Sleeping Multi-Armed Bandit Learning for Fast Uplink Grant Allocation in Machine Type Communications

Samad Ali, *Student Member, IEEE*, Aidin Ferdowsi, *Student Member, IEEE*, Walid Saad, *Senior Member, IEEE*, Nandana Rajatheva, *Senior Member, IEEE*, and Jussi Haapola, *Member, IEEE*

***Abstract*—Scheduling fast uplink grant transmissions for machine type communications (MTCs) is one of the main challenges of future wireless systems. In this paper, a novel fast uplink grant scheduling method based on the theory of multi-armed bandits (MABs) is proposed. First, a single quality-of-service metric is defined as a combination of the value of data packets, maximum tolerable access delay, and data rate. Since full knowledge of these metrics for all machine type devices (MTDs) cannot be known in advance at the base station (BS) and the set of active MTDs changes over time, the problem is modeled as a sleeping MAB with stochastic availability and a stochastic reward function. In particular, given that, at each time step, the knowledge on the set of active MTDs is probabilistic, a novel probabilistic sleeping MAB algorithm is proposed to maximize the defined metric. Analysis of the regret is presented and the effect of the prediction error of the source traffic prediction algorithm on the performance of the proposed sleeping MAB algorithm is investigated. Moreover, to enable fast uplink allocation for multiple MTDs at each time, a novel method is proposed based on the concept of best arms ordering in the MAB setting. Simulation results show that the proposed framework yields a three-fold reduction in latency compared to a maximum probability scheduling policy since it prioritizes the scheduling of MTDs that have stricter latency requirements. Moreover, by properly balancing the exploration versus exploitation tradeoff, the proposed algorithm selects the most important MTDs more often by exploitation. During exploration, the sub-optimal MTDs will be selected, which increases the fairness in the system, and, also provides a better estimate of the reward of the sub-optimal MTD.**

***Index Terms*—Machine Type Communications, Scheduling, Fast Uplink Grant, Multi-armed Bandits, Internet of Things**

I. Introduction

The next-generation of wireless networks is expected to support Internet of Things (IoT) [2], [3] services and applications such as autonomous vehicles [4] and unmanned aerial vehicles [5]. To enable such emerging IoT applications, next-generation wireless systems must have native support for machine type communications (MTCs). In MTC, a large number of machine-type-devices (MTDs) must communicate small data packets [6]. Due to the heterogeneous nature of IoT applications, MTC data packets have fundamentally novel requirements in terms of latency, reliability, and security [7]. In particular, reducing the signaling overhead and latency, while avoiding random access channel congestion is an important open problem for MTC.

MTC access schemes can be categorized into three groups: a) *Coordinated transmission*, in which MTDs send scheduling requests to the BS by using a random access process and the BS schedules MTDs. This approach can be inefficient for MTC since the data packets are small and, hence, the signaling to data packet size ratio is large, b) *grant-free transmission*, in which MTDs choose a random uplink radio resource and transmit their data without sending any scheduling request, to reduce the signaling overhead, and c) *fast uplink grant*, in which the MTDs do not send random access based scheduling requests, and, instead, the BS sends an uplink grant to MTDs based on a prediction of the set of active MTDs. Schemes a) and b) can suffer from severe collisions among transmissions because the number of MTDs is often much larger than the number of available resources. In a massive MTC [8] scenario, collision problems become even more challenging to address. The authors in [9] and [10] provide an extensive overview of several proposed solutions for such problems. The authors in [11] use non-orthogonal multiple access for the random access process so as to identify random access requests from multiple MTDs with the same preamble. Correlation between transmission patterns of different MTDs is exploited in [12] to optimize the random access process by reducing the collisions. To avoid wasting radio resources in random access collisions, the authors in [13] propose to attach the MTD identity information in the physical random access channel which will prevent the BS from allocating uplink resources to devices that collided. In summary, the works [11]–[13] are focused on optimizing random access process for MTC and solving problems associated with collisions. For grant-free transmission, in [14], the authors present a resource allocation approach for a massive number of devices with reliability and latency guarantees. Meanwhile, the work in [15] presents a game-theoretic model for optimizing the coexistence of MTDs with cellular users in the uplink period. Since IoT applications have diverse range of QoS requirements, dynamic resource allocation is used in [16] for mission critical MTC. The authors in [17] provide a dynamic QoS aware resource allocation for narrow band IoT networks. Although these prior solutions can

This work was supported by the Academy of Finland 6Genesis Flagship under grant 318927 and by the U.S. National Science Foundation under Grant CNS-1836802. A preliminary version of this work appeared in the IEEE GLOBECOM 2018 Workshops [1].

S. Ali, N. Rajatheva and J. Haapola are with the Centre for Wireless Communications (CWC), University of Oulu, Finland. Emails: {samad.ali, nandana.rajatheva, jussi.haapola}@oulu.fi. A. Ferdowsi and W. Saad are with Wireless@VT, Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA, USA, Emails: {aidin, walids}@vt.edu.

improve the performance of MTCs, coordinated access still suffer from heavy signaling overheard and collisions [8], [9], [12], [13], [18]. Moreover, grant-free transmissions still also experience non-negligible collisions, particularly in massive access scenarios [14], [15], [19], [20]. The main drawback of this prior art is that it relies solely on random access process (for sending scheduling requests in coordinated transmission or sending data packets in the grant-free scheme) whose performance is optimal only when the number of competing devices is equal to the number of available resources [21]. This clearly does not hold in massive MTC cases since the number of radio resources is limited, and hence, novel solutions are needed to address the uplink resource allocation problem for massive MTCs.

To address the challenges of random access congestion, collisions, and high signaling overhead, a middle ground from the point of view of the uplink resource allocation method between a) fully coordinated transmission by using random access based scheduling requests and b) grant-free transmission, can be achieved by using the concept the *fast uplink grant* [22] and [23]. In the fast uplink grant scheme, if the MTDs have data to transmit, they proceed with the transmission, otherwise, the radio resource is wasted [23], [24]. An overview of challenges and opportunities of the fast uplink grant is provided in [25]. As a first step to implement the fast uplink grant, one must investigate the problem of *source traffic prediction*. In this regard, in [26], an MTD traffic prediction method based on the so-called directed information is presented for source traffic prediction. By using the method proposed in [26] upon detection of an irregular transmission, a set of future active MTDs facing the same event can be detected. The authors in [27] propose a predictive resource allocation scheme for event-driven MTC in which MTDs are physically located across a line where their traffic pattern can be predicted.

The second step after predicting the source traffic is the optimal allocation of the fast uplink grants which is the focus of this work. If the BS has full knowledge of the QoS requirements of all the MTDs, this task is rather trivial. However, in practice, the MTDs might not reveal the nature of the application to the BS. Moreover, the QoS requirements of the MTDs might change at different times, due to changes in channel quality between the MTDs and the BS and the presence of various applications that must send data through a single MTD. Therefore, the BS must perform fast uplink grant allocation, with limited or no prior knowledge about the QoS requirements of the MTDs, and use the information revealed to the BS after the transmission for future fast uplink grant allocation purposes. One suitable tool for such a task is multi-armed bandit (MAB) theory. MABs are a class of reinforcement learning problems [28] in which an agent interacts with an environment and learns from its actions. MABs have been previously used in wireless communications problems (e.g., see [29] where a review of applications of MABs in small cells is provided.) The authors in [30] use MABs for channel selection in device-to-device (D2D) communications and in [31], MABs are used for distributed user association in energy harvesting small cell networks. MAB is also proposed for multi-user channel allocation for cognitive radio networks in [32].

The main contribution of this paper is to address the problem of optimal fast uplink grant allocation when the number of active devices is larger than the number of available resources, there is no prior information about QoS requirement of the MTDs, and the source traffic prediction algorithm is imperfect. We consider that the BS is not able to perfectly predict the set of the active MTDs and hence, a probability of activity is associated with each MTD at any given time. Therefore, the BS has probabilistic knowledge on the set of active MTDs and we propose a novel MAB algorithm for allocating the fast uplink grant under these conditions. The contributions of this paper can, therefore, be summarized as follows:

- In order to capture a diverse set of QoS metrics during scheduling, we introduce a compound QoS metric that is a combination of three MTD-specific metrics: a) the value of the data packets, b) maximum tolerable access delay, and c) the data rate. We concretely define this metric by proposing a novel method to model the access delay by mapping it to a value between zero and one using a sigmoid function known as Gompertz function. To find the optimal MTD that the BS must schedule at each time slot, a novel probabilistic sleeping MAB algorithm is proposed. Sleeping MABs are appropriate to address problems in which the set of of active MTDs change over time. To account for imperfect source traffic prediction algorithm, we introduce a Bayesian inference mechanism at the output of the source traffic prediction algorithm to learn prediction errors. The posterior probabilities of the Bayesian inference method are then combined with the concept of upper confidence bound (UCB) in the context of sleeping MABs.
- We rigorously analyze the regret of the proposed MAB algorithm and decouple the effect of the MTC source traffic prediction errors and the learning process on the regret. We analytically derive the conditions under which the errors in MTC source traffic prediction lead to selecting an MTD with lower utility value thereby increasing the regret of the proposed MAB algorithm.
- Simulation results show that for any source traffic prediction algorithm with good accuracy, the proposed algorithm is optimal since it achieves logarithmic regret. For example, the proposed framework achieves up to three-fold improvement in the access delay compared to a baseline random scheduling policy.
- We extend the proposed probabilistic sleeping MAB from single MTD selection to several MTD selection by using the concept of best ordering of bandits and provide an algorithm for scenarios where multiple MTDs can be scheduled at any given time. In this method, MTDs with highest UCB value are selected for transmission, which achieves much better performance in terms of delay and throughput compared to the baseline maximum prediction probability allocation policy. Here, our simulation results show two-fold performance improvement in terms of

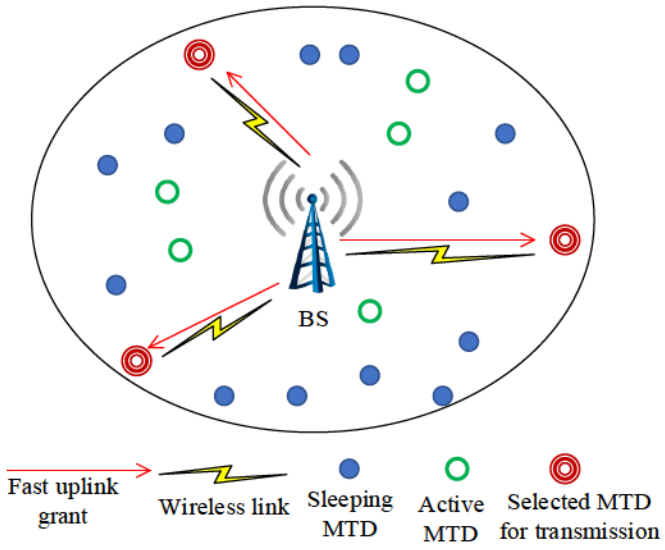


Fig. 1: Illustration of system model. First, the set of active MTDs are predicted. Next, selected MTDs receive the fast uplink grants and transmit their data.

latency compared to a baseline maximum prediction probability allocation policy.

The rest of the paper is organized as follows. Section II presents the system model and problem formulation. In Section III, we introduce the proposed probabilistic sleeping MAB solution and its extension to multiple MTDs and provide the regret analysis and study the effect of the source traffic prediction accuracy on the performance of the MAB algorithm. Numerical results are presented in Section IV and conclusions are drawn in Section V.

II. System Model and Problem Formulation

Consider the uplink of a cellular system composed of one BS and a set $\mathcal{M}$ of M MTDs that use a fast uplink grant. Scheduling is done at the BS and a fast uplink grant is sent to each scheduled MTD. We assume that the total available bandwidth is divided into resource blocks, each of which is of size W and duration τ. Without loss of generality, we consider the problem of selecting one MTD for the fast uplink grant at each time duration τ. Hereinafter, we use i for indexing MTDs and t for time. Due to the heterogeneous nature of IoT applications, packets are assumed to have different QoS requirements. The system model is presented in Fig. 1.

A. Performance Metrics

We now define three performance metrics that are combined to build a single metric that is used in the problem formulation.

1) Value of information: At time t, for each MTD i, we define the *value of information* as the assessment of the utility of an information product in a specific usage context [33]. Hence, each packet that arrives at the queue of an MTD i will have an associated value $v_i(t)$. According to [33], this value can be determined by relative pairwise comparison of all IoT applications and the use of the so-called analytic hierarchy process (AHP) to calculate the importance weight for each packet. This normalized value is derived in the form of a percentage of importance, and hence we choose $v_i(t) \in [0, 1]$.

2) Maximum tolerable access delay: Delay in a wireless communication network consists of different components: Processing delay T_p which is a function of hardware and software used by the MTDs, queuing delay T_q, and transmission delay T_t which pertains to the delay for the transmission of the data packets through the physical medium. Once the data is transmitted and received at the BS, the time needed for the packet to travel to the final destination through a network of wireless, wired, or fiber link is called routing delay T_r. Finally, the access delay T_a, which is the main focus of this work, is the time duration from the moment that the packet is ready for transmission, until the MTD receives the uplink resource blocks to transmit the packet. For each data packet of MTD i, we consider a *maximum tolerable access delay* $d_i(t_s)$ defined as the total delay that can be tolerated from the time instance t_s at which the data packet is ready to be transmitted at the MTD queue until it is scheduled to be sent. To calculate the total access delay that can be tolerated for each MTD, we first assume that all the other delay components are modeled and subtracted from the total tolerable delay of the packet. We assume T_p and T_t to be constant since the packets are small and always generated by the same devices, and the MTDs are either stationary or have low mobility. Most of the MTDs have sparse packet arrivals at their local buffer and we can consider that the service time is considerably shorter than the packet inter-arrival times. Therefore, the queuing time resulting from other packets in the buffer of each MTD is considered to be negligible.Moreover, each IoT device might be transmitting data from various applications. For example, there are IoT sensors that transmit five different data such as temperature, humidity, light, movement and CO2 levels. Each of these applications can have different delay requirements. Once all the delay components are modeled, we can calculate the maximum tolerable access delay as follows:

$$T_a = T_{\text{total}} - T_t - T_p. \tag{1}$$

Due to the fact that the values of T_{total}, T_r, T_t, and T_p are constant and that each application that is transmitting through the MTD might have different QoS requirements, the maximum tolerable access delay for each MTD will be different at any given time. Moreover, once a packet is in the MTD queue and waits for to access the channel, after each time step of waiting, its tolerable access delay will be shorter. Therefore, the packets of each MTD might have different tolerable access requirements at different times.

3) Throughput: Once each signal is received at the BS, the signal-to-noise ratio (SNR) is:

$$\gamma_i(t) = \frac{q_i(t)|h_i(t)|^2}{WN_0}, \tag{2}$$

where $h_i(t)$ represent the channel between MTD node i and the BS. N_0 is the power spectral density of the noise, W is the bandwidth of the transmission channel, and $q_i(t)$ is the transmit power of MTD i. The channel is modeled as $h_i(t) = a_i(t).g_i(t)$ where $g_i(t) \sim \mathcal{CN}(0, 1)$ represents the small-scale Rayleigh fading, assumed to be independent at different times.

Large scale fading is included in $a_i(t) = 10^{\frac{a_{i,dB}(t)}{10}}$ where $a_{i,dB}(t) = PL_{dB} + X_\sigma$ with PL_{dB} and X_σ denoting the path loss and log-normal shadowing with variance σ. We use the 3GPP path loss model from the BS to MTDs [34] which is given by $PL_{\text{dB}} = 128.1 + 37.6\log(d)$. Subsequently, the rate is given by:

$$C_i(t) = W \log\left(1 + \frac{q_i(t)|h_i(t)|^2}{WN_0}\right). \tag{3}$$

B. Problem Formulation

We first normalize $C_i(t)$ as well as the maximum tolerable access delay to a value within the range $[0, 1]$. We define the normalized rate using the following order-preserving mapping function from $[0, \infty]$ to $[0, 1]$:

$$C_i^n(t) = \frac{C_i(t)}{\phi + C_i(t))}, \tag{4}$$

where ϕ can be any positive number. We use $\phi = C_{\text{av}}$ where C_{av} is the approximate the average rate of the system that is calculated by averaging the approximate minimum and maximum possible rate from MTDs to the BS. We use the closest MTD to the BS with a line-of-sight fading model for the maximum rate. For the minimum rate, we use the path loss model with maximum shadow fading for the farthest MTD. Note that one could use any positive number for ϕ in this normalization equation, however, since smaller values would push the normalized throughput towards one, we chose C_{av} which leads to a better spreading of values in $[0, 1]$.

To normalize the maximum tolerable access delay, we use a mapping from maximum tolerable access delay to a number in $[0, 1]$ using a function $g(d_i(t))$. To do this, we use Gompertz function [35] with slight modifications, which is an asymmetric sigmoid function that is widely used in growth modeling. The rationale behind using this function is that it is possible to control the point at which the value of the function starts to decrease as well as the steepness of the curve. Gompertz function [35] is given by $w(t) = ae^{-be^{-ct}}$, where parameter a defines the asymptote of the function, b sets the displacement along the time axis, and c determines the growth rate or the steepness of the function. The Gompertz function is an increasing function in time. Moreover, since smaller values of the maximum tolerable access delay mean that the MTD has delay-sensitive data to transmit, and hence, it should have a higher value in the utility function, we modify the Gompertz function to create a new function that is decreasing with time, as follows:

$$g(d_i(t)) = a - ae^{-be^{-cd_i(t)}}. \tag{5}$$

Fig. 2 shows the plot of the modified Gompertz function for some different values of the control parameters. Any scheduling algorithm performs better in terms of delay if it selects MTDs with smaller maximum tolerable access delay, which is the one that maximizes function $g(d_i(t))$. For each MTD $i \in \mathcal{M}$, we can now define a utility function that combines all of the QoS metrics:

$$U_i(t) = \alpha v_i(t) + \beta C_i^n(t) + \gamma g(d_i(t)). \tag{6}$$

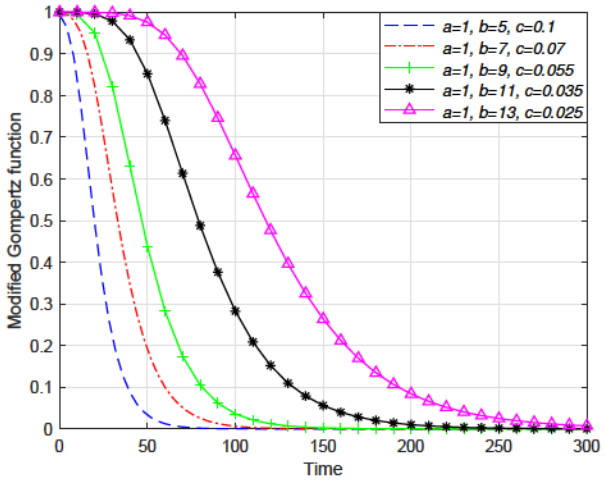


Fig. 2: Modified Gompertz function for modeling latency for different values of the control parameters.

In (6), α, β, and γ are weight parameters used to modify the importance of each metric with $\alpha+\beta+\gamma = 1$. This combined QoS metric is used to handle multiple objectives and tradeoffs between them [36]. Values of α, β, and γ can be derived by using pairwise comparison of different QoS metrics and using Analytic Hierarchy Process [37]. The best performance at time t is achieved if an MTD $k \in \mathcal{K} \subseteq \mathcal{M}$ is selected such that:

$$\begin{aligned} k = &\underset{i\in\mathcal{K}}{\text{argmax}} \quad U_i(t), \\ &\text{s.t.} \qquad\quad C_i(t) \geq \rho, \\ &\qquad\quad\; d_i \geq t - t_s, \end{aligned} \tag{7}$$

where $\mathcal{K}$ is the set of active MTDs and ρ is rate threshold required for data transmission. If $v_i(t)$, $h_i(t)$, $d_i(t)$, and the set of active MTDs are available to the BS, solving (7) is straightforward. However, in real-world networks, having such information at the BS is impractical due to the following reasons. First, MTDs should send a scheduling request to the BS using periodically available random access slots. Sending scheduling requests in MTC is not optimal since it: a) will most likely fail in massive access scenario, b) requires large signaling overhead compared to the small data packet size, and c) increases the latency. This motivates the development of a predictive resource allocation scheme, where the set of active MTDs is predicted at the BS. Second, for optimal performance in the system, the BS must know the channel state information (CSI) of the MTDs, their data values, and their exact latency requirements. Clearly, in practical MTD networks, the BS does not have full knowledge on the parameters of the metric defined in (6). For example, since the data packets are small, having instantaneous CSI at the BS requires signaling overhead that is almost equal to the data size, which is naturally inefficient. Moreover, as discussed earlier, the tolerable access delay and value of the data packets can be different each time. Therefore, it is appropriate to solve problem (7) using online learning methods with limited or no information [28] at the BS. In this case the learning algorithm can learn the statistical properties of the CSI, the tolerable access delay, and the value

of the data packets over time. Next, we propose a novel online algorithm based on MAB theory [28] to solve (7).

III. Proposed Multi-Armed Bandit Framework and Algorithm

A. MAB theory and MAB problem formulation

In a multi-armed bandit problem, a player (decision maker), pulls an arm from a set of available arms (selects an action from a set of available actions). Each arm generates a reward after being played, based on a distribution that is not known to the decision maker – the decision maker only observes the reward of the selected arm. The aim of the player is to maximize a cumulative reward or minimize a cumulative regret. Regret is defined as the difference between the reward of the best possible arm at each game instant, and the generated reward of the arm that is played.

Let $\theta(t)$ be the reward of playing an arm from the set of arms $\mathcal{K}$ at time t, and let $\theta^*(t) = \max_{i \in \mathcal{K}} \theta_i(t)$ be the highest possible reward that could be achieved at time t from the set of all arms $i \in \mathcal{K}$. The regret up to time T is defined as [28]:

$$R(T) = \mathbb{E}\left[\sum_{t=1}^{T} \theta^*(t) - \sum_{t=1}^{T} \theta(t)\right], \tag{8}$$

where t is the discrete time index and the expectation is taken over the random choices of the algorithm as well as the randomness in reward allocation. In our problem, each MTD is seen as an arm in the MAB settings and the BS is the player that selects the best arm at each time and after playing that arm, receives a reward that is generated by the metric defined in (6). Hence, the reward that is generated by each MTD $i \in \mathcal{M}$ is:

$$\theta_i(t) = 1[d_i > t - t_s]1[C_i(t) > \rho]U_i(t), \tag{9}$$

where $1(.)$ is an indicator function that is equal to 1 when the argument of the function holds and 0 otherwise. Indicator functions are used to show that the reward of the algorithm at time step t for selecting MTD i is 0 under the following conditions:

- $C_i(t) < \rho$, i.e, the achieved throughput falls below the defined threshold and the packet cannot be transmitted successfully. This often happens when the channel quality between MTD i and the BS is below a certain level.
- $d_i(t) < t - t_s$. Here, t_i is the time that MTD i is selected for transmission and t_s is the time when MTD i had a packet ready for transmission. Hence, $t - t_s$ is the number of time steps that MTD i has waited to receive the uplink grant. Naturally, if $d_i(t) < t - t_s$, then the MTD packets will be dropped and the reward at the BS for selecting MTD i will be 0.

The goal of the BS is to maximize its cumulative reward over time. To solve such a problem, the natural solution is to find the best possible arm and play it all the time. This requires playing all of the available arms for many times to find their expected value. However, randomly selecting arms in the process of learning is highly suboptimal. Hence, an MAB algorithm finds the arms with higher rewards and chooses them more often, which is known as *exploitation* of those arms. At the same time, an MAB algorithm should *explore* all the other arms enough times to find their expected value more precisely. This is known as the exploration versus exploitation tradeoff. Several methods exist to solve the problem of exploration/exploitation. One of the most popular solution approaches for the MAB problem is based on the concept of upper-confidence bound (UCB). In this method, the MAB algorithm at each time t plays an arm $x(t)$ such that:

$$x(t) = \arg\max_{i \in \mathcal{K}} \frac{z_i(t)}{n_i(t)} + \sqrt{\frac{\psi \ln t}{n_i(t)}} \tag{10}$$

where t is the time step, $n_i(t)$ is the number of the times that arm i was played in the previous time steps up to $t-1$, $z_i(t)$ is the sum of the rewards of playing arm i up to time t, and ψ is a parameter that provides a tradeoff between exploration and exploitation. Larger values of ψ lead to a higher amount of exploration. We will next use the UCB concept in our proposed probabilistic sleeping MAB algorithm to provide a tradeoff between exploration and exploitation. In the UCB method, an interval is defined around the average of the received rewards from each arm. This confidence interval $\sqrt{\frac{\psi \ln t}{n_i(t)}}$ depends on the number of the times that an arm was played and the total number of the times that algorithm is running. The more one arm is played, the UCB value becomes smaller. This means that the empirical mean is closer to the real expected value of the arm.

B. Sleeping Bandits and Proposed Algorithm

In classical MAB problems, it is assumed that all of the arms are available to be played at all time instants. However, for the MTC fast uplink grant scheduling problem, this assumption is not valid since MTDs will have a small number of packets and usually, after each transmission, they become idle for some time. Hence, we consider a scenario in which, the set of available arms varies over time. This type of problems are called *sleeping MAB* problems [38]. In our problem, since the availability of the MTDs follows the distribution of their traffic, and the reward can be described by (9), we have *sleeping bandits with stochastic action availability and stochastic rewards*. The authors in [38] provide an algorithm named AUER that addresses such problems and achieves optimal regret. However, AUER is only applicable to sleeping MAB problems in which the set of available arms is perfectly known to the decision maker in advance. In our problem formulation, such an assumption will not hold. Therefore, we propose a novel solution, summarized in Algorithm 1. Here, we consider that the BS has a prediction algorithm (e.g., such as those proposed in [39], [26], and [40]) to determine the set of active MTDs at each given time. This algorithm provides the set of active MTDs with a certain probability. That is, each MTD i has a probability $\lambda_i(t)$ of being active at time t. In this problem, since the availability of the MTDs is probabilistic, the selected MTD might not be active, which will lead to 0 reward and a waste of resources. Therefore, to solve the optimization problem in (7) we propose an MAB algorithm that takes such a probability of being active into account. Any

Algorithm 1 The probabilistic sleeping MAB algorithm.

Initialize z_i, n_i for all $i \in [n]$, initialize $t^{'}$
for $t = 1$ *to* T **do**
  if $\exists j \in \mathcal{K}_t$ s.t. $n_j = 0$ **then**
    Play arm $x(t) = j$
  else
    Calculate the posterior probability $\Lambda_i(t)$ for all MTDs
    Play arm $x(t) = \arg\max_{i \in \mathcal{K}_t} (\Lambda_i(t)) \left(\frac{z_i(t)}{n_i^{'}(t)} + \sqrt{\frac{\psi \log t^{'}}{n_i^{'}(t)}} \right)$
  end
  if $x(t)$ is an available arm $(x(t) \neq 0)$ **then**
  observe payoff $\theta_{x(t)}$
  $z_{x(t)}^{'} \leftarrow z_{x(t)}^{'} + \theta_{x(t)}(t)$
  $n_{x(t)}^{'} \leftarrow n_{x(t)}^{'} + 1$
  $t^{'} \leftarrow t^{'} + 1$
  else
  $z_{x(t)}^{'} \leftarrow z_{x(t)}^{'}$
  $n_{x(t)}^{'} \leftarrow n_{x(t)}^{'}$
  $t^{'} \leftarrow t^{'}$
end

error in the source traffic prediction algorithm that provides the set of active MTDs will affect the performance of the proposed sleeping MAB algorithm. We first define two types of prediction errors that will later be used in the proposed algorithm and the regret analysis in Section III-C:

1) For any MTD that is active at time t with probability of being available $\lambda_i(t)$, the prediction error will be $1 - \lambda_i(t)$. If an optimal MTD is active and has high prediction error, that MTD might not be scheduled and some sub-optimal MTD j will be scheduled instead, which will lead to regret $\mu_i(t) - \mu_j(t)$. $\mu_i(t)$ and $\mu_j(t)$ represent the rewards of arm i and arm j respectively. We use e_1 to capture this event.
2) For any non-active MTD j that is in the set $\mathcal{K}_t$, the prediction error is $\lambda_j(t)$. If any non-active MTD j is improperly selected due to high prediction error instead of an optimal MTD i, then the returned reward is zero, and, the hence, regret is $\mu_i(t)$. This is the highest amount of regret that can happen at any given time. We denote this event by e_2.

For any MTD that is selected for transmission, e_2 will be immediately observed. This means that if the non-active arm is played due to a high probability of error, it will be seen by the BS. Moreover, e_2 in previous time steps can also be observed by the BS by simply observing the time step when the received packet was generated. For example, if a packet was received at time step t and was generated k time steps earlier, then, the BS can observe this and infer that, in the k previous time steps, the prediction algorithm has made a mistake in providing the probability of activity for MTD i in all cases for which $\lambda_i(t) \neq 1$. Moreover, whenever the MAB algorithm selects an MTD with prediction probability $\lambda_i(t)$ and receives a reward, it can observe that the prediction algorithm had an error $1 - \lambda_i(t)$. Here, we consider the probability of being active as side information to help in selecting the best possible MTD for transmission. To thwart the prediction errors e_1 and e_2 and achieve higher accuracy, we propose to use a Bayesian approach to infer the activity status of each MTD at any given time. For this, let for each MTD i, we define the following posterior probability of being active as:

$$\Lambda_i(t) := P_{i,a}(i \text{ is active} | \lambda_i(t)) = \frac{P(\lambda_i(t)|\text{active}) P(\text{active})}{P(\lambda_i(t))}, \tag{11}$$

where the *posterior probability* $P_{i,a}(i \text{ is active} | \lambda_i(t))$ represents the probability that the arm i is active at time t given that the probability of being active that is provided by the prediction algorithm is $\lambda_i(t)$, the *likelihood* $P(\lambda_i(t)|\text{active})$ represents the probability that the prediction algorithm will provide $\lambda_i(t)$ for MTD i while the MTD is active, the *prior* probability $P(\text{active})$ is the probability that MTD i is active, and the *marginal likelihood* $P(\lambda_i(t))$ is the probability of providing $\lambda_i(t)$ for MTD i by the source traffic prediction algorithm. Since $\lambda_i(t)$ can be a continuous variable, we use binning to convert it to a discrete value which is then used in calculating the posterior probabilities. In our proposed algorithm, the BS at each time selects an MTD $x(t)$ such that:

$$x(t) = \arg\max_{i \in \mathcal{K}_t} (\Lambda_i(t)) \left(\frac{z_i(t)}{n_i^{'}(t)} + \sqrt{\frac{\psi \log t^{'}}{n_i^{'}(t)}} \right), \tag{12}$$

where $z_i(t)$ is the sum of rewards of MTD i, $n_i^{'}(t)$ is the number of the times that MTD i was selected and was active, and $t^{'}$ is the total number of the times that the selected MTD was active. $\mathcal{K}_t$ is defined as the set of active MTDs at time t. In contrast to the original UCB method, we only count the number of times that the selected MTD was active. This ensures that the statistical average and the UCB values are calculated correctly. Since the availability of the MTDs in set $\mathcal{K}_t$ have associated probabilities, the error of the prediction at the BS will propagate to the MAB. This means that the performance of the sleeping MAB will suffer since some selected MTDs for the fast uplink grant might not be active. Less error in the prediction algorithm will lead to a better performance of the probabilistic sleeping MAB. However, since the error in the prediction algorithm can lead to selecting the sub-optimal arm, and if the prediction algorithm makes the same error on the optimal arm many times in a row, the sub-optimal arm will be played which can lead to a linear increase in regret. Term $\Lambda_i(t)$ ensures that the previous mistakes of the source traffic prediction are taken into account while using the current probability of being active that is provided by the source traffic prediction algorithm. This will alleviate the performance degradation due to possible consecutive mistakes by the source traffic prediction algorithm. For example, if an optimal arm i has a small $\lambda_i(t)$ for many time steps, once it is played due to higher UCB value, (recall that the UCB value increases by time), the proposed Bayesian approach will infer that the MTD i might be active and therefore it will have higher $\Lambda_i(t)$. In a similar manner, if the prediction algorithm

assigns a high probability of being active many times to a non-active MTD, value of $\Lambda_i(t)$ for that MTD will be low, and therefore, it will have a lower $x(t)$ and will be selected less often. Note that since the Bayesian approach minimizes the error [41], then, for any given prediction algorithm, our proposed approach will achieve the best possible performance. This algorithm will eventually select MTDs with higher values of the utility function and a higher chance of being active while balancing the tradeoff between exploration and exploitation.

C. Regret Analysis of the Proposed Algorithm

Next, we provide the analytical regret analysis of the proposed probabilistic sleeping MAB. We derive the upper bound of the regret and derive the relation between the accuracy of the source traffic prediction method and the regret of our proposed algorithm. Throughout this section, we use the following setup. Consider a MAB scenario with n arms, where $\mu_1 > \mu_2 > ... > \mu_n$, with μ_i being the expected value of the rewards of arm i. We define the random variable $N_{i,j}$ as the number of times arm j was played while some arm in set $\mathcal{I} = \{1, \ldots, i\}, (i < j)$ could have been played. We define $\Delta_{i,j} = \mu_i - \mu_j$, which is always positive. The expected value of the regret can be expressed as:

$$\begin{aligned} R(T) =& \mathbb{E}\left[\sum_{j=2}^{n}\sum_{i=1}^{j-1}\left(N_{i,j}-N_{i-1,j}\right)\Delta_{i,j}\right] + \mathbb{E}\left[\mu_1(t)\right] f(e_2)T \\ =& \sum_{j=2}^{n}\sum_{i=1}^{j-1}\mathbb{E}\left[N_{i,j}\right]\left(\Delta_{i,j}-\Delta_{i+1,j}\right) + \mathbb{E}\left[\mu_1(t)\right] f(e_2)T. \end{aligned} \tag{13}$$

$N_{0,j} = 0$ and $\Delta_{j,j} = 0$ for all j [38]. In the following, $n_i(t)$ is the number of times that arm i is played until time t and T is the total time that the algorithm has been running. Moreover, in the following, $\hat{\mu}_k(t)$ shows the average received reward of arm k up to time t. Next, we derive the number of times that prediction error event e_2 happens with function $f(e_2)$. First, we present the concentration bounds that are used in the regret analysis. These bounds show that the probability that the estimated value of the reward of an arm will be within the confidence bounds used by the sleeping MAB algorithm.

Lemma 1. *Given the definitions of μ_k, $\hat{\mu}_k(t)$, and $n_k(t)$, the following holds:*

$$\begin{aligned} &\mathbb{P}\left[\hat{\mu}_k(t) - \sqrt{\frac{\psi \ln t'}{n_k'(t)}} \leq \mu_k \leq \hat{\mu}_k(t) + \sqrt{\frac{\psi \ln t'}{n_k'(t)}}\right] = \\ &\mathbb{P}\left[\mu_k - \sqrt{\frac{\psi \ln t'}{n_k'(t)}} \leq \hat{\mu}_k(t) \leq \mu_k + \sqrt{\frac{\psi \ln t'}{n_k'(t)}}\right] \geq 1 - \frac{2}{t^{2\psi}}. \end{aligned} \tag{14}$$

Proof. We start from Chernoff-Hoeffding inequality where $\hat{\mu_k}(t)$ are strictly bounded by the intervals $[0, 1]$ and considering the confidence bound $\sqrt{\frac{\psi \ln t'}{n_k'(t)}}$, the inequality can be given by

$$\mathbb{P}\left[|\hat{\mu}_k(t) - \mu_k| \geq \sqrt{\frac{\psi \ln t'}{n_k'(t)}}\right] \leq 2\exp\left(-n_k'(t)\left(\sqrt{\frac{\psi \ln t'}{n_k'(t)}}\right)^2\right). \tag{15}$$

After simplifications, we prove the lemma. ■

This lemma is used in Theorem 1 where we analyze the regret bounds of the proposed probabilistic sleeping MAB solution presented in Algorithm 1. In our proposed MAB algorithm, a suboptimal arm is selected instead of the optimal arm in the following cases: a) The MAB algorithm does not have an accurate estimate of the rewards of each arm. This mostly happens during the initial learning phase, b) A suboptimal arm is selected because of prediction error e_1, or c) Zero reward is returned due to prediction error e_2. Clearly, cases a) and b) for the regret are a function of the accuracy of the prediction algorithm. We decouple the effect of the prediction errors of the source prediction algorithm from the uncertainty of the MAB algorithm about the expected values of the rewards of each MTD. We show that prediction errors can lead to linear regret with respect to the total running time of the algorithm with a coefficient that is a function of the prediction error. However, such a coefficient becomes very small for a source traffic prediction algorithm with high accuracy, and therefore, make the linear term very small.

Theorem 1. *The regret of the probabilistic sleeping MAB algorithm is at most:*

$$\begin{aligned} R(T) \leq& \left(4\psi \ln T \Lambda_{av} + \mathcal{O}(1) + f(e_1)T\right) \\ &\times \sum_{j=2}^{n}\sum_{i=1}^{j-1}\left(\frac{1}{\left(\Lambda_i\mu_i - \Lambda_j\mu_j\right)^2}\right)\left(\Lambda_i\mu_i - \Lambda_{i+1}\mu_{i+1}\right)^2 \\ &+ \mathbb{E}\left[\mu_1(t)\right] f(e_2)T \\ \leq& \left(8\psi \ln T \Lambda_{av} + \mathcal{O}(1) + f(e_1)T\right) \\ &\times \sum_{j=1}^{n-1}\left(\frac{1}{\left(\Lambda_{j+1}\mu_{j+1} - \Lambda_j\mu_j\right)^2}\right) + \mathbb{E}\left[\mu_1(t)\right] f(e_2)T. \end{aligned} \tag{16}$$

where $f(e_2)$ and $f(e_2)$ become very small as a result of Bayesian inference, Λ_{av} is the average value of the posterior probability given by the Bayesian inference method for all MTDs and Λ_j is the average value of the posterior probability for arm j.

Proof. The proof is given in Appendix A. ■

This theorem shows that the performance of the proposed sleeping MAB algorithm is a function of the accuracy of the predictions that are done in the previous step. This theorem shows that, for a source traffic prediction algorithm with good accuracy, after the learning period, the sleeping MAB will be able to select the most important MTD and it can achieve logarithmic regret. In MAB problems, logarithmic regret, as compared to linear regret shows that the algorithms has been able to learn the arms with higher reward and the gap between the selected arm and the best arm has become smaller [28].

In our MTC setting, this means that, our of the set of active MTDs, the one with best combination of latency requirements, wireless channel quality, and high value will be selected. Clearly, we can change the coefficients of the reward that we have defined in (6) to give higher priority to the QoS of interest.

D. Multiple MTD selection

In the previous sections, we have studied the sleeping MAB algorithm for the fast uplink grant allocation problem. Most MAB algorithms are developed for selecting one arm at a time. However, in practical wireless systems, at any given time, there are multiple radio resources block that could be allocated to the MTDs, and hence, the network may need to select more than one MTD for resource allocation. Here, we extend the proposed sleeping MAB algorithm for multiple arms. We assume that there are l radio resource blocks in the frequency domain that can be allocated for l MTDs. In order to do this, since the criteria in selecting the best MTD in the probabilistic sleeping MAB algorithm was the MTD with highest UCB value, we extend our methods by selecting l highest UCB values at each time step. This method follows the concept of best ordering of arms in MAB theory, in which, the arms are ordered based on their importance to be selected [28]. If we assume that the arms are selected one by one, after selecting the best MTD, for the next selection, we must choose the next arm with highest UCB value. Hence, the ordering of the UCB values and selecting the best l MTDs is a very natural extension to the proposed probabilistic sleeping MAB. We should mention that at each time step, all of the MTDs that are active for the first time are selected first, and then other MTDs are sorted based on their UCB value. This proposed method of multiple MTD selection is summarized in Algorithm 2.

Algorithm 2 Multiple MTD selection

Initialize z_i, n_i, $n_i^{'}$ for all $i \in [n]$, initialize $t^{'}$
for $t = 1$ to T **do**
  if $\exists i \in A_t$ *s.t.* $n_i = 0$ and $n_i^{'} = 0$ **then**
    Play all arms with $x(t) = i$ or set b = number of arms with $n_i = 0$ and $n_i^{'} = 0$
  else
    Calculate the posterior probability $\Lambda_i(t)$ for all MTDs
    Order the arms in descending oder by $\Lambda_i(t)\left(\frac{z_i}{n_i^{'}} + \sqrt{\frac{\psi \log t^{'}}{n_i^{'}}}\right)$ and select the $(l-b)$ first arms
  end
  if $x(t)$ is an available arm $(x(t) \neq 0)$ **then**
  For all available arms, observe payoff $\theta_{x(t)}$
  $z_{x(t)}^{'} \leftarrow z_{x(t)}^{'} + \theta_{x(t)}$
  $n_{x(t)}^{'} \leftarrow n_{x(t)}^{'} + 1$
  $t^{'} \leftarrow t^{'} + 1$
  else
  for all non-available arms **do**
  $z_{x(t)}^{'} \leftarrow z_{x(t)}^{'}$
  $n_{x(t)}^{'} \leftarrow n_{x(t)}^{'}$
  $t^{'} \leftarrow t^{'}$
end

IV. Simulation Results

In this section, we present simulation results to show the ability of the proposed methods to learn the QoS requirements of the MTDs with no prior knowledge about them.

A. Single MTD selection

We consider a single circular cell system where the simulation parameters are given in Table I. All statistical results are averaged over a large number of independent runs. Each MTD has a reward distribution with expected value $U_i \in (0, 1)$ that should be estimated at the BS. The value of the reward function changes due to the following reasons. First, the achieved rate at each time changes due to changes in the channel quality. Second, the maximum tolerable access delay might change at different times since the packet in the MTD might face various delays. Moreover, each MTD can send packets from various applications with different data values. In the utility function, values $\alpha = 0.2$ $\beta = 0.3$, and $\gamma = 0.5$ are initially used. As needed, we change the parameters of the modified Gompertz function from Fig. 2 based on the maximum access delay required in the system to have an accurate modeling of the latency.

Table I: Simulation parameters.

Parameter	Value
Cell radius	500 m
Bandwidth	360 kHz
Total number of MTDs	500
Number of active MTDs	50
Noise figure at BS and MTA	2 dB
CU to BS path loss model	128.1 + 36.7log(d[km])
Noise spectral density	-174 dBm/Hz
Log-normal shadow fading	10 dB

In Fig. 3, we set $a = 1$, $b = 8$, and $c = 0.0.3$, and we show the regret resulting from the proposed Bayesian sleeping MAB algorithm. Our results are compared to: a) A highest probability scheduling policy, b) The case when the availability of the MTDs is not taken into account in the selection process of (12) and only UCB values are used, c) A scenario in which the prediction is error free, and d) when the value $\lambda_i(t)$ is directly multiplied to the UCB value, and e) a case where a higher prediction error is considered for the source traffic prediction. Fig. 3 clearly shows that the maximum probability allocation of radio resources has linear regret which is much worse compared to the logarithmic regret achieved by the proposed solution. Fig. 3 also shows that the proposed enhancement of our algorithm done by using the Bayesian method provides up to three-fold improvement in the performance compared to using the sleeping MAB without modification. Moreover, Fig. 3 shows that the Bayesian approach can find the mistake of the source traffic prediction algorithm as it has much better performance compared to the case when we only multiply the side information to the UCB values. The baseline maximum probability policy performs very poorly in terms of regret as seen from Fig. 3 since its regret increases linearly with time.

In Fig. 4, we consider $\alpha = \beta = 0$ and $\gamma = 1$ to study

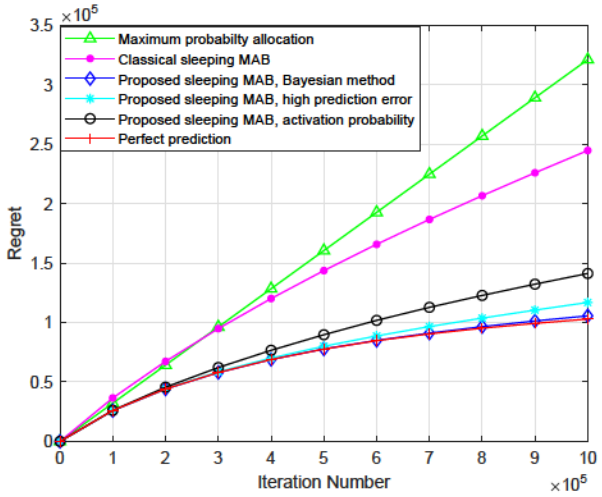


Fig. 3: Regret resulting from the proposed Bayesian sleeping MAB compared to the case in which source traffic prediction is multiplied with the UCB value, Bayesian sleeping MAB with higher prediction errror, classic sleeping MAB with no probability taken into account, sleeping MAB with perfect prediction, and maximum probability allocation. ($a = 1$, $b = 8$, and $c = 0.0.3$)

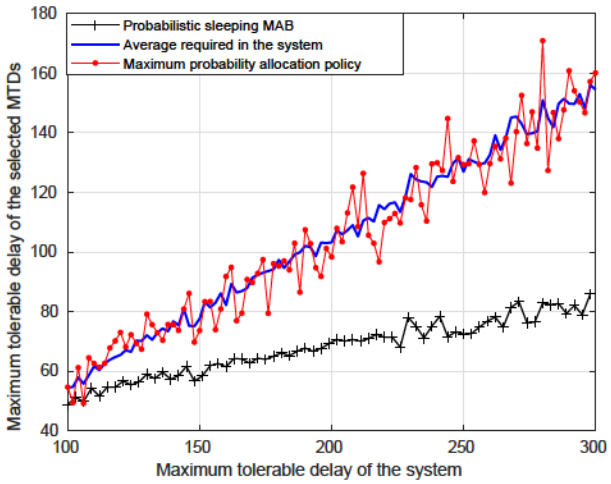


Fig. 4: Average maximum tolerable access delay of the selected MTD in fast uplink grant allocation using sleeping MABs compared to random allocation of uplink grant.

the performance in terms of latency. The maximum tolerable access delay is considered to be a value in $[1, 300]$ ms and we set the parameters of the modified Gompertz function to $a = 1, b = 13$, and $c = 0.025$ with the time horizon $T = 10^6$. For every value of the maximum tolerable access delay in the system, the average tolerable access delay that is achieved by a maximum probability allocation policy is compared to the Bayesian sleeping MAB algorithm. From Fig. 4, we can see that the maximum probability allocation of the fast uplink grant achieves a delay that is equal to the average delay of the network. This is due to the law of large numbers when the average value of the random selections approaches the expected value of the sampled experiment. In contrast, the proposed algorithm is able to select MTDs with stricter latency requirements. The maximum tolerable access delay of the MTD selected by the proposed algorithm is almost two times smaller than that of the randomly selected MTD. Note that this scheduling policy not only decreases the average latency of the system but is also able to satisfy the individual latency requirements of each MTD by prioritizing the scheduling of MTDs with strict requirements.

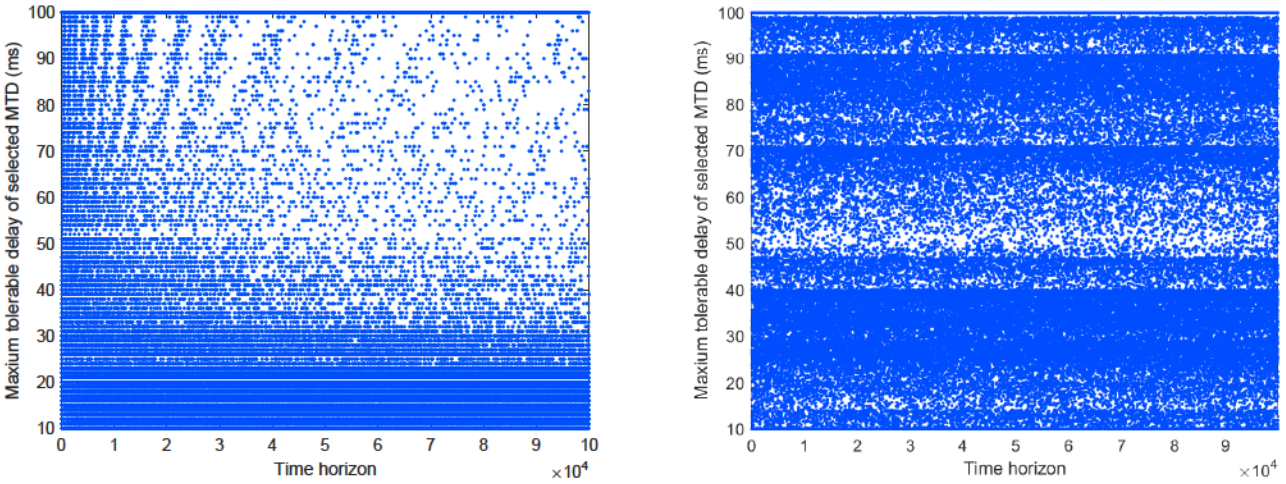
The scatter plot of the latency of the selected MTD at each time is presented in Fig. 5(a) for the proposed sleeping MAB, and in Fig. 5(b) for the maximum probability allocation case. We set the maximum tolerable access delay to 100 ms and the parameters of the modified Gompertz function to $a = 1, b = 7$, and $c = 0.07$. Each dot in these figures corresponds to the maximum tolerable access delay of the selected MTD. For any time step during which the selected MTD was not active, the maximum latency of 100 ms is plotted. Fig. 5(a) shows the effectiveness of the sleeping MAB algorithm in optimizing latency while providing fairness in the system. From Fig. 5(a), we can see that, initially, the dots are uniformly distributed which means that the MTDs are randomly selected. However, after learning, the intensity of the dots for MTDs with stricter latency requirements is much higher than that of the MTDs with larger delay requirement. Clearly delay sensitive MTDs are scheduled more often. However, after the learning period, the algorithm will keep scheduling MTDs with larger latency requirements. This increases the accuracy of the information at the BS about the latency requirements of all MTDs and also provides fairness. Moreover, if the latency requirements of an MTD has changed over time, the algorithm can discover that and start scheduling that MTD accordingly. From Fig. 5(b), we can see that a maximum probability scheduling algorithm selects the latency completely randomly at all times and the performance of the system is much worse than the proposed sleeping MAB.

In Fig. 6, we present the scatter plot of the achieved throughput of the system at each time step for the proposed sleeping MAB and the maximum probability allocation policy. Here, we have set $\alpha = \gamma = 0$ and $\beta = 1$. The bandwidth is considered to be 360 kHz and the transmit power of all the MTDs is set to 10 dBm. It is clear from Fig. 6 that the maximum prediction probability allocation policy, on average, achieves a lower rate and the proposed method yields much better average performance.

B. Multiple Resource Blocks

In this section, we provide the results for selecting multiple MTDs by using Algorithm 2. Here, we consider that all the devices require the same amount of resources and one resource block is enough for transmitting the packet of each MTD. For each failed transmission, we consider the device to be available in the next time step.

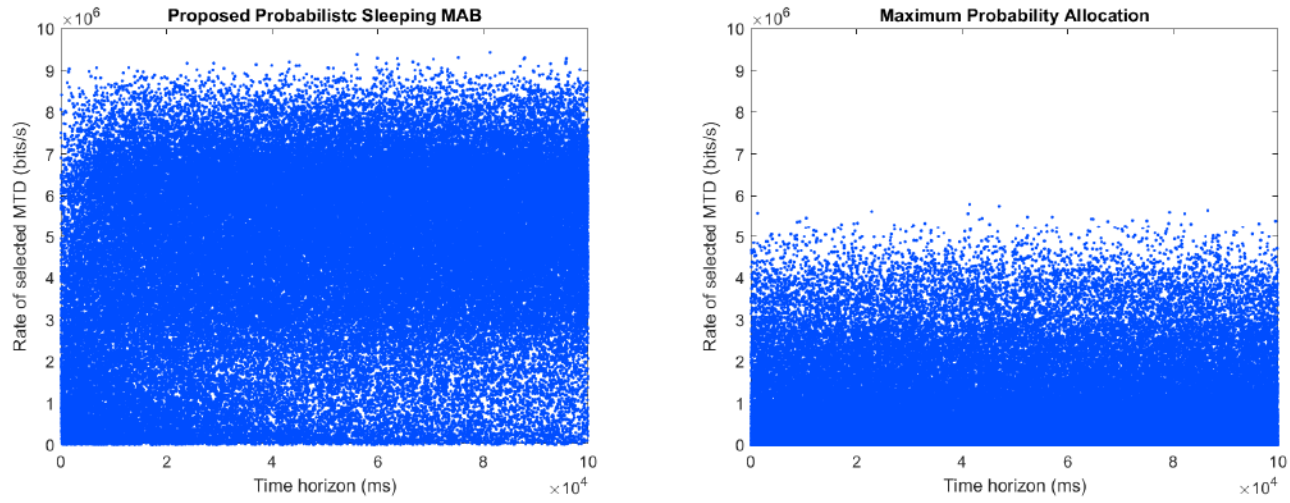
First, we provide the regret of the algorithm to study its performance. We set $a = 1, b = 8$, and $c = 0.03$, and the utility function values $\alpha = 0.2$ $\beta = 0.3$, and $\gamma = 0.5$. We assume that there are 500 MTDs in the system and, at each time, 50 MTDs are active and 20 MTDs can be scheduled at each time. The



(a) Probabilistic sleeping MAB with $\psi = 1$

(b) Maximum probability allocation

Fig. 5: Required access delay of the selected MTD at each time during the entire learning period. This figure shows how our proposed method can optimize the system while providing fairness. ($a = 1, b = 7$, and $c = 0.07$)



(a) Proposed probabilistic sleeping MAB

(b) Maximum probability allocation

Fig. 6: Scatter plot of the achieved throughput of the system at each time step for the proposed sleeping MAB compared to maximum probability allocation policy.

regret of the proposed probabilistic sleeping MAB is compared to the maximum probability baseline scenario, the scenario when the probability of being active is directly multiplied to the UCB value, sleeping MAB with only using UCB values, and perfect prediction. Fig. 7 shows that the proposed method achieves logarithmic regret. We observe that compared to the maximum probability allocation policy, the regret achieved by our proposed probabilistic sleeping MAB is nearly three and four times lower for the two different probability intervals that we considered. Fig. 7 naturally confirms that the perfect prediction scheme achieves the best performance. We can clearly see from this figure that the proposed Bayesian approach can minimize the errors of the source traffic prediction algorithm and therefore, achieve near optimal performance.

In Fig. 8, we present the average delay of the selected MTDs with $\alpha = \beta = 0$ and $\gamma = 1$. The maximum tolerable access delay is considered to be a value in $[1, 300]$ ms and we set the parameters of the modified Gompertz function to $a = 1, b = 13$, and $c = 0.025$ with the time horizon $T = 10^6$. It is clear that the proposed probabilistic sleeping MAB algorithm is able to provide much better average achieved access delay in the system. One must note that the achieved average access delay is almost constant for any value of the maximum tolerable delay, since the select MTDs are averaged. This shows that, in real-time systems, by increasing the number of MTDs, our proposed solution achieves almost a two-fold improved performance compared to baseline methods. This is an interesting result since it shows that, for a massive access scenario, our proposed method is able to achieve very low access delay. In contrast, conventional random access based systems experience excessive delays due to collisions.

In Fig. 9, a scatter plot of the average access delay of

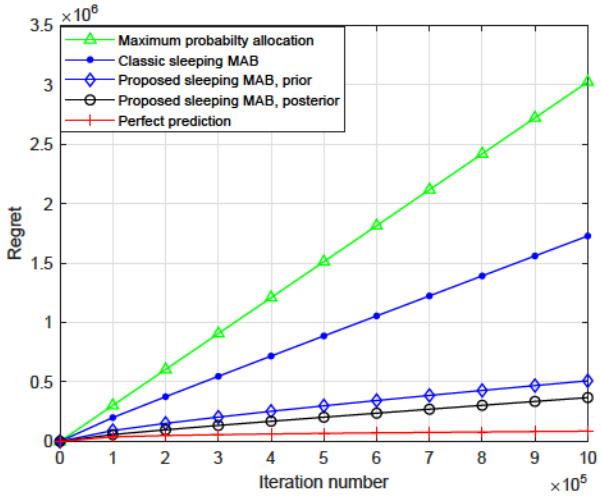


Fig. 7: Regret resulting from the proposed probabilistic sleeping MAB compared to sleeping MAB for multiple resource blocks with prediction, sleeping MAB with perfect prediction, and the maximum prediction probability allocation policy. ($a = 1, b = 8$, and $c = 0.03$)

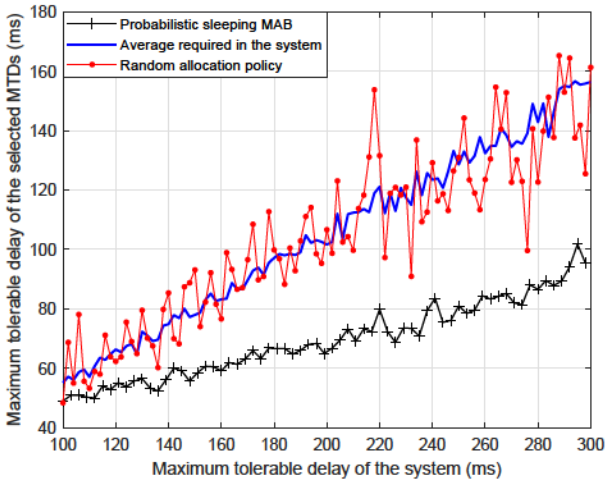


Fig. 8: Average achieve access delay in the system for all the selected MTDs. ($a = 1, b = 13$, and $c = 0.025$)

the selected MTDs is presented for our proposed solution compared to a maximum prediction probability allocation policy. We set the maximum tolerable access delay to 100 ms, and the parameters of the modified Gompertz function to $a = 1, b = 7$, and $c = 0.07$. Each dot in Fig. 9 captures the average of the maximum tolerable access delay of the selected MTDs. From 9, we can clearly observe that the proposed sleeping MAB achieves a better performance and can improve the average latency in the system. Moreover, there is a balance between selecting the MTDs with the most strict access delay requirements and exploring other MTDs to provide fairness, which can be done by changing ψ.

V. Conclusions

In this paper, we have introduced a novel sleeping MAB framework for optimal scheduling of MTDs using the fast uplink grant. First, we have devised a mixed QoS metric based on a combination of the value of the data, rate of the link, and maximum tolerable access delay of each MTD. Second, we have used that metric as a reward function in a MAB framework whose goal is to find the best MTD at each time for scheduling. Moreover, we have considered an imperfect source traffic prediction where each MTD in the set of active MTDs has a probability of being active. Then, we have proposed a probabilistic sleeping MAB framework to solve the problem of fast uplink grant allocation. We have analytically studied the regret of the proposed probabilistic sleeping MAB and we have shown how the errors in the source traffic prediction algorithm impact the performance of the proposed sleeping MAB compared to the case of perfect source traffic prediction. Moreover, we have extended the sleeping MAB algorithm for selecting multiple arms at each time to use it in scenarios where more than one MTD is scheduled at each time. Simulation results have shown that the proposed algorithm performs much better than the maximum prediction probability allocation policy, and can achieve almost three-fold performance gain in terms of latency and throughput. To the best of our knowledge, this is the first paper that addresses the optimal allocation of the fast uplink grant for MTC.

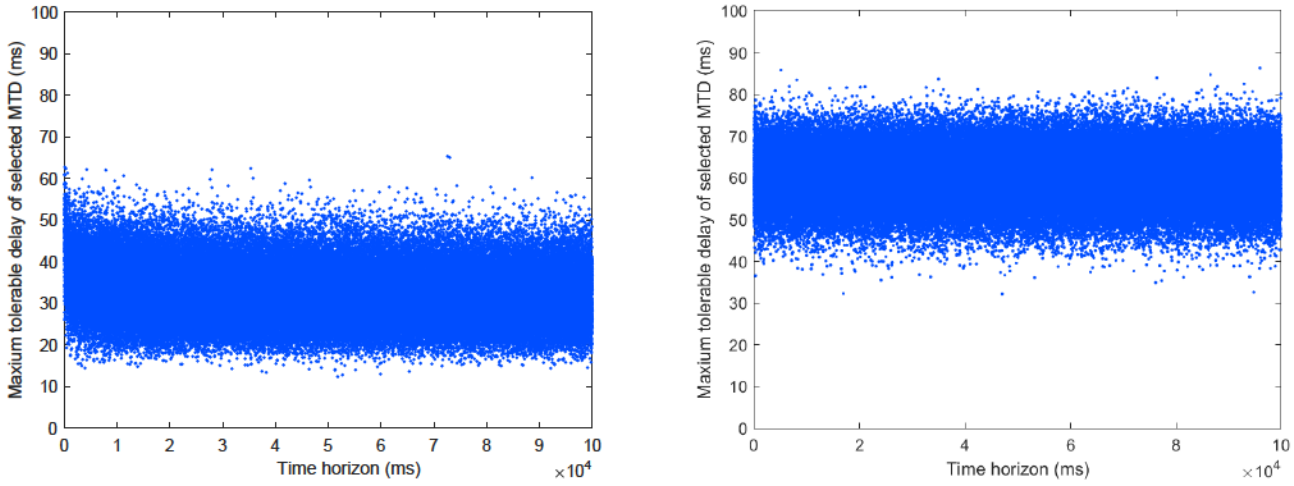
Appendix A
Proof of Theorem 1

Proof. To derive the regret for our algorithm, we need to bound the regret arm by arm. We need to find the expected value of the number of the times that each arm was played, when that arm was suboptimal. That is, what is the expected number of times $N_{i,j}$ that arm j was played while some other arm $i \in \mathcal{I}$, $i \neq j$ could have been played. Assume that arm j was already played $Q_{i,j}$ times while some other arm $i \in \mathcal{I}$ was available. The total number of times that arm j was played after it has already been played $Q_{i,j}$ is given by $L_{i,j}$ and can be written as:

$$
\begin{aligned}
L_{i,j} = & \sum_{t=Q_{i,j}+1}^{T} \sum_{s=Q_{i,j}+1}^{t} \mathbb{P}[(x_t = j) \wedge (j \text{ is played } s \text{ times}) \wedge (\mathcal{I} \neq \emptyset)] \\
& \leq \sum_{t=Q_{i,j}+1}^{T} \sum_{s=Q_{i,j}+1}^{t} \mathbb{P}\Bigg[(x_t = j) \wedge (n_j'(t) = s) \\
& \wedge \left(\forall_{k=1}^{i} \left(\Lambda_k(t)\hat{\mu}_k(t) + \sqrt{\frac{\psi \ln t'}{n_k'(t)}}\right) \leq \left(\Lambda_j(t)\hat{\mu}_j(t) + \sqrt{\frac{\psi \ln t'}{s}}\right)\right)\Bigg] \\
& \leq \sum_{t=Q_{i,j}+1}^{T} \sum_{s=Q_{i,j}+1}^{t} \mathbb{P}\Bigg[\forall_{k=1}^{i} \left(\Lambda_k(t) \left(\hat{\mu}_k(t) + \sqrt{\frac{\psi \ln t'}{n_k'(t)}}\right)\right) \leq \\
& \Lambda_j(t) \left(\hat{\mu}_j(t) + \sqrt{\frac{\psi \ln t'}{s}}\right)\Bigg]
\end{aligned}
\tag{17}
$$

To analyze this, we define two events E_1 and E_2 as follows:

$$
\begin{aligned}
E_1 := & \Bigg[\forall_{k=1}^{i} \left(\Lambda_k(t) \left(\hat{\mu}_k(t) + \sqrt{\frac{\psi \ln t'}{n_k'(t)}}\right)\right) \leq \\
& \Lambda_j(t) \left(\hat{\mu}_j(t) + \sqrt{\frac{\psi \ln t'}{n_j'(t)}}\right)\Bigg],
\end{aligned}
\tag{18}
$$



(a) Probabilistic sleeping MAB

(b) Maximum probability allocation

Fig. 9: Scatter plot of the average access delay of the system. ($a = 1, b = 7$, and $c = 0.07$)

and, for all $k \in \{j\} \cup \mathcal{I}$:

$$E_2 := \hat{\mu}_k(t) \in \left[\mu_k - \sqrt{\frac{\psi \ln t}{n_k(t)}}, \mu_k + \sqrt{\frac{\psi \ln t}{n_k(t)}}\right]. \tag{19}$$

E_2 means that average received reward for each arm is not further away than the real value of expected value of the reward of each arm, within a margin of the UCB value. We have defined E_2 since it will help us in evaluating the accuracy of our estimation of the reward for each arm. After conditioning $E1$ on $E2$, we have:

$$P[E_1] = P[E_1|E_2]P[E_2] + P[E_1|E_2^c]P[E_2^c] \leq P[E_1|E_2] + P[E_2^c]. \tag{20}$$

From Lemma 1, the probability of occurrence of E_2 for each arm is $1 - 1/t^{2\psi}$, and, thus, for all $k \in \{j\} \cup \mathcal{I}$ we have:

$$P[E_2^c] = \frac{2(i+1)}{t^{2\psi}}. \tag{21}$$

Now we can evaluate E_1 after conditioning on E_2. Event E_1 will happen if at least one of the following conditions hold [42]:

I) We are grossly overestimating the value of arm j:

$$A_1 := \Lambda_j(t)\hat{\mu}_j(t) > \mu_j + \sqrt{\frac{\psi \ln t'}{n'_j(t)}}. \tag{22}$$

By carefully evaluating events E_1 and E_2, we can observe that $P[A_1|E_2] = 0$. Note that this overestimation is evaluated considering the worst case scenario with $\Lambda_j(t) = 1$.

II) We are grossly underestimating the values of all of the arms in $\mathcal{I}$, which can be captured by the following event:

$$A_{2,1} := \forall_{k=1}^{i} \left(\Lambda_k(t)\hat{\mu}_k(t) < \mu_k - \sqrt{\frac{\psi \ln t'}{n'_k(t)}}\right). \tag{23}$$

For $P_k(t) \simeq 1$, this term never holds when conditioned on E_2, i.e, $P[A_{2,1}|E_2, P_k(t) \simeq 1] = 0$. However, for $P_k < 1$, arm k will be grossly underestimated under the following condition:

$$A_{2,2} := \forall_{k=1}^{i} \left(\Lambda_k(t) < \frac{\mu_k - \sqrt{\frac{\psi \ln t'}{n'_k(t)}}}{\hat{\mu}_k(t)}\right). \tag{24}$$

This means that, for all arms in $\mathcal{I}$, the probability of being active (while the arm is actually active) so low that the probabilistic UCB value is lower than the real expected value of the arm. However, $A_{2,2}$ is not sufficient for incurring regret and another condition must hold for arm j: the probability of being active must be high enough such that its probabilistic UCB value is within the confidence interval around the real expected value, i.e., we must have:

$$A_{2,3} := \Lambda_j(t) > \frac{\mu_j - \sqrt{\frac{\psi \ln t'}{n'_j(t)}}}{\hat{\mu}_j(t)}. \tag{25}$$

Since for selecting the suboptimal arm j both $A_{2,2}$ and $A_{2,3}$ must hold, we define the event:

$$A_{2,4} = A_{2,2} \wedge A_{2,3}, \tag{26}$$

and, thus, if $A_{2,4}$ occurs, a suboptimal arm j might be played which lead to increase in the accumulated regret. We should state that $A_{2,4}$ is independent of E_2.

III) The expected value of the arms j and k are nearly equal. When the expected values of two arms are close to each other, following two conditions will lead to choosing a suboptimal arm: a) whenever the confidence interval of the suboptimal arm is large and, hence, the suboptimal arm has higher UCB value compared to the optimal arm, or b) When the UCB value of the optimal arm is larger than the suboptimal, but the optimal arm has lower probability, and, therefore the suboptimal arm is selected. These two conditions can be expressed by:

$$A_{3,1} := \Lambda_j\mu_j + 2\sqrt{\frac{\psi \ln t'}{n'_j(t)}} > \Lambda_k\mu_k. \tag{27}$$

After rearranging (27), to choose the optimal arm, the following condition is needed for the confidence interval:

$$\sqrt{\frac{\psi \ln t'}{n_j'(t)}} < \frac{\Lambda_k \mu_k - \Lambda_j \mu_j}{2}. \tag{28}$$

Now, in order for the condition in (28) to hold, we must play arm j enough times to have an exact estimate of its value:

$$Q_{i,j}' > \left\lceil \frac{4\psi \ln T'}{\left(\Lambda_k \mu_k - \Lambda_j \mu_j\right)^2} \right\rceil. \tag{29}$$

This means that, conditioned on E_2, after playing arm j for $Q_{i,j}'$ times, $A_{3,1}$ will never happen since the confidence intervals are small enough. Therefore we have $P[A_{3,1}|E_2] = 0$.

Now, we can write (17) as:

$$L_{i,j} \leq \sum_{t=Q_{i,j}+1}^{T} \sum_{s=Q_{i,j}+1}^{t} \Big[P[A_1|E_2] + P[A_{2,1}|E_2] + P[A_{2,4}|E_2] + P[A_{3,1}|E_2] + P[E_2^c] \Big]. \tag{30}$$

We have already seen that $P[A_1|E_2] = 0$, $P[A_{2,1}|E_2] = 0$, and $P[A_{3,1}|E_2] = 0$. Moreover, $P[A_{2,4}|E_2] = P[A_{2,4}]$ since the probability of an MTD being active is independent of the event E_2. As observed from Lemma 1, we have $P[E_2^c] = 2(i+1)/t^{2\psi}$, and since E_3 has occurred, (30) simplifies to:

$$L_{i,j} \leq \sum_{t=Q_{i,j}+1}^{T} \sum_{s=Q_{i,j}+1}^{t} \Big[P[A_{2,4}] \Big] + \sum_{t=Q_{i,j}+1}^{T} \sum_{s=Q_{i,j}+1}^{t} \frac{2(i+1)}{t^{2\psi}} \tag{31}$$

$$= \sum_{t=Q_{i,j}+1}^{T} \sum_{s=Q_{i,j}+1}^{t} \Big[P[A_{2,4}] \Big] + \mathcal{O}(nT^{-\psi}) \tag{32}$$

$$= \mathcal{O}(1) + \sum_{t=Q_{i,j}+1}^{T} \sum_{s=Q_{i,j}+1}^{t} \Big[P[A_{2,4}] \Big]. \tag{33}$$

It is impossible to derive a closed-form expression for the number of times that event $A_{2,4}$ happens since the confidence interval and accuracy of the estimated average for each arm changes at each time. However, we can conclude that the number of times that $A_{2,4}$ happens is a linear function of time T multiplied by a coefficient that is the function of the prediction error $f(e_1)$. This coefficient, will be a very small value since Bayesian inference minimizes the error, and the posterior probabilities will be as accurate as possible and events in $A_{2,2}$ and $A_{2,3}$ will have small probabilities. Therefore, the probability of event $A_{2,4}$, which is a multiplication of probabilities of $A_{2,2}$ and $A_{2,3}$ will be very small. Clearly, as time increases, the Bayesian method will be able to make very small mistakes in inferring the values of $\Lambda_i(t)$, and therefore, $f(e_2)$ will eventually be very small. Therefore, we have:

$$\mathbb{E}[N_{i,j}] \leq \left\lceil \frac{4\psi \ln T'}{\left(\Lambda_i \mu_i - \Lambda_j \mu_j\right)^2} \right\rceil + \mathcal{O}(1) + f(e_1)T \tag{34}$$

We can now exactly calculate $Q_{i,j}'$. To this end, we need to count the total number of times that a given arm was selected that this arm was available, which is equal to the total number of times that we have selected an arm multiplied by probability of the availability of that arm, i.e.,

$$\begin{aligned} \frac{n_j'}{n_j} &= \mathbb{E}[\Lambda_j(t)] = \Lambda_j \Rightarrow n_j' = n_j \Lambda_j \quad \forall j \in \mathcal{A}, \\ \frac{t'}{t} &= \mathbb{E}[\Lambda_j] = \Lambda_{\text{av}} \Rightarrow t' = t\Lambda_{\text{av}}. \end{aligned} \tag{35}$$

Therefore, we can upper bound $N_{i,j}$ as:

$$\mathbb{E}[N_{i,j}] \leq \left\lceil \frac{4\psi \ln(TP_{av})}{\left(\Lambda_j \mu_i - \Lambda_j \mu_j\right)^2} \right\rceil + \mathcal{O}(1) + f(e_1)T. \tag{36}$$

By plugging this in (13), we can conclude:

$$\begin{aligned} R(T) \leq & \Big(4\psi \ln T \Lambda_{\text{av}} + \mathcal{O}(1) + f(e_1)T \Big) \\ & \times \sum_{j=2}^{n} \sum_{i=1}^{j-1} \left(\frac{1}{\left(\Lambda_i \mu_i - \Lambda_j \mu_j\right)^2} \right) \Big(\Lambda_i \mu_i - \Lambda_{i+1} \mu_{i+1} \Big)^2 \\ & + \mathbb{E}\left[\mu_1(t)\right] f(e_2) T \\ \leq & \Big(8\psi \ln T \Lambda_{\text{av}} + \mathcal{O}(1) + f(e_1)T \Big) \\ & \sum_{j=1}^{n-1} \left(\frac{1}{\left(\Lambda_{j+1} \mu_{j+1} - \Lambda_j \mu_j\right)^2} \right) + \mathbb{E}\left[\mu_1(t)\right] f(e_2) T. \end{aligned} \tag{37}$$

Since the Bayesian inference will eventually optimize the accuracy of the source traffic prediction algorithm, and, therefore, the error of the posterior probabilities will be minimized. As a results, the probability of selecting an MTD that is not available will become very small, and term to $f(e_2)$ will be a very small number. This concludes the proof. ■

REFERENCES

[1] S. Ali, A. Ferdowsi, W. Saad, and N. Rajatheva, "Sleeping multi-armed bandits for uplink grant allocation in machine type communications," in *Proc. IEEE Global Communications Conference (GLOBECOM), Workshop on Ultra-High Speed, Low Latency and Massive Connectivity Communication for 5G/B5G*, Abu Dhabi, UAE, Dec 2018, pp. 1–6.

[2] W. Saad, M. Bennis, and M. Chen, "A vision of 6G wireless systems: Applications, trends, technologies, and open research problems," *arXiv preprint arXiv:1902.10265*, 2019.

[3] M. R. Palattella, M. Dohler, A. Grieco, G. Rizzo, J. Torsner, T. Engel, and L. Ladid, "Internet of things in the 5G era: Enablers, architecture, and business models," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 3, pp. 510–527, March 2016.

[4] A. Ferdowsi, U. Challita, and W. Saad, "Deep learning for reliable mobile edge analytics in intelligent transportation systems," *CoRR*, vol. abs/1712.04135, 2017. [Online]. Available: http://arxiv.org/abs/1712.04135

[5] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Unmanned aerial vehicle with underlaid device-to-device communications: Performance and tradeoffs," *IEEE Transactions on Wireless Communications*, vol. 15, no. 6, pp. 3949–3963, June 2016.

[6] Z. Dawy, W. Saad, A. Ghosh, J. G. Andrews, and E. Yaacoub, "Toward massive machine type cellular communications," *IEEE Wireless Communications*, vol. 24, no. 1, pp. 120–128, February 2017.

[7] A. Ferdowsi and W. Saad, "Deep learning for signal authentication and security in massive Internet of Things systems," *IEEE Transactions on Communications*, vol. to appear, pp. 1–1, 2018.

[8] C. Bockelmann, N. Pratas, H. Nikopour, K. Au, T. Svensson, C. Stefanovic, P. Popovski, and A. Dekorsy, "Massive machine-type communications in 5G: physical and MAC-layer solutions," *IEEE Communications Magazine*, vol. 54, no. 9, pp. 59–65, September 2016.

[9] M. T. Islam, A. e. M. Taha, and S. Akl, "A survey of access management techniques in machine type communications," *IEEE Communications Magazine*, vol. 52, no. 4, pp. 74–81, April 2014.

[10] 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 Communications Surveys Tutorials*, vol. 16, no. 1, pp. 4–16, December 2013.

[11] Y. Liang, X. Li, J. Zhang, and Z. Ding, "Non-orthogonal random access for 5G networks," *IEEE Transactions on Wireless Communications*, vol. 16, no. 7, pp. 4817–4831, July 2017.

[12] A. E. Kalor, O. A. Hanna, and P. Popovski, "Random access schemes in wireless systems with correlated user activity," in *Proc. of IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, June 2018, pp. 1–5.

[13] N. Zhang, G. Kang, J. Wang, Y. Guo, and F. Labeau, "Resource allocation in a new random access for M2M communications," *IEEE Communications Letters*, vol. 19, no. 5, pp. 843–846, May 2015.

[14] G. C. Madueño, Č. Stefanović, and P. Popovski, "Reliable reporting for massive M2M communications with periodic resource pooling," *IEEE Wireless Communications Letters*, vol. 3, no. 4, pp. 429–432, Aug 2014.

[15] N. Abuzainab, W. Saad, C. S. Hong, and H. V. Poor, "Cognitive hierarchy theory for distributed resource allocation in the Internet of Things," *IEEE Transactions on Wireless Communications*, vol. 16, no. 12, pp. 7687–7702, Dec 2017.

[16] T. Salam, W. ur Rehman, R. Khan, I. Khan, and X. Tao, "Dynamic resource allocation and mobile aggregator selection in mission critical MTC networks," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, April 2018, pp. 64–69.

[17] W. Chen, H. Zhang, H. Ji, and X. Li, "Dynamic QoS-aware resource allocation for narrow band internet of things," in *2018 IEEE/CIC International Conference on Communications in China (ICCC Workshops)*, Aug 2018, pp. 107–111.

[18] Z. Wang and V. W. S. Wong, "Optimal access class barring for stationary machine type communication devices with timing advance information," *IEEE Transactions on Wireless Communications*, vol. 14, no. 10, pp. 5374–5387, Oct 2015.

[19] R. Abbas, M. Shirvanimoghaddam, Y. Li, and B. Vucetic, "Grant-free massive NOMA: Outage probability and throughput," *arXiv preprint arXiv:1707.07401*, 2017.

[20] B. Wang, L. Dai, Y. Zhang, T. Mir, and J. Li, "Dynamic compressive sensing-based multi-user detection for uplink grant-free noma," *IEEE Communications Letters*, vol. 20, no. 11, pp. 2320–2323, Nov 2016.

[21] J. F. Kurose and K. W. Ross, "Computer networking: a top-down approach," *Addison Wesley*, vol. 4, p. 8, 2007.

[22] 3GPP, "Study on latency reduction techniques for LTE," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 36.881.

[23] C. Hoymann, D. Astely, M. Stattin, G. Wikstrom, J. F. Cheng, A. Hoglund, M. Frenne, R. Blasco, J. Huschke, and F. Gunnarsson, "LTE release 14 outlook," *IEEE Communications Magazine*, vol. 54, no. 6, pp. 44–49, June 2016.

[24] P. Schulz, M. Matthe, H. Klessig, M. Simsek, G. Fettweis, J. Ansari, S. A. Ashraf, B. Almeroth, J. Voigt, I. Riedel, A. Puschmann, A. Mitschele-Thiel, M. Muller, T. Elste, and M. Windisch, "Latency critical IoT applications in 5G: Perspective on the design of radio interface and network architecture," *IEEE Communications Magazine*, vol. 55, no. 2, pp. 70–78, February 2017.

[25] S. Ali, N. Rajatheva, and W. Saad, "Fast uplink grant for machine type communications: Challenges and opportunities," *IEEE Communications Magazine*, vol. 57, no. 3, pp. 97–103, March 2019.

[26] S. Ali, W. Saad, and N. Rajatheva, "A directed information learning framework for event-driven M2M traffic prediction," *IEEE Communications Letters*, vol. 22, no. 11, pp. 2378–2381, Nov 2018.

[27] J. Brown and J. Y. Khan, "A predictive resource allocation algorithm in the LTE uplink for event based M2M applications," *IEEE Transactions on Mobile Computing*, vol. 14, no. 12, pp. 2433–2446, Dec 2015.

[28] R. S. Sutton, A. G. Barto *et al.*, *Reinforcement learning: An introduction*. MIT press, 1998.

[29] S. Maghsudi and E. Hossain, "Multi-armed bandits with application to 5G small cells," *IEEE Wireless Communications*, vol. 23, no. 3, pp. 64–73, June 2016.

[30] S. Maghsudi and S. Stańczak, "Channel selection for network-assisted D2D communication via no-regret bandit learning with calibrated forecasting," *IEEE Transactions on Wireless Communications*, vol. 14, no. 3, pp. 1309–1322, March 2015.

[31] S. Maghsudi and E. Hossain, "Distributed user association in energy harvesting small cell networks: A probabilistic bandit model," *IEEE Transactions on Wireless Communications*, vol. 16, no. 3, pp. 1549–1563, March 2017.

[32] Y. Gai, B. Krishnamachari, and R. Jain, "Learning multiuser channel allocations in cognitive radio networks: A combinatorial multi-armed bandit formulation," in *Proc. of IEEE Symposium on New Frontiers in Dynamic Spectrum (DySPAN)*, Singapore, Singapore, April 2010, pp. 1–9.

[33] C. Bisdikian, L. M. Kaplan, and M. B. Srivastava, "On the quality and value of information in sensor networks," *ACM Trans. Sen. Netw.*, vol. 9, no. 4, pp. 48:1–48:26, Jul. 2013. [Online]. Available: http://doi.acm.org/10.1145/2489253.2489265

[34] 3GPP, "Radio frequency (RF) requirements for LTE pico node B," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 36.931.

[35] D. Jukić, G. Kralik, and R. Scitovski, "Least-squares fitting Gompertz curve," *Journal of Computational and Applied Mathematics*, vol. 169, no. 2, pp. 359–375, 2004.

[36] E. Bjornson, E. A. Jorswieck, M. Debbah, and B. Ottersten, "Multiobjective signal processing optimization: The way to balance conflicting metrics in 5g systems," *IEEE Signal Processing Magazine*, vol. 31, no. 6, pp. 14–23, Nov 2014.

[37] R. Saaty, "The analytic hierarchy process: what it is and how it is used," *Mathematical Modelling*, vol. 9, no. 3, pp. 161 – 176, 1987. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0270025587904738

[38] R. Kleinberg, A. Niculescu-Mizil, and Y. Sharma, "Regret bounds for sleeping experts and bandits," *Machine learning*, vol. 80, no. 2-3, pp. 245–272, 2010.

[39] M. Laner, P. Svoboda, N. Nikaein, and M. Rupp, "Traffic models for machine type communications," in *Proc. of the Tenth International Symposium on Wireless Communication Systems*, Ilmenau, Germany, Aug 2013, pp. 1–5.

[40] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, "Machine learning for wireless networks with artificial intelligence: A tutorial on neural networks," *CoRR*, vol. abs/1710.02913, 2017. [Online]. Available: http://arxiv.org/abs/1710.02913

[41] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. New York, NY, USA: Wiley-Interscience, 2000.

[42] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine learning*, vol. 47, no. 2-3, pp. 235–256, 2002.