually, the bytes

Table 2
Pseudo-code of the algorithm.

```
Variables:
A: number of bytes transmitted in the current sojourn of Y(n) in "good" state;
N: number of PDUs transmitted in the current sojourn;
H: control information transmitted in the current sojourn;
W: EWMA of the values of A associated with current and previous sojourns;
Initialization:
A = 0;    N = 0;    H = 0;    W = 0.
-----------------------------------------------------------------
Algorithm:
Upon arrival of an acknowledgment for an ARQ block which is associated with the identifier id do:
If there is an element for the identifier id in the list
    If the acknowledgment is positive
        A = A + s_id;
        H = H + h_id;
        N = N + 1;
    Else If N > 0
        A = A + A/N;
        If W > 0
            W = αW + (1 − α)A;
        Else
            W = A;
        End If
        l_opt = sqrt(WH/N);
        A = 0;
        H = 0;
        N = 0;
    End If
    Remove the element from the list;
End If
```

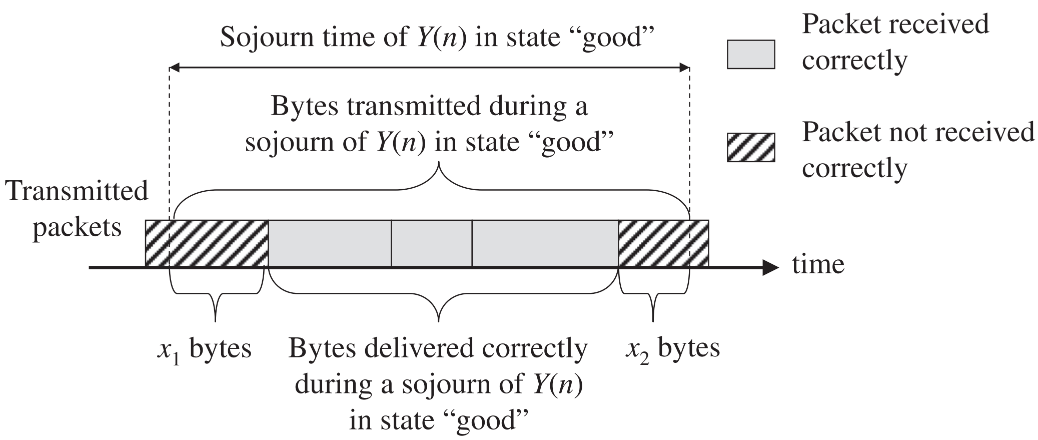

Fig. 6. Bytes transmitted and bytes delivered correctly during a sojourn of $Y(n)$ in "good" state.

transmitted during the same period and carried by PDUs received in error are not taken into account as the value of A includes only the bytes correctly delivered during the sojourn time (Fig. 6). For this reason, the action (a) is executed.

6. Simulation models

The efficiency models in Sections 4 and 5 have been validated by simulating a scenario in which a greedy source on top of MAC layer in a Base Station (BS) [1] sends packets to a MSS. Note that, since an IEEE 802.16e connection is unidirectional, two connections have to be established, one for the BS to send data packets to MSS and another for the MSS to send ARQ Feedback messages to BS.

After the validation of the ARQ protocol efficiency models, the effectiveness of the proposed algorithm in improving TCP performance when a MSS accesses an Internet FTP server through an IEEE 802.16e link has been evaluated. To this, the scenario in Fig. 7 has been simulated, considering, in particular, a reliable bidirectional data transfer on the wireless link.

Network Simulator v2 (NS-2) tool has been adopted. The protocol stack of a NS-2 node has been extended with an implementation of MAC and PHY layers of IEEE 802.16e.

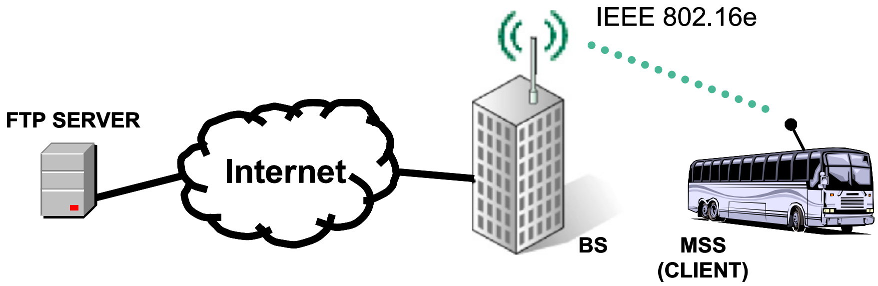

Fig. 7. Network model: a mobile user accesses an Internet server.

The added modules implement TDD/TDMA, FEC block creation, MAC PDU construction. The channel model proposed in [7] has been implemented. In particular, the constant c has been set to 3 (see Section 3). The source code is available at [11].

According to the standard specifications [1], the frame duration has been set to 2.5 ms, the bandwidth to 7 MHz, the duration of an OFDM symbol to 32 μs and the guard time to 1 μs (thus, the slot time T_S is equal to 33 μs) and, hence, a frame consists of $V = 75$ slots. The number U of slots per frame assigned to each IEEE 802.16e connection has been set to 5. The MAC layer exploits an AMC system and the bit rate varies in the range of [190; 1704] kb/s. Both fragmentation and packing have been adopted in order to format PDUs carrying a given number of ARQ blocks (i.e. a given packet size).

We consider an acknowledgment policy which provides for an ARQ Feedback to be sent whenever a MAC PDU which carries out-of-order ARQ blocks is received. In this way, the error recovery process is speeded up, as lost blocks can be retransmitted before the retry timer expires. An ARQ Feedback is also sent if an amount of time T_{ACK} has elapsed since the previous transmission of an ARQF. T_{ACK} has to be set to a value such that the objective of avoiding useless retransmissions due to ARQF losses is pursued.

The minimum time interval which a transmitter has to wait before retransmitting an unacknowledged ARQ block (*ARQ Retry Timeout*) has been set to the maximum value allowed by the standard. Thus, useless retransmissions due to premature timeouts are avoided and, in addition, the rate of useless retransmissions due to ARQF losses results very low, as a higher number of ARQ Feedbacks can be sent before the retry timer expires.

Each ARQ Feedback contains one only IE. Whenever an ARQF has to be sent, the receiver considers the IE types which allow to acknowledge the highest number of ARQ blocks and selects the one that introduces the lowest overhead.

To enable sending of MAC PDUs back-to-back, the *ARQ Block Size* and the maximum number of unacknowledged ARQ blocks at any given time (*ARQ Window Size*) have been set in such a way that their product is higher than the bandwidth-delay product. IEEE 802.16e specifications restrict the *ARQ Block Size* to powers of two ranging from 16 to 1024 bytes. Based on this range, the lowest value which would avoid the MAC window to be smaller than the TCP window and, thus, a bottleneck, is 128. Aiming at evaluating the adaptive algorithm effectiveness while maximizing the ARQ protocol performance, we have set the block size to 72 bytes instead of 128 bytes despite the standard specifications. In fact, the former value, besides avoiding the above bottleneck, also gives the adaptive algorithm a greater flexibility in the choice of the MAC PDU size.

Table 3
Main simulation parameters.

Parameter	Value
Size of the packets generated by the greedy source	576 bytes
TCP Maximum Segment Size	536 bytes
Frame Duration	2.5 ms
U	5 slots
V	75 slots
Bandwidth	7 MHz
Carrier frequency	2.4 GHz
Slot Time T_S	33 μs
MAC PDU Header size	6 bytes
Fragmentation SubHeader size	2 bytes
Packing SubHeader size	3 bytes
CRC size	4 bytes
ARQ Block Size	72 bytes
ARQ Retry Timeout	655.35 ms
ARQ Window Size	1024 blocks
T_{ACK}	100 ms
Distance between BS and MSS	2 km
MSS speed	50 km/h
Propagation time on wired portion	50 ms

The TCP performance has been expressed in terms of Goodput normalized with respect to the Goodput achieved in case the maximum PDU size (2048 bytes), the burst profile with the highest spectral efficiency and an error-free channel are considered.

All the results of the simulations are characterized by a 95% confidence interval whose maximum relative error is equal to 1%. Finally, Table 3 reports the main simulation parameters.

7. Simulation results

The curves in Fig. 8 refer to the validation of models in (17) and (31) and report the efficiency η versus the PDU size for several values of γ_0 in range [15; 30] dB. The curves have been obtained assuming that the source can use all the slots in a frame, the burst profile can vary from slot to slot and the *target FER* is equal to 0.3. They show that the model in (17) is very accurate. The approximated model is clearly less accurate than the model in (17), but nevertheless the related curves show its validity in estimating the optimal packet size. In fact, for a given γ_0, their maximum point is close enough to the maximum points of curves obtained for model in (17) and for simulation model.

The curves in Fig. 9 illustrate the influence of the parameter α on TCP performance when U consecutive slots of every frame are available for transmissions and the burst profile can vary once a frame. Normalized TCP Goodput versus the *target FER* for several values of γ_0 is reported. The curves show that, provided that γ_0 is not too high, the value of α does not significantly affect TCP performance, even though EWMA is not used (namely, for $\alpha = 0$, so $W = A$). Following this analysis, the parameter α has been set to 0.875, a value which allows to obtain high performance in the considered scenarios.

TCP performance obtained when the proposed algorithm is adopted (dynamic configuration, *Adaptive ARQ*) has been compared with that evaluated setting PDU size D to 2048 bytes (static configuration) and with the maximum achievable one (*Ideal ARQ*). Each point of the *Ideal*

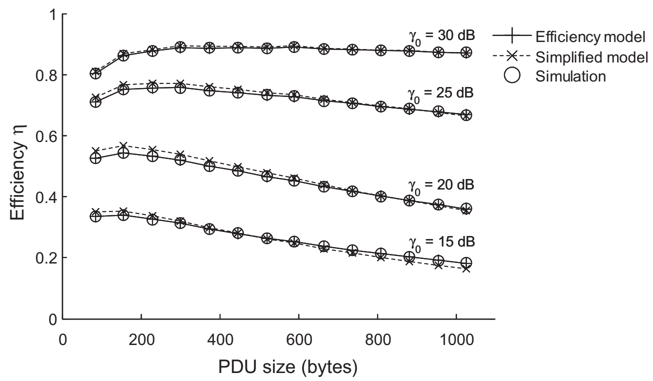

Fig. 8. Comparison between the efficiency model described in Section 4 and the simplified model in Section 5; *target FER* is equal to 0.3.

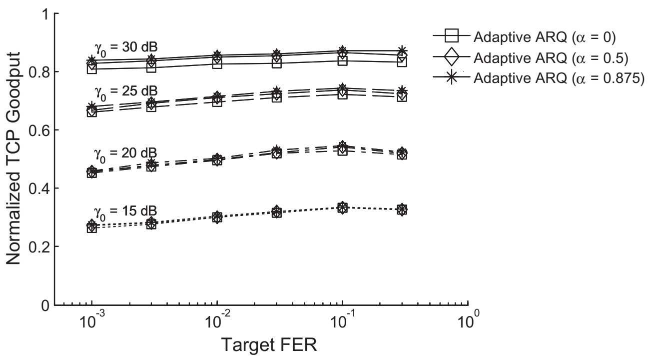

Fig. 9. Impact of α on TCP Goodput in case data link layer exploits the defined algorithm.

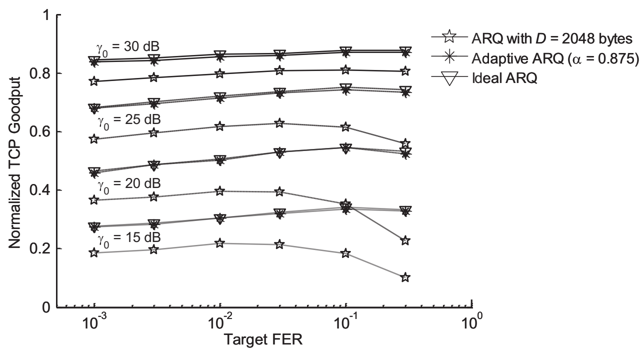

Fig. 10. Performance comparison between ARQ protocols with static and dynamic configuration exploiting the defined algorithm for $\alpha = 0.875$.

ARQ solution curves is associated to a given scenario and corresponds to the normalized TCP Goodput obtained choosing the optimal packet size in that scenario.

In Fig. 10 the normalized TCP Goodput versus *target FER* has been plotted for several values of *target FER* from 0.003 to 0.3 and for γ_0 in range $[15;30]$ dB. As expected, the curves show that TCP performance achieved by adopting the defined algorithm is much better than that obtained by setting D to 2048 bytes, especially for low γ_0 and for high *target FER*. TCP performance is very good and very close to the maximum achievable one (obtained by *Ideal ARQ*). This result is considerable and shows the effectiveness of the algorithm in estimating the optimal size.

8. Conclusion and future works

In order to improve network performance in a scenario including an IEEE 802.16e link, we have proposed a novel algorithm which can be used to adapt MAC PDU size to wireless channel behavior when ARQ is adopted at MAC layer. An analytical approach has been used aiming at dynamically evaluating the optimal packet size. The latter is derived from a model of the ARQ protocol efficiency which has been obtained by exploiting a finite-state Markov error model which also considers Adaptive Modulation/Coding. The effectiveness of the proposed algorithm in improving TCP performance has been shown.

Future research activities will address an analytical formulation of the ARQ protocol efficiency when OFDMA is employed and the analysis of the impact of the dynamic bandwidth allocation schemes adopted by IEEE 802.16e.

References

[1] IEEE 802.16e-2005, Air interface for fixed and mobile broadband wireless access systems – amendment for physical and medium access control layers for combined fixed and mobile operation in licensed bands, IEEE Standard, February 2006.
[2] E. Modiano, An adaptive algorithm for optimizing the packet size used in wireless ARQ protocols, Wireless Networks 5 (4) (1999) 279–286.
[3] E.I. Kim, J.R. Lee, D.H. Cho, Performance evaluation of data link protocol with adaptive frame length in satellite networks, IEICE Transactions on Communications E87-B (6) (2004) 1730–1736.
[4] J. Xiao, J. Qiu, S. Cheng, A joint adaptive packet size and modulation scheme combined with SR-ARQ over correlated fading channels, Proceedings of the Wireless Communications, Networking and Computing 1 (2005) 478–483.
[5] H. Martikainen, A. Sayenko, O. Alanen, V. Tykhomyrov, Optimal MAC PDU Size in IEEE 802.16, in: Proceedings of IT-NEWS 2008, 2008, pp. 66–71.
[6] J. Yun, M. Kavehrad, Markov error structure for throughput analysis of adaptive modulation systems combined with ARQ over correlated fading channels, IEEE Transactions on Vehicular Technology 54 (1) (2005) 235–245.
[7] Q. Zhang, S.A. Kassam, Finite-state Markov model for Rayleigh fading channels, IEEE Transactions on Communications 47 (11) (1999) 1688–1692.
[8] D. Bertsekas, R. Gallager, Data Networks, second ed., Prentice Hall, 1992.
[9] Q. Liu, S. Zhou, G.B. Giannakis, Cross-layer combining of adaptive modulation and coding with truncated ARQ over wireless links, IEEE Transactions on Wireless Communications 3 (5) (2004) 1746–1755.
[10] J. Yun, M. Kavehrad, Corrections to "Markov error structure for throughput analysis of adaptive modulation systems combined with ARQ over correlated fading channels", IEEE Transactions on Vehicular Technology 56 (2) (2007) 991.
[11] NS-2 extensions for IEEE 802.16e. <http://conetlab.unisalento.it/>.
[12] G. Ciccarese, M. De Blasi, P. Marra, C. Palazzo, L. Patrono, A packet size control algorithm for IEEE 802.16e, in: Proceedings of the IEEE Wireless Communications & Networking Conference 2008, 2008, pp. 1420–1425.

Giovanni Ciccarese received the Laurea degree in Electronic Engineering from the Politecnico di Torino, Italy, in 1989. He is currently an Assistant Professor at the Innovation Engineering Department of Faculty of Engineering (Lecce). His research interests include design, modelling and performance evaluation of communication protocols for wireless networks, with particular emphasis on protocols for wireless vehicular networks and underwater networks.

Mario De Blasi is full professor of Computer Networks at the University of Lecce, Italy. His research and teaching activities involved computer architecture, computer networks and distance learning. He has published papers and he has written some books within these areas. He has coordinated or participated in several large national and international projects (Med-Campus Project of EU, Interactive Satellite multimedia Information System – ISIS in ACTS of EU, ESA SkyNet and ESA MODUS). Furthermore, he has coordinated industrial research projects with Alenia Spazio S.p.a. and ST Microelectronics.
The main areas of the research group he coordinates are: modelling and performance analysis, mobile Internet, IEEE 802.11e, IEEE 802.16e, vehicular networks.
De Blasi has been a Chair of a Symposium on Future Wireless Systems organized within the Conference on Software in Telecommunications and Computer Networks (SoftCOM) since 2000.

Pierluigi Marra received the Laurea degree in Computer Engineering in July 2003 and Ph.D. in Computer Engineering in 2009 from the University of Salento, Italy. His research areas include design, modelling and performance evaluation of communication protocols over wireless networks. Recently his research interests have been focused on vehicular networking.

Cosimo Palazzo received the Laurea degree in Computer Engineering in 2003 and Ph.D. in Computer Engineering in 2007 from the University of Salento, Italy. His research areas include design, modelling and performance evaluation of communication protocols over wireless networks based on IEEE 802.11, IEEE 802.16 and related QoS issues. Recently, his activities have also concerned vehicular networks and underwater networks.

Luigi Patrono received the Laurea degree in Computer Engineering from the University of Lecce, Italy, in 1999. He received Ph.D. in Innovative Materials and Technologies at the ISUFI of Lecce, Italy, in 2003. He is currently an Assistant Professor at the Faculty of Engineering of the University of Lecce (Italy) in the Computer Network group. His research interests include design, modelling and performance evaluation of wireless communication protocols. In particular, main research topics are: mobile Internet, satellite networks, power saving in IEEE 802.11e, and IEEE 802.16e.