Priority based Adaptive Access Barring for M2M Communications in LTE Networks using Learning Automata

Faezeh Morvari and Abdorasoul Ghasemi

Faculty of Computer Engineering, K.N. Toosi University of Technology, Tehran, Iran

Email: morvari@ee.kntu.ac.ir, arghasemi@kntu.ac.ir

Abstract

Supporting a huge number of Machine-to-Machine (M2M) devices with different priorities in LTE networks is addressed in this paper. We propose a Learning Automaton (LA) based scheme for dynamically allocating Random Access (RA) resources to different classes of M2M devices according to their priorities and their demands in each cycle. We then use another LA based scheme to adjust the barring factor for each class to control the possible overload. We show that by appropriate updating procedure for these LAs, the system performance asymptotically converges to the optimal performance in which the evolved Node B (eNB) knows the number of access-attempting devices from each class a priori. Simulation results are provided to show the performance of the proposed scheme in RA resource allocation to defined classes and adjusting the barring factor for each of them.

Index Terms

Machine-to-machine communications; Access barring; Learning automaton; Random access;

I. INTRODUCTION

Machine-to-Machine (M2M) or Machine Type Communication (MTC) refers to an emerging communication technology in which the key elements for constituting new communication paradigms such as smart city and Internet of Things (IoT) are addressed [1]. It involves a large number of autonomous devices that exchange information or data with each other or with the MTC server through a wireless area network without human intervention [2]. Smart grids, city automation, and infrastructure management are the typical examples of M2M applications which are widely adopted in our daily life [3]. The demand for M2M communications is continuously growing and it is expected that there will be 50 billion devices by 2020 [4].

Currently, cellular networks and in particular, the Third Generation Partnership Project (3GPP) Long Term Evolution (LTE) network are considered as a suitable infrastructure for deployment of MTC Devices (MTCDs) due to the advantages of providing the possibility of a ubiquitous and transparent communications for MTCDs [5]. However, cellular networks are mainly designed for human type communication which generally characterized by bursts of data during a limited number of active periods. Hence, the required signaling traffic for resource management is negligible. M2M communications, instead, involves a huge amount of MTCDs that need to transmit typically a small amount of data, most of the time [6]. That is the generated signaling traffic by a massive amount of MTCDs is significant and may cause a risk to the traditional operation of the cellular networks [7]. Therefore, deployment of the MTC in LTE infrastructure raises new challenges.

Specifically, when a massive number of MTCDs try to access the network within a short interval, the Radio Access Network (RAN) becomes congested which leads to decrease in the access success probability and heavy access delay for MTCDs. Therefore, handling the massive access requests of MTCDs is one of the main challenges for MTC in LTE [8]. So far, several methods have been proposed to alleviate congestion in the RAN. Among them, the Access Class Barring (ACB) scheme has attracted more attention due to its simplicity in deployment [9]. In the ACB scheme, the access of MTCDs are barred according to a barring factor which is broadcasted by the evolved node B (eNB).

On the other hand, since MTCDs belong to various applications with different priorities, the network should consider the priorities of devices in access granting for connecting to the network [10]. In this paper, we address the prioritized massive access of MTCDs in LTE networks and how to allocate Random Access CHannel (RACH) resources to them. The Random Access (RA) procedure is the first step for connecting to the cellular network which is done through RACH resources.

We propose a prioritized random access scheme using Learning Automaton (LA) in which the MTCDs are classified into different classes according to the priorities of the corresponding applications. Two LA modules are deployed. The first LA dynamically determines the amount of RACH resources which must be allocated to each class according to its priority. The second LA is used for determining the transmission probability for access-attempting MTCDs of each class to prevent from a huge amount of simultaneous RA attempts. It is shown that by proper adjustment of learning parameters, the asymptotic behavior of the proposed scheme tends to the optimal scheme in which the eNB has priori information about the number of access-attempting devices from each class.

The rest of this paper is organized as follows. The related works are presented in section II. In section III, the preliminaries and system model are explained. The proposed scheme is presented in section IV. Performance analysis and simulation results are provided in section V and VI, and the paper is concluded in section VII.

TABLE I

SUMMARY OF RACH OVERLOAD CONTROL TECHNIQUES.

Techniques	**References**	**Idea**
Separation of RACH resources	Split preambles [11, 12] Split PRACH occasions [9] Prioritized random access [13]	Split preambles between M2M and H2H users. Pre-allocates RACH resources to different MTC classes.
Slotted Access	Slotted access schemes [9, 14]	Dedicated slots for each MTCD.
Access Class Barring	Extended ACB [9] Dynamic ACB [15] Cooperative ACB [16]	Selectively control the access attempts of UEs which configured for EAB. The ACB factor is adjusted by a heuristic algorithm in each time slot, dynamically. Controls the RAN overload by dispersing MTCDs among neighboring cells that overlapped with each other.
MTC-specific backoff	Backoff tunning [17] Backoff timer method [12, 18]	PRACH overload is controlled by proper adjusting backoff times of MTCDs.
Other solutions	pull based scheme [9, 19]	Allows MTCDs to access the PRACH when paged by the eNB.
	Q-learning [20]	Uses Q-learning based RACH scheme slot assignment to MTCDs.
	Self optimizing overload control (SOOC) [21]	A self-optimizing mechanism for configuring the RACH resources based on load condition.

II. RELATED WORKS

In 3GPP LTE release 11, i.e., the LTE-A system, several approaches are proposed to counteract the RACH overload such as separate RACH resource allocation for M2M and non-M2M communications, slotted Access, ACB scheme, the MTC-specific backoff scheme, and pull based scheme [9]. In table I, a summary of the different RACH overload control techniques is presented. Particulary, in the ACB scheme, the eNB broadcasts the ACB or barring factor. Each device which has an access request selects a uniform distributed random number between 0 and 1, and compares it with the ACB factor. If this number is less than the ACB factor, the device can participate in contention by selecting a preamble, otherwise, it barred for a barring time. In this scheme, eNB controls the number of access-attempting MTCDs or the congestion level by adjusting the ACB factor. In addition to ACB, 3GPP also proposed the extended access barring (EAB) scheme. In EAB, eNB considers 16 access classes and in the case of RAN overload, only one or more of these classes which belong to the high priority applications are allowed to participate in the RA procedure and others become barred [9], [22].

The ACB factor should be adjusted according to the number and priorities of different MTCDs. In [15], a heuristic algorithm for adaptive adjustment of ACB factor using the number of successful and collided transmissions in the previous time slots is proposed. Also, the authors derive an analytical model for determining the total expected access delay for MTCDs. The proposed scheme in [23] uses available information in the eNB for accurate estimation of the number of M2M devices using Kalman filtering and adjusts the ACB factor based on this estimation. In order to reduce the RAN overload caused by MTCDs, the authors in [24] proposed a scheme which jointly utilizes from timing advance information and ACB. In this scheme by selecting the optimal value for ACB factor, the number of MTCDs which can be served in each time slot is maximized.

In [16] the authors proposed a cooperative ACB scheme for access load sharing among neighboring cells that overlapped with each other. The MTCDs which located in the coverage area of eNBs can select one of the eNBs to access such that the load is balanced among overlapping cells. This scheme improves the congestion delay for M2M communications. A Q-learning based scheme is proposed in [20] to avoid collision between M2M devices and enhance the throughput of the RACH resources. Using this scheme the performance loss of H2H devices that can be caused by massive access requests of M2M devices is reduced.

In these works, the RACH overload problem caused by massive access requests of MTCDs is discussed and less attention has been paid to the priorities and quality of service (QoS) requirements of them. Since different applications with different access priorities should be handled in MTC scenarios, the RACH overload control solutions should take into account the tolerable access delay of each MTC class. In order to satisfy the QoS requirements, the authors in [13] presented a prioritized random access mechanism that pre allocates RACH resources to different MTC classes according to their priorities. Furthermore, this mechanism prevents a large number of concurrent random accesses by dynamic access barring (DAB). However, the resources are not allocated to different priority classes in a dynamic manner which may lead to resource wasting. In this paper, in contrast, we propose an LA based scheme in which the available RACH resources are dynamically allocated to the priority classes of MTCDs according to their current demands where the ACB factor for each class is adjusted properly in the massive access case.

III. Preliminaries and System Model

A. Random Access Procedure in LTE Networks

In LTE networks, a User Equipment (UE) can be scheduled for uplink transmission if its uplink transmission timing is synchronized. The Random Access (RA) procedure is the first step for connecting to the LTE networks which is done through RACH resources. Therefore, the RA procedure plays a key role as an interface between non-synchronized UEs and the orthogonal transmission scheduling scheme through the LTE uplink radio resources [25]. That is the eNB can schedule UEs for uplink transmissions provided that they successfully passed the RA procedure. Notice that the RA procedure can be performed in a contention-free or contention-based manner [2]. In contention-free RA procedure, the eNB allocates a unique RA preamble to a specific UE and hence guarantee its access to the network. This access scheme is not typically used for massive access of M2M applications and deployed for time critical usages such as handover. However, the contention-based RA procedure which is also adopted in this paper, is much more appropriate for M2M traffic. That is, a certain number of assigned preamble sequences to each LTE cell is reserved for contention-free RA procedure and the remaining ones are used in the contention-based RA. The information about these preambles which are used by MTCDs is broadcasted by eNB through downlink control channel [8]. Then each access-attempting UE selects a preamble randomly and transmits its request to the eNB through the RA slot which is a time-frequency radio resource of the Physical RACH (PRACH). The contention-based RA procedure consists of four steps as follows [7], [25]:

Step 1: The MTCD transmits a randomly selected RA preamble through the next available RA slots of the PRACH. Due to the orthogonality of the available preambles, an eNB can decode multiple transmitted access requests by MTCDs which select different preambles in the same RA slot.

Step 2: For each successfully detected preamble, the eNB sends a random access response (RAR) through the Physical Downlink Shared Channel (PDSCH) which includes a random access preamble identifier (ID), an uplink (UL) grant that will be used for transmitting the third step of the RA procedure, a temporary cell identifier (C-RNTI), and a time alignment (TA) command.

Step 3: When the MTCD receives a RAR corresponding to the transmitted preamble in a specific RA slot, it sends the connection setup request message to the eNB using the assigned

UL grant in the received RAR.

Step 4: The eNB sends the contention resolution message to the MTCD provided that it can successfully decode the transmitted third message by the device in the specific UL grant. Otherwise, the eNB will not transmit any response and the device assumes that failed and schedules for a new RA procedure.

Collision occurs if one preamble is selected by two or more MTCDs in the same RA slot. In the situation of undetected preamble collision in step 1 of the RA procedure, more than one MTCD transmit connection setup request messages or data through the same UL grant and eNB cannot decode the received data successfully in step 3 and collision occurs [26]. We assume that for a collided preamble the eNB cannot decode any of the transmitted data in the third message of the RA procedure and all of the devices corresponding to such preambles must retry in subsequent cycles. Specifically, at the end of each cycle the eNB can divide the preambles into three groups including: 1) successful preambles: preambles which are selected by one device, 2) idle preambles: preambles which are not selected by any device, and 3) collided preambles: preambles which are selected by more than one device.

B. Learning Automata

Learning automata is a self-operating learning model which aims at operating in the environments with unknown characteristics. This learning model is useful in many applications involving adaptive decision making. An LA is an automaton that enhances its functionality by acquiring knowledge about the behavior of the random environment. It uses the acquired knowledge for adaptive decision making in the future. The response of the environment to the selected action by the LA feedbacks as a reward or penalty to the LA for updating the selection probability of the action as it is shown in Fig. 1 [27].

That is, the LA interacts with the random environment in repetitive cycles so as to find among the set of actions the one that maximizes the average reward the system receives by the environment. The environment is represented by a triple $E = \{a, b, p\}$ where $a = \{a_1, a_2, \ldots, a_r\}$ is the environment input set, $b = \{b_1, b_2, \ldots, b_r\}$ represents the environment output set and $p = \{p_1, p_2, \ldots, p_r\}$ represents the probability distribution for the r actions at t^{th} cycle where $\sum_{i=1}^{r} p_i(t) = 1$. The automaton is known as a P-model one, if the set of environmental responses take only the values 1 and 0, representing penalty and reward, respectively [28], [29].

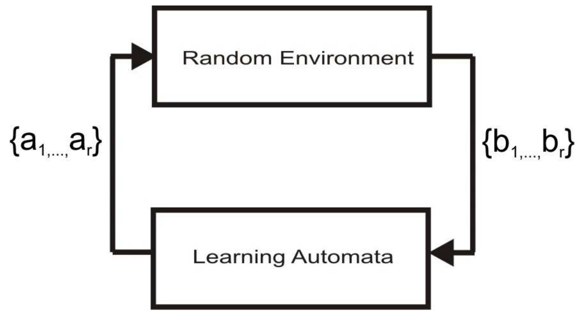

Fig. 1. An example of learning automata.

Assume that in cycle t the selected action and the corresponding normalized environmental response by the automaton are denoted by a_i and $c(t)$ respectively. The probabilities of actions are then updated in a reinforcement manner according to (1).

$$p_i(t+1) = \begin{cases} p_i(t) - (1-c(t))g_i(p(t)) + c(t)h_i(p(t)), & if \quad a(t) \neq a_i \\ \\ p_i(t) + (1-c(t))\sum_{j\neq i} g_j(p(t)) - c(t)\sum_{j\neq i} h_j(p(t)), & if \quad a(t) = a_i \end{cases} \tag{1}$$

where functions g_i and h_i are associated with reward and penalty for the selected action a_i.

C. System Model

We consider a system with N MTCDs corresponding to applications with different priorities in the coverage area of an eNB in a cell of LTE networks. The MTCDs are grouped into three priority classes including high, medium, and low according to their QoS requirements which are indicated by H, M and L in this paper, respectively. The corresponding numbers of MTCDs in each class are denoted by N_H, N_M, and N_L. We consider each MTCD will be activated at the interval $[0, T_s]$ with probability $g(t)$. In [9] two different probability distributions for $g(t)$ are proposed including uniform and beta distributions. In this paper, in order to consider the massive access scenario in which a large number of MTCDs try to access the network simultaneously, we assume that the activation of MTCDs in the interval $[0, T_s]$ follows beta distribution with parameters $\alpha = 3$ and $\beta = 4$, as follows:

$$g(t) = \frac{t^{\alpha-1}(T_s - t)^{\beta-1}}{T_s^{\alpha+\beta-1}\mathcal{B}(\alpha,\beta)} \tag{2}$$

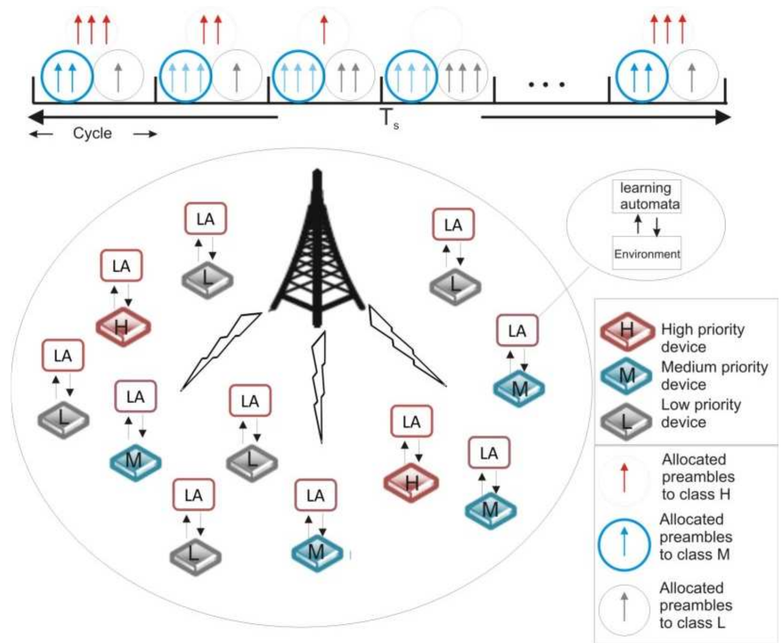

Fig. 2. M2M devices with different priorities in LTE networks

where $\mathcal{B}(\alpha, \beta)$ is the beta function [30].

Since most of the M2M applications have small sized data for transmission, we assume each activated device has only one small data packet for transmitting in a T_s interval. T_s is divided into Z_s cycles each of them consists of two parts. The first part is used for transmission of the preambles and the second part is used for transmission of the third messages of the RA procedure, see Fig. 2. In this paper, in order to avoid the signalling burden, we assume that the small data packets of MTCDs are transmitted to the eNB during the RA procedure. We also assume that eNB only knows the average number of access requests from each priority class in $[0, T_s]$ and it does not know the number and the start times of traffic bursts as well as the access request probabilities of MTCDs in each cycle. The total number of cycles which are required for serving all the MTCDs corresponding to each class in the activation interval is called Total Service Time (TST).

Let M be the number of available preambles for MTCDs in each cycle. To provide QoS for different priority classes one can divide the available RACH resources among them according to their average resource requirements. However, determining a fixed amount of resources for each priority class may cause a significant degradation in the network throughput when a class does not utilize the allocated resources in some cycles and another class has more data for transmission rather than the corresponding allocated RACH resources. We use an LA based scheme for dynamic assignment of the RACH resources to classes. As mentioned before, LA is a useful structure that can provide adaptation to systems operating in environments with changing and/or unknown characteristics [28]. On the other hand, the number of contending MTCDs in each cycle is unknown and depends on the stochastic arrival process of random access requests of the UEs. Furthermore, these access-attempting UEs have different priorities and demands for uplink resources [7]. We deploy LA to followup the number of contending MTCDs in each priority class and then adjust the ACB and RACH allocation probabilities for them. In the proposed scheme, the following prioritization rules for allocating RACH resources must be satisfied:

1. Each priority class can utilize a certain amount of available resources which is determined statically based on its priority and average requirement.

2. The unused resources of each priority class should be proportionally allocated to other priority classes which require more resources.

The initial probability of RACH resource allocation and the corresponding amount of allocated RACH resources, i.e., the number of allocated preambles, to priority class $x \in \{H, M, L\}$ in the t^{th} cycle are denoted by $q_x(t)$ and $M_x(t)$, respectively. Also, the maximum value of $q_x(t)$ is denoted by C_x. According to the priority and the average number of access-attempting devices of class x in a T_s interval, the value of C_x is determined statically by eNB and broadcasted at the beginning of the T_s interval. The MTCDs acquire this information through reading the broadcasted system information blocks (SIBs).

Although, a certain amount of the RACH resources are dedicated to each class, the number of access-attempting MTCDs can be much greater than the assigned resources in the massive access scenario. Hence, we use an LA based ACB scheme for each class to control the possible overload. The ACB parameter for priority class $x \in \{H, M, L\}$ in the t^{th} cycle is denoted by $p_x(t)$. The key mathematical symbols and their definitions are presented in table II.

IV. Learning automata based random access Scheme

For the proposed LA based scheme, two LAs are used in each MTCD. The first LA is responsible for adjusting the value of $q_x(t)$ and the second LA is used to adjust the barring

TABLE II

TABLE OF KEY MATEMATICAL SYMBOLS

Symbol	Definition
N	Total number of MTCDs in the coverage area of eNB
N_H, N_M, N_L	Number of MTCDs in high, medium and low priority classes
T_s	Activation interval of all MTCDs
M	Number of available preambles for MTCDs in each cycle
$q_x(t)$	Probability of RACH resource allocation to priority class x in t^{th} cycle
$M_x(t)$	Number of allocated preambles to priority class x in t^{th} cycle
C_x	Maximum value of $q_x(t)$
$p_x(t)$	ACB parameter for priority class x in t^{th} cycle
$N_x(t)$	Number of access-attempting devices from priority class x in t^{th} cycle
$p_x^{idle}(t)$	Probability that a preamble from class x remains idle in t^{th} cycle
$p_x^{succ}(t)$	Probability that a preamble from class x becomes successful in t^{th} cycle
$P_x^{coll}(t)$	Probability that a preamble from class x becomes collided in t^{th} cycle
$r(t)$	Feedback array in t^{th} cycle

factor, i.e., $p_x(t)$. Consider priority class x and let the number of access-attempting devices which belong to this class in t^{th} cycle is denoted by $N_x(t)$. Each MTCD from priority class x participates in the RA procedure with probability $p_x(t)$ and randomly selects a preamble from the available $M_x(t)$ preambles by probability $\frac{1}{M_x(t)}$. Hence, the probability that a certain preamble is selected by an MTCD from priority class x is given by $\frac{p_x(t)}{M_x(t)}$.

Therefore, the probability that this preamble remains idle, successfully exploited by one device, or encounters collision are given by (3), (4), and (5) respectively.

$$p_x^{idle}(t) = \left(1 - \frac{p_x(t)}{M_x(t)}\right)^{N_x(t)}. \tag{3}$$

$$p_x^{succ}(t) = \binom{N_x(t)}{1} \frac{p_x(t)}{M_x(t)} \left(1 - \frac{p_x(t)}{M_x(t)}\right)^{N_x(t)-1}. \tag{4}$$

$$p_x^{coll}(t) = 1 - \frac{N_x(t)p_x(t)}{M_x(t)} \left(1 - \frac{p_x(t)}{M_x(t)}\right)^{N_x(t)-1} - \left(1 - \frac{p_x(t)}{M_x(t)}\right)^{N_x(t)}. \tag{5}$$

The objective of the proposed scheme is adjusting the values of $q_x(t)$ and $p_x(t)$ such that the optimal performance is achieved.

If the number of access-attempting devices from priority class x is less than the maximum allowable RACH resources for this class, the optimal performance is achieved when the number of allocated preambles to this class is equal to $N_x(t)$, i.e., $M_x(t) = N_x(t)$. On the other hand, if the number of access-attempting devices from priority class x is greater than the maximum allowable RACH resources for this class, the optimal performance is achieved when the maximum allowable RACH resources are allocated to class x and the ACB factor is adjusted such that $p_x(t) = \frac{M_x(t)}{N_x(t)}$.

The eNB does not know the number of access-attempting devices of MTC classes in each cycle. The available information for the eNB are the number of successful, collided, and idle preambles at the end of each cycle which are denoted by $p_x^{succ}(t)$, $p_x^{coll}(t)$, and $p_x^{idle}(t)$, respectively. Notice that by optimal adjusting the values of $q_x(t)$ and $p_x(t)$ in a massive access scenario, the probability that eNB finds each preamble in successful, idle, and collision states would converge to e^{-1}, e^{-1} and $1-2e^{-1}$, respectively.

In the proposed scheme, we use p_x^{coll} as an indicator in order to determine the feedback for each class. This feedback which is received by all devices' LAs is denoted by the array $\mathbf{r}(t) = (r_H(t), r_M(t), r_L(t))$. $r_x(t)$ for each class takes a binary value as reward or penalty. At the end of each cycle, eNB monitors the value of p_x^{coll} for class x and generates $r_x(t)$ by comparing it with the optimal expected value of $v = 1-2e^{-1}$. That is:

$$r_x(t) = \begin{cases} 0 & if \quad p_x^{coll}(t) < v \\ 1 & if \quad p_x^{coll}(t) \geq v \end{cases} \tag{6}$$

The eNB broadcasts the generated feedback array $r(t)$ at the end of each cycle through the downlink broadcast channel.

A. Dynamic RACH Resource Allocation

Assume that each MTCD is empowered with a P-model LA. The LA must update $q_x(t)$ after receiving the feedback array. $r_x(t) = 1$ is occurred when the percentage of collision in class x is greater than the optimal value. It means that the allocated RACH resources to this class is less than the optimal value, therefore, $q_x(t)$ should be increased. On the other hand, $r_x(t) = 0$ indicates that $q_x(t)$ should be decreased. Note that, in order to simplify the analysis of the proposed scheme, we assume $p_x^{coll} = 1 - 2e^{-1}$. The general updating procedure of $q_x(t)$ which is used by LAs in the proposed scheme is given by:

$$q_x(t+1) = \begin{cases} q_x(t) + \Delta_1 & if \quad r_x(t) = 1 \\ \\ q_x(t) - \Delta_2 & if \quad r_x(t) = 0 \end{cases} \tag{7}$$

Where $0 < \Delta_1 < C_x - q_x(t)$ and $0 < \Delta_2 < q_x(t) - a_1$. a_1 takes a small value and is used to ensure that the percentage of allocated resources to each class be greater than zero when that class has no access request. In the proposed scheme, the LA starts with the maximum probability of allocating RACH resources to each class, i.e., $q_x(t) = C_x$. After updating the values of $q_x(t)$ at the end of t^{th} cycle , the values of $q_x(t)$ are normalized by each LA according to (8).

$$\sigma_x(t) = \frac{q_x(t)}{q_H(t) + q_M(t) + q_L(t)}, \qquad for \;\; x \in \{H, M, L\}. \tag{8}$$

It is clear that $\sum_{x\in\{H,M,L\}} \sigma_x(t) = 1$. The normalization of the probabilities is used in the LA based schemes [31]-[33].

The number of preambles that priority class x can use in the t^{th} cycle is determined according to the normalized probability $\sigma_x(t)$ as given in (9).

$$M_x(t) = M \times \sigma_x(t) \tag{9}$$

Therefore, the range of preambles which can be used by MTCDs in each priority class is determined.

To ensure the convergence of $q_x(t)$ to the optimal value, the values of Δ_1 and Δ_2 in (7) should be selected appropriately. According to (5) and (9) $p_x^{coll}(t)$ is a function of $q_x(t)$ and the optimal value of $p_x^{coll}(t)$ will be achieved by proper increasing or decreasing of $q_x(t)$. We have

$$\begin{aligned} \delta q_x(t) &= E[q_x(t+1) - q_x(t)] = p_x^{coll}(t)\Delta_1 - (1 - p_x^{coll}(t))\Delta_2 \\ &= p_x^{coll}(t)\Delta_1 + p_x^{coll}(t)\Delta_2 - \Delta_2 = p_x^{coll}(t)(\Delta_2 + \Delta_1) - \Delta_2 \\ &= (\Delta_1 + \Delta_2)(p_x^{coll}(t) - \frac{\Delta_2}{\Delta_1 + \Delta_2}). \end{aligned} \tag{10}$$

In fact, $q_x(t)$ increases with probability $p_x^{coll}(t)$ and decreases with probability $1 - p_x^{coll}(t)$. As we mentioned before, when the number of access-attempting devices is less than the maximum amount of RACH resources which can be allocated to priority class x, if $q_x(t)$ is adjusted by optimal value, we should have $p_x^{coll}(t) = v$. To asymptotically converge to the optimal case, the allocation procedure should be updated according to the following conditions.

1. If $p_x^{coll}(t) < v$ then $\delta q_x(t) < 0$ and therefore $\delta p_x^{coll}(t) > 0$.
2. If $p_x^{coll}(t) > v$ then $\delta q_x(t) > 0$ and therefore $\delta p_x^{coll}(t) < 0$.
3. If $p_x^{coll}(t) = v$ then $\delta q_x(t) = 0$ and therefore $\delta p_x^{coll}(t) = 0$.

According to (10), these conditions are satisfied and the optimal case is achieved provided that $\delta q_x = 0$ and $\frac{\Delta_2}{\Delta_2+\Delta_1} = v$. Therefore,

$$\begin{aligned} \Delta_1 &= \frac{1-v}{v}\Delta_2 = d_1\Delta_2, \\ d_1 &= \frac{1-v}{v} = 2.77. \end{aligned} \tag{11}$$

By considering $\Delta_2 = \Delta$ where $0 < \Delta < \frac{C_x - q_x(t)}{d_1}$ and $0 < \Delta < q_x(t) - a_1$, we adjust Δ by:

$$\Delta = L_1\big(C_x - q_x(t)\big)\big(q_x(t) - a_1\big), \qquad \textit{where} \qquad L_1 \in (0,1). \tag{12}$$

In sum, the RACH allocation updating procedure is given by (13).

$$q_x(t+1) = \begin{cases} q_x(t) + d_1 L_1\big(C_x - q_x(t)\big)\big(q_x(t) - a_1\big) & if \quad r_x(t) = 1 \\ \\ q_x(t) - L_1\big(C_x - q_x(t)\big)\big(q_x(t) - a_1\big) & if \quad r_x(t) = 0 \end{cases} \tag{13}$$

Where $L_1 \in (0,1)$ is the step size of probability updating procedure. The convergence speed as well as the estimation accuracy of the automaton depend on the value of L_1. By the updating procedure in (13), $q_x(t)$ is changed according to the requirements of each class and takes a value in the interval (a_1, C_x). Also, note that the two mentioned priority rules which is discussed in the system model section are satisfied.

As a special case consider a scenario in which each class experiences massive access by a lot of access-attempting devices. In this case, $q_x(t)$ will converge to C_x and we have

$$\sigma_x(t) = \frac{C_x}{C_L + C_M + C_H}, \qquad \textit{for} \quad x \in \{H, M, L\}. \tag{14}$$

That is all classes use from the maximum preassigned RACH resources.

Now, consider a scenario in which class H has no traffic for transmission, however, the other two classes are in massive access mode. In this case, $q_x(t)$ for class H, M, and L converge to a_1, C_M, and C_L respectively and we have:

$$\sigma_H(t) = \frac{a_1}{C_L + C_M + a_1}, \tag{15}$$

and

$$\sigma_x(t) = \frac{C_x}{C_L + C_M + a_1}, \qquad for \quad x \in \{M, L\}. \tag{16}$$

Therefore, the unused RACH resources by class H is proportionally allocated to the other two classes as expected.

B. Dynamic Adjusting of ACB Factors

When the number of access-attempting devices in a class is greater than the maximum amount of allocatable RACH resources, the ACB probability should be adjusted properly in order to reduce the chance of collisions. We use another P-model based LA in each MTCD to adjust the corresponding ACB factor. For this purpose, similar to the updating procedure for $q_x(t)$, at cycle t each MTCD updates $p_x(t)$ using the broadcasted $r_x(t)$. Notice that if $p_x(t)$ is adjusted appropriately, $p_x^{coll}(t)$ would converge to $v = 1 - 2e^{-1}$. The updating procedure is defined as follows.

$$p_x(t+1) = \begin{cases} p_x(t) + \Delta_1 & if \quad r_x(t) = 0 \\ \\ p_x(t) - \Delta_2 & if \quad r_x(t) = 1 \end{cases} \tag{17}$$

where $0 < \Delta_1 < 1 - p_x(t)$ and $0 < \Delta_2 < p_x(t) - a_2$. Again, a_2 is an appropriate small value and Δ_1 and Δ_2 should be selected such that the updating procedure for the ACB factor converges to the optimal value asymptotically. According to (17) we have:

$$\begin{aligned} \delta p_x(t) &= E\big[p_x(t+1) - p_x(t)\big] = \big(1 - p_x^{coll}(t)\big)\Delta_1 - p_x^{coll}(t)\Delta_2 \\ &= \Delta_1 - p_x^{coll}(t)\Delta_1 - p_x^{coll}(t)\Delta_2 = \Delta_1 - p_x^{coll}(t)(\Delta_1 + \Delta_2) = \\ &= (\Delta_1 + \Delta_2)\Big(- p_x^{coll}(t) + \frac{\Delta_1}{\Delta_1 + \Delta_2}\Big). \end{aligned} \tag{18}$$

To ensure that $p_x^{coll}(t) = v$, the ACB factor updating procedure must satisfy the following conditions.

1. If $p_x^{coll}(t) < v$ then $\delta p_x(t) > 0$ and therefore $\delta p_x^{coll}(t) > 0$.
2. If $p_x^{coll}(t) > v$ then $\delta p_x(t) < 0$ and therefore $\delta p_x^{coll}(t) < 0$.
3. If $p_x^{coll}(t) = v$ then $\delta p_x(t) = 0$ and therefore $\delta p_x^{coll}(t) = 0$.

According to (18), these conditions are satisfied and the optimal case is achieved when $\delta p_x(t) = 0$ and $\frac{\Delta_1}{\Delta_1+\Delta_2} = v$. Therefore, we have:

$$\Delta_1 = \frac{v}{1-v}\Delta_2 = d_2\Delta_2,$$
$$d_2 = \frac{v}{1-v} = 0.359. \tag{19}$$

By considering $\Delta_2 = \Delta$ where $0 < \Delta < \frac{1-p_x(t)}{d_2}$ and $0 < \Delta < p_x(t) - a_2$, we adjust Δ as follows

$$\Delta = L_2\big(1 - p_x(t)\big)\big(p_x(t) - a_2\big), \qquad \textit{where} \qquad L_2 \in (0,1). \tag{20}$$

In sum, the ACB updating procedure is given by (21).

$$p_x(t+1) = \begin{cases} p_x(t) + d_2L_2\big(1 - p_x(t)\big)\big(p_x(t) - a_2\big) & if \quad r_x(t) = 0 \\ \\ p_x(t) - L_2\big(1 - p_x(t)\big)\big(p_x(t) - a_2\big) & if \quad r_x(t) = 1 \end{cases} \tag{21}$$

C. State Diagram of the LA Based Scheme

The state diagram of the proposed scheme is illustrated in Fig. 3. Consider an MTCD from class x. According to Fig. 3, at the first step the values of $q_x(t)$ and $p_x(t)$ for this device are initialized by the corresponding maximum values, i.e., C_x and 1, respectively. Then, if the received feedback is 0, $p_x(t)$ remains constant and $q_x(t)$ is decreased according to (13). Therefore, the percentage of allocated RACH resources to priority class x is decreased and may be used for other priority classes which require more resources. Otherwise, if the received feedback is 1, the value of $q_x(t)$ is increased until it reaches to its maximum value and then the value of $p_x(t)$ is decreased to bare the massive access of this class taking into account the maximum allocatable RACH resources.

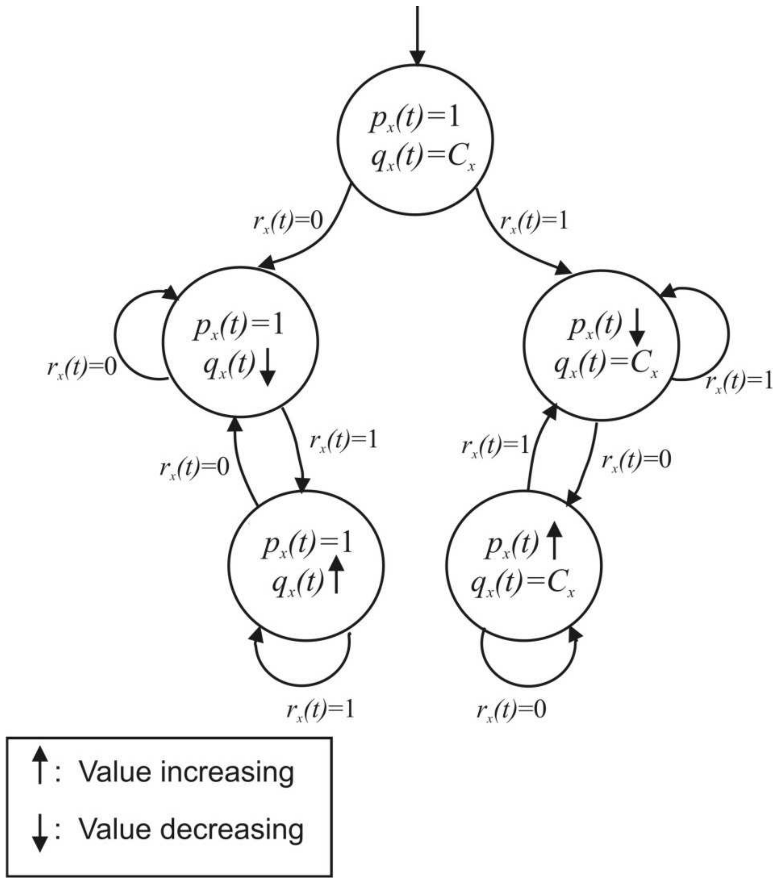

Fig. 3. The state transition diagram of the LA based scheme.

In this state, if the received feedback changes to 0, at the first the value of $p_x(t)$ is increased until it reaches to its maximum value and then the value of $q_x(t)$ is decreased. Note that, the value of $p_x(t)$ can be decreased only when $q_x(t)$ is adjusted by its maximum value, i.e., C_x. Also, the value of $q_x(t)$ can be decreased only when $p_x(t)$ is adjusted by its maximum value, i.e., 1.

Put together, the probability updating procedures for $q_x(t)$ and $p_x(t)$ are given by (22) and (23).

$$q_x(t+1) = \begin{cases} q_x(t) + d_1 L_1 \big(C_x - q_x(t)\big)\big(q_x(t) - a_1\big) & if \quad r_x(t) = 1 \\ \\ q_x(t) - L_1 \big(C_x - q_x(t)\big)\big(q_x(t) - a_1\big) & if \quad r_x(t) = 0 \;\; and \;\; p_x(t) = 1 \end{cases} \tag{22}$$

and

$$p_x(t+1) = \begin{cases} p_x(t) + d_2 L_2 \big(1 - p_x(t)\big)\big(p_x(t) - a_2\big) & if \quad r_x(t) = 0 \\ \\ p_x(t) - L_2 \big(1 - p_x(t)\big)\big(p_x(t) - a_2\big) & if \quad r_x(t) = 1 \ \ and \ \ q_x(t) = C_x \end{cases} \tag{23}$$

where $L_1, L_2 \in (0,1)$.

V. Performance Analysis

According to the number of access-attempting devices for each class two different situations can occur in the t^{th} cycle as follows:

1. If the number of access-attempting devices from class x is less than the maximum amount of RACH resources which can be assigned to this class, the optimal performance is achieved when the number of allocated preambles to class x is equal to the $N_x(t)$, i.e., $M_x(t) = N_x(t)$. Therefore, the optimal value of $\sigma_x(t)$ would be $\frac{N_x(t)}{M}$ as follows.

$$\sigma_X(t) = \frac{q_x(t)}{\sum_{i=1}^{k} q_i(t)} = \frac{N_x(t)}{M}. \tag{24}$$

Hence, the optimal value of $q_x(t)$ is given by (25).

$$q_x(t) = \frac{N_x(t)}{M}\Big(\sum_{i=1, i\neq x}^{k} q_i(t) + q_x(t)\Big). \tag{25}$$

According to (25) and since the maximum value of $q_x(t)$ is bounded by C_x we conclude that the optimal value for $q_x(t)$ is:

$$min\Big\{\frac{\frac{N_x(t)}{M}\sum_{i=1, i\neq x}^{k} q_i(t)}{1 - \frac{N_x(t)}{M}}, C_x\Big\}. \tag{26}$$

2. In the case that the number of access-attempting devices from class x is greater than the maximum allowable RACH resources for this class, the optimal performance is achieved when the maximum allowable RACH resources are allocated to class x, i.e., $q_x(t) = C_x$ and the number of participating MTCDs is limited by optimal value of $p_x(t)$ as is given by (27).

$$p_x(t) = \frac{M_x(t)}{N_x(t)}. \tag{27}$$

According to (27) and since the maximum value of $p_x(t)$ is 1, we conclude that the optimal value for $p_x(t)$ is:

$$min\Big\{\frac{M_x(t)}{N_x(t)}, 1\Big\}. \tag{28}$$

The asymptotic behaviors of $q_x(t)$ and $p_x(t)$ are given by Lemma 1 and Lemma 2.

Lemma 1: If the number of access-attempting devices in priority class x in the t^{th} cycle is $N_x(t)$ which is less than the maximum amount of RACH resources which can be assigned to this class and the probability updating procedure (13) is used, we have:

$$\lim_{L_1\to 0, a_1\to 0, t\to\infty} q_x(t) = min\Big\{\frac{\frac{N_x(t)}{M}\sum_{i=1, i\neq x}^{k} q_i(t)}{1-\frac{N_x(t)}{M}}, C_x\Big\}. \tag{29}$$

Proof: see appendix A.

Lemma 2: If the number of access-attempting devices in priority class x in t^{th} cycle is $N_x(t)$ and this class uses from the maximum allocatable resources, using the updating procedure of (21) we have:

$$\lim_{L_2\to 0, a_2\to 0, t\to\infty} p_x(t) = min\Big\{\frac{M_x(t)}{N_x(t)}, 1\Big\} \tag{30}$$

Proof: see appendix A.

VI. Simulation Results

In this section, we evaluate the performance of the proposed LA based scheme and compare it against the optimal and fixed allocation schemes. In optimal allocation, we assume that eNB knows the number of access-attempting devices from each class in each cycle and allocates preambles to them taking into account the maximum allocatable RACH resources to each class. Hence, the RACH allocation and ACB probabilities are assigned in the optimal manner. In the fixed allocation, a fixed number of preambles are pre-allocated to each class statically by the eNB according to the priority and the average number of access-attempting devices in that class in a T_s interval as given by (31).

$$M_x = \frac{N_x C_x M}{N_H C_H + N_M C_M + N_L C_L}, \qquad for \;\; x \in \{H, M, L\}. \tag{31}$$

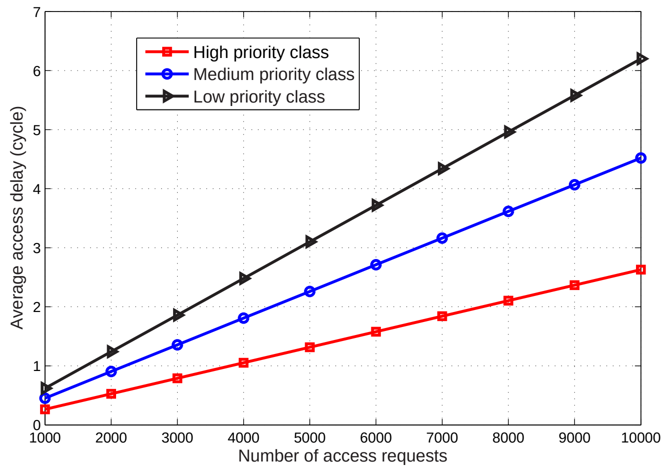

Fig. 4. The average access delay vs. the number of MTCDs for three priority classes with Z_s =200.

We assume that one RA slot occurs in each cycle and 50 preambles are reserved in each RA slot for using by three priority classes. The values of C_H, C_M, and C_L are set to 0.5, 0.3, and 0.2, respectively.

In Fig. 4, the average access delay of a typical MTCD in each class for different number of MTCDs in three classes for the proposed LA based scheme is shown. The number of cycles in the activation interval is $Z_s = 200$. As expected, in massive access scenario the average access delay for each priority class depends on the percentage of resources which considered for that class. That is each class exploits from the maximum allocatable RACH resources and the MTCDs which belong to the high priority class incur less average access delay as the number of access requests from each class is increased.

We then evaluate the number of allocated preambles to different classes in consecutive cycles of the TST interval in the proposed scheme as shown in Fig. 5. In this simulation, at first the number of MTCDs in three classes is equal to 1000 and $Z_s = 20$. It is clear that when each class has data for transmission, the number of allocated preambles to that class is proportional to the maximum percentage of resources which can be allocated to it. However, when all of the access requests from priority class x are served, the corresponding RACH resources for this class should allocate to other classes proportionably. For example, at cycle 535, when the priority class H has no more access request, the RACH resources which can be used by it are

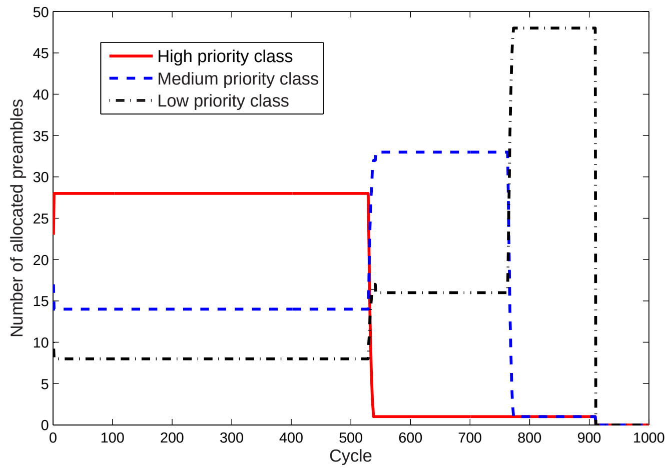

Fig. 5. The number of allocated preambles to three priority classes when $N_H = N_M = N_L = 1000$ and Z_s=20.

allocated to priority class M and L proportionally. Also, as expected and it can be seen in Fig. 5, in this case a small percentage of RACH resources is still allocated to priority class H which corresponds to parameter a_1 in the LA scheme.

In order to compare the proposed scheme with the optimal and fixed allocation schemes, we consider a $\mathcal{B}(3, 4)$ traffic model for each class.

We consider three traffic bursts for class H in the T_s interval. The first burst is started at the cycle 0^{th} and last for 20 cycles with 500 requesting devices. The second and third bursts of this class are started at 200^{th} and 400^{th} cycles, with 20 cycles duration and 250 requesting devices in each burst, respectively. Also, we consider two traffic bursts for medium priority starting at 0^{th} and 500^{th} cycles with 100 cycles duration respectively. The number of requesting devices in two bursts is equal to 2500. Finally, a traffic burst is generated by 10000 low priority devices at 0^{th} cycle with 100 cycles duration in T_s interval. The number of allocated preambles in consecutive cycles of the TST interval for the proposed LA based scheme, optimal allocation, and fixed allocation schemes for the priority class H, M, and L are depicted in Fig. 6 (a), (b) and (c), respectively. In this simulation $Z_s = 600$.

As it is expected, when there is no request from class H, the minimum allowable number of preambles are allocated to this class and the remaining preambles are allocated to other classes proportionally. However, as the second burst of this class starts, the allocated preambles

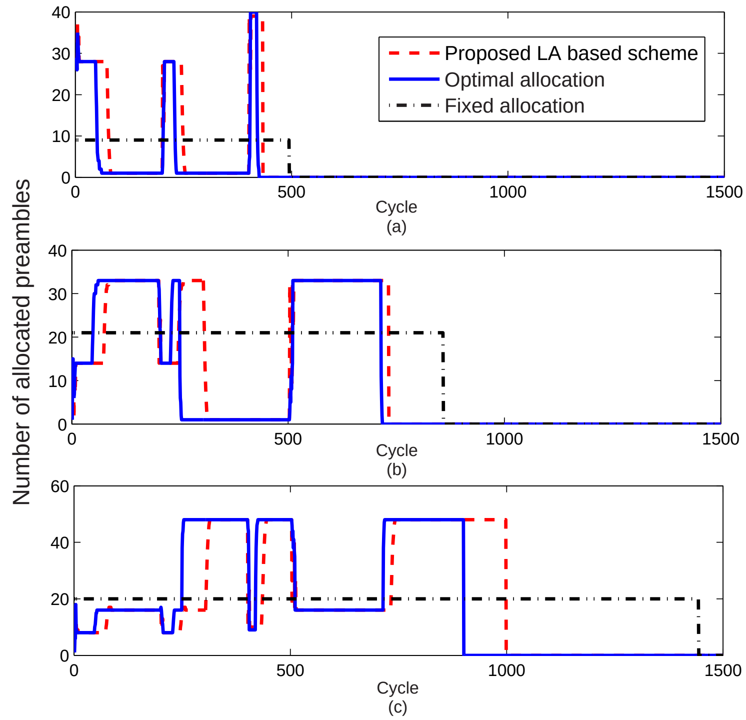

Fig. 6. (a) The number of allocated preambles to priority class H in different cycles of the TST interval. (b) The number of allocated preambles to priority class M in different cycles of the TST interval. (c) The number of allocated preambles to priority class L in different cycles of the TST interval.

is increased again and the MTCDs in class H exploit from the maximum allocatable RACH resources, i.e., C_H. The same trend is observed for other two classes.

Also, the proposed LA based scheme follows the optimal scheme in which we assume that eNB knows the number of access requests in each cycle and also, this scheme has better performance in terms of decreasing the TST compared to the fixed allocation scheme. Note that, the reason of the observed small differences between the proposed LA based scheme and optimal allocation is that the learning process in the proposed scheme is done in two steps including learning the RACH allocation and ACB factors. For this scenario, the corresponding variation in the ACB factors in different cycles of the TST intervals for the proposed LA based scheme and optimal allocation scheme are shown in Fig. 7(a), 7(b), and 7(c). The results show that the proposed LA based scheme can successfully follow the optimal decisions.

The average access delay versus the number of allocated preambles for M2M communications

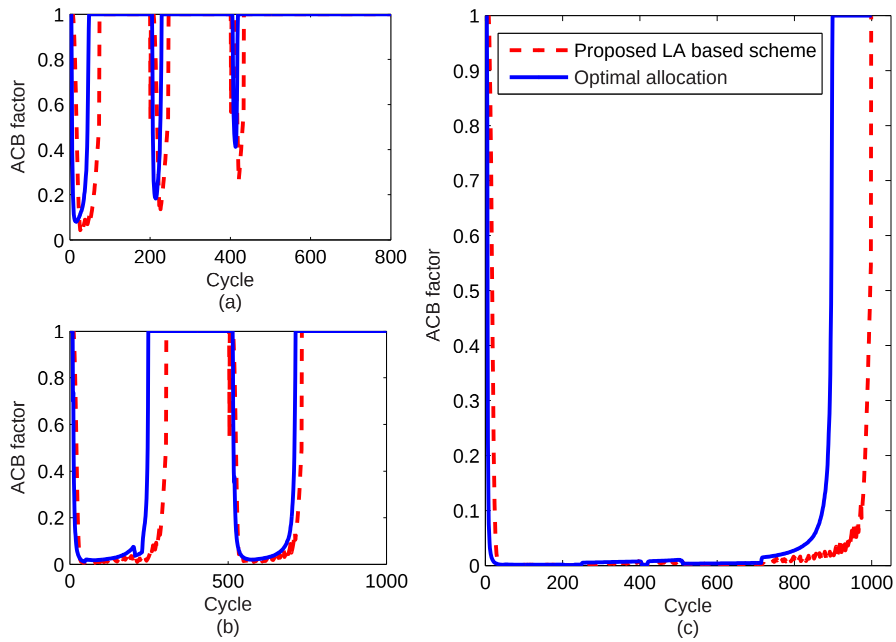

Fig. 7. The values of ACB factor in different cycles of TST interval, (a) class H, (b) class M, (c) class L.

for each of requesting devices in priority class H, M and L are depicted in Fig. 8(a), 8(b) and 8(c) respectively. In this simulation $N_H = 1000$, $N_M = 5000$ and $N_L = 10000$ and $Z_s = 600$. The traffic bursts of each class follows the beta distribution where the start times of the burst is uniformly distributed in T_s interval. Also, the number of access-attempting devices in each traffic burst is taken by a uniform random value between 1 and the number of access-attempting devices. The simulation is performed for 200 runs and the averages are reported. We find that the average access delay is decreased when the number of allocated preambles increases. Also, the proposed scheme performs close to the optimal case and has better performance compared to the fixed allocation scheme.

In Fig. 9, we provide the sensitivity analysis of the proposed scheme for variation in the proper value of learning parameter L. In this figure the cumulative distribution function (CDF) of the average access delay for different priority classes are shown for tuned learning parameter, i.e., proper L, and for $L \pm 0.15L$. The simulation is performed for 300 runs and the averages are reported. The results of this simulation show that the proposed scheme are not much sensitive to learning parameter and loading.

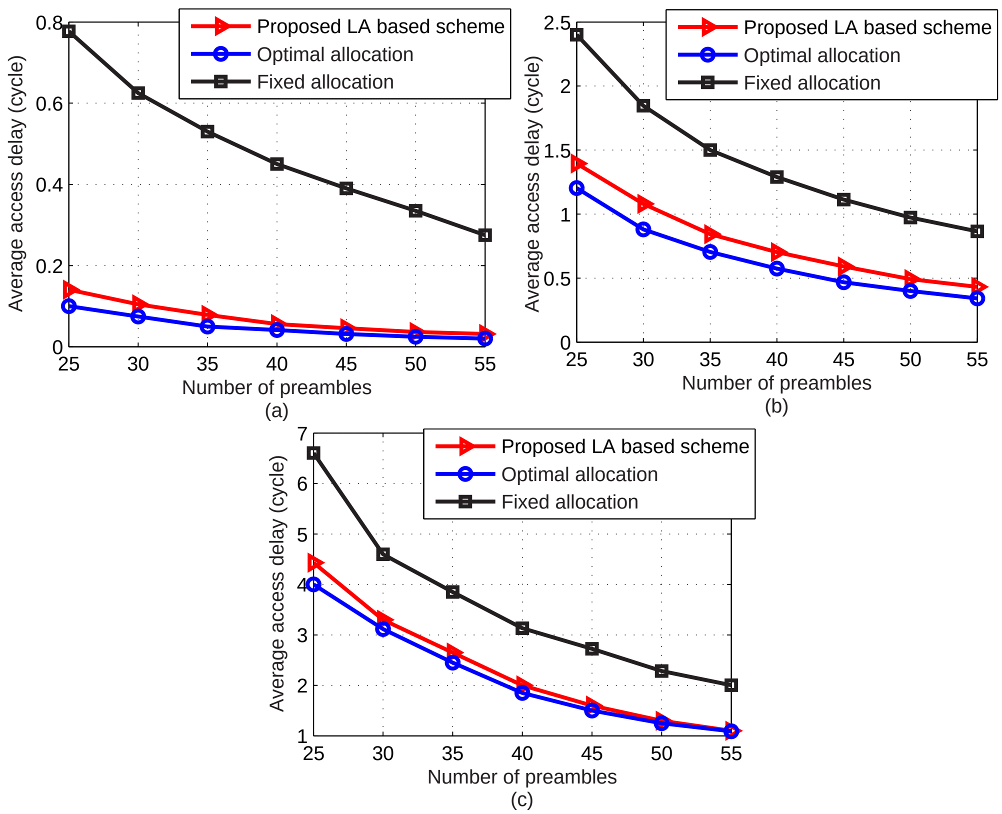

Fig. 8. Average access delay vs. the number of preambles for (a) class H , (b) class M, (c) class L.

VII. Conclusion

In this paper, we focused on supporting different priority classes of MTC devices in resource allocation procedure. We presented an LA based scheme for allocating RACH resources and adjusting the ACB factors for classes of MTC devices. Simulation results show that the propose scheme allocates the RACH resources and adjusts the ACB factors of each priority class properly. Also, it has better performance compared to the fixed allocation and follows the optimal scheme in which the eNB know the number of access requests in each cycle.

VIII. Appendix A

We use the following theorem from [34] for proving the asymptotic behaviors of the proposed LAs.

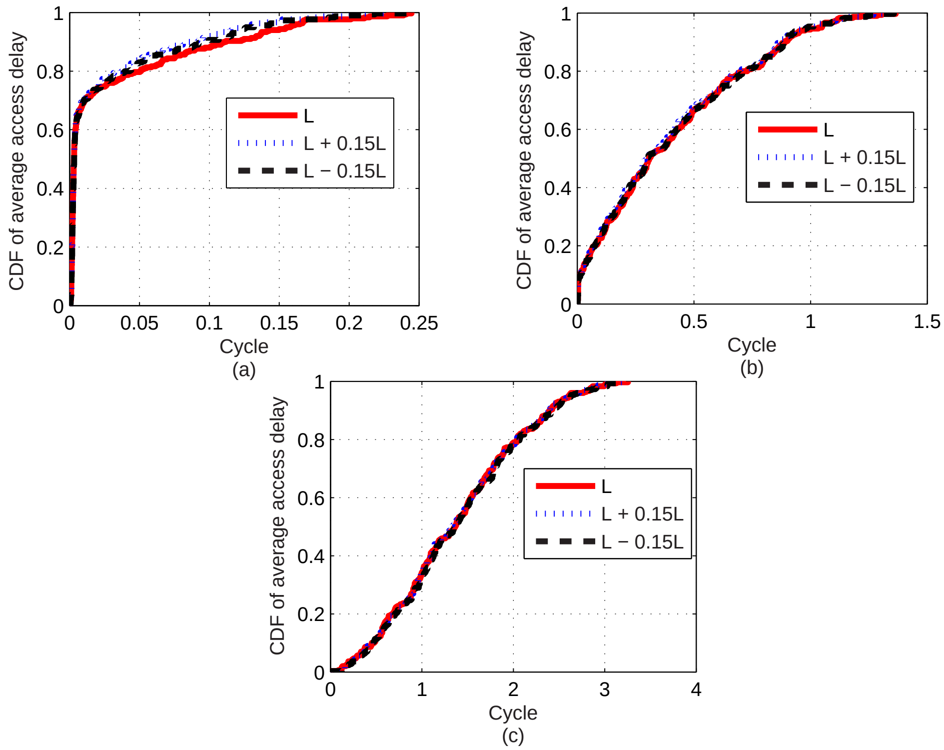

Fig. 9. CDF of the average access delay for (a) class H , (b) class M, (c) class L.

Theorem 1 [34]: Let $\{x(t)\}_{n\geq 0}$ be a stationary Markov process dependent on a constant parameter $\theta \in [0,1]$. Each $x(t) \in I$, where I is a subset of the real line. Let $\delta x(t) = x(t+1) - x(t)$. The followings are assumed to hold:

i. I is compact

ii. $E[\delta x(t)|x(t) = y] = \theta\omega(y) + O(\theta^2)$

iii. $E[|\delta x(t)|^2|x(t) = y] = \theta^2 b(y) + O(\theta^2)$.

Where $sup\frac{O(\theta^k)}{\theta^k} < \infty$ for $k >= 2$ and $sup\frac{O(\theta^2)}{\theta^2} \to 0$ as $\theta \to 0$.

iv. $\omega(y)$ has a Lipschitz in I.

v. $b(y)$ is continuous in I.

If assumptions (i)-(v) hold, for small values of the parameter θ, $\omega(y)$ has a unique root y^* in I and $d\omega/dy|_{(y=y^*)} < 0$.

Proof of Lemma 2: To use theorem 1, identify $x(t)$ with $q_x(t)$, θ with L_1, and I with (0,1). We have:

$$\begin{aligned}
E[\delta q_x(t)|q_x(t)] &= p_x^{coll}(t)(d_1 L_1(C_x - q_x(t))(q_x - a_1)) + (1 - p_x^{coll}(t))(-L_1(C_x - q_x(t))(q_x(t) - a_1)) \\
&= L_1(1 + d_1)(C_x - q_x(t))(q_x(t) - a_1)(p_x^{coll}(t) - v) \\
&= L_1\omega(q_x(t)) \qquad (32)
\end{aligned}$$

and

$$\begin{aligned}
E[|\delta q_x(t)|^2|q_x(t)] &= p_x^{coll}(t)(d_1 L_1(C_x - q_x(t))(q_x(t) - a_1))^2 + (1 - p_x^{coll}(t))(-L_1(C_x - q_x(t))(q_x(t) - a_1))^2 \\
&= L_1^2((C_x - q_x(t))(q_x(t) - a_1))^2(1 + p_x^{coll}(d_1^2 - 1)) \\
&= L_1^2 b(q_x(t)) + O(L_1^2). \qquad (33)
\end{aligned}$$

The function $\omega(q_x(t))$ and $b(q_x(t))$ are defined as follows:

$$\omega(q_x(t)) = L_1(1 + d_1)(C_x - q_x(t))(q_x(t) - a_1)(p_x^{coll}(t) - v) \qquad (34)$$

$$b(q_x(t)) = L_1^2(C_x - q_x(t))(q_x(t) - a_1)^2(1 + p_x^{coll}(d_1^2 - 1)) \qquad (35)$$

As it can be seen in (34) and (35), $\omega(q_x(t))$ is a Lipschitz function in (0,1) and $b(q_x(t))$ is a continuous function in (0,1). Therefore, assumptions (i)-(v) are satisfied for small values of L_1. For the convergence of $q_x(t)$ to the optimal point, $E[\delta q_x(t)|q_x(t)]$ must converge to 0. According to this, we should have

$$\omega(q_x(t)) = 0, \qquad (36)$$

Therefore,

$$L_1(1 + d_1)(C_x - q_x(t))(q_x(t) - a_1)(p_x^{coll}(t) - v) = 0, \qquad (37)$$

There are three possible roots for $\omega(q_x(t))$. The first root is $q_x(t) = C_x$ which means that we use from maximum percentage of allocatable resources for class x. In this case, the updating procedure in (13) does not affect $q_x(t)$ and the system is stable. The second root is happened when $q_x(t) = a_1$, but again means that there are no available resources for class x and hence the updating procedure does not affect $q_x(t)$ and the system is stable. The third root is happened when

$$p_x^{coll}(t) = v, \qquad (38)$$

where in this case the updating procedure is running. Therefore,

$$1-\frac{N_x(t)}{M\sigma_x(t)}\Big(1-\frac{1}{M\sigma_x(t)}\Big)^{(N_x(t)-1)}-\Big(1-\frac{1}{M\sigma_x(t)}\Big)^{N_x(t)}=v. \tag{39}$$

Consequently,

$$\sigma_x(t)=\frac{N_x(t)}{M}. \tag{40}$$

And therefore,

$$q_x^*(t)=\frac{\frac{N_x(t)}{M}\sum_{i=1,i\neq x}^{k}q_i(t)}{1-\frac{N_x(t)}{M}}. \tag{41}$$

If the updating procedure (13) is used, the optimal value for $q_x(t)$ is obtained according to (41). Since the value of $q_x(t)$ can not be greater than C_x, we have:

$$\lim_{l\to 0,a\to 0,t\to\infty} q_x(t)=min\Big\{\frac{\frac{N_x(t)}{M}\sum_{i=1,i\neq x}^{k}q_i(t)}{1-\frac{N_x(t)}{M}},C_x\Big\} \tag{42}$$

Proof of Lemma 1: In order to use theorem1, identify $x(t)$ with $p_x(t)$, θ with L_2, and I with $(0,1)$. We have

$$\begin{aligned} E[\delta p_x(t)|p_x(t)] &= p_x^{coll}(t)(d_2L_2(1-p_x(t))(p_x(t)-a_2))+(1-p_x^{coll}(t))(-L_2(1-p_x(t))(p_x(t)-a_2)) \\ &= L_2(1+d_2)(1-p_x(t))(p_x(t)-a_2)(p_x^{coll}(t)-v) \\ &= L_2\omega(p_x(t)), \end{aligned} \tag{43}$$

and

$$\begin{aligned} E[|\delta p_x(t)|^2|p_x(t)] &= p_x^{coll}(t)(d_2L_2(1-p_x(t))(p_x(t)-a_2))^2+(1-p_x^{coll}(t))(-L_2(1-p_x(t))(p_x(t)-a_2))^2 \\ &= L_2^2((1-p_x(t))(p_x(t)-a_2))^2(1+p_x^{coll}(d_2^2-1)) \\ &= L_2^2 b(p_x(t))+O(L_2^2). \end{aligned} \tag{44}$$

The function $\omega(p_x(t))$ and $b(p_x(t))$ are defined as follows:

$$\omega(p_x(t))=L_2(1+d_2)(1-p_x(t))(p_x(t)-a_2)(p_x^{coll}(t)-v) \tag{45}$$

$$b(p_x(t))=L_2^2((1-p_x(t))(p_x(t)-a_2))^2(1+p_x^{coll}(1+(d_2^2-1))). \tag{46}$$

As it can be seen in (45) and (46), $\omega(p_x(t))$ is a Lipschitz function in (0,1) and $b(p_x(t))$ is a continuous function in (0,1). Therefore, assumptions (i)-(v) are satisfied for small values of L_2.

For the convergence of $p_x(t)$ to the optimal point, $E[\delta p_x(t)|p_x(t)]$ must converge to 0. According to this, we have

$$\omega(p_x(t)) = 0 \tag{47}$$

Therefore,

$$L_2(1+d_2)((1-p_x(t))(p_x(t)-a_2)((p_x^{coll}(t)-v)=0 \tag{48}$$

There are three possible roots for $\omega(p_x(t))$. For $p_x(t) = 1$, we use maximum allocatable resources for class x and the updating procedure in (21) does not affect $p_x(t)$ and the system remains stable. The second root is $p_x(t) = a_2$, but same as before, this value means that there are no request for class x and therefore the updating procedure does not affect $p_x(t)$ and the system is stable. The third root happens for:

$$p_x^{coll}(t) = v, \tag{49}$$

$$1 - \frac{N_x(t)p_x(t)}{M_x(t)}\Big(1-\frac{p_x(t)}{M_x(t)}\Big)^{N_x(t)-1} - \Big(1-\frac{p_x(t)}{M_x(t)}\Big)^{N_x(t)} = v. \tag{50}$$

$$p_x(t) = \frac{M_x(t)}{N_x(t)}. \tag{51}$$

If the updating procedure (21) is used, the optimal value for $p_x(t)$ is obtained according to (50). Since the maximum value for $p_x(t)$ is 1, we have:

$$p_x^{*}(t) = min\Big\{\frac{M_x(t)}{N_x(t)}, 1\Big\}. \tag{52}$$

REFERENCES

[1] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787-2805, Oct. 2010.

[2] A. Laya, L. Alonso, and J. Alonso-Zarate, "Is the Random Access Channel of LTE and LTE-A Suitable for M2M Communications? A Survey of Alternatives," *IEEE Commun. Surv. and Tuto.*, vol. 16 , no. 1, pp. 4-16, first quarter, 2014.

[3] L. Ferdouse, A. Anpalagan and S. Misra, "Congestion and overload control techniques in massive M2M systems: a survey," *Transactions on Emerging Telecommunications Technologies* (2015).

[4] Cisco, San Jose, CA, USA, "Cisco visual networking index: Global mobile data traffic forecast update, 2014-2019 technical report," Feb. 2015.

[5] H. Shariatmadari, R. Ratasuk, S. Iraji, "Machine-type communications: current status and future perspectives toward 5G systems," *IEEE Com. Magazine*, vol. 53, no. 9, pp.10 -17, Sep. 2015.

[6] M. Tauhidul Islam, A. M. Taha, "A Survey of Access Management Techniques in Machine Type Communications,"*IEEE Communications Magazine*, VOL. 52, PP. 74 - 81, April 2014.

[7] F. Hussain, A. Anpalagan, and R. Vannithamby, "Medium access control techniques in M2M communication: survey and critical review," *European Trans. Emerging Tel. Tech*. doi: 10.1002/ett.2869, 2015.

[8] A. Biral, M. Centenaro, A. Zanella, L. Vangelista and M. Zorzi, "The challenges of M2M massive access in wireless cellular networks," *Digit. Commun. Netw*. , vol. 1, no. 1, pp.1 -19, 2015.

[9] 3GPPTS 37.868, "Study on RAN Improvements for Machine-type Communications," v11.0.0, Sep. 2011.

[10] A. Rajandekar and B. Sikdar, "A survey of MAC layer issues and protocols for machine-to-machine communications," *IEEE Internet of Things Journal*, vol. 2, no. 2, pp. 175-186, Apr 2015.

[11] K.-D. Lee, S. Kim, B. Yi, "Throughput Comparison of Random Access Methods for M2M Service over LTE Networks," *Proc. 2011 IEEE GLOBECOM Wksps.*, pp. 373-77, 2011-Dec.

[12] ZTE R2104662: "MTC Simulation Results with Specific Solutions," Aug. 2011.

[13] T.-M. Lin, C.-H.Lee, J.-P.Cheng, "PRADA: Prioritized Random Access with Dynamic Access Barring for MTC in 3GPPLTE-A Networks," *IEEE Trans. Veh. Technology*, vol. 63, no. 5, pp. 2467-2472, Jun. 2014.

[14] M. Cheng, G. Lin, H. Wei, A. Hsu, "Overload control for machine-type-communications in LTE-advanced system," *IEEE Commun. Mag.*, vol. 50, no. 6, pp. 38-45, Jun. 2012.

[15] S. Duan, V. Shah-Mansouri, and V.W.S. Wong, "Dynamic access class barring for M2M communications in LTE networks," *in Proc. Of IEEE GLOBECOM, Altlanta*, GA, Dec. 2013.

[16] S.-Y. Lien, T.-H. Liau, C.-Y. Kao and K.-C. Chen, "Cooperative access class barring for machine-to-machine communications," *IEEE Trans. Wireless Commun.*, vol. 11, no. 1, pp. 27-32, Jan. 2012.

[17] X. Yang, A. Fapojuwo, E. Egbogah, "Performance analysis and parameter optimization of random access backoff algorithm in LTE," *Proc. IEEE Veh. Technol. Conf. (VTC-Fall12)*, pp. 1-5, Sep. 2012.

[18] S.-Y. Lien, K.-C. Chen, Y. Lin, "Toward Ubiquitous Massive Accesses in 3GPP Machine-to-Machine Commu-nications," *IEEE Commun. Mag.*, vol. 49, no. 4, pp. 66-74, Apr. 2011.

[19] M.-Y. Cheng , G.-Y. Lin , H.-Y. Wei , and C.-C. Hsu, "Performance evaluation of radio access network overloading from machine type communications in LTE-A networks," *in Proc. IEEE WCNC Workshops* , Paris, France , Apr. 2012 , pp. 248 252 .

[20] M L. M. Bello, P. Mitchell, and D. Grace, "Application of Q-Learning for RACH Access to Support M2M Traffic over a Cellular Network," *in European Wireless Conference*, May 2014, pp. 1-6.

[21] A. Lo, Y. W. Law, M. Jacobsson, "Enhanced LTE-Advanced Random-Access Mechanism for Massive Machine-to-Machine (M2M) Communications," *Proc. 27th Meeting of Wireless World Research Forum*, 2011-Oct.

[22] 3GPP, "Service requirements for machine-type communications," 3GPP TS 22.368 V13.0.0, 2014.

[23] M. Tavana, V. Shah-Mansouri, and V.W.S. Wong, "Congestion control for bursty M2M traffic in LTE networks," *in Proc. IEEE ICC*, London, June 2015.

[24] Z. Wang and V.W.S. Wong, "Joint access class barring and timing advance model for machine-type communications," *in Proc. of IEEE ICC, Sydney,* Australia, June 2014.

[25] S. Sesia, I. Toufik, and M. Baker, *LTE - The UMTS Long Term Evolution.* West Sussex, UK: John Wiley & Sons, Ltd, 2009.

[26] A. Ali, W. Hamouda, "Next Generation M2M Cellular Networks: Challenges and Practical Considerations," *IEEE Com. Magazine*, vol. 53, no. 9, pp.18 -24, Sep. 2015.

[27] K. S. Narendra, and M. A. L. Thathachar, Learning automata: An introduction. Englewood Cliffs NJ: Prentice Hall, (1989).

[28] P. Nicopolitidis, G. I. Papadimitriou and A. S. Pomportsis, "Distributed protocols for Ad-Hoc wireless LANs: A learning-automata-based approach," *Ad Hoc Netw.*, vol. 2, no. 4, pp.419 -431 2004.

[29] S. Misra, V. Tiwari and M. S. Obaidat, "LACAS: Learning Automata-based Congestion Avoidance Scheme for Healthcare Wireless Sensor Networks," *IEEE JSAC,* vol. 27, no. 4, pp. 466-79, May 2009.

[30] 3GPP, "LTE: MTC LTE simulations," 3rd Generation Partnership Project (3GPP), TSG RAN WG2 v11.0.0, Sep. 2011.

[31] P. Nicopolitidis , G. I. Papadimitriou and A. S. Pomportsis, "Learning-automata-based polling protocols for wireless LANs", *IEEE Trans. Commun.,* vol. 51, pp. 453-463, 2003.

[32] S. Pediaditaki, P. Arrieta and M.K. Marina, "A Learning-Based Approach for Distributed Multi-Radio Channel Allocation in Wireless Mesh Networks," *Proc. IEEE Int',l Conf. Network Protocols (ICNP),* 2009.

[33] T. D. Lagkas, G. I. Papadimitriou, P. Nicopolitidis, and A. S. Pomportsis, "POAC-QG: Priority Oriented Adaptive Control with QoS Guarantee for Wireless Networks," *Proceedings of IEEE EUROCON 2005, Belgrade, Serbia & Montenegro,* Nov. 2005, pp. 1858-1861.

[34] G. I. Papadimitriou and D. G. Maritsas, "Self-adaptive random-access protocols for WDM passive star networks," *IEE Proc.-Comput. Digit. Tech.,* vol. 142, no. 4, pp.306 -312 1995.