Revision of ISJ_1703

On the Use of Learning Automata in Tuning the Channel Split Ratio of WiMax Networks

Antonios Sarigiannidis, *Student Member, IEEE*, Petros Nicopolitids, *Senior Member, IEEE*, Georgios Papadimitriou, *Senior Member, IEEE*, Panagiotis Sarigiannidis, *Member, IEEE*, Malamati Louta, *Member, IEEE* and Andreas Pomportsis

***Abstract*—Worldwide Interoperability for Microwave Access (WiMAX) family of standards have introduced a flexible, efficient, and robust wireless interface. Among other interesting features, WiMAX access networks bring into play a flexible determination of the ratio between the downlink and the uplink directions, allowing a relation width from 3:1 to 1:1 respectively. However, this promising feature is not properly utilized, since hitherto scheduling and mapping schemes proposed neglect it. In this work, this challenging issue is effectively addressed by proposing an adaptive model that attempts to adequately adjust the downlink-to-uplink sub-frame width ratio according to the current traffic conditions. In the context of a mobile WiMAX wireless access network, the Base Station is enhanced with an error-aware Learning Automaton in order to be able to identify the magnitude of the incoming and the outgoing traffic flows and in turn to suitably define the ratio on a frame-by-frame basis. The model designed is extensively evaluated under realistic and dynamic scenarios and the results indicate that its performance is clearly improved compared to schemes having predefined, fixed ratio values.**

***Keywords*-IEEE 802.16, Learning Automata, mapping, OFDMA, WiMAX**

I. INTRODUCTION

In recent years, Broadband Wireless Access (BWA) systems have been gaining popularity as the next generation of wireless access infrastructure [1]. A broad industry consortium, the Worldwide Interoperability for Microwave Access (WiMAX) Forum has begun certifying broadband wireless products for interoperability and compliance with the WiMAX standard [2]. WiMAX is based on wireless metropolitan area networking (WMAN) standards developed by the IEEE 802.16 group. The IEEE 802.16 group was formed in 1998 to develop an air-interface standard for wireless broadband.

Manuscript received April 1, 2013.

A. Sarigiannidis, P. Nicopolitidis, G. Papadimitriou and A.Pomportsis are with the Department of Informatics, Aristotle University, 54124, Thessaloniki, Greece (e-mail: {asarigia, petros, gp, apombo}@csd.auth.gr). (corresponding author to provide phone: +30-2310998538; fax: +30-2310998221; e-mail: petros@csd.auth.gr). P. Sarigiannidis and M. Louta are with the Department of Informatics and Telecommunications Engineering, University of Western Macedonia, 50100, Kozani, Greece (e-mail: {psarigiannidis, louta}@uowm.gr).

In 2005, the IEEE group completed and approved IEEE 802.16e-2005, an amendment to the earlier IEEE 802.16-2004 standard that added mobility support. The IEEE 802.16e-2005 forms the basis for the WiMAX solution for nomadic and mobile applications and is often referred to as mobile WiMAX [3, 4].

The WiMAX Physical (PHY) layer is based on Orthogonal Frequency Division Multiple Access (OFDMA). OFDMA enables high-speed data, video, and multimedia communications and it is used by a variety of commercial broadband systems. OFDMA combines the Time Division Multiple Access (TDMA) and the Frequency Division Multiple Access (FDMA) schemes, resulting in allowing multiple subscribers to make use of different bandwidth regions in both the time and the frequency domains. The time domain is segmented into groups of OFDMA symbols and each symbol is segmented into sub-carriers. The minimum time-frequency resource unit that can be allocated by a WiMAX system to a given link is called a slot. Each slot consists of one sub-channel over one, two, or three OFDMA symbols, depending on the particular sub-channelization scheme used. In essence, the OFDMA technique is responsible for allocating PHY resources to Medium Access Control (MAC) requests [2].

Figure 1 depicts a sample TDD frame structure for mobile WiMAX. The frame is divided into two sub-frames: a downlink sub-frame followed by an uplink sub-frame after a small guard interval. This interval is used for signal interference avoidance between uplink and downlink signals. The downlink sub-frame begins with a preamble, which is used for synchronization and channel estimation. After the preamble, the Frame Control Header (FCH) is transmitted, which is a downlink control information used in providing frame configuration information such as coding and modulation information. Multiple users are allocated with data regions within the frame. These allocations are specified in the uplink and downlink MAP messages (DL-MAP and UL-MAP) that are broadcast following the FCH in the downlink sub-frame. Moreover, the MAP messages carry user-specific control information, required for the next sub-frames, such as the sub-channel and the symbol of users' transmission and reception. It is worth to note that the length relation between the downlink and uplink sub-frames is variable and defined by the downlink-to-uplink ratio. This feature could follow dynamic modifications of communication requirements in an efficient manner, altering the respective ratio on a frame-by-frame basis.

A typical IEEE 802.16e wireless access network provides a

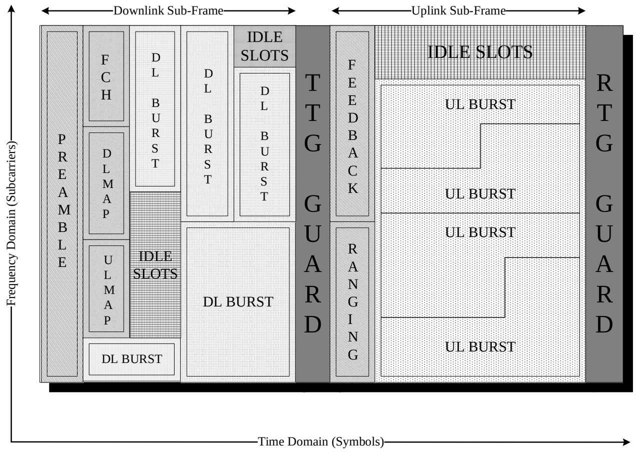

Figure 1. A TDD frame instance of mobile WiMAX wireless communication.

full-duplex communication, in which the Base Station (BS) and the connected Mobile Stations (MSs) are the two major entities that are involved. Firstly, the BS provides the air interface to the MSs. Additional functions that may be part of the BS are the establishment, preservation and maintenance of a feasible communication.

The two-way communication between the BS and the various MSs is realized via the downlink and the uplink flows. In the downlink direction, the BS is responsible for delivering data to the MSs, while in the uplink direction the MSs send data to the BS. Both directions are organized in time frames. Each frame is composed of two distinct regions dedicated to the downlink and the uplink communication, the uplink sub-frame and the downlink sub-frame respectively. The way of organizing the two sub-frames within the time frame is defined by the division duplexing technique employed. Fixed and mobile WiMAX standards support Frequency Division Duplexing (FDD) and Time Division Duplexing (TDD) techniques. In the FDD technique both downlink and uplink sub-frames are transmitted simultaneously over different carriers, so they have the same length. Some WiMAX deployments are likely to employ the TDD mode, since it offers simplicity and flexibility, hence the TDD technique is adopted in the proposed scheme for handling the two directions ordering within the OFDMA frame, focusing on TDD-based WiMAX wireless access networks [2].

Due to the fact that the standard does not provide specific algorithms for utilizing the available bandwidth through the OFDMA technique, an open research area is revealed, which focuses on effectively exploiting the available bandwidth. Nonetheless, the allocation and usage of bandwidth must abide by some specific rules and restrictions. Due to OFDMA's nature the available bandwidth is formed as a two-dimensional allocation bin in a rectangular shape, having the one dimension associated with frequency (height) and the other associated with time (width). Moreover, each downlink request (or a set of requests that share common PHY characteristics and is referred to as a burst) must follow the rectangular shape. More specifically, each downlink request should be formed as a two-dimensional rectangular within the OFDMA allocation bin.

OFDMA fastens the PHY layer and the MAC layer, employing two major processes, the QoS scheduling and the mapping [5]. Regarding the downlink sub-frame, the QoS scheduler is aware of the QoS requirements of the MAC data units and forwards these requests to the mapper. Then, a downlink program is constructed, upon the collection of the downlink bandwidth requests and each MS is informed about the time and the sub-frequency of its dedicated data. Concerning the uplink sub-frame, the mapper constructs an uplink program, which defines the exact time and the frequency of all uplink MSs' transmissions. Actually, the operation of the uplink QoS scheduler depends on the call admission policy of the system, which is out of the scope of this work. The efficiency of the scheduling and the mapping processes dramatically affects the system performance, since requests that fail to be mapped are returned to the respective scheduler and their transmission is postponed for at least one entire frame. Recognizing this issue, the standard supports flexible and dynamic adjustment of the downlink-to-uplink sub-frame width ratio, which may be varied from 3:1 to 1:1. This feature could be noticeably beneficial considering different traffic profiles, variant multimedia traffic etc.

Having in mind that most scheduling and mapping approaches found in literature consider a stable, predetermined, and likely arbitrary defined downlink-to-uplink sub-frame width ratio, in this paper we endeavor to effectively cover this gap, by proposing a fully adaptive model. The model presented takes into account both downlink and uplink feedback obtained by the mapping processes and applies a learning from experience framework in order to adequately adopt to the traffic requirements. The learning tool designed comes from the Learning Automata (LA) research field and it appropriately processes the feedback obtained along with the past recorded actions [6]. The model efficiency is calculated by its performance in terms of OFDMA resources utilization, service ratio, and allocation effectiveness. The novelty of our approach lies in the fact that it constitutes a dynamic and adaptive scheme for adjusting the downlink-to-uplink ratio based on the short-term network dynamics.

The rest of the paper is organized as follows. Section II presents the fundamental concepts of LA and highlights the characteristics that make their application to networking protocols an attractive approach. In order to substantiate this claim, Section II also briefly overviews representative LA-based networking protocols that have recently appeared in the literature, focusing mainly on wireless ones, as this is the area addressed by this paper. Section III describes related mapping techniques proposed in research literature. Section IV describes the introduced adaptive model. Section V evaluates the performance of the proposed model. Finally, Section VI concludes this paper.

II. Applications of Learning Automata in Communication Networks

Many networks operate in environments with unknown and time-varying characteristics. For wireless networks, representative examples of such characteristics are the network topology, the channel qualities, the location and total number of network nodes and the inter-node distances. The changing nature of such characteristics will profoundly affect network performance. This fact significantly affects the design of efficient networking protocols and as a consequence, adaptivity arises as one of the most important properties of such protocols.

LAs are artificial intelligence tools that can provide adaptation to systems operating in changing and/or unknown environments. A LA is a finite state machine that interacts with a stochastic environment and tries to learn the optimal action offered by the environment via a learning process. There is a bidirectional information exchange between the environment and the LA module. Specifically, the LA acts by selecting a specific action from a pool of possible actions and the environment reacts to the action taken by producing a feedback. The feedback is received by the LA and it is appropriately processed so as to compute the next action to perform. The process is repeated and finally leads the LA to select the best possible action from the pool of possible ones. The low computational complexity that a LA exhibits enables it to rapidly converge to the best action of the environment with which it interacts.

LAs have found use in a large number of wireless networking protocols, enabling them to efficiently operate in environments with unknown and time-varying properties [6-25].

A brief background of the LAs follows. Let $A = \{a_1, a_2, \ldots, a_m\}, m < \infty$ be the set of actions available. At each instant k, the automaton chooses an action $a(k) \in A$ based on its current action probability distribution $p(k) = \{p_1(k), p_2(k), \ldots, p_m(k)\}, k = 0,1, \ldots.$ It holds that $\sum_{i=1}^{m} p_i(k) = 1, \forall k$. The action chosen by the automaton is the input to the environment which responds with a stochastic reaction or reinforcement, $\beta(k) \in R \subseteq [0,1]$ where R is the set of possible reactions. Higher values of the reinforcement feedback are assumed more desirable. Let d_i denote the expected value of $\beta(k)$ given $a(k) = a_i$. Then d_i is called the reward probability associated with action $a_i, 1 \leq i \leq m$. Define the index r by $d_r = max_i\{d_i\}$. Then the action a_r is called the optimal action [24].

III. RELATED WORK

Existing scheduling and mapping techniques proposed in related research literature are inflexibly designed and define static bandwidth allocations. Present mapping schemes seem to ignore the downlink to uplink load balance ratio, proposing algorithms unaware of the efficiency of adaptively setting the downlink and uplink allocations. Various mapping schemes designed for the downlink sub-frame have been proposed identifying the downlink bandwidth distribution as more challenging issue than the uplink one because of the shape rule, i.e., the requests ought to be shaped as a two-dimensional rectangular. Indicatively, the reader could see the efforts in [26-30]. One of the earliest efforts in mapping could be found in [26], whereby the Simple Packing Algorithm (SPA) is designed to accommodate the incoming bandwidth requests within the downlink sub-frame. The scheme involves a top to bottom and left to right slot allocation, accommodating symbols (rows) and sub-channels (columns) for each request in a First In First Out discipline, until the requested number of slots is met. If this number is not an integer multiple of the frequency or the time dimension respectively, the remaining unallocated space remains idle. The authors in [27] apply a persistent mapping technique using a binary-tree full search tree, but the final result indicates complex operation, limited to eight users. Other attempts involve as a first step an initial request sorting in terms of the number of requested slots and as a second step either bucket definition and accommodation, where the combined bursts define buckets [28] that are accommodated in a column by column basis into the allocation bin, or heuristic packing algorithms, allocating the incoming requests in a two step procedure, comprising first a horizontal mapping and then vertical accommodation [29].

Our previous work includes the design of various mapping schemes giving emphasis to the fairness issue [30] and the QoS provisioning issue [31]. Nonetheless, recent studies on determining the channel split ratio in WiMAX wireless networks [32-34] indicate a compelling research area in OFDMA-based BWA networks. In [32] the authors study the determination of downlink and uplink channel split ratio for TDD-based IEEE 802.16 wireless networks. They focus on Transmission Control Protocol (TCP) based traffic and explore the impact of improper bandwidth allocation to downlink and uplink channels on the performance of TCP. In [33], the Dynamic Ratio Determination (DRD) algorithm was presented. This work constitutes an initial attempt on studying the impact of efficiently determining the ratio, but the employed technique needs to be adequately tuned since it dramatically depends on the environment. Furthermore, in [34] the Extrapolated Ratio Determination (ERD) algorithm was introduced, which applies a cubic spline extrapolation technique in order to estimate the appropriate ratio for the next frame. The ERD algorithm maintains a fixed-size history vector that is used so as to extrapolate the next value based on the past valued stored in the history vector. The problem of the adequate tuning also holds here, since the history vector is unaware of the traffic dynamics that take place in modern wireless networks.

The mapping algorithm designed in [35] presents considerable assets in view of performance. The so-called Adaptive Horizon Burst Mapping (AHBM) algorithm applies horizon-based allocation, creating initial pilots for the forthcoming requests. Large requests are accommodated first, leaving minimum remaining idle space, while pilots are formed in a right to left and bottom to top manner. In the sequel, the remaining requests are mapped based on the horizons. Based on extensive evaluation experiments, the performance of the AHBM scheme seems to be improved with respect to other leading schemes. Thus, it is adopted as the main downlink mapping technique for the rest of our study. Even though, the AHBM involves a prediction tool based on hidden markovian chains in order to redefine the length of the downlink sub-frame in accordance with downlink traffic profiles, it neglects the uplink part of mapping process.

On the other hand, the mapping process of the uplink sub-frame is much simpler, since the rectangular restriction does not apply in that case. The standard applies a simple accommodation technique that operates in a row-by-row basis. One after the other, uplink requests are scheduled into the uplink sub-frame, without occurrence of row cuts. Upon the accommodation of an uplink request, the following one is sided directly next to it, without leaving gaps (i.e., idle slots). In this manner, the set of uplink requests uniformly fills the allocation bin, until either all requests are mapped or the bin becomes full. This fixed and simple uplink mapping technique is also adopted in this study.

IV. ADAPTIVE MODEL PROPOSED

A. Motivation

Wireless networks constitute a highly dynamic environment and movement estimation following users' preferences, requirements and constraints as encoded in the user profiles is a quite cumbersome task. Thus, it may be easily concluded that a predetermined traffic ratio definition with respect to the downlink and the uplink directions would lead to negative phenomena such as resource non optimal allocation, inefficient bandwidth distribution, subscriber service outages, data loss and transmission and reception delays. Let us for example consider the case of a fixed 1:1 downlink-to-uplink sub-frame width ratio. In such a case, the network supports a potential high-speed uplink flow, which could be quite useful at specific time periods during which high uplink traffic demands occur. However, this situation may lead to underutilization of the available network resources and poor quality of service provisioning, since subscribers that need bandwidth for downlink-oriented services (i.e., downloading, video and audio demand, online gaming, tele-conference) are inadequately served, even though bandwidth resources potentially exist. In the light of the aforementioned aspects, the decision on favoring each time a traffic direction depends on the burstiness and the duration of the traffic demands. Although the subscriber traffic profiles' prediction is an arduous process, the consideration and adjustment to short-term traffic dynamics constitutes an efficient and quite accurate way for providing flexibility and effectiveness on the decision making. The current work follows this direction; specifically, its main contribution is the provision of a rigorous, adaptive, and effective scheme for adjusting the downlink-to-uplink width ratio in the BS side, following the short-term network dynamics.

Regarding the use of LA, in the context of this study, the environment is represented by the OFDMA resource allocation process (i.e., the mapping process), while the pool of actions is expressed by the available downlink-to-uplink width values in accordance with the adopted coding and sub-channelization techniques. The feedback produced is fed to the LA module, supporting the learning process towards the optimal action, while it is associated with the error rate of the action taken in order to further adjust the LA's convergence speed.

Our work allows for dynamic tuning of the convergence speed of the LA mechanism, thus designing an error-aware LA structure. Specifically, the learning process periodically calculates the learning error rate; the convergence speed is accelerated if the error rate is high, while it is decelerated if continuing correct decisions are observed. In this manner, the learning process becomes more accurate, the system performance is improved, and the adaptive nature of the applied learning module is strengthened.

B. Adaptive Model Formulation

In this study, we exploit LA as the main adaptation mechanism in order to determine, in the context of each frame, the most appropriate downlink-to-uplink width ratio on the basis of the traffic observed in the past.

Let us assume that DW^f and UW^f denote the width values of the downlink and uplink sub-frames of frame f, respectively. Also, let $A = \{a_1, a_2, \dots, a_m\}$ denote the pool of m possible actions. Due to the fact that each action involves two distinct values (i.e., the downlink and the uplink sub-frame width values) the A set could be reformed as follows, $A = \{a_1 \to (DW_1^f, UW_1^f), a_2 \to (DW_2^f, UW_2^f), \dots, a_m \to (DW_m^f, UW_m^f)\}$, where the symbol $\to$ stands for 'corresponds to'. Obviously, the pool of actions includes all possible width values that the downlink (or the uplink) may receive in relation to the uplink (or the downlink) sub-frame.

In order for the reader to better comprehend the aforementioned aspect, hereafter we provide an illustrative example. Assuming a frame structure comprising 42 symbols (i.e., 42 allocation columns), it holds that $DW_i^f + UW_i^f = 42$, where $1 \leq i \leq m$. Furthermore, the A set is now $A = \{a_1 \to (21{,}21), a_2 \to (22{,}20), \dots, a_{13} \to (33{,}9)\}$. It is clear that the action pool reflects to a downlink-to-uplink width ratio from 1:1 (i.e., $a_1 \to (21{,}21)$) to 3:1 (i.e., $a_{13} \to (33{,}9)$).

For each frame f the LA decides an action from the ones available in the pool, determining the downlink-to-uplink width ratio for the forthcoming frame. The decision at each frame f is supported by a probability vector, which represents the probability distribution for selecting one of the possible actions from the A set. The probability vector at frame f is defined as follows: $p(f) = \{p_1(f), p_2(f), \dots, p_m(f)\}$. It holds that $\sum_{i=1}^{m} p_i(f) = 1$.

C. Feedback Definition

Each action taken by the LA is followed by a feedback generated by the mapper as a reaction to the mapping process performed. Figure 2 illustrates the assumed architecture. In essence, the feedback directs the LA towards optimal action selection (i.e., selection of the most appropriate downlink-to-uplink ratio) aiming to accomplish two main objectives: a) dynamic reallocation of the available space in accordance with the traffic requests, considering both uplink and downlink directions and b) load balancing. Dynamic adjustment of the allocation space results in bandwidth reassignment from the one sub-frame to the other. Specifically, the adaptive model proposed is able to sense whether the one sub-frame requires more bandwidth than the portion it was granted and also to check whether the other sub-frame could provide the extra space required. In other words, the model proposed is able to reallocate more space, in terms of symbols, to the direction that needs it more on a frame-by-frame basis. For instance, if the adaptive model senses that the downlink sub-frame produces large portion of idle slots, while the uplink

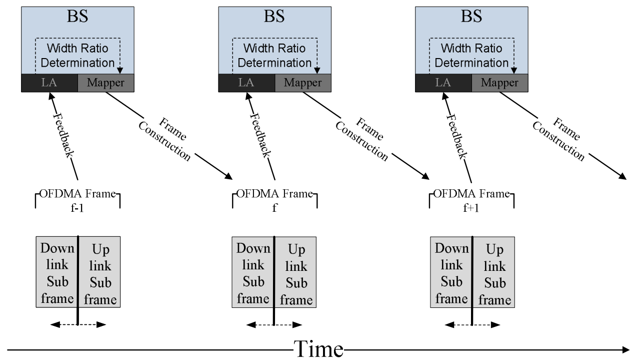

Figure 2. The proposed architecture.

bandwidth requests fail to be accommodated due to allocation space insufficiency, then it reallocates the available space by readjusting the downlink-to-uplink ratio, granting more symbols to the uplink direction in accordance, of course, with the standard restrictions (i.e., within the 3:1 to 1:1 range).

Load balancing is the second objective of the adaptive model proposed. In case both directions exhibit similar traffic demands, they are fairly treated by the proposed model. To be more specific, if the model senses traffic load similarities amongst the two directions, it modifies the downlink-to-uplink width ratio in order to provide load balancing. For example, under heavy traffic conditions regarding both directions, i.e., both sub-frames require more allocation space, the model proposed favors the sub-frame that needs larger portion of bandwidth, so that the amount of bandwidth granted to the sub-frames proportional to their traffic needs.

Thus, feedback definition is related to the output of the mapping processes. Both mappers endeavor to accommodate all (downlink / uplink) requests to the allocation space given. Upon the completion of the request allocation process, each mapper informs the adaptive model about the process results. The results are formed based on two critical performance metrics: a) the $unserved\_slots^f$, which refer to the cumulative number of requests that fail to get accommodated at each sub-frame of frame f and b) the $idle\_slots^f$, which denote the total number of wasted slots within the corresponding sub-frame of frame f. Let $feedback_d^f$ and $feedback_u^f$ denote the feedback generated considering the downlink and uplink direction, respectively. Based on the above, the feedback provided by the system at each frame f is given by the following equations:

$$feedback_d^f = \left\lceil \frac{unserved\_slots_d^f - idle\_slots_d^f}{H} \right\rceil,$$

$$feedback_u^f = \left\lceil \frac{unserved\_slots_u^f - idle\_slots_u^f}{H} \right\rceil$$

Where H symbolizes the allocation bin height. The allocation bin height defines the height of the available allocation OFDMA bin. In essence, the parameter H defines the number of available subcarriers in the frequency domain.

At frame $f-1$ the Automaton decides on an action from the set A of the possible actions, which corresponds to the definition of the width values for each sub-frame, for the following frame f. In the following, we have assumed that $current\_action^f$ corresponds to the selected width of the downlink/uplink sub-frames. For example, considering 42 total slots as presented in the example in Section IV.B, if $current\_action^f = 1$, which means that $current\_action^f$ corresponds to α_1, the downlink-to-uplink width ratio is 1:1 and each sub-frame shares 21 symbols. Then after frame's f transmission, the Automaton is able to compute the $next\_best\_action^f$ (i.e., the most suitable action on the basis of the feedback produced from the system). For instance, if the Automaton decides a $next\_best\_action^f$ greater than $current\_action^f$ by one, then $next\_best\_action^f = 2$, hence $next\_best\_action^f$ corresponds to α_2, fact that alters the ratio in such a way that 22 symbols are destined to the downlink sub-frame and 20 to the uplink one. For frame f the following *if-elseif* block describes the way of associating the $current\_action^f$ and feedback provided $next\_best\_action^f$.

IF $feedback_d^f > 0\ AND\ feedback_u^f < 0$ THEN
$$next_{best_{action}}{}^f = current\_action^f + \min(|feedback_d^f|, |feedback_u^f|)$$
ELSE IF $feedback_d^f < 0\ AND\ feedback_u^f > 0$ THEN
$$next\_best\_action^f = current\_action^f - \min(|feedback_d^f|, |feedback_u^f|)$$
ELSE IF $feedback_d^f > feedback_u^f$ THEN
$$next\_best\_action^f = current\_action^f + \left\lceil \frac{|feedback_d^f - feedback_u^f|}{2} \right\rceil$$
ELSE
$$next\_best\_action^f = current\_action^f - \left\lceil \frac{|feedback_d^f - feedback_u^f|}{2} \right\rceil$$
END IF

The block describes the conditions that should hold for assigning allocation space from the downlink to uplink sub-frame and vice versa. The first two conditions check whether the downlink (first *if*) and the uplink (second *else-if*) could offer extra allocation space to the uplink and the downlink sub-frame, respectively, in line with the dynamic reallocation objective. This is possible in case negative feedback is returned with respect to one sub-frame, meaning that it comprises at least one idle symbol. The two latter conditions balance the relation of both sub-frames in case there is not sufficient allocation space for satisfying all requests at both downlink and uplink directions following the load balancing objective. It is clear that no negative values are permitted for both $current\_action^f$ and $next\_best\_action^f$ parameters. Furthermore, both parameters are not allowed to exceed the maximum value of *A* set that could violate the ratio rules.

D. Probability Updating Algorithm

The adaptivity of the proposed model lies in the incorporated learning mechanism, which enables the adjustment of the action probability vector based on past experience. Upon the reception of the feedback provided by the system at each frame f, the Automaton calculates the $next\_best\_action^f$ and then updates its action probability distribution. The method adopted is based on the updating scheme of an S-model linear Reward Inaction (SL_{R-I}) LA [35]. The $next\_best\_action^f$ should be rewarded, increasing, thus, the probability of selecting the specific action in the future, whereas all other actions should be penalized by decreasing the respective probabilities. Initially, all probabilities are taken equal to $\frac{1}{m}$, where m denotes the total number of possible actions. For each frame, the adaptive model chooses the action associated with the largest probability. The procedure is depicted in *Algorithm1*. At this point it should be mentioned that the parameters L and a are associated with the convergence speed of the Automaton and they will be discussed in a detailed manner in the following subsection.

Algorithm1: Probability Updating
Set $p_i = \frac{1}{m}$
FOR each frame f
  Calculate the $next\_best\_action^f$
  FOR each action m
    IF $action \neq next\_best\_action^f$ THEN
      Set $p_i(f+1) = p_i(f) - L(p_i(f) - a)$
    ELSE
      Set $P_{next\_best\_action^{(f+1)}} = P_{next\_best\_action^{(f)}} + L(P_{next\_best\_action^{(f)}} - a)$
    END IF
  Choose the maximum probability and
  set it $p_{current\_action^{f+1}}$
  END FOR
END FOR

E. Convergence Speed

In *Algorithm1*, parameter a is employed in order to prevent reaching zero probabilities; in the context of this study it assumes a very small, fixed value (e.g., 10^{-4}). L governs the convergence speed of the learning process; it constitutes a critical parameter, especially in a highly dynamic and unpredictable environment such as a broadband wireless access network supporting mobile and nomadic users with unknown and unpredictable profiles, thus, it should be carefully handled. Setting L to an arbitrary fixed value, as many research works do, leads to constraining the capabilities of the automaton, as the convergence speed remains stable, not following the environment dynamics. A poor selection of the L parameter's value may lead to multiple continuous erroneous decisions, leading to performance degradation. In this work, L parameter is dynamically adjusted, allowing the Automaton to adapt to network dynamics according to the following effective rule [36]: increase the speed as long as the decisions are erroneous and decrease the speed as long as the decisions are successively correct. In this manner, erroneous decisions are quickly corrected by increasing the learning speed, minimizing, thus, their negative impact to the system, while correct decisions are rewarded, decreasing the learning speed when the best action is reached.

In order to adjust parameter L, in case of erroneous actions taken, a normalized error function $l(model\_error^f)$ is designed, taking into account the portion of the model error after the transmission of the current frame f:

$$l(model\_error^f) = \frac{1}{1-e}[1 - e^{(1-model\_error^f)}]$$

where $model\_error^f$ lies within [0,1] and receives the zero value when the decision reached by the automaton is correct. Specifically, $model\_error^f$ is given by the following equation:

$$model\_error^f = \frac{|current\_action^f - next\_best\_\_action^f|}{m-1}$$

At this point it should be mentioned that in case of correct decision taken for frame f, it stands that $current\_action^f = next\_best\_action^f$. It may be easily observed that the following equations hold:

$$l(\lim_{model\_error^f \to 0} model\_error^f) \to 1$$
$$l(\lim_{model\_error^f \to 1} model\_error^f) \to 0$$

In order to adjust parameter L in case of correct decisions, a correctness factor is defined, expressing the portion of correct decisions during a specific time window. Supposing that the number of the correct decisions considering the previous W frames is denoted by Z, the correctness factor is given by the following equation:

$$model\_correctness^f = \frac{Z}{W}$$

The $model\_correctness^f$ factor ranges from 0 to 1 and represents the model's accuracy during the previous W decisions made. In a similar manner to the definition of $l(model\_error^f)$, the $l(model\_correctness^f)$ is defined as follows:

$$l(model\_correctness^f) = \frac{1}{1-e}[1 - e^{(1-model\_correctness^f)}]$$

Similarly, it holds:

$$l(\lim_{model\_correctness^f \to 0} model\_correctness^f) \to 1$$
$$l(\lim_{model\_correctness^f \to 1} model\_correctness^f) \to 0$$

According to the designed model, a higher number of correct decisions taken during the previous W frames corresponds to a higher decrease in the value of parameter L, as it is indicated that the best action is reached with respect to the current traffic situation observed. Finally, L parameter is determined on a frame-by-frame basis according to *Algorithm2*:

Algorithm2: Determination of the L parameter
Set $L = initial\_value$
FOR each frame f
  IF $current\_action^f = next\_best\_action^f$ THEN
    Set $L = L - Ll(model\_correctness^f)$
  ELSE
    Set $L = L + Ll(model\_error^f)$
  END IF
END FOR

TABLE I

LEARNING AUTOMATON PARAMETERS

L $initial\_value$	0.015
a	0.0001
W	100

F. Complexity Analysis

In this section the complexity of the proposed adaptive model is examined. Low complexity is a primary design goal keeping the operation simple, flexible, and scalable. The operation of the proposed model includes the probability updating algorithm (*Algorithm1*) and the determination of the *L* parameter (*Algorithm2*). By investigating the computational requirements of the probability updating algorithm, we can easily infer that the complexity of the whole operation is *O(m)*, since the algorithm is responsible of updating the probability vector of *m* size. On the other hand, the complexity of the determination of the *L* parameter process is *O(1)*, because it includes an if-else block. Hence, the total complexity of the proposed model is quite low and grows linearly to the number of the possible Automaton actions (*m*). Given that a typical value of the parameter *m* is 13, as in subsection IV.B presented, the complexity of the proposed model is not a limitation for real-time applications.

V. EVALUATION OF THE MODEL PROPOSED

A. Motivation

The performance of the proposed adaptive model is evaluated in this section. The evaluation environment has been designed in Matlab. In particular, the simulator has been implemented in Matlab 7.11.0 (R2010b). The simulation experiments have been conducted in a 64-bit laptop computer with Core i7 Processor and 6Gb RAM. Comparative results have been obtained regarding the effectiveness of the model designed with that of schemes keeping static and predefined the downlink-to-uplink width sub-frame ratio. Furthermore, evaluation results have been obtained by comparing the proposed model with similar works such as the DRD [33] and the ERD [34] algorithm. Therefore, six algorithms have been implemented in the simulation experiments conducted: a) the adaptive, error-aware LA model proposed, which is capable of adjusting the ratio from 3:1 to 1:1 on a frame-by-frame basis, b) the static1:1 approach, which keeps the ratio stable and equal to 1:1, c) the static2:1 approach, which maintains a fixed ratio of 2:1, d) the static3:1 approach, which employs a static ratio of 3:1, e) the DRD algorithm, which applies the cubic spline extrapolation in order to determine the most appropriate downlink-to-uplink width ratio, and f) the ERD algorithm, which uses a static automaton to estimate the downlink-to-uplink width ratio. For each algorithm, both downlink and uplink directions have been considered, while all schemes adopt the AHBM algorithm [35], as the mapping algorithm concerning the downlink sub-frame and the standard uplink mapping scheme, as described in Section IV, regarding the uplink sub-frame.

The simulation environment has been designed in accordance with the following mobile IEEE 802.16 network parameters: The well-known Partially Used Sub-Channelization (PUSC) mode (the most common frequency diversity mode for practical mobile communications environments) is adopted, hence the frame defines 30 different channels. Furthermore, the time is subdivided into fixed frames, whereby each frame lasts for 10 ms. Under this assumption, three symbols are reserved for control in formation (one symbol for preamble, and two symbols for MAP and FCH fields) and are excluded from the available slots for satisfying allocation needs, while the available symbols for the formation of the downlink and the uplink sub-frames depend on the downlink-to-uplink ratio. Following the standard restrictions, the ratio may vary from 1:1 to 3:1, allowing 21 to 33 available symbols to be utilized for the downlink sub-frame and 21 to 9 symbols available to uplink sub-frame. Since the allocation bin is constructed by 30 channels, the downlink sub-frame defines a rectangular allocation space of 630 (30×21) slots, under 1:1 ratio, to 990 (30×33) slots, under 3:1 ratio, while the uplink sub-frame defines a rectangular allocation space of 630 (30×21) slots, under 1:1 ratio, to 270 (30×9) slots, under 3:1 ratio. Therefore, the static1:1 scheme offers 21 available symbols to each sub-frame, resulting in 630 slots allocation space each, the static2:1 gives 27 and 15 available symbols to downlink and uplink sub-frame respectively, and static3:1 allocates 33 and 9 available symbols to downlink and uplink sub-frame respectively.

TABLE II

SIMULATION ASSUMPTIONS

Channel Mode	PUSC
Frame Length	10 ms
Preamble Size	1 Symbol
MAP, FCH Sizes	2 Symbols
Downlink sub-frame Symbols	21 to 33 (1:1 to 3:1 ratio)
Uplink sub-frame Symbols	21 to 9 (1:1 to 3:1 ratio)
Downlink sub-frame capacity	630 to 990 slots (1:1 to 3:1 ratio)
Uplink sub-frame capacity	630 to 270 slots (1:1 to 3:1 ratio)
Request length	64-1518 Bytes
Buffer Length	1 MB
Shape Parameter a_{ON}	1.6
Shape Parameter a_{OFF}	1.4

The proposed error-aware LA operation involves the following tuning parameters, the $initial\_value$ of L parameter, which governs the convergence speed of the learning process, the a parameter, which prevents the probabilities of receiving the zero value, and the W parameter, which constitutes the history window of defining the $model\_correcteness^f$ factor. During the simulation experiments conducted the above parameters have been set as follows: $initial\_value = 0.015$, $a = 10^{-4}$, and $W = 100$. Furthermore, the pool of possible actions defines 13 possible selections ($m = 13$). LA parameters are summarized in Table I.

The traffic generated for both flows follows the self-similar traffic model, generated as an aggregation of multiple sources. Multiple Pareto-distributed ON/OFF periods are cycled producing the traffic requests of each source with shape parameter $a_{ON} = 1.6$ and $a_{OFF} = 1.4$. The scheduler forwards

TABLE III
WIRELESS CHANNEL PARAMETERS

Probability	Modulation and	Bits per slot
0.05	Outage	0
0.15	QPSK-1/2	48
0.2	QPSK-3/4	72
0.3	16QAM-1/2	96
0.3	16QAM-3/4	144

Ethernet frames to both mappers, whereas each frame length varies from 64 to 1518 Bytes. Two buffers, one for each direction, temporarily store the traffic requests from the scheduler to the mapper. The buffer capacity for both directions has been set equal to 1 MB. The main simulation assumptions are summarized in Table II.

The number of bits per slot a MS receives depends on the chosen modulation and coding scheme, which in turn depends on its radio channel condition. In essence, the number of bits per slot is given considering specific modulation and coding schemes obtained by the adaptive modulation scheme supported by the standard. For each frame the BS ranges the channel condition of each MS and determines the modulation and coding schemes. We consider different probabilities for MSs to receiving modulation and coding schemes. In this manner, for each frame, MSs' requests in bytes are associated to slots based on specific probabilities [37, 38]. For example, it is assumed that a MS receives QPSK-1/2 channel status with probability equal to 0.15. Additionally, on stimulating the realism of the simulation environment we consider that a MS may experience outage, having probability equal to 0.05, if its channel condition is too bad. The distribution of probabilities in each modulation and coding state changes across MS forming a practical wireless environment. In practice, for each data packet being transmitting either in the downlink or the uplink direction the wireless channel parameters are randomized for each MS. Table III summarizes the wireless channel parameters.

It is assumed that each MS requests one burst per frame [5]. This assumption is justified by the fact that each MS may request or send a batch of data packets sharing the same physical characteristics (i.e., modulation and coding), so all could be formed as a whole burst. It is considered that a burst

TABLE IV
SCENARIO I & IV TRAFFIC ASSUMPTIONS

Time	0 – 10 secs	10-20 secs	20-30 secs	30-40 secs	40-50 secs	50-60 secs
Mean Downlink Traffic Load (per MS)	1 Mbps	0.75 Mbps	0.5 Mbps	0.25 Mbps	0.25 Mbps	0.1 Mbps
Mean Uplink Traffic Load (per MS)	0.1 Mbps	0.5 Mbps	0.75 Mbps	0.75 Mbps	1 Mbps	1 Mbps

TABLE V
SCENARIO II & III TRAFFIC ASSUMPTIONS

Scenario II		Scenario III	
Mean Uplink Traffic Load (per MS)	Mean Downlink Traffic Load (per MS)	Mean Downlink Traffic Load (per MS)	Mean Uplink Traffic Load (per MS)
0.05 Mbps	0.05-0.5 Mbps	0.05 Mbps	0.05-0.5 Mbps
0.1 Mbps	0.05-0.5 Mbps	0.1 Mbps	0.05-0.5 Mbps
0.15 Mbps	0.05-0.5 Mbps	0.15 Mbps	0.05-0.5 Mbps
0.2 Mbps	0.05-0.5 Mbps	0.2 Mbps	0.05-0.5 Mbps
0.25 Mbps	0.05-0.5 Mbps	0.25 Mbps	0.05-0.5 Mbps
0.3 Mbps	0.05-0.5 Mbps	0.3 Mbps	0.05-0.5 Mbps
0.35 Mbps	0.05-0.5 Mbps	0.35 Mbps	0.05-0.5 Mbps
0.4 Mbps	0.05-0.5 Mbps	0.4 Mbps	0.05-0.5 Mbps
0.45 Mbps	0.05-0.5 Mbps	0.45 Mbps	0.05-0.5 Mbps
0.5 Mbps	0.05-0.5 Mbps	0.5 Mbps	0.05-0.5 Mbps

length may varies from 1 to $6 \cdot H \cdot DW_1^f$ and 1 to $6 \cdot H \cdot UW_1^f$ slots, which approximately correspond to 1600 Bytes under QPSK-1/2 status (minimum operational channel conditions), regarding the downlink and the uplink direction respectively.

The performance evaluation involves three metrics: a) the mean number of unserved MSs, which expresses the portion of MSs that fail to be accommodated in both downlink and uplink sub-frame due to lack of resources, b) the mean number of unserved slots, which denotes the total number of slots that fail to find allocation space in both sub-frames due to lack of resources, and c) the mean number of idle slots, which indicates the utilization of the available allocation bin.

B. Evaluation Scenarios

Four evaluation scenarios have been designed in order to extensively study the performance of the adaptive, error-aware model proposed. In the first scenario the performance of the proposed adaptive model is compared with the three static schemes under dynamic traffic conditions.

Specifically, it is assumed that the downlink and the uplink request traffic flows change every 10 secs, during which the traffic flows are continuously shifted. The number of connected MSs alters from 1 to 20. Each experiment has been conducted for 10 minutes of continuous network operation.

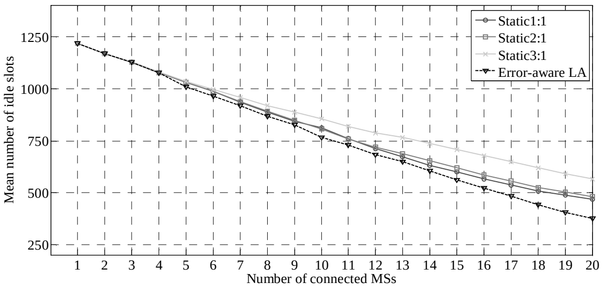

Figure 3. 1st Scenario: Mean number of idle slots as the number of connected MSs increases.

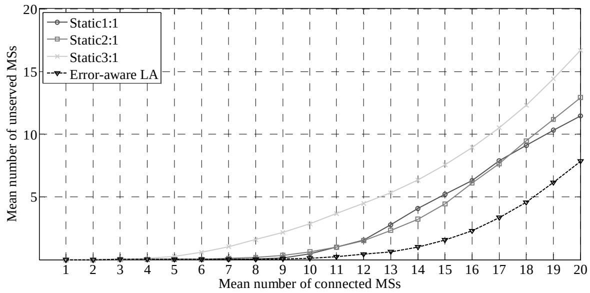

Figure 4. 1st Scenario: Mean number of unserved MSs as the number of connected MSs increases.

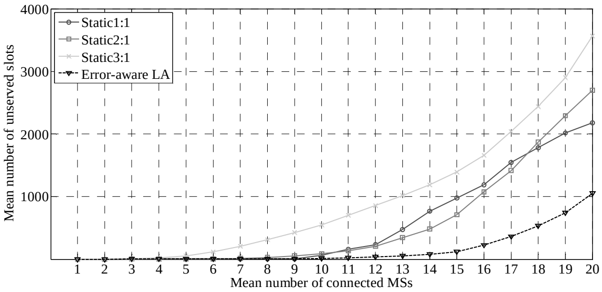

Figure 5. 1st Scenario: Mean number of unserved slots as the number of connected MSs increases.

This time corresponds to 60000 consecutive frames. The mean execution time of each frame was 0.00971697 sec approximately. The detailed assumptions for the first scenario are clarified in Table IV.

The second and the third scenario explore the efficiency of the model suggested as the traffic load changes. In both cases, the number of connected MSs is the same and equals to 10. In the second scenario each figure is composed of the results obtained from 10 different simulation runs, where the uplink traffic load per MS varies from 0.05 to 0.5 Mbps with a step of 0.05 Mbps. The uplink traffic load values are depicted in the *x*-axis. Concurrently, for each one of the total ten simulation runs, the downlink traffic load alters from 0.05 to 0.5 Mbps with a step of 0.05 Mbps for every 60 sec, meaning that each simulation run carried out for 10 minutes of continuous simulation time. Hence, for each one of the ten uplink values the downlink traffic load changes and the mean value of the performance metric is depicted. Similarly, in the third scenario each figure is defined based on the results obtained from 10 different simulation runs, where the downlink traffic load per MS varies from 0.05 to 0.5 Mbps with a step of 0.05 Mbps for every 60 sec, meaning that each simulation run carried out for 10 minutes of continuous simulation time. Here, the *x*-axis shows the downlink traffic load values. For each one of the total ten simulation runs, the uplink traffic load alters from 0.05 to 0.5 Mbps with a step of 0.05 Mbps for every 60 sec. Thus, for each one of the ten downlink values the uplink traffic load changes and the mean value of the performance metric is depicted. The selection of the traffic load variations aims at covering a wide range of possible traffic load conditions. For example, the mean traffic load per MS varies from 0.05 Mbps (low traffic load) to 0.5 Mbps (high traffic load). Traffic load (per MS) more than 0.5 Mbps have no beneficial impact in the evaluation, since it causes high portion of unserved MSs (per frame), and therefore the assessment of the various schemes becomes vague. On the other hand, applying 0.05 Mbps as a low traffic load profile gives the opportunity to investigate the performance of the various schemes in an environment with low traffic conditions, where the BS is able to manage all MSs without losses. Table V summarizes the traffic assumptions of the second and the third scenario.

Lastly, the performance of the adaptive schemes is investigated in the fourth scenario. The proposed error-aware LA model is compared with the DRD and the ERD scheme in order to infer about their efficiency. In particular, it is assumed that the downlink and the uplink request traffic flows change every 10 secs, during which the traffic flows are continuously shifted (Table IV). The number of connected MSs alters from 1 to 20 as in the first scenario. The fixed value of the L parameter adopted by the DRD algorithm is equal to 0.015, while the a parameter was set $a = 10^{-4}$. On the other hand, the ERD algorithm uses a history of $V = 100$ past values in order to apply the cubic Spline extrapolation technique.

C. Evaluation Results

The results of the first simulation scenario are illustrated in Figure 3, Figure 4, and Figure 5, which depict the mean number of idle slots, the mean number of unserved MSs, and the mean number of unserved slots respectively. It is clear that the adaptive scheme succeeds to a) reduce the portion of the wasted (frame) allocation space, in terms of slots, b) reduce the mean number of the MSs that fail to find accommodation space per frame, and c) increase the service ratio of the system by reducing the portion of bandwidth requests that return to the scheduler. To be more specific, as depicted in Figure 3, the adaptive scheme presents the lowest mean number of idle slots compared to all other static schemes, resulting in notable improvements in the channel utilization, as the number of the connected MSs to the system increases. In the same manner, as depicted in Figure 4, the adaptive scheme achieves to reduce the number of unserved MSs as the number of the connected MSs increases. In some case, e.g., when the number of the connected MSs is near five, the achieved improvement reaches approximately 50% compared to the static3:1. It is worth mentioning that this metric examined is quite important, since it expresses the pure system service ratio, i.e., the mean

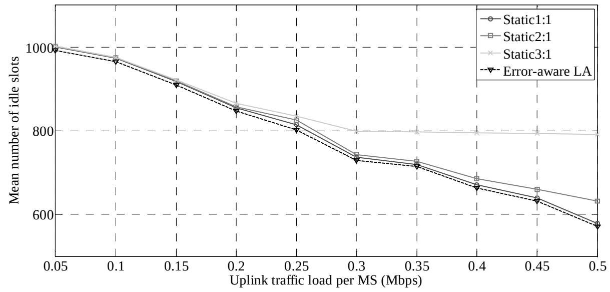

Figure 6. 2nd Scenario: Mean number of idle slots as the uplink load per MS increases. For each one of the ten uplink load values the downlink traffic load per MS varies from 0.05 to 0.5 Mbps with a step of 0.05 Mbps for every 60 sec and the mean number of idle slots is depicted.

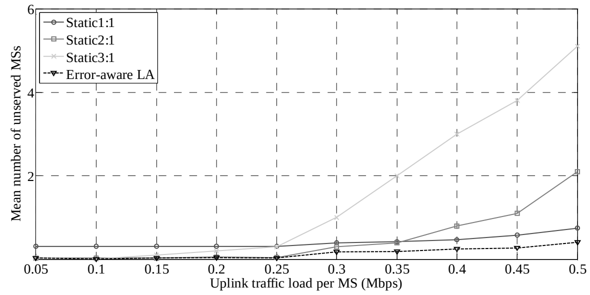

Figure 7. 2nd Scenario: Mean number of unserved MSs as the uplink load per MS increases. For each one of the ten uplink load values the downlink traffic load per MS varies from 0.05 to 0.5 Mbps with a step of 0.05 Mbps for every 60 sec and the mean number of idle slots is depicted.

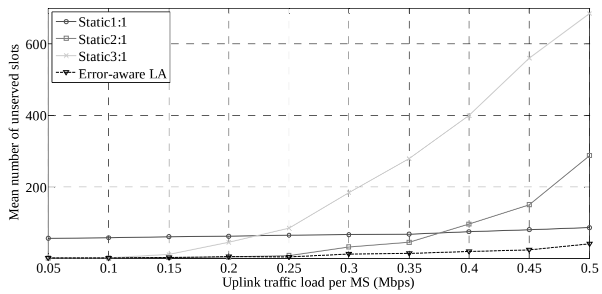

Figure 8. 2nd Scenario: Mean number of unserved slots as the uplink load per MS increases. For each one of the ten uplink load values the downlink traffic load per MS varies from 0.05 to 0.5 Mbps with a step of 0.05 Mbps for every 60 sec and the mean number of unserved slots is depicted.

number of MSs that successfully got serviced by the system. Subsequently, similar results are recorded in Figure 5, where the adaptive scheme allows more bandwidth requests in terms of slots to be successfully allocated per frame compared to the static schemes. As expected, the adaptive nature of the model proposed emerges as the key feature that effectively affects its behavior. As the static schemes maintain a predetermined and fixed downlink-to-uplink sub-frame width ratio, the adaptive model is able to adjust the ratio according to traffic dynamics. Hence, it manages to sense the relation of incoming and outgoing bandwidth demands and therefore to determine the most beneficial ratio in order to progressively improve the

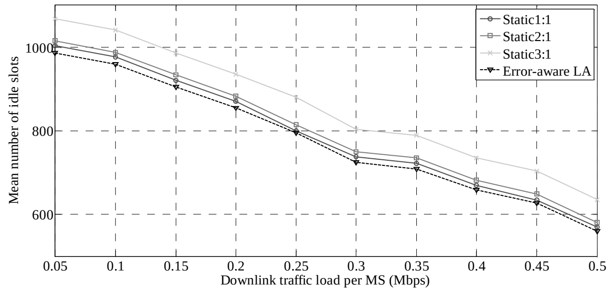

Figure 9. 3rd Scenario: Mean number of idle slots as the downlink load per MS increases. For each one of the ten downlink load values the uplink traffic load per MS varies from 0.05 to 0.5 Mbps with a step of 0.05 Mbps for every 60 sec and the mean number of idle slots is depicted.

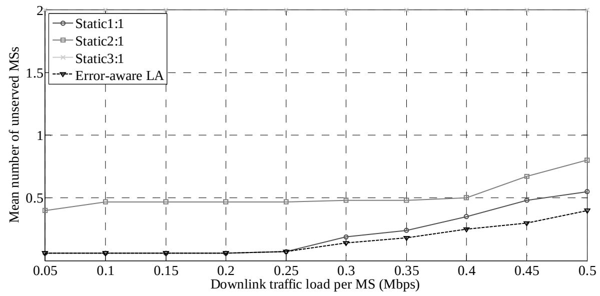

Figure 10. 3rd Scenario: Mean number of unserved MSs as the downlink load per MS increases. For each one of the ten downlink load values the uplink traffic load per MS varies from 0.05 to 0.5 Mbps with a step of 0.05 Mbps for every 60 sec and the mean number of unserved MSs is depicted.

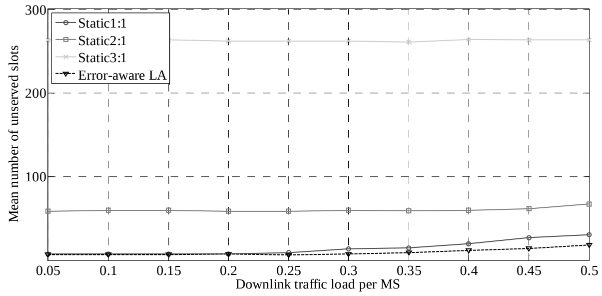

Figure 11. 3rd Scenario: Mean number of unserved slots as the downlink load per MS increases. For each one of the ten downlink load values the uplink traffic load per MS varies from 0.05 to 0.5 Mbps with a step of 0.05 Mbps for every 60 sec and the mean number of unserved slots is depicted.

system performance.

Figures 6, 7, and 8 verify the superiority of the adaptive scheme with respect to uplink traffic load alterations. Figure 6 shows the mean number of idle slots as the uplink load per MS increases, Figure 7 depicts the mean number of unserved MSs as the uplink load per MS increases, and Figure 8 illustrates the mean number of unserved slots as the uplink load per MS increases for second scenario. As the second scenario investigates the model efficiency considering stable uplink traffic conditions and variable downlink traffic conditions, it is

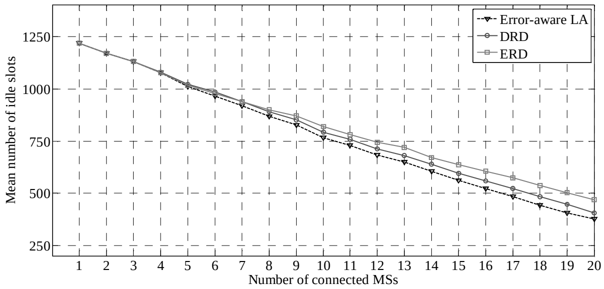

Figure 12. 4th Scenario: Mean number of idle slots as the number of connected MSs increases.

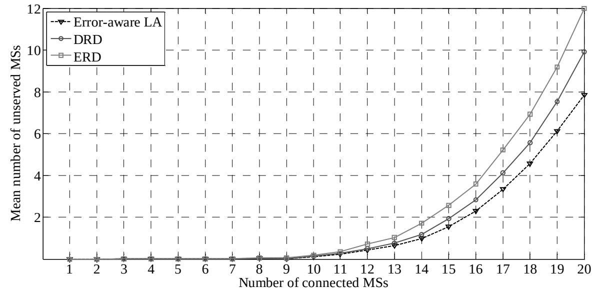

Figure 13. 4th Scenario: Mean number of unserved MSs as the number of connected MSs increases.

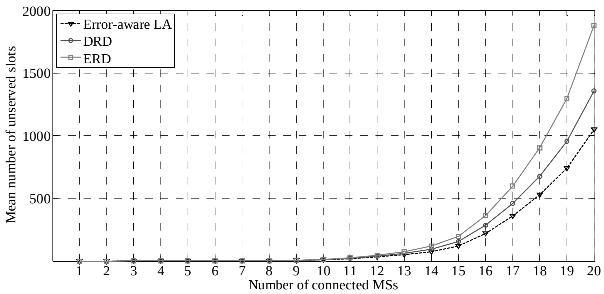

Figure 14. 4th Scenario: Mean number of unserved slots as the number of connected MSs increases.

clear that the proposed scheme contributes beneficially towards increasing network performance, since the target of increasing the network throughput by carrying more subscribers' requests is fulfilled. Indeed, this target is achieved without overshadowing the channel utilization that is expressed by the portion of idle slots, since the Automaton keeps the mean number of idle slots low. It is obvious that the performance of the adaptive scheme is always better than the static ones independently of the traffic relation between the downlink and uplink load. This fact verifies the ability of the model designed to quickly adapt to the new traffic conditions. The obtained improvement becomes evident considering at least eight connected MSs. This happens due to the fact that eight or more MSs begin to press for extra bandwidth (either for uplink or downlink flow) and the system should take sophisticated decisions to cover those needs. Keeping fixed downlink-to-uplink ratio, as the three static schemes apply, causes deficiencies regarding the system service ability. On the other hand, the model proposed effectively manages the bandwidth demands, by efficiently determine the ratio.

Furthermore, as the number of the connected MSs that fail to be accommodated increases as the uplink traffic load increases, the adaptive scheme succeeds to retain its superior performance, offering considerable improvements compared to the static1:1 (i.e., 0.22 to 0.43 unserved MSs for 0.4 Mbps uplink traffic load), to the static 2:1 (i.e., 0.22 to 0.8 unserved MSs for 0.4 Mbps uplink traffic load), and to the static3:1 (i.e, 0.22 to 3 unserved MSs for 0.4 Mbps uplink traffic load). It should be noted that the improvement comes without sacrificing idle slots, since as the Figure 5 indicates the model proposed succeeds to keep the portion of slot wastage low and slightly lowest compared to all other schemes.

The third scenario results set confirm the above findings. Figure 9 depicts the mean number of idle slots as the downlink traffic load increases, whereby the adaptive scheme presents a slight improvement regarding the channel utilization. The improvement in terms of channel utilization is marginal since the adaptive model tries to grant symbols and therefore slots to the direction that needs it more. In this manner and considering that in the downlink sub-frame the rectangular rule holds sacrificing slots in order to server more MSs, there is a trade-off between the utilization factor and the serving ability of the system. Nevertheless, the scheme proposed achieves better results for both performance metrics. Figure 10 illustrates the mean number of unserved MSs as the downlink traffic load increases, and again the static schemes seem to be unable to absorb the impact occurred from the dynamic traffic changes, while the adaptive scheme limits the performance losses. Finally, Figure 11 shows the mean number of unserved slots produced as the downlink traffic load gains ground. As expected, the findings are quite promising, since the bandwidth losses per frame are limited compared to the static schemes that present major insufficiencies (e.g., the static3:1).

Finally, the fourth scenario confirms the superiority of the proposed error-aware LA scheme in comparison with other algorithms that are capable of determining the width ratio in an adaptive way. Figure 12 shows the mean number of idle slots as the number of the involved MSs increases. Even though the difference between the three schemes is slight, the proposed error-aware model succeeds to improve the system utilization, by exploiting the potential resources in a more efficient way. Similarly, Figures 13 and 14 verify the above findings. The error-aware scheme achieves more accurate downlink-to-uplink width ratio estimations than the other adaptive schemes. This is obtained by exploring the mean number of unserved MSs in Figure 13, where the proposed scheme attains minimum losses. In terms of unserved slots, Figure 14 verifies the advantage of the error-aware LA. Due to its error-aware determination of the convergence speed, the proposed model is able to provide the mapper with accurate decisions. Moreover, it is capable of adapting to the network load diversities. On the contrary, the DRD algorithm maintains a stable speed of adopting, resulting in either premature estimations or delayed adopting. Accordingly, the ERD scheme is dramatically depends on its history vector that remains fixed and unaltered to the various traffic changes within the network. As a result, a portion of ERD predictions is erroneous and induce network performance degradation.

Overall, the proposed error-aware, adaptive scheme seems to be the most efficient one with respect to network resources utilization, while the incorporation of the LA beneficially affects the mapping process, offering quick and effective adoption to the dynamic traffic conditions. In addition, it achieves to effectively adjust the downlink-to-uplink ratio to satisfy extra needs for bandwidth, without overshadowing allocation space, i.e., channel resources. Finally, the scheme proposed succeeds in considerably reducing the portion of the requests returned back to the scheduler in terms of slots, increasing, thus, the network service rate.

VI. CONCLUSION

An adaptive model has been presented, aiming at providing an accurate determination of the downlink-to-uplink sub-frame width ratio on a frame-by-frame basis for mobile IEEE 802.16 wireless networks. The design of the model proposed is based on an error-aware LA, which is able to interact with the environment, to receive a feedback stemming from the mapping processes and to decide about the ratio value by taking into account the dynamic traffic conditions. The idea behind the proposed scheme lies in the efficient bandwidth reallocation between the downlink and the uplink sub-frame, whereby potential idle allocation space is granted to the sub-frame that needs it more. Furthermore, a bandwidth allocation balancing mechanism is employed. According to numerous evaluation experiments conducted, the behavior of the model suggested could beneficially affect the performance of the applied scheduling and mapping schemes, resulting in notable improvements in terms of system service ratio and channel utilization.

REFERENCES

[1] A.Ghosh, D.R.Wolter, J.G.Andrews, R.Chen, "Broadband wireless access with WiMax/802.16: current performance benchmarks and future potential," *IEEE Communications Magazine*, vol.43, no.2, pp.129,136, Feb. 2005.

[2] J.Andrews, A.Ghosh and R.Muhamed, Fundamentals of WiMAX, Understanding Broadband Wireless Networking, *Prentice Hall*, 2007.

[3] IEEE 802.16-2004, "IEEE Std 802.16-2004 (Revision of IEEE Std 802.16-2001) IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems," 2004.

[4] IEEE 802.16e-2005, "IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands and Corrigendum 1," 2005.

[5] C.So-In, R.Jain and A.-K.Tamimi, "Scheduling in IEEE 802.16e mobile WiMAX networks: key issues and a survey," *IEEE Journal on Selected Areas in Communications*, vol.27, no.2, pp.156-171, February 2009.

[6] P.Nicopolitidis, G.I.Papadimitriou, P.G.Sarigiannidis, M.S.Obaidat and A.S.Pomportsis, "Adaptive Wireless Networks Using Learning Automata", *IEEE Wireless Communications Magazine*, vol.18, no.2, April 2011, pp.75-81.

[7] B.J.Oommen, "Recent advances in Learning Automata systems", in *Proc. 2nd International Conference on Computer Engineering and Technology (ICCET 2010)*, pp.V1-724-V1-735, 16-18 April 2010.

[8] T.Joshi *et al.*, "SARA: Stochastic Automata Rate Adaptation for IEEE 802.11 Networks," *IEEE Trans. Parallel and Distrib.Sys.*, vol. 19, no. 10, Oct. 2008, pp.157–90.

[9] P.Nicopolitidis, G.I.Papadimitriou and A.S.Pomportsis, "Learning-Automata-Based Polling Protocols for Wireless LANs," *IEEE Trans. Commun.*, vol. 51, no.3, Mar. 2003, pp.453–63.

[10] P.Nicopolitidis, G.I.Papadimitriou and A.S.Pomportsis, "Distributed Protocols for Ad-Hoc Wireless LANs: A Learning-Automata-Based Approach," *Ad HocNetworks*, Elsevier, vol. 2, no. 4, Oct. 2004, pp.419–31

[11] J.A.Torkestani and M.R.Meybodi, "Clustering the Wireless Ad-Hoc Networks: A Distributed Learning Automata Approach," *Parallel and Distrib. Comp., Elsevier*,vol. 70, no. 4, Apr. 2010, pp.394–405.

[12] A.E.Eraghi , J.A.Torkestani, M.R.Meybodi and A.H.F.Navid, "Cellular Learning Automata-based Channel Assignment Algorithm for Wireless Mobile Ad Hoc Networks", *2009 International Conference on Machine Learning and Computing IPCSIT vol.3*, 2011.

[13] M.Esnaashari and M.R.Meybodi, "Data Aggregation in Sensor Networks using Learning Automata," *Wireless Networks*, Springer, vol. 16, no. 3, Apr. 2010, pp.687–99.

[14] M.Klantary, M.R.Meybobi, "Mobile Sensor Network Deployment Using Cellular Learning Automata Approach", in *Proc. International Congress on Ultra Modern Telecommunications and Control Systems and Workshops(ICUMT)*,2010.

[15] Y.Song, Y.Fang and Y. Zhang, "Stochastic channel selection in cognitive radio networks", in *Proc. IEEE Global Telecomm. Conf.(GLOBECOM)*, 2007, pp.4878–4882.

[16] P.Nicopolitidis, G.I.Papadimitriou and A. S. Pomportsis,"Using Learning Automata for Adaptive Push-Based Data Broadcasting in Asymmetric Wireless Environments", *IEEE Trans. Vehic. Tech.*, vol. 51, no. 6, Nov. 2002, pp.1652–60.

[17] P.Nicopolitidis, G.I.Papadimitriou and A.S.Pomportsis, "Adaptive Data Broadcasting in Underwater Wireless Networks", *IEEE Journal of Oceanic Engineering*, vol.35, no.3, pp.623-634, July 2010.

[18] G.I.Papadimitriou and D.G.Maritsas, "Learning automata-based receiver conflict avoidance algorithms for WDM broadcast-and-select star networks", *IEEE/ACM Trans Networking*, vol. 4, pp.407–412,June 1996.

[19] G.I.Papadimitriou and A.S.Pomportsis, "On the use of stochastic estimator learning automata for dynamic channel allocation in broadcast networks", in *Proc. IEEE/CEC*, San Diego, CA, July 2000

[20] S.Misra, B.J.Oommen, S.Yanamandra and M.S.Obaidat, "Random Early Detection for Congestion Avoidance in Wired Networks: A Discretized Pursuit Learning-Automata-Like Solution", *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol.40, no.1, pp.66-76, Feb. 2010.

[21] A.A.Economides, "Learning automata routing in connection-oriented networks", *Int. J. Commun. Syst.*, vol. 8, no. 4, pp.225–237, Jul.–Aug. 1995.

[22] K.S.Narendra and M.A.L.Thathachar, "Learning Automata: An Introduction". Englewood Cliffs, NJ: Prentice-Hall, 1989.

[23] N. Kaddour, and A. S. Poznyak. "Learning automata: theory and applications". Pergamon Press, Inc., 1994.

[24] M.A.L.Thathachar, and P. Shanti Sastry. "Varieties of learning automata: an overview." IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, vol. 32, no. 6, pp. 711-722, 2002.

[25] H.L.Liao, and Q.H.Wu. "Multi-objective optimization by learning automata." *Springer, Journal of Global Optimization* vol. 55, no. 2, pp. 459-487, 2013.

[26] C.Desset, E.B.de Lima Filho and G. Lenoir, "WiMAX Downlink OFDMA Burst Placement for Otimized Receiver Duty-Cycling", in *Proc. IEEE Int. Conf. Comm.*, pp.5149-5154, 2007.

[27] T.Ohseki, M.Morita and T.Inoue, "Burst Construction and Packet Mapping Scheme for OFDMA Downlinks in IEEE 802.16 Systems", in *Proc. IEEE Global telecom Conf.,* pp.4307-4311, 2007.

[28] T.Wang, H.Feng, and B.Hu, "Two-Dimensional Resource Allocation for OFDMA System", in P*roc. IEEE International Conference on Communications (ICC)*, pp.1-5, 19-23 May 2008.

[29] C.So-In, R.Jain and A.Al-Tamimi, "eOCSA: An Algorithm for Burst Mapping with Strict QoS requirements in IEEE 802.16e Mobile WiMAX Networks", *2nd Inter. Conf. on Comp. and Automation Engin. (ICCAE 2010)*, February 2010.

[30] P.G.Sarigiannidis, M.Louta and P.Chatzimisions, "A Novel Fair Mapping Scheme for IEEE 802.16 Downlink Sub-Frame", in *Proc. 2010 International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, pp.763-768, 18-20 Oct. 2010.

[31] P.G.Sarigiannidis, M.Louta, D.Stratogiannis and G.Tsiropoulos, "Towards a QoS-Aware IEEE 802.16 Downlink Sub-Frame Mapping Scheme", in *Proc. IEEE GLOBECOM Workshops (GC Wkshps 2010),* pp.1243-1247, 6-10 Dec. 2010.

[32] C.-H. Chiang, W. Liao, T. Liu, I. Chan, H.-L. Chao, "Adaptive downlink and uplink channel split ratio determination for TCP-based best effort traffic in TDD-based WiMAX networks," *IEEE Journal on Selected Areas in Communications*, vol.27, no.2, pp.182-190, February 2009.

[33] A.Sarigiannidis, P.Nicopolitidis, G.Papadimitriou, P.Sarigiannidis, and M.Louta, "Using Learning Automata for Adaptively Adjusting the Downlink-Uplink Ratio in IEEE 802.16e Wireless Networks", in *Proc. IEEE Symposium on Computers and Communications (ISCC'11)*, pp.353-358, June 28 2011-July 1 2011.

[34] P.Sarigiannidis, M.Louta, A.Michalas, "On Effectively Determining the Downlink-to-Uplink Sub-frame Width Ratio for Mobile WiMAX Networks Using Spline Extrapolation," In Proc. of IEEE 15th Panhellenic Conference on Informatics (PCI), pp.139,143, Sept. 30 2011-Oct. 2 2011.

[35] S.Misra, K.I.Abraham, M.S.Obaidat and P.Venkata Krishna, "Intrusion Detection in Wireless Sensor Networks: The S-Model Learning Automata Approach", in *Proc. IEEE International Conference on Wireless and Mobile Computing Networking and Communications*, (WIMOB '08), pp.603-607, 12-14 Oct. 2008.

[36] R.S.Sutton and A.G.Barto, "Reinforcement Learning: An Introduction," Cambridge University Press, 1998.

[37] S.Doirieux, B.Baynat, M.Maqbool, M.Coupechoux, "An efficient analytical model for the dimensioning of WiMAX networks supporting multi-profile best effort traffic", Elsevier *Computer Communications*, vol. 33, no. 10, pp.1162-1179, June 2010.

[38] B.Baynat, G.Nogueira, M.Maqbool and M. Coupechoux, "An Efficient Analytical Model for the Dimensioning of WiMAX Networks", in *Lecture Notes in Computer Science,* 2009, vol. 5550, pp.512-534, 2009.