

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

Samad Ali^{*}, Aidin Ferdowsi[†], Walid Saad[†], and Nandana Rajatheva^{*}

^{*} Center for Wireless Communications (CWC), University of Oulu, Oulu, Finland,
Emails: {samad.ali, nandana.rajatheva}@oulu.fi

[†] Wireless@VT, Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA, USA,
Email: {aidin, walids}@vt.edu

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. Numerical results show that the proposed algorithm has logarithmic regret, and hence is optimal. The results also show that the proposed framework yields a three-fold reduction in latency compared to a random 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 is able to provide system fairness by allowing the most important MTDs to be scheduled more often while also allowing the less important MTDs to be selected enough times to ensure the accuracy of estimation of their importance.

I. INTRODUCTION

The fifth generation of cellular communication networks (5G) is expected to support Internet of Things (IoT) [1] services and applications such as virtual reality, autonomous vehicles, and unmanned aerial vehicles [2]. To enable such emerging IoT applications, 5G systems must have native support for machine type communications (MTCs). In contrast to enhanced mobile broadband (eMBB) services that require high data rates for large data packets, in MTC, a large number of machine-type-devices (MTDs) must communicate small data packets [3]. Due to the heterogeneous nature of IoT applications, MTC data packets have fundamentally novel requirements in terms of latency, reliability, and security [4]. These requirements bring forward new challenges to cellular communication systems that include random access channel congestion, signaling overhead management, and need for

satisfying various quality-of-service (QoS) requirements for different IoT applications [5].

MTC can be categorized into two groups based on whether scheduling requests are sent by MTDs or not. The first MTC group is *coordinated transmission*, in which MTDs perform a random access process and the base station (BS) schedules MTDs, similar to conventional cellular systems. In the second method, known as *uncoordinated transmission*, to reduce the signaling overhead, MTDs pick a random uplink radio resource and transmit their data without sending any scheduling request. Both approaches can suffer from severe collisions among transmissions due to the fact that the number of MTDs is often much larger than the number of available resources. In a coordinated transmission, collisions can occur during random access while in the uncoordinated method, they occur during packet transmission. In a massive MTC [6] scenario, such problems become even more challenging to address. The authors in [7] and [8] provide an extensive overview of several proposed solutions for such problems. One possible solution is known as access class barring (ACB) [9] where different access classes are assigned to MTDs and in massive access scenario, MTDs with lower class are barred from a transmission. For uncoordinated transmission, [10] presents a resource allocation approach for a massive number of devices with reliability and latency guarantees. Even though these prior solutions can improve the performance of MTCs, coordinated access still suffer from heavy signaling overhead and collisions. Moreover, uncoordinated transmissions still also experience non-negligible collisions, particularly in massive access scenarios. The main drawback of this prior art is that it relies solely on random access whose performance is optimal only when the number of competing devices is equal to the number of available resources. This clearly does not hold in massive MTC cases since the number of radio resources are limited, and hence, novel solutions are needed.

To address the challenges of random access congestion, collisions and high signaling overhead, a middle ground between fully scheduled and uncoordinated transmission can be achieved by using the so-called *fast uplink grant* [11], [12]. In the fast uplink grant, a BS sends an uplink grant to MTDs without MTDs sending scheduling requests. If the MTDs have data to transmit, they proceed with the transmission, otherwise,

This research was supported by the Academy of Finland 6Genesis Flagship under Grant 318927 and, in part, by the U.S. National Science Foundation under Grants AST-1506297 and CNS-1460316.

the radio resource is wasted [12]. However, the fast uplink grant has two main challenges. First, the set of the MTDs that have data to transmit should be known to the BS which is known as *source traffic prediction*. To address this problem, in [13], the use of directed information for event-driven source traffic prediction is presented. Second, when the number of active MTDs is larger than the number of fast uplink grants that can be allocated, an optimal allocation policy must be developed. An overview of challenges and opportunities of the fast uplink grant is provided in [14] where some solutions are proposed for both of the problems associated with the fast uplink grant.

The main contribution of this paper is to provide a novel solution for the problem of allocating fast uplink grants to MTDs. Since the BS has to take different QoS metrics into account while scheduling, first, we define 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. Then, we propose a novel method to model the delay by mapping it to a value in $[0, 1]$ using a sigmoid function. Finally, to find the optimal MTD for scheduling at each time slot, a probabilistic sleeping multi-armed bandit (MAB) algorithm is proposed. The proposed MAB can effectively schedule MTDs with low signaling overhead and no collision. Simulation results show that the algorithm is optimal since it achieves logarithmic regret, and it is able to find the best MTD to schedule. For example, the proposed framework achieves up to three-fold improvement in the access delay compared to a baseline random scheduling 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. 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 have different QoS requirements. The system model is presented in Fig. 1.

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 [15]. Hence, each packet that arrives at the queue of an MTD i will have an associated value $v_i(t)$. According to [15], 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

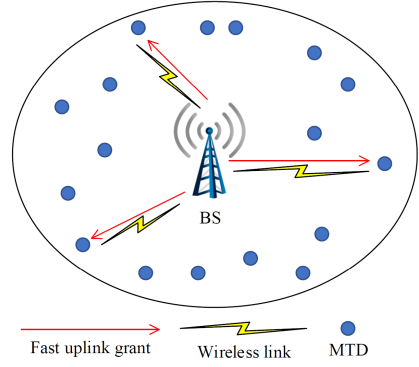


Fig. 1: Illustration of system model.

derived in the form of a percentage of importance, and hence we choose $v_i(t) \in [0, 1]$.

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. Queuing, transmission and propagation delays are assumed to be already subtracted from the total latency requirements of each IoT application that has generated the data packets. Such a maximum tolerable access delay might change for each MTD over time since, at different times, data packets might have had different queueing times, or even different latency requirements due to various applications that are generating the packets.

Once each signal is received at the BS, the throughput of each link is:

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

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 .

A. Problem Formulation

We first normalize $C_i(t)$ as well as the maximum tolerable access delay to a value within the range $[0, 1]$. For the rate $C_i(t)$, we simply divide the achieved rate by the maximum rate C_{\max} that can be achieved by the node having the best channel to the BS. We fix C_{\max} for the entire period by using the knowledge of the set of all the MTDs that are registered in the network. Thus, we use a normalized rate $C_i^n(t) = C_i(t)/C_{\max}$.

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 $f(d_i(t))$. To do this, we use Gompertz function [16] 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 is given by [16]:

$$g(t) = ae^{-be^{-ct}}, \quad (2)$$

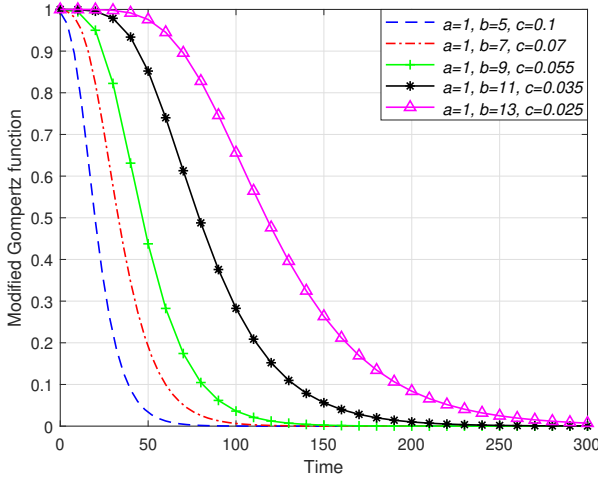


Fig. 2: Modified Gompertz function for modeling latency for different values of the control parameters.

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 in time, as follows:

$$f(d_i(t)) = a - ae^{-be^{-cd_i(t)}}. \quad (3)$$

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 $f(d_i(t))$. We can now define a utility function that combines all QoS metrics:

$$U_i(t) = \alpha v_i(t) + \beta C_i^n(t) + \gamma f(d_i(t)). \quad (4)$$

In (4), α , β , and γ are weight parameters used to modify the importance of each metric with $\alpha + \beta + \gamma = 1$. The best performance at time t is achieved if an MTD $k \in \mathcal{K} \subseteq \mathcal{M}$ is selected such that:

$$\begin{aligned} k = \operatorname{argmax}_{i \in \mathcal{K}} \quad & U_i(t), \\ \text{subject to} \quad & C_i(t) \geq \rho, \\ & d_i \geq t - t_s, \end{aligned} \quad (5)$$

where \mathcal{K} is the set of MTDs with non-empty queue buffer and ρ is rate threshold required for data transmission. If $v_i(t)$, $h_i(t)$, $d_i(t)$, and the set \mathcal{K} of active MTDs, \mathcal{K} are available to the BS, solving (5) 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 terms of the achieved throughput of the system, the BS must know the channel state information (CSI) of the MTDs. since the data packets are small having instantaneous CSI at the BS requires signaling which is inefficient. Hence, having a statistical knowledge of the channel at the BS is desirable. Clearly, in practical MTD networks, the BS does not have the knowledge on the parameters of the defined metric in (4) and hence must learn the utility function over time. Therefore, the problem in (5) must be solved using online methods with limited or no information in the BS. In the next section, we propose an online algorithm based on MAB to solve (5).

III. PROPOSED MULTI-ARMED BANDIT FRAMEWORK AND ALGORITHM

In an MAB 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 $\theta_k(t)$ be the reward of playing arm k at time t , and $\theta^*(t) = \max_{i \in \mathcal{I}} \theta_i(t)$ to be the highest possible reward that could be achieved at time t from the set of all arms $i \in \mathcal{I}$. Regret up to time T is defined as [17]:

$$R(T) = \mathbb{E} \left[\sum_{t=1}^T \theta^*(t) - \sum_{t=1}^T \theta_k(t) \right], \quad (6)$$

where the expectation is taken over the random choices of the algorithm and 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 (4). Hence, the reward that is generated by each MTD is:

$$\theta_i(t) = \mathbb{1}[d_i > t - t_s] \mathbb{1}[R_i(t) > \rho] U_i(t), \quad (7)$$

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

- $C_i(t) < \rho$, i.e, the achieved falls below the defined threshold and the packet cannot be transmitted successfully. This often happens when channel quality between MTD and BS is below a certain level.

- $d_i(t) < t_i - t_s$. Here, t_i is the time that node i is selected for transmission and t_s is the time when node 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 fast uplink grant. Naturally, if $d_i(t) < t_i - t_s$, 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 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 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 dilemma. Several methods exist to solve the problem of exploration/exploitation. One of the most recognized methods for the MAB problem is to use the concept of upper-confidence bound (UCB). In this method, the MAB algorithm at each time t plays an arm k such that:

$$I(t) = \arg \max_k \frac{1}{n_k} \sum_{i=1}^{n_k} R_i + \sqrt{\frac{s \ln t}{n_k}} \quad (8)$$

where n_k is the number of the times that arm k was played before and s is a parameter that provides a trade-off between exploration and exploitation. Larger values of s lead to the 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.

A. 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 are silent for some time. Hence, we consider a scenario in which, the set of available arms varies over time. This type of problems is called *sleeping MAB* problems. In our problem, since the availability of the MTDs follows the distribution of their traffic, and the reward can be described by (7), we have *sleeping bandits with stochastic action availability and stochastic rewards*. The authors in [18] provide an algorithm named AUER for such problems that achieve optimal regret. However, AUER is suitable for 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. Here, we consider that the BS has a prediction algorithm 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 $P_i(t)$ of being active at time t . In this problem, since the availability of the MTDs are probabilistic, the selected MTD might not be

Algorithm 1 The Probabilistic Sleeping MAB Algorithm.

```

Initialize  $z_i, n_i$  for all  $i \in \mathcal{I}$ , 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
    Play arm
     $x(t) = \arg \max_{i \in \mathcal{K}_t} P_i(t) \times \left( \frac{z_i}{n_i} + \sqrt{\frac{8 \log t'}{n_i}} \right)$ 
  end
  if  $x(t)$  is an available arm ( $x(t) \neq 0$ ) then
    observe payoff  $r_{x(t)}$ 
     $z_{x(t)} \leftarrow z_{x(t)} + r_{x(t)}$ 
     $n_{x(t)} \leftarrow n_{x(t)} + 1$ 
     $t' \leftarrow t' + 1$ 
  else
     $z_{x(t)} \leftarrow z_{x(t)}$ 
     $n_{x(t)} \leftarrow n_{x(t)}$ 
     $t' \leftarrow t'$ 
  end
end

```

active, which will lead to 0 reward and a waste of resources. Hence, to solve the optimization problem in (5) we propose an algorithm that takes such a probability of being active into account. In this algorithm, the BS at each time selects an MTD $x(t)$ such that:

$$x(t) = \arg \max_{i \in \mathcal{K}_t} P_i(t) \times \left(\frac{z_i}{n_i} + \sqrt{\frac{8 \log t'}{n_i}} \right) \quad (9)$$

where z_i is the sum of rewards of MTD i , n_i 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 presents 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. The proposed algorithm is shown in Algorithm 1. This algorithm will select MTDs with higher values of the utility function and higher probability of being active while balancing the tradeoff between exploration and exploitation.

IV. SIMULATION RESULTS

We consider a single circular cell with radius 500 meters consisting of 100 MTDs with 10 MTDs being active at each time. The probability of being active for each MTD is selected randomly to be a number in $[0.8, 1]$ and is assumed to be

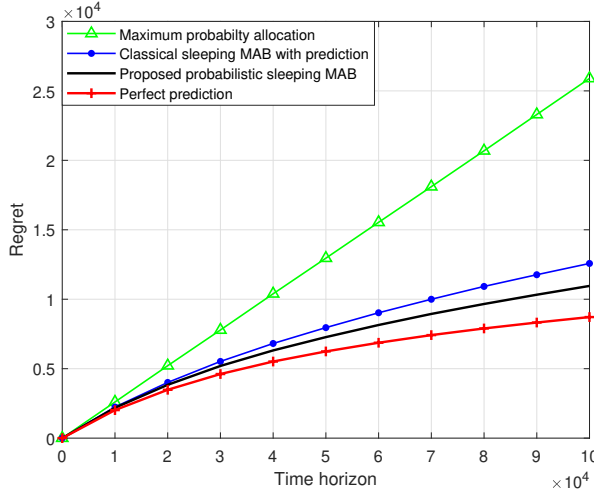


Fig. 3: Regret resulting from the proposed probabilistic sleeping MAB compared to sleeping MAB with prediction, sleeping MAB with perfect prediction, and random allocation.

known to the BS at each time. The noise power is considered to be -174 dBm/Hz and bandwidth is 360 kHz. Each MTD has a reward distribution with expected value $U_i \in [0, 1]$. 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 queuing 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.

In Fig. 3, we set $a = 1$, $b = 8$, and $c = 0.03$, and we present the regret resulting from the proposed sleeping MAB algorithm. The result is compared to: a) a random scheduling policy, b) the case when the availability of the MTDs is not taken into account in the selection process of (9) and only UCB values are used, and c) a scenario in which the prediction is error free. Fig. 3 clearly shows that the random allocation of radio resources has linear regret which is much worse compared to the logarithmic regret achieved by the proposed solution. The regret of the proposed algorithm is logarithmic which is considered optimal for MAB problems. Fig. 3 also shows that the slight modification to the algorithm by adding the probability in (9) improves the performance compared to using the sleeping MAB without modification. Moreover, Fig. 3 shows that perfect prediction has the best performance.

In Fig. 4, we consider $\alpha = \beta = 0$ and $\gamma = 1$ to study 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 $T = 10^6$ ms. For every value of the maximum tolerable

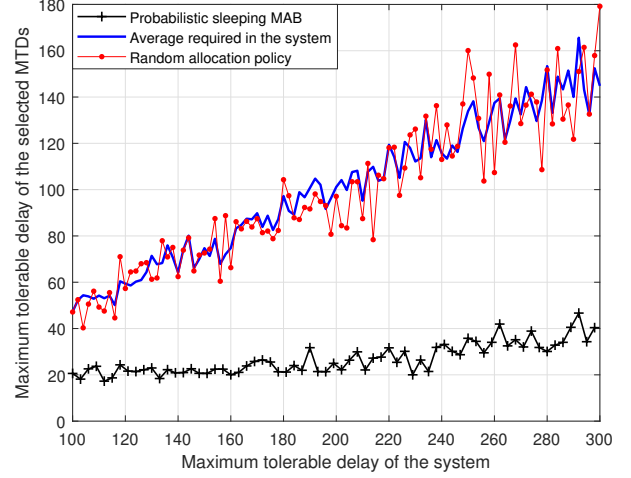
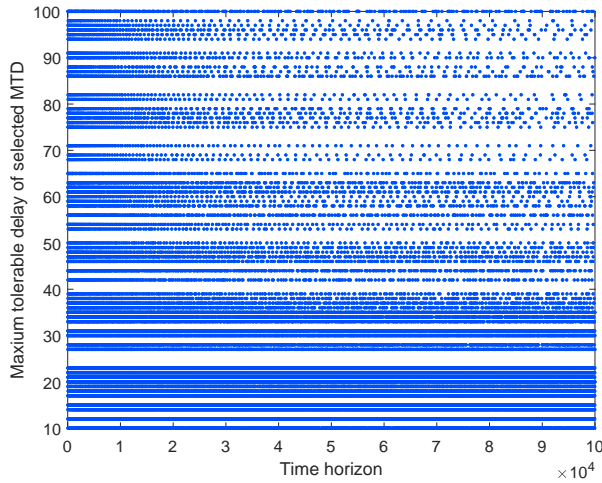


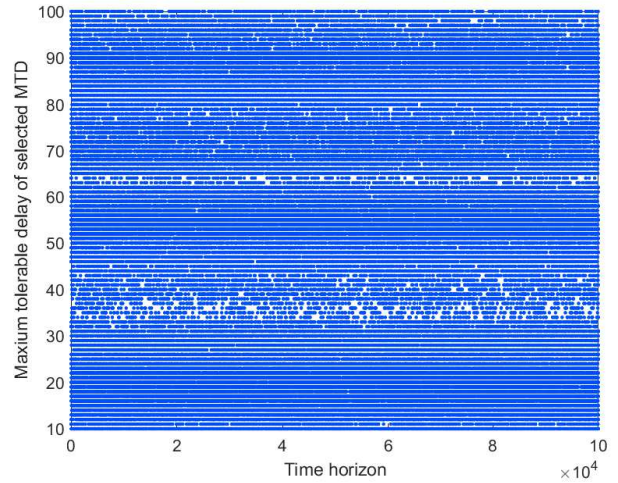
Fig. 4: Average maximum tolerable access delay of the selected MTDs.

access delay in the system, the average maximum tolerable access delay of a random allocation policy is compared to the sleeping MAB algorithm. From Fig. 4, we can see that the random allocation of the fast uplink grant achieves a delay that is equal to the average delay of the network. 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 three-fold less 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 random allocation. We set the maximum tolerable access delay to 100 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. Fig. 5(a) shows the capability of the sleeping MAB algorithm in optimizing the latency while providing fairness in the system. From Fig. 5(a), we can see that, in the beginning, the dots are uniformly distributed for all values of delay requirements, which means that the MTDs are randomly selected. However, after the learning, the intensity of the dots for MTDs with stricter latency requirements is much higher than that of the MTDs with larger delay requirement, which means that delay sensitive MTDs are scheduled more often. However, after learning period, the algorithm still keeps 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. Such a behavior shows that the algorithm is able to keep the



(a) Probabilistic sleeping MAB



(b) Random allocation

Fig. 5: Required access delay of the selected MTD at each time during the entire learning period. This plot shows how the sleeping MAB algorithm optimizes the system while providing fairness.

tradeoff between exploration and exploitation. From Fig. 5(b), we can see that a random scheduling algorithm selects the latency completely randomly all the time and the performance of the system is much worse than the proposed sleeping MAB.

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 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 the reward in a multi-armed bandit framework. Finally, we have proposed a sleeping MAB framework that achieves optimal regret for our problem formulation. Simulation results have shown that the proposed algorithm performs much better than a random allocation policy. To the best of our knowledge, this is the first paper that addresses the optimal allocation of the fast uplink grant for MTC.

REFERENCES

- [1] 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.
- [2] 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.
- [3] 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.
- [4] A. Ferdowsi and W. Saad, "Deep learning for signal authentication and security in massive Internet of Things systems," *CoRR*, vol. abs/1803.00916, 2018. [Online]. Available: <http://arxiv.org/abs/1803.00916>
- [5] 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.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] 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.
- [10] 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.
- [11] 3GPP, "Study on latency reduction techniques for LTE," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 36.881.
- [12] 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.
- [13] S. Ali, W. Saad, and N. Rajatheva, "A directed information learning framework for event-driven M2M traffic prediction," *IEEE Communications Letters*, vol. early access, 2018.
- [14] S. Ali, N. Rajatheva, and W. Saad, "Fast uplink grant for machine type communications: Challenges and opportunities," *arXiv preprint arXiv:1801.04953*, 2018.
- [15] 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>
- [16] 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.
- [17] R. S. Sutton, A. G. Barto *et al.*, *Reinforcement learning: An introduction*. MIT press, 1998.
- [18] 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.