Meta-Learning-aided Activity Detection for massive Machine-Type Communication Networks

Ivana Nikoloska, *Student Member, IEEE*, Nikola Zlatanov, *Member, IEEE*

Abstract

Signalling overhead proves to be a significant cause for concern in massive Machine-Type Communication (mMTC) networks. To alleviate this issue, in this paper, we propose a multiple-access protocol based on deep reinforcement learning (DRL), which is applicable to mMTC nodes with computational power and energy consumption constraints. In particular, for a system comprised of an access point (AP) and K mMTC nodes, a DRL agent placed at the AP is able to learn the communication activity of each mMTC node and then grant channel access to the nodes classified as active. The conventional Random Access (RA) procedure is only used as a back-up access mechanism for misclassified, unseen or new nodes in the cell. We propose leveraging expert knowledge in order to signifficantly reduce the amount of live data which needs to be acquired from the mMTC network for training. By using publicly available data-sets, we show that the proposed algorithm drastically decreases the probability of access failure compared to the conventional RA scheme. In fact, the worst possible performance of the proposed scheme is upper-bounded by the performance of the RA scheme. The proposed scheme leads to improvements in the latency, as well as in the energy consumption of the mMTC nodes.

I. Introduction

Given its ubiquitous coverage, the 5-th Generation of cellular networks (5G) has great potential to support diverse wireless technologies. These wireless technologies are expected to be crucial in next generation smart cities, smart homes, automated factories, automated health management systems, and many other applications, some of which can not even be foreseen today [1]. The

Ivana Nikoloska and Nikola Zlatanov are with the Department of Electrical and Computer Systems Engineering, Monash University, Melbourne, Australia. Emails: ivana.nikoloska@monash.edu, nikola.zlatanov@monash.edu

heterogeneous ecosystem of applications will result in different and often conflicting demands on the 5G radio and, as a result, the air-interface must be capable of supporting both high and low data rates, mobility, (ultra) low latency, as well as many different types of Quality of Service (QoS). In order for this QoS diversity to be achieved, improved medium access control (MAC) protocols have to be developed. As the state of the art MAC protocols for cellular networks have been designed and optimized to support primarily Human-to-Human (H2H) communication, these MAC protocols are not optimal for massive Machine-Type Communication (mMTC) devices, due to the unique characteristics of the mMTC traffic.

In mMTC networks, communication devices can operate autonomously with little or no human intervention. The amount of data which is generated by the mMTC nodes is usually small and the communication activity of these devices is heterogeneous [2]. The heterogeneous communication activity of the mMTC nodes makes any attempt to pre-allocate network resources to each node spectrally inefficient [3]. Currently, in order for a mMTC device to gain access to the channel and thereby transmit information, it has to perform the contention-based random access (RA) procedure [4], which requires a heavy exchange of signalling messages. Since the signalling messages have a similar length as the payload itself, the RA signalling overhead can easily become the bottleneck for mMTC networks. In addition, the amount of energy spent for signalling is comparable to the amount of energy spent for information transmission, making RA energy inefficient.

One promising research direction for MAC related problems in wireless communication is machine learning (ML). Reinforcement Learning (RL) is one of many ML paradigms, where agents mimic the human learning process and learn optimal strategies by trial-and-error interactions with the environment [5]. RL has been implemented in the development of MAC schemes for cognitive radios [6], where the authors have developed a RL based MAC scheme which allows each autonomous cognitive radio to distributively learn its own spectrum sensing policy. In [7], the authors add intelligence to sensor nodes to improve the performance of Slotted ALOHA. In addition, solving MAC problems with multi agent Deep Reinforcement Learning (DRL) has been proposed in [8],[9],[10]. Specifically, in [8], the authors propose a DRL MAC protocol for wireless networks in which multiple agents learn when to access the channel. A DRL MAC protocol for wireless networks in which several different MAC protocols co-exist has been studied in [9]. The authors of [10] have proposed a multi-agent, DRL based MAC scheme for wireless sensor networks with multiple frequency channels. Another distributed MAC scheme has been investigated in [11], where authors embed learning mechanisms to the mMTC nodes in order to control mMTC traffic and consequently reduce its impact on any cellular network. In [12], [13], [14], the authors reduce the access congestion by adapting the parameters of the access class barring (ACB) mechanism to different mMTC traffic conditions, via DRL. A different approach is proposed in [15], [16], where the authors investigate learning mechanisms to aid the MAC in mMTC networks, via dynamic access point (AP) selection schemes, in order to avoid overloading a single AP.

In spite of being highly promising, the proposed schemes in [7] - [11], do not necessarilly account for the severe device constraints in terms of energy availablity and computing power for running on-device inferences [17]. On the other hand, the schemes in [12] - [14] still rely on RA as a primary access mechanism. In this paper, we propose a DRL aided MAC protocol, whilst accounting for such on-device constraints. In particular, we consider a mMTC network comprised of an AP and K mMTC nodes, which transmit information towards the AP. In the proposed scheme, a single DRL agent placed at the AP is able to learn the communication activity of each mMTC node and grant transmission access to the active nodes. The standard RA scheme [4], is only used as a back-up access mechanism for misclassified, unseen or new nodes in the cell. In addition, we leverage expert knowledge from the available theoretical models and thereby reduce the amount of live data which needs to be acquired from the mMTC network for training. Our numerical results show that the proposed algorithm drastically decreases the probability of access failure, and implicitly decreases the energy consumption of the nodes. In fact, we show that the worst possible performance of the proposed scheme is upper-bounded by the performance of the conventional RA scheme. As the intelligence is concentrated at the AP, the nodes do not need signifficant computational power, or energy, for the on-device inferences, and thereby the proposed scheme can be deployed in cells with generic mMTC nodes that have limited computational capabilities.

The rest of the paper is organized as follows. Section II provides the network model. Section III presents the proposed DRL algorithm. In Section IV, we provide numerical evaluation, and a short conclusion concludes the paper in Section V.

II. System Model

We consider a mMTC network comprised of K mMTC nodes and an AP, as illustrated in Fig. 1. The distances from the AP to the nodes are uniformly distributed at random. The locations of the mMTC nodes are assumed to be fixed and not to change with time. The transmission time is assumed to be divided into T time slots of equal duration. The mMTC nodes are assumed to have conventional power sources and limited computational capabilities.

Each mMTC node is assumed to sporadically become active in order to sense its environment and generate data packets. When a mMTC node is active in time slot t, the objective of the node is to transmit its generated data packet to the AP. To do so, the mMTC node needs to be allocated time-frequency blocks, also known as resource blocks (RBs). In the considered network, the AP is assumed to have available N RBs. Since the packet size is equal for all nodes and is typically small for mMTC nodes, a RB is assumed to be sufficient for the transmission of any generated data packet. A conventional approach for an active mMTC node to obtain a RB is by employing the RA procedure [4]. However, due to the large signalling overhead, the RA protocol is inefficient for mMTC networks, as discussed previously. Due to its inefficiency, in this paper, we propose a new hybrid protocol that employs activity detection of the mMTC nodes based on DRL, where the RA procedure is used as a back-up access mechanism for misclassified or unseen nodes. The role of activity detection is to detect the set of active mMTC nodes in the network in order for the AP to grant RBs to the active nodes in each time slot. To this end, the AP hosts a DRL agent that interacts with the mMTC nodes in each time slot in order to learn to predict the active nodes in a given time slot. The interaction process between the DRL agent and the environment is described in detail in the following section.

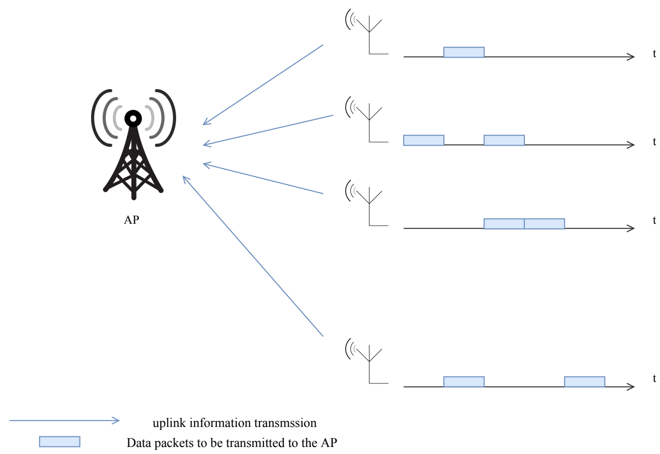

Fig. 1. Network model.

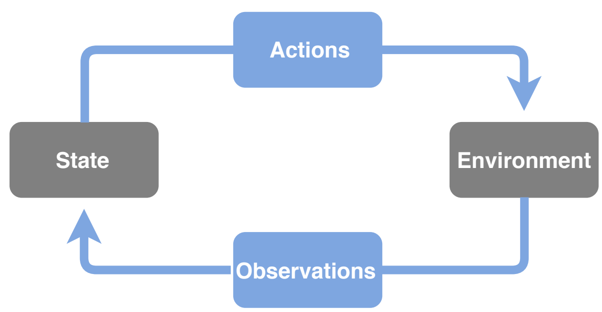

Fig. 2. Learning model.

III. Proposed DRL Algorithm

The DRL agent at the AP learns the communication activity of the mMTC nodes based on a trial and error bases by interacting with the environment, as shown in Fig. 2. The interaction process, which is repeated in each time slot, is described in Subsection III-A. In Subsection III-B, we show how expert knowledge can be leveraged in order to decrease the time needed for the DRL agent to be trained.

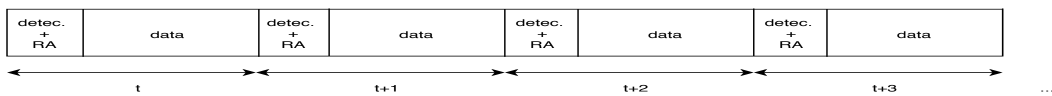

Fig. 3. Time frame.

A. Interaction Process Between the Agent and the Environment

- We define a state in time slot t, denoted by s_t. The state s_t represents a set comprised of the nodes which have been active during the previous t_h time slots, i.e., $t - t_h, ..., t - 1$, where t_h denotes the history that the agent "remembers".
- We define an action in time slot t, denoted by a_t. The action a_t represents a set of nodes which have been classified as active by the agent in time slot t. The action is taken based on a decision policy π, which maps states into actions, i.e., given state s_t, the agent takes action a_t in time slot t according to the policy π.
- Based on a_t, the following scenarios can be distinguished:
  - If a node has been correctly classified as active, then the respective node will be granted a RB to transmit to the AP, see Fig 3.
  - If a node has been misclassified as active and as a result has been granted a RB to transmit to the AP, then the misclassified node will stay silent since the misclassified node will not have data to transmit to the AP.
  - If a node has been correctly classified as inactive the node is not granted a RB and the node stays silent.
  - If a node has been misclassified as inactive and as a result it has not been granted a RB, the misclassified node attempts to gain access to a RB using the RA procedure in [4]. To

this end, the node transmits a preamble to the AP using a Random Access Opportunity (RAO) to initiate the RA procedure.

- Next, the agent makes an observation of the network. Thereby, the agent observes:
  - If the AP has received data from the nodes classified as active.
  - If the AP has not received data from the nodes classified as active.
  - If the AP has received RA preambles from any of the nodes classified as inactive.
- Based on the taken action a_t and the successive observation, the agent receives a reward in time slot t, denoted by r_{t+1}, such that if the agent correctly classifies all K nodes, then $r_{t+1} = K$, else $r_{t+1} = 0$, i.e.,

$$r_{t+1} = \begin{cases} K, & \text{iff all nodes are correctly classified,} \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

- Based on the observation of the network in time slot t, the agent transitions to state s_{t+1} and the whole process described above is repeated.

We use the standard DRL framework for training, which is already available in the literature. Briefly, the goal of the agent is to maximize the accumulated return, given by $R_t = r_{t+1} + \gamma R_{t+1}$, where γ is referred to as the discount factor and is used for weighting future rewards [5]. The value of each action (i.e., the return from taking action a_t in state s_t) is described by the action-value function $Q(s_t, a_t)$. To avoid estimating separately the values of all actions in all states, i.e., $Q(s_t, a_t)$, $\forall(s_t, a_t)$, one can learn a parameterized action value function, $q(s_t, a_t; \theta)$ by using a neural network with parameters θ as proposed in [18]. The updates on $q^*(s_t, a_t)$ are performed using the following iterative expression [5]

$$\begin{aligned} q^*(s_t, a_t) \leftarrow & q^*(s_t, a_t) \\ & + \alpha[r_{t+1} + \gamma \max_a q^*(s_{t+1}, a) - q^*(s_t, a_t)], \end{aligned} \tag{2}$$

where $0 < \alpha < 1$ denotes the learning rate. The algorithm converges to the optimal action value function, $Q^*(s_t, a_t)$, as the number of iterations goes to infinity. The reader is kindly referred to [5] and [18], the references therein and the references in this paper for further details.

In the spirit of reproducible science, the codes for the algorithm will be made available on [19].

B. Learning to Learn from Expert Knowledge

In order for the training process of the agent to be succesful, the agent needs to obtain a sufficient number of ”live” data samples from the interaction with the nodes. In fact, a deep neural network (NN) can always achieve the highest inference accuracy, so long as sufficiently large amount of data samples are fed for training [20], [5]. Let the live data sample at time slot t be represented by the quintuples $(s_t, a_t, r_{t+1}, s_{t+1})$, obtained from the agent's interaction with the nodes, as per Subsection III-A. In practice, the acquisition of sufficient number of live samples can be impractical and ultimately prohibitive, as it is going to take an excessively long amount of time to acquire the data. In these cases, transfer learning can be leveraged in order to accelerate the training process. Transfer learning is a recent trend in the ML community, where available prior knowledge about the considered problem stemming from theoretical models is embedded in the neural networks [21]-[22]. Transfer learning (alternatively learning to learn or meta-learning), dramatically reduces the number of live data samples that are needed for the training process to be successful. Transfer learning allows the neural network to be trained using a priorly known mismatched model, and then to refine the result with fewer live data samples.

The mMTC networks research community has provided many theoretical models for the activity of the mMTC nodes, such as those in [2], [23]. We chose to leverage the model in [23], where the authors use a Coupled Markov Modulated Poisson Process (CMMPP) to capture the activity of the nodes in a mMTC cell. The CMMPP model captures both regular and alarm reporting, as well as the correlated activity behavior among the nodes. In addition, the model provides the aggregate number of arrivals per node during a given time period. Hence, we use the CMMPP model in [23] to train the DRL agent at the AP. To this end, we first synthetize artificial activity patterns of the mMTC nodes, according to the CMMPP model. Let the artificial activity of the mMTC nodes be of length T_m slots, where the subscript m is used to distinguish between slots in which meta-interaction is performed (i.e., slots in which the agent ”interacts” with the artificial arrivals), and slots in which the actual interaction with the nodes is performed.

Thereby, in each time slot t_m, the agent interacts with and learns from the artificial arrivals patters as per Subsection III-A.

Once the prior knowledge has been transferred, we correct the mismatch between the CMMPP model and the activity in the actual mMTC cell by using a small number of actual live samples from the mMTC nodes. To this end, the agent is allowed to interact with the mMTC nodes to continue the training process. The obtained weights for the neural network during the training with artificial arrivals are used as initial weights, and the rest of the interaction and training processes do not deviate from Subsection III-A.

IV. Numerical Results

In this section, we compare the performance of the proposed DRL algorithm with the RA scheme in [4]. In the following, we first present the data sets in Section IV-A and the hyper parameters of the proposed algorithm are given in Section IV-B. The RA scheme used as benchmark is given in more detail in Section IV-C, and the numerical results are given in Section IV-D.

A. Data Sets

For the purposes of this section, the live data samples are drawn from the publicly available data sets in [24], [25], [26], and [27]. We assume that all nodes operate during the same time period and in the same mMTC cell. We use the provided meta-information to determine when the nodes have been active. The data sets in [24]-[27] are comprised of nodes which have different reporting intervals and different arrival rates [2]. In particular, the reporting intervals in this case range from few seconds for some nodes up to 1 hour for other. In total our mMTC cell is comprised of $K = 222$ nodes, which report 35448 data arrivals during one hour.

B. Algorithm Hyper-Parameters

The architecture of the neural network used in the proposed DRL algorithm is consisted of a three-layer, fully connected neural network with 6000 hidden units in the hidden layer. The activation functions for the neurons are ReLU functions [28], given by

TABLE I

ALGORITHM HYPER-PARAMETERS

Parameter	Value
No. of hidden layers	3
No. of hidden units	6000
Experience-replay memory size	200
Experience-replay mini-batch size	32
Discount factor γ	0.05
Learning rate α	0.001
ϵ	1 to 0.01

$$f(x) = \begin{cases} 0, & \text{for } x < 0, \\ x, & \text{for } x \geq 0. \end{cases} \quad . \tag{3}$$

Next, let $\mathcal{E}$ denote the experience set (see [18]), which is comprised of 32 experience samples (i.e., quintuples $(s_t, a_t, r_{t+1}, s_{t+1})$), that are randomly chosen from the memory buffer that can store the last 200 experience samples. The discount factor is set to $\gamma = 0.05$ and the learning rate is set to $\alpha = 0.001$. The exploration-exploitation trade-off [28] is controlled via the ϵ-greedy algorithm, where ϵ_t is decreasing from $\epsilon_t = 1$ to $\epsilon_t = 0.01$ as

$$\epsilon_{t+1} \leftarrow \epsilon_t * \epsilon_{dec.}, \tag{4}$$

where $\epsilon_{dec} = 0.995$. The parameters of the proposed algorithm are summarized in Table I.

C. Contention Based RA Scheme

The RA scheme is presented in Fig. 4 and is explained in the following.

- In the first step, the node randomly selects a preamble reserved for contention-based RA by the AP and the node transmits the corresponding ZC sequence to the AP.

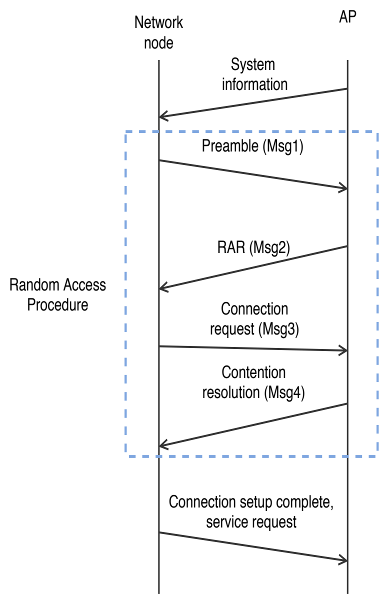

Fig. 4. Random access procedure.

- Upon the detection of the preambles, a random access response (RAR) is transmitted by the AP containing uplink grants (i.e., time-frequency resource block) for the successfully detected preambles and temporary identifiers (C-RNTI) for the detected preambles.
- In the third step, after receiving the corresponding RAR, the node transmits its allocated device identifier (C-RNTI) in the allocated time-frequency resource block.
- In the fourth step, the AP transmits a contention resolution message for the successfully decoded device identifiers, which completes the RA procedure.

If several mMTC nodes transmit the same preamble in step one, a collision will occur. We assume that the packets of those nodes are dropped. The maximum number of available preambles is 64, however some of the preambles are reserved by the AP, hence we assume that there are 54 preambles available for contention and the AP has 50 RBs at its disposal.

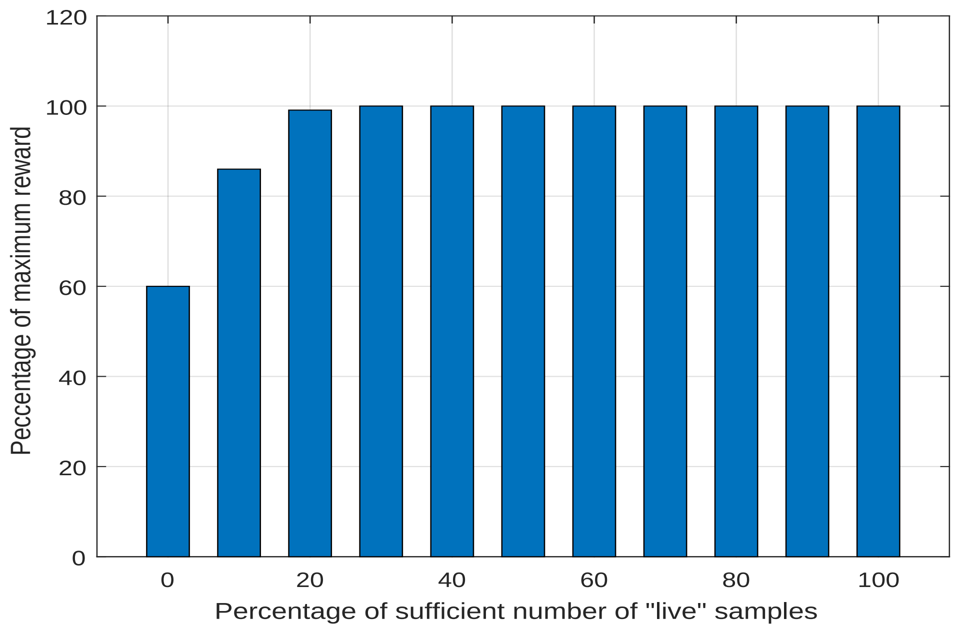

Fig. 5. Comparison between transfer learning and data driven learning. The percentage of artificial samples can be found as $100 - X$, where X denotes the percentage of sufficient number of live samples.

D. Performance Evaluation

1) Proof of Concept: To illustrate the benefits of transfer learning, we present Fig. 5, where the percentage of maximum possible reward is illustrated as a function of the percentage of sufficient live samples. The sufficient number of live samples is defined as the number of live samples needed for the agent to obtain the maximum possible reward, and thereby achieve the maximum possible inference accuracy. The maximum reward is defined as the reward obtained by using 100% of live samples. Fig. 5 shows that the maximum reward can be obtained by using 20% of live samples and 80% of artificial samples. Note that, using an insufficient number of live samples, and no artificial samples, leads to a reward that is significantly lower than the maximum possible reward, as the agent does not have enough data for the training process. In addition, the obtained reward is significantly lower if only artificial samples, without any live samples, are used which is a consequence of the mismatch between the artificial model and the activity in the mMTC cell. For the considered model, optimal performance can be achieved by using 20% of live samples and 80% of artificial samples from the theoretical model. This in turn significantly decreases the time required for the agent to be trained, i.e., by up to 80% in our case.

2) Access Failure: In both the proposed hybrid scheme and the conventional RA scheme, an access failure in time slot t occurs as a consequence of preamble collision. A preamble collision occurs if two or more mMTC nodes, which attempt RA, choose the same preamble to transmit in time slot t. In the proposed DRL scheme, a preamble collision occurs if two or more nodes are miscalssified in time slot t and those nodes attempt RA as a back up and end up choosing the same preamble to transmit.

In Fig. 6, we illustrate the ratio between the number of collisions and total number of RA attempts in each time slot during a period of 1 hour (3600 s). In the RA case, the peaks correspond to time slots with many data arrivals, which results in an increase of the number of preamble collisions. As for the proposed DRL+RA procedure, the ratio is much lower than in the RA case irrespective of the number of data arrivals. This is a consequence of the fact that the agent is able to correctly classify some of the active nodes, and therefore the number of nodes that are misclassified as inactive and attempt the RA procedure is much lower compared to the pure RA scheme. In Fig. 7 we illustrate the average ratio between the number of collisions and total number of RA attempts for different number of nodes in the cell, up to the maximum number of nodes $K = 222$. As the number of nodes increases, so does the average ratio between the number of collisions and total number of RA attempts. Again, the average ratio is much lower for the hybrid scheme, for any number of nodes in the cell.

V. Conclusion

In this paper, we proposed a DRL based activity detection algorithm for a network comprised of K mMTC nodes and an AP. The AP hosts an intelligent agent, which learns to detect the active mMTC nodes in the network in each time slot in order to grant them access to transmit to the AP. The standard RA scheme is reduced to a back-up access mechanism for potentially misclassified, unseen or new nodes in the cell. By using publicly available data sets, we show

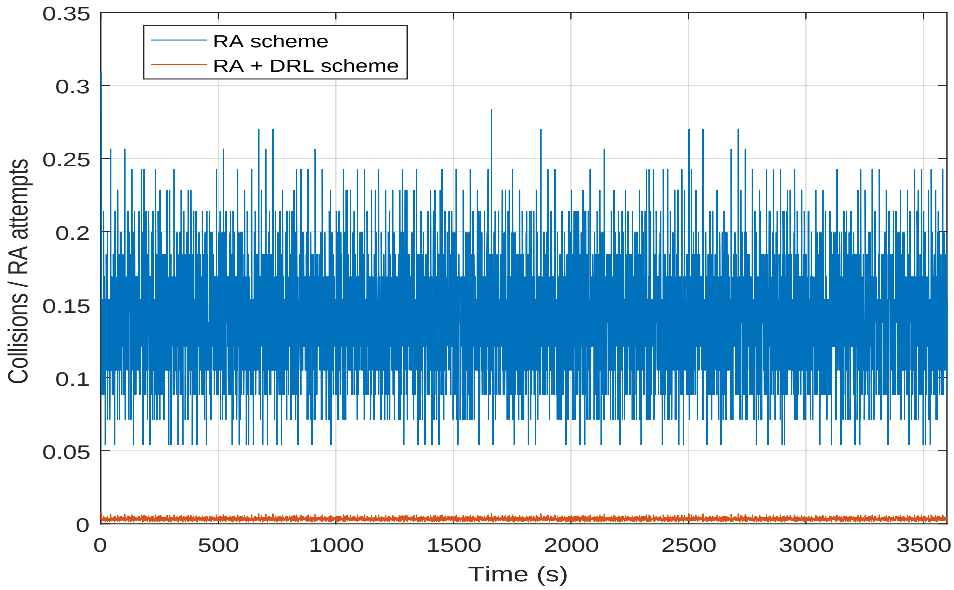

Fig. 6. Ratio between the number of collisions and total number of RA attempts in each time slot during one hour.

signifficant improvements in terms of access failure, when the proposed DRL is implemented, compared to the pure RA scheme.

REFERENCES

[1] R. Ratasuk, A. Prasad, Z. Li, A. Ghosh, and M. A. Uusitalo, "Recent advancements in m2m communications in 4g networks and evolution towards 5g," in *2015 18th International Conference on Intelligence in Next Generation Networks*, Feb 2015.

[2] D. W. K. N. V. W. Wong, R. Schober and L.-C. Wang, *Key Technologies for 5G Wireless Systems*. Cambridge, U.K.: Cambridge Univ. Press, 2017.

[3] 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 and Tutorials*, vol. 16, no. 1, pp. 4–16, 2014.

[4] 3GPP, "Medium access control (mac) protocol specification," *3GPP TS 38.321 V0.0.3*, May 2017.

[5] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press Cambridge, 1998, vol. 1, no. 1.

[6] M. Bkassiny, S. K. Jayaweera, and K. A. Avery, "Distributed reinforcement learning based mac protocols for autonomous cognitive secondary users," in *Wireless and Optical Communications Conference (WOCC), 2011 20th Annual*. IEEE, 2011, pp. 1–6.

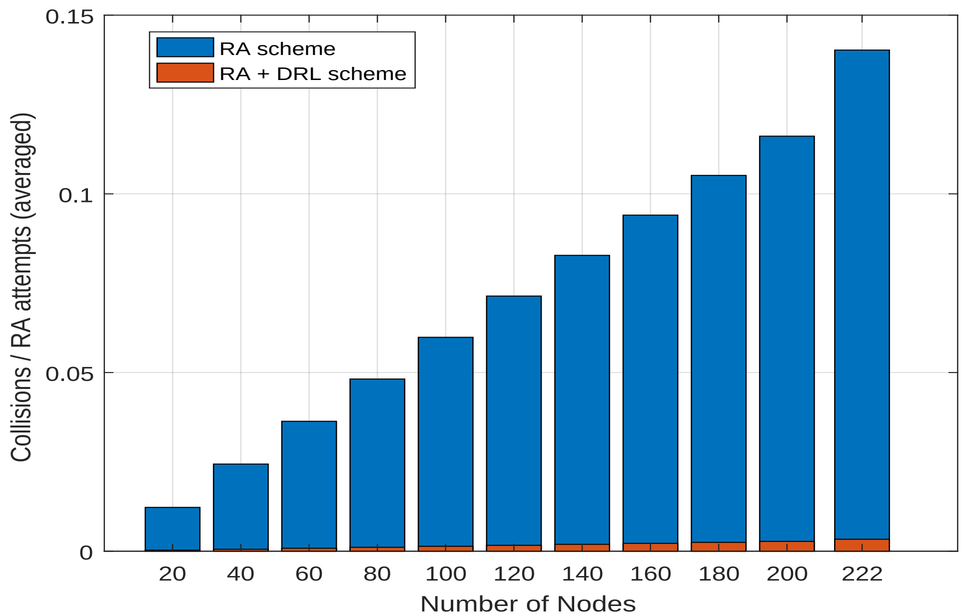

Fig. 7. Average ratio between the number of collisions and total number of RA attempts as a function of the number of nodes in the cell.

[7] Y. Chu, P. D. Mitchell, and D. Grace, "Aloha and q-learning based medium access control for wireless sensor networks," in *Wireless Communication Systems (ISWCS), 2012 International Symposium on*. IEEE, 2012, pp. 511–515.

[8] O. Naparstek and K. Cohen, "Deep multi-user reinforcement learning for dynamic spectrum access in multichannel wireless networks," *arXiv preprint arXiv:1704.02613*, 2017.

[9] Y. Yu, T. Wang, and S. C. Liew, "Deep-reinforcement learning multiple access for heterogeneous wireless networks," *arXiv preprint arXiv:1712.00162*, 2017.

[10] S. Wang, H. Liu, P. H. Gomes, and B. Krishnamachari, "Deep reinforcement learning for dynamic multichannel access," in *International Conference on Computing, Networking and Communications (ICNC)*, 2017.

[11] L. M. Bello, P. Mitchell, and D. Grace, "Application of q-learning for rach access to support m2m traffic over a cellular network," in *European Wireless 2014; 20th European Wireless Conference*. VDE, 2014, pp. 1–6.

[12] J. Moon and Y. Lim, "Access control of mtc devices using reinforcement learning approach," in *2017 International Conference on Information Networking (ICOIN)*. IEEE, 2017, pp. 641–643.

[13] ——, "A reinforcement learning approach to access management in wireless cellular networks," *Wireless Communications and Mobile Computing*, vol. 2017, 2017.

[14] L. Tello-Oquendo, D. Pacheco-Paramo, V. Pla, and J. Martinez-Bauset, "Reinforcement learning-based acb in lte-a networks for handling massive m2m and h2h communications," in *2018 IEEE International Conference on Communications (ICC)*.

IEEE, 2018, pp. 1–7.

[15] Y.-J. Liu, S.-M. Cheng, and Y.-L. Hsueh, "enb selection for machine type communications using reinforcement learning based markov decision process," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 12, pp. 11 330–11 338, 2017.

[16] A. Mohammed, A. S. Khwaja, A. Anpalagan, and I. Woungang, "Base station selection in m2m communication using q-learning algorithm in lte-a networks," in *2015 IEEE 29th International Conference on Advanced Information Networking and Applications*. IEEE, 2015, pp. 17–22.

[17] T. Park, N. Abuzainab, and W. Saad, "Learning how to communicate in the internet of things: Finite resources and heterogeneity," *IEEE Access*, vol. 4, pp. 7063–7073, 2016.

[18] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, p. 529, 2015.

[19] "Ra x drl," *https://github.com/ikoloska*.

[20] A. A. L. S. Becker, Y. Zhang, "Geometry of energy landscapes and the optimizability of deep neural networks," *arXiv preprint arXiv: 1808.00408*, 2018.

[21] T. Inoue, S. Choudhury, G. De Magistris, and S. Dasgupta, "Transfer learning from synthetic to real images using variational autoencoders for precise position detection," in *2018 25th IEEE International Conference on Image Processing (ICIP)*, Oct 2018, pp. 2725–2729.

[22] A. N. C. Kim, E. Variani and M. Bacchiani, "Efficient implementation of the room simulator for training deep neural network acoustic models," 2019.

[23] G. C. Madueño, Č. Stefanović, and P. Popovski, "Reliable and efficient access for alarm-initiated and regular m2m in ieee 802.11 ah systems," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 673–682, 2016.

[24] D. Murray and L. Stankovic, "Refit," *available on https://pureportal.strath.ac.uk/en/datasets/refit-electrical-load-measurements-cleaned*, 2016.

[25] M. P. L. M. G. D. F. S. De Vito, E. Massera, "On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario," *Sensors and Actuators B: Chemical*, vol. 129, no. 2, pp. 750 – 757, 2008.

[26] N. Batra, O. Parson, M. Berges, A. Singh, and A. Rogers, "A comparison of non-intrusive load monitoring methods for commercial and residential buildings," *available on arXiv:1408.6595*, 2014.

[27] "Individual household electric power consumption data set."

[28] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016, vol. 1.