Multi-Agent Proximal Policy Optimization for Dynamic Multi-Channel URLLC Access

Benoît-Marie Robaglia†, Marceau Coupechoux†, Dimitrios Tsilimantos*
†LTCI, Telecom Paris, Institut Polytechnique de Paris
*Advanced Wireless Technology Lab, Paris Research Center, Huawei Technologies Co. Ltd.

***Abstract*—This work addresses the challenge of Dynamic Multi-Channel Access (DMCA) in the context of Ultra Reliable Low Latency Communications (URLLC), a framework subjected to notably stringent constraints, required by numerous Internet of Things (IoT) applications across various sectors. We introduce a theoretically grounded approach, leveraging Deep Multi-Agent Reinforcement Learning (MARL) to tackle this problem. While prior research has not fully addressed the DMCA problem in URLLC networks under time-varying heterogeneous channels and traffic profiles, nor provided robust theoretical guarantees in the multi-agent context, this paper adapts the recent theoretical framework of Trust Region Policy Optimization (TRPO) in MARL to meet the specific challenges and requirements of the URLLC-DMCA problem. Specifically, we introduce Multi Channel Access - Proximal Policy Optimization (MCA-PPO), a MARL algorithm that benefits from theoretical guarantees and effectively handles the partial observability and the combinatorial nature of the DMCA challenge. We validate the superiority of our proposed method across a variety of heterogeneous scenarios, in terms of traffic models and system parameters, and show that we outperform the traditional multiple access benchmark and learning algorithms.**

***Index Terms*—Distributed Multiple Access, Deep Multi-Agent Reinforcement Learning, Internet of Things, URLLC.**

I. Introduction

The proliferation of industrial automation, underpinned by the Internet-of-Things (IoT), is reshaping the landscape of next-generation wireless networks [1]. Domains like factory automation, motion control and vehicle networks are clear examples of this transition. These applications present stringent communication constraints, characterized by the Third Generation Partnership Project (3GPP) standard [2] as Ultra Reliable Low Latency Communications (URLLC). This paradigm not only demands high reliability, but also insists on minimal latency and has to deal with strict deadlines where a packet is lost beyond this delay. However, conventional multiple access protocols struggle to meet these URLLC standards, especially on the uplink, largely due to the four handshake protocol that exacerbates signalling overhead and latency. An effective method to improve reliability and to reduce the transmission latency is the multi-channel technology [3]. However, this approach brings its own set of complexities, as selecting the optimal channels can be challenging, especially in time-varying environments. In order to overcome this limitation, we present a theoretically justified approach, using Deep Multi-Agent Reinforcement Learning (MARL) for the dynamic multichannel access (DMCA) problem.

A. Related Work

Traditional random access protocols have been extended to the DMCA problem in order to meet the URLLC requirements. For example, the work of [4] proposes a multi-channel ALOHA-type grant-free (GF) algorithm. However, the authors assume that the users are aware of all the channel states and thus only good channels are selected. Furthermore, this approach does not adapt to the dynamics of the environment and is therefore sub-optimal.

Deep Reinforcement Learning (DRL) approaches have also been considered to tackle the DMCA problem. The most natural way to extend single-agent DRL to multi-agent DRL is *independent learning* (IL) [5]. The idea is to equip each agent by a single agent DRL algorithm and to consider the other agents as part of the environment. The most notable examples of IL applied to the DMCA problem are the work of [6] that introduces an actor critic algorithm for DMCA, the P-DDPG algorithm [7], where the authors predict the channel state with a Channel Prediction Module and use this predicted value as prior information for the DRL agent. Some other works tackle the MCA aspect without the time-varying aspect of the channels. Some other works address the Multi-Channel Access (MCA) aspect without considering the time-varying nature of the channels. For instance, the study by [8] introduces a branching architecture that enables users to access multiple channels within a single frame, while [9] integrates Q-learning with a Recurrent Neural Network (RNN) to address the MCA challenge in heterogeneous networks. However, these approaches do not take into account the dynamic nature of channels and the stringent deadlines associated with URLLC, and they also lack the ability to opportunistically access the channel. In addition, these studies assume a single multiple-frequency channel, i.e. all users observe the same channel state which is not realistic in a wireless context. Regarding their RL model, they do not address the theoretical limitations of IL such as the non-stationarity caused by the concurrent learning of all agents and do not provide any convergence guarantees to their approaches.

Futhermore, IL has been applied to Dynamic Spectrum Access (DSA) where the channel state is good when it is not used by a primary user and bad otherwise. For instance, the work of [10] proposes Q-learning based agents equipped with a RNN similarly to [9] to maximize the network utility in DSA. One alternative to address the limitations of IL is *Cen-*

tralized Training with Decentralized Execution (CTDE) where agents are allowed to exchange information during training in order to reduce the issues of non-stationarity. The authors of [11] apply CTDE to the DSA problem. However, the users adopt the listen-before-talk mechanism to access the spectrum which is not suitable to URLLC networks due to the strigent latency requirements and the small packet size. Besides, even if CTDE approaches can help agents learn more effectively by leveraging global information, convergence guarantees are still an active area of research and agents do not necessarily converge to an optimal or even stable policies.

To summarize, prior research has not tackled the DMCA problem in URLLC networks under time-varying heterogeneous channels and traffic profiles. In addition, existing studies have predominantly explored off-policy algorithms (such as Q-learning or actor-critic approaches) for the DMCA problem and these methods suffer from theoretical guarantees in the multi-agent context. In contrast, Trust Region Policy Optimization (TRPO) techniques have showcased superior performance across a variety of tasks in both single-agent [12] and multi-agent DRL, be it in a IL paradigm with independent PPO [13] or a CTDE framework with MAPPO [14]. Yet, those approaches still lack the *monotonic improvement guarantees*, characterizing the TRPO methods. One reason for their good empirical success may be the parameter sharing and homogeneous agents. Recent work of [15], provides a TRPO algorithm with monotonic improvement guarantees within the multi-agent framework. In this paper, we adapt their theoretical model to address the URLLC-DMCA problem, tailoring the approach to meet its specific challenges and requirements.

B. Contribution and outline

Our contributions can be summarized as follows:

- We formulate a DMCA problem in a URLLC network with heterogeneous users that need to deliver a short packet within a strict deadline on the uplink as a Decentralized Partially Observable Markov Decision Process (Dec-POMDP). To the best of our knowledge, this is the first study using this DMCA problem formulation.
- We present MCA-PPO, a theoretically-robust solution designed to address the complexities inherent in the DMCA challenge. This method stands out by leveraging the theoretical results from [15]: it incorporates the monotonic improvement guarantee, ensuring a consistent enhancement in performance with each policy update. Besides, MCA-PPO combines three technical contributions: 1) It handles the combinatorial action space, a complexity arising from the selection of subchannel subsets in each frame within the DMCA framework, with a branching policy network. 2) It is able to leverage historical observation-action data and effectively deals with partial observability thanks to a RNN. 3) We introduce a communication protocol specifically designed to satisfy the theoretical constraints outlined in [15], thereby achieving theoretical guarantees.

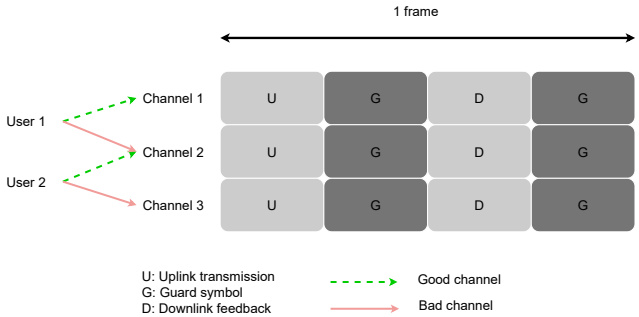


Fig. 1: System model and Slot Structure

- Finally, we validate the superiority of our proposed methods on different scenarios. Our results consistently surpass traditional multiple access benchmarks and established off-policy DRL algorithms.

Notations: For a finite set X, $\Delta(X)$ denotes the set of all probability distributions over X. The indicator function is denoted $\mathbb{1}\{\cdot\}$. The matrices are written in bold upper case and the vectors in bold lower case. $[\cdot]$ refers to the modulo operator and $k_{1:m}$ denotes an ordered subset $k_1, \ldots, k_m$ of $[\![1, K]\!]$ and $-k_{1:m}$ refers to its complement.

II. System Model

A. Network model

We consider a time-slotted wireless network with K users communicating with a Base Station (BS) over N time-varying orthogonal wireless channels on the uplink. Time is divided into radio frames, and every frame is divided into four time-slots (see Fig. 1). This division represents the minimum time required for the processes of transmitting and acknowledging. We further assume that a single packet can be transmitted within the duration of one time slot. At every frame, a user can select one or several channels to send one or several replicas of its packet along with a pilot to facilitate the decoding process.

Finally, at the end of every frame, the BS broadcasts a feedback message $\boldsymbol{\alpha} \in \{-1, 0, 1\}^N$ to the users detailing the outcomes of the transmissions on every channel. In particular, they receive an ACK ($\alpha^n = 1$) if a packet has been successfully transmitted with channel n, a NACK message ($\alpha^n = -1$) if the BS did not manage to decode the packet, and an IDLE message ($\alpha^n = 0$) to indicate that the channel was idle. Operating within a Time Division Duplexing (TDD) framework, our system is able to utilize channel reciprocity, a principle asserting that the wireless channel characteristics are symmetric, meaning that they provide identical responses in both forward and reverse communication directions [16]. By leveraging this principle, the feedback message from the BS enables users to ascertain the state of their respective communication channels during a given frame.

B. Traffic model

Devices initiate packet generation following either a deterministic or probabilistic traffic.

Periodic traffic: In this model inspired by [17], each device k generates packets periodically every N_p radio frames with probability q_k. Devices are not synchronous, and are assigned an offset parameter $f_k \in [0, N_p]$ such that, at every radio frame $t \geq 0$, the probability for a device k of generating a new packet is: $\bar{q}_k(t|f_k, q_k, N_p) = \mathbb{1}_{\{t[N_p]=f_k\}} q_k$.

Aperiodic traffic: This model comes from 3GPP specifications and is based on the File Transfer Protocol (FTP) model 3 defined in [18], but with a fixed size for each packet. At every device k, packets are generated according to a Poisson process of rate λ_k.

In order to model the strict latency constraint of URLLC networks, each user k needs to deliver its packet within an individual air interface latency constraint, δ^k; if a packet has not been transmitted before δ^k slots after its arrival in the buffer, it is discarded. If a transmission fails (due to a collision or a decoding error), the packet can be retransmitted up until its deadline is met.

We assume that the packet queue operates on a "first come, first served" principle. For any device k, we define the buffer status at a time t by the vector $\boldsymbol{b}_t^k \in \mathbb{N}^{\delta^k}$, where $b_t^{k,d} = i$ indicates that device k has i packets with a deadline duration of d at time t. The matrix of the buffers of all devices at time t is represented by $\boldsymbol{B}_t$. The buffer status of a device k transits as follows: (a) Successfully decoded packets are removed from the buffer (b) Other packets see their time-to-deadline decreased by one. Expired packets are removed from the buffers; (c) New generated packets enter the buffer with a deadline δ_k.

C. Channel model

Every channel between a user and the BS follows the Gilbert-Elliot channel model [19]: at any given slot each channel can be in one of two states: a good channel state, ensuring successful transmission, or a bad channel state, leading to a transmission failure.

The state switching pattern is represented by a Markov chain. For each slot t, the channel state is represented by $\boldsymbol{\eta_t} \in \{0,1\}^{K \times N}$ where $\eta_t^{k,n}$ represents the state of the n-th channel of user k in slot t. We assume that each channel state can only change at the beginning of each frame and remains constant during the frame. The channel n of a user k evolves according to the transition matrix: $\begin{pmatrix} 1-p_{k,n} & p_{k,n} \\ \tilde{p}_{k,n} & 1-\tilde{p}_{k,n} \end{pmatrix}$ where $p_{k,n}$ is the probability that the state of the n-th channel of k changes from bad to good and $\tilde{p}_{k,n}$ the probability that it switches from good to bad.

We adopt a collision channel model: when the BS receives a single packet via the good channel resource n, it can successfully decode it. When several users transmit on the same channel during the same frame, a collision occurs and no packet is delivered whatever their respective channel state.

III. Problem formulation

This problem can be formulated by a Decentralized Partially Observable Markov Decision Process (Dec-POMDP) [20]. A Dec-POMDP is a cooperative Markov Game where agents take decisions based on individual observations about the environment. In our problem, the Dec-POMDP elements are defined as follows:

- The state $\boldsymbol{s}_t \in \mathcal{S}$ is the concatenation of the current buffer status and the current channel state i.e. $\boldsymbol{s}_t = (\boldsymbol{B}_t, \boldsymbol{\eta}_t)$.
- Each user k observes $\boldsymbol{o}_t^k$, which is made of its own buffer $\boldsymbol{b}_t^k$, the last channel observation $\boldsymbol{\eta}_{t-1}^k$ and the last ACK/NACK from the BS $\boldsymbol{\alpha}_{t-1}$: $\boldsymbol{o}_t^k = (\boldsymbol{b}_t^k, \boldsymbol{\eta}_{t-1}^k, \boldsymbol{\alpha}_{t-1})$.
- At every slot t, each agent k selects an action $\boldsymbol{a}_t^k \in \{0,1\}^N$ based on its observation $\boldsymbol{o}_t^k$ and policy $\pi^k(\cdot|\boldsymbol{o}_t^k)$ where $a_{n,t}^k = 1$ if agent k transmits a packet using the channel n. We denote the global action $\boldsymbol{A}_t \in \{0,1\}^{K \times N}$, this is the concatenation of all individual actions.
- The next state of the system is drawn with the transition function $\mathcal{T} : \mathcal{S} \times \mathcal{A} \mapsto \Delta(\mathcal{S})$. This function follows the buffer and channel state's dynamics defined in Sections II-B and II-C.
- Finally, we define the reward r_t at frame t as the total number of successful transmissions across all channels.

$$r_t = \sum_{n=1}^{N} 1_{\{\alpha_t^n = 1\}} \tag{1}$$

Each user k aims to optimize the following objective:

$$\mathbb{E}_{\boldsymbol{s}_{t+1} \sim \mathcal{T}, a^{-k} \sim \pi^{-k}} \left[\sum_{t=0}^{T} \gamma^t r_t | a_t^k \sim \pi^k(\cdot|\boldsymbol{o}_t^k), \boldsymbol{s}_0 \right] \tag{2}$$

where $\gamma \in (0, 1]$ is the discount factor that allows the agents to balance immediate rewards with future ones.

In order to compare the performance of the MAC protocols for the URLLC problem, we define a *URLLC score* as the ratio between the number of packets delivered before expiration and the number of generated packets.

IV. Multi-Agent Deep Reinforcement Learning

A. Policy Gradient for Multi-Agent Reinforcement Learning

Trust Region Policy Optimization (TRPO) algorithms [21], have been introduced in single-agent RL in order to ensure the *monotonic improvement* property, which guarantees a non-decreasing performance of the policy at every iteration.

This property has been extended to MARL in [15] thanks to the Multi-Agent Advantage Decomposition lemma:

Lemma 1 (Multi-Agent Advantage Decomposition [15])**.** *In any cooperative Markov game, given a joint policy $\boldsymbol{\pi}$, for any state $\boldsymbol{s}$, and any agent subset $k_{1:m}$, the global advantage $A_{\boldsymbol{\pi}}^{k_{1:m}}(\boldsymbol{s}, \boldsymbol{a}^{k_{1:m}})$ can be decomposed into a summation of each agent's local advantages $A_{\boldsymbol{\pi}}^{k_j}$:*

$$A_{\boldsymbol{\pi}}^{k_{1:m}}(\boldsymbol{s}, \boldsymbol{a}^{k_{1:m}}) = \sum_{j=1}^{m} A_{\boldsymbol{\pi}}^{k_j}(\boldsymbol{s}, \boldsymbol{a}^{k_{1:j-1}}, a^{k_j}) \tag{3}$$

Lemma 1 gives us a methodology for the agents to update their local policies while guaranteeing monotonic improvement. Indeed, let agents take actions sequentially by following an arbitrary order $k_{1:K}$. Agent k_1 takes action $\bar{a}^{k_1}$ such that $A^{k_1}_{\boldsymbol{\pi}}(\boldsymbol{s}, \bar{a}^{k_1}) > 0$. Agent k_2 selects action $\bar{a}^{k_2}$ such that $A^{k_2}_{\boldsymbol{\pi}}(\boldsymbol{s}, \bar{a}^{k_1}, \bar{a}^{k_2}) > 0$. For the remaining $m = 3, \dots, K$, each agent k_m selects an action $\bar{a}^{k_m}$ such that $A^{k_m}_{\boldsymbol{\pi}}(\boldsymbol{s}, \bar{a}^{k_{1:m-1}}, \bar{a}^{k_m}) > 0$. Thus, Lemma 1 guarantees that the global advantage $A_{\boldsymbol{\pi}}(\boldsymbol{s}, \boldsymbol{A})$ is positive and therefore the performance is guaranteed to improve.

To summarize, the work of [15] demonstrates that the monotonic improvement property holds in Multi-Agent TRPO when each agent updates its local policy sequentially and taking into account all previous agents' updates.

B. Multi-Channel Access Proximal Policy Optimization

The Proximal Policy Optimization (PPO) algorithm [22] is a TRPO algorithm that leverages the principle of limiting the magnitude of policy updates, using first-order optimization techniques only. In a multi-agent setting where the monotonic improvement property holds, each agent k_m is equipped with a PPO algorithm and aims to maximize the following objective with respect to parameters θ^{k_m} and ϕ^{k_m}:

$$\mathbb{E}_{\boldsymbol{s},\boldsymbol{a}\sim(\pi_{\theta_{\text{old}}},\mathcal{T})}\left[\min\left(\frac{\pi^{k_m}_{\theta^{k_m}}(\boldsymbol{a}^{k_m}|\boldsymbol{o}^{k_m})}{\pi^{k_m}_{\text{old}}(\boldsymbol{a}^{k_m}|\boldsymbol{o}^{k_m})}\hat{A}_{\phi^{k_m}}, g(\nu)\hat{A}_{\phi^{k_m}}\right)\right] \tag{4}$$

with $g(\nu) = \text{clip}\left(\frac{\pi_{\theta^{k_m}}(\boldsymbol{a}^{k_m}|\boldsymbol{o}^{k_m})}{\pi_{\text{old}}(\boldsymbol{a}^{k_m}|\boldsymbol{o}^{k_m})}, 1-\nu, 1+\nu\right)$ and $\nu \in [0, 1)$ a hyperparameter that indicates how far away the new policy can deviate from the old one and where $\hat{A}_{\phi^{k_m}}$ is a global advantage estimator with parameters ϕ^{k_m}.

In MCA-PPO, the advantage of an agent k_m is estimated as follows:

$$\hat{A}_{\phi^{k_m}} = M^{k_{1:m}} \tag{5}$$

where $M^{k_{1:m}}$ is the compound policy ratio introduced by [15]. It is a joint advantage estimator that takes into account the previous policy updates and allow us to apply Lemma 1.

It is defined as follows:

$$M^{k_{1:m}} = \frac{\pi^{k_{1:m-1}}(\boldsymbol{a}^{k_{1:m-1}}|\boldsymbol{o}^{k_{1:m-1}})}{\pi^{k_{1:m-1}}_{\text{old}}(\boldsymbol{a}^{k_{1:m-1}}|\boldsymbol{o}^{k_{1:m-1}})}\hat{A}^{\text{GAE}}_{\phi}(\boldsymbol{s}, \boldsymbol{A}) \tag{6}$$

with $\hat{A}^{\text{GAE}}_{\phi}(\boldsymbol{s}, \boldsymbol{A})$ the global advantage estimate, parameterized by ϕ and computed at the BS with Generalized Advantage Estimation (GAE) [23]. This method uses the temporal difference residuals $\delta^{V_\phi}_t = r_t - \gamma V_\phi(\boldsymbol{s}_{t+1}) - V_\phi(\boldsymbol{s}_t)$ and defines $\hat{A}^{GAE}_{\phi_t}$ as follows:

$$\hat{A}^{GAE}_{\phi_t} = \sum_{l=0}^{\infty}(\gamma\lambda_{GAE})^l\delta^{V_\phi}(t+l) \tag{7}$$

Details of the algorithm are provided in Algorithm 1. Note that contrary to IL and CTDE versions, only one global advantage estimate is required for all agents.

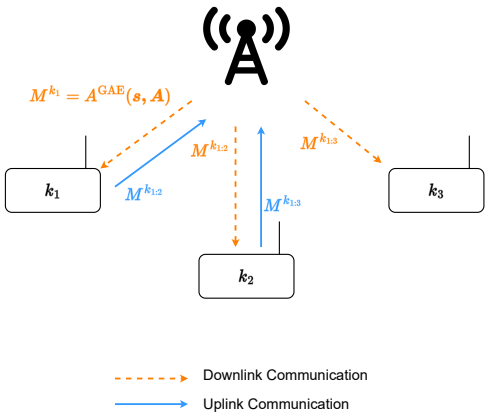


Fig. 2: Training phase of MCA-PPO.

C. Communication Protocol To Train MCA-PPO

Training the MCA-PPO algorithm is more complex than traditional methods described in the literature because updating the agents sequentially is necessary to ensure the monotonic improvement guarantee. In the context of IoT, this process is challenging as devices must not only collaborate and effectively utilize limited resources but also meet the constraints of the MARL training.

In order to train MCA-PPO, we use the on-policy property of policy gradient algorithms. This property divides the algorithm's operation into 2 phases: an *execution phase* and a *training phase*. During the execution phase, MCA-PPO runs the current joint policy and stores trajectories while using the communication resources for data transmission to the BS. During the training phase, data transmission is stopped and the communication resources are used to share the compounded policy ratios between devices in order to update the policy. The training phase is described Fig. 2: 1) The BS draws a permutation of the agents $k_{1:3}$, computes the global advantage function, and sends it to user k_1; 2) User k_1 updates its policy, computes $M^{k_{1:2}}$ and sends it to the BS; 3) The BS sends $M^{k_{1:2}}$ to user k_2 which updates its policy, computes $M^{k_{1:3}}$, and transmits it to the BS; 4) The BS sends $M^{k_{1:3}}$ to user k_3 that updates its policy, knowing all previous updates.

D. Addressing Partial Observability with a Recurrent Neural Network (RNN)

In order to tackle the partially observable aspect of our problem, we use a RNN [25] to enable agents to take actions based on their previous actions and observations. The intuition is that the RNN's hidden states estimate a belief state over the underlying system state. In Dec-POMDP, the Gated Recurrent Unit (GRU) architecture [26] is the most commonly used and this is the one that we use in MCA-PPO.

V. Simulation Results

A. Simulation settings

The parameters of our traffic model are adopted from the factory automation use case of the 3GPP 5G NR specifications

Algorithm 1: MCA-PPO

Initialize policy parameters $\theta_0^1, \dots, \theta_0^K$ for each agent and the global value function parameters ϕ

for *iteration* $i = 1, \dots, I$ **do**

  Switch the devices to *execution mode* and execute the joint policy $\boldsymbol{\pi_{\theta_i}}(\pi_{\theta_i}^1, \dots, \pi_{\theta_i}^K)$.

  Save trajectories $\{(\boldsymbol{o}_{b,t}^k, \boldsymbol{a}_{b,t}^k, \boldsymbol{o}_{b,t+1}^k, r_{b,t})\}_{b=1,\dots,\beta} \forall k \in [\![1, K]\!], \forall t \in [\![1, T]\!]$ in the buffer.

  Compute the rewards-to-go $\hat{R}_{b,t}$ for each trajectory: $\hat{R}_{b,t} = \sum_{t'=t}^{T} \gamma^{t'} r_{b,t'}$

  Switch the devices to *training mode*.

  for *epoch* $e = 1, \dots, E$ **do**

    Compute the global advantage function $\hat{A}^{GAE}(\boldsymbol{s}, \boldsymbol{A})$ with V_ϕ and GAE.

    Draw a random permutation of the agents $k_{1:K}$

    Set $M^{k_1} = \hat{A}^{GAE}(\boldsymbol{s}, \boldsymbol{A})$.

    for *agent* $k_m = k_1, \dots, k_K$ **do**

      Update actor k_m and derive $\theta_{i+1}^{k_m}$ by maximizing the following objective with the Adam algorithm [24]

$$\max_{\theta} \frac{1}{\beta T} \left[\sum_{b=1}^{\beta} \sum_{t=1}^{T} \min \left(\frac{\pi_{\theta}^{k_m}(\boldsymbol{a}_{b,t}^{k_m} | \boldsymbol{o}_{b,t}^{k_m})}{\pi_{\theta_i^{k_m}}^{k_m}(\boldsymbol{a}_{b,t}^{k_m} | \boldsymbol{o}_{b,t}^{k_m})} M_{b,t}^{k_{1:m}}, g(\nu) M_{b,t}^{k_{1:m}} \right) \right] \tag{8}$$

      Compute (unless $m = K$):

$$M^{k_{1:m+1}} = \frac{\pi_{\theta_{i+1}^{k_m}}^{k_m}(\boldsymbol{a}_{b,t}^{k_m} | \boldsymbol{o}_{b,t}^{k_m})}{\pi_{\theta_i^{k_m}}^{k_m}(\boldsymbol{a}_{b,t}^{k_m} | \boldsymbol{o}_{b,t}^{k_m})} M^{k_{1:m}}$$

    Update the global value network by minimizing the mean-squared error with the Adam algorithm:

$$\phi_{i+1} = \arg\min_{\varphi} \frac{1}{\beta T} \sum_{b=1}^{\beta} \sum_{t=1}^{T} \left(V_\phi(\boldsymbol{s}_{b,t}) - \hat{R}_b(t) \right)^2$$

on URLLC [27] where we consider deadlines of 1ms and an inter-arrival time of 2ms. Every slot is made of 1 OFDM symbol for a subcarrier spacing of 30kHz so that its time duration is equal to $T_s = 35.67\mu$s [28]. We consider the two following settings for our experiments:

- An *homogeneous setting* where all users' traffic is aperiodic with the same rate λ and deadlines δ. Given that our radio frame is made of four time-slots, we can express realistic values of λ and δ in terms of number of frames i.e. $\lambda = 1/7 = 0.14$ packet per user and per frame and $\delta = 7$ frames. We consider $N = 6$ channels and channel switch probabilities $p_{k,n}$ and $\tilde{p}_{k,n}$ equal to 0.8.
- An *heterogeneous setting* with K=6 users and N=16 channels. Devices have a deadline chosen uniformly in the subset $\{1\text{ms}, 2\text{ms}\}$. Half of them have a periodic traffic with arrival probabilities chosen uniformly in $\{0.2, 0.4, 0.6, 0.8\}$ and no offset. The other half an aperiodic one. The channel switch probabilities $p_{k,n}$ and $\tilde{p}_{k,n}$ are equal and chosen uniformly in $\{0.2, 0.4, 0.6, 0.8\}$. The parameters of the algorithms are given in Table I.

TABLE I: Parameters of the learning algorithms

Parameter	Value	Algorithm
Discount factor (γ)	0.4	All
E	5	MCA-PPO, MCA-iPPO
I	2000	MCA-PPO, MCA-iPPO
Learning rate policy	$3 \cdot 10^{-4}$	MCA-PPO, MCA-iPPO
Learning rate critic	10^{-3}	MCA-PPO, MCA-iPPO
Batch size	64	All
History length	K	All
Update target frequency	100	iDRQN
Episode length (T)	200 slots	All
Final exploration rate	0.1	iDRQN

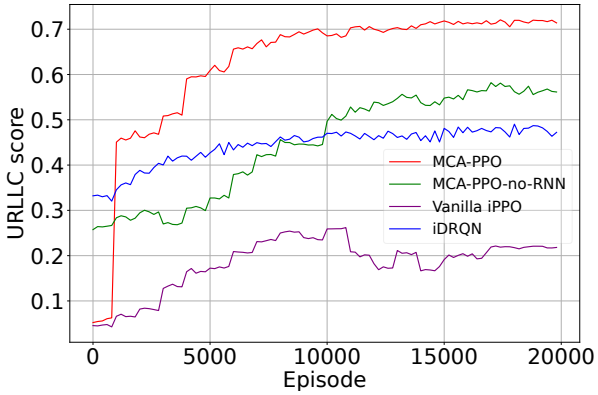
B. Baselines

In order to assess the performance of our algorithms, we introduce the following baselines:

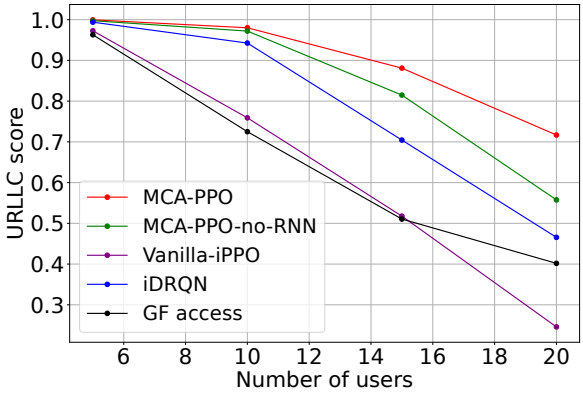
- **Contention-based grant-free access (GF access)**: All devices with a packet to transmit can simultaneously access multiple channels. Each channel is accessed with the same probability p. This access probability is empirically optimized for every user at every experiment in order to maximize the URLLC score. We employ a reactive scheme: when a device receives a NACK feedback from the BS, it will re-transmit the same packet with probability p until an ACK feedback is received or the time to deadline of the packet expires.
- **Independent Deep Recurrent Q-Networks (iDRQN)**: This baseline represents a widely-adopted DRL algorithm where each agent is modeled by a Deep Q-network and selects what channel to access through to a RNN specifically composed of a GRU layer. This standard approach has been previously employed in [9] for example. As this algorithm is not equipped to select combinatorial actions, we assume that a iDRQN agent can only send one replica of its packet in one channel at each frame.
- **Independent Proximal Policy Optimization (Vanilla iPPO)**: This baseline represents the standard iPPO algorithm as detailed in the existing literature. Compared to MCA-PPO, this baseline does not have a RNN, a branching architecture and lacks theoretical guarantees. The action space covers all possible combinations i.e. 2^n where n represents the number of subchannels.
- **MCA-PPO-no-RNN**: This baseline is MCA-PPO without RNN but with branching architecture.

C. Convergence speed of the algorithms

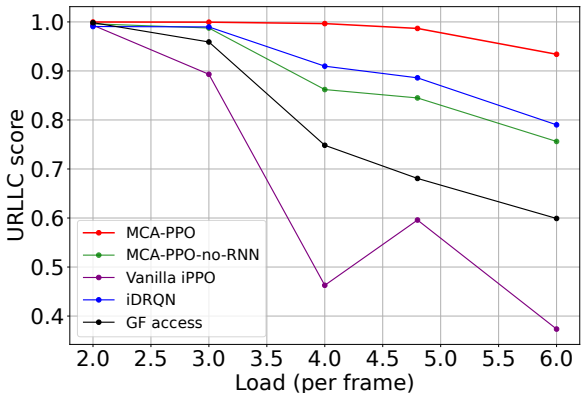
In Fig. 3a, we present the evolution of the URLLC score during the training of 20 DRL agents in a homogeneous environment. Initially, we can see that MCA-PPO outperforms all



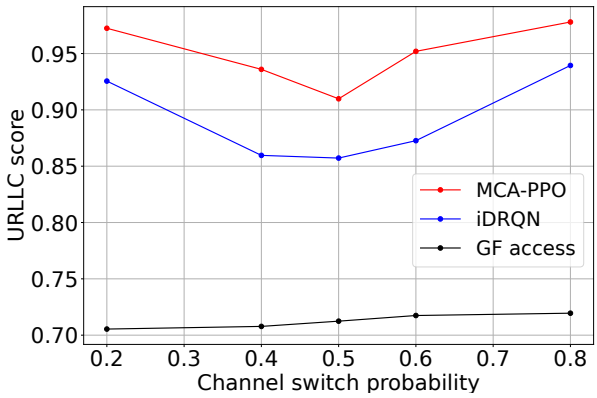
(a) URLLC score during training for 20 homogeneous users.



(b) URLLC score w.r.t. the number of homogeneous users.



(c) URLLC score w.r.t. the load in the heterogeneous setting.



(d) URLLC score w.r.t. the channel switch probability.

Fig. 3: Performance metrics in the 3GPP scenario.

baselines both in terms of convergence speed and asymptotic URLLC score. Furthermore, the variant of MCA-PPO without an RNN ranks second, illustrating its inability to match the full performance of MCA-PPO. This discrepancy highlights the RNN's role in enhancing collaboration through improved management of partial observability. Despite this, MCA-PPO without RNN still outperforms Vanilla iPPO and iDRQN. Besides, we observe that the iDRQN algorithm achieves rapid convergence. This can be explained by its limited number of copies to send and thus a much smaller action space than the others. However, this strategy results in a lower asymptotic value, highlighting the significance of sending multiple copies in a single frame. In contrast, Vanilla iPPO shows the least favorable outcomes, primarily due to its absence of a branching architecture, RNN, and theoretical performance guarantees.

D. Performance in Different Scenarios

In this section, we evaluate the performance of our proposed approach under various conditions. On the one hand, we explore a homogeneous environment in Fig. 3b, where the number of users varies from 5 to 20. On the other hand, a heterogeneous environment is studied in Fig. 3c, characterized by a load ranging from 2 to 6 packets per frame.

Across all scenarios, MCA-PPO consistently surpasses the benchmark models. Specifically, within the homogeneous setting depicted in Fig. 3b, MCA-PPO without RNN still outperforms iDRQN, the GF access method, and Vanilla iPPO. This result shows the importance of incorporating a branching architecture and having theoretical guarantees for addressing our DMCA challenge. Additionally, the iDRQN strategy demonstrates superior transmission protocol learning over Vanilla iPPO and GF access, even though it is restricted to a single channel per frame. This could be attributed to GF access experiencing increased collisions with a higher number of replicas and Vanilla iPPO's inability to navigate the combinatorial action space without a branching architecture.

Moreover, the ability to handle the combinatorial action space is tested in the heterogeneous setting where the results are shown in Fig. 3c. Indeed, we consider 16 channels which correspond to an action space of $2^{16} = 65,536$ for each agent. Here, Vanilla iPPO exhibits the worst performance, falling behind GF access across all loads. The reason is that the algorithm does not manage to converge due to the large action

space and its deterministic nature. Meanwhile, iDRQN secures the second position, outperforming MCA-PPO-no-RNN yet trailing behind MCA-PPO. Indeed, in the dense environment scenario, where the load is high, the strategy of allowing agents to send multiple copies might not be advantageous due to the increased necessity for precise collaboration. This is because a higher density of transmissions leads to a greater likelihood of collisions occurring. In such contexts, we observe that using a RNN is necessary for MCA-PPO to manage the combinatorial action space and find the optimal strategy to minimize collisions.

E. Study of the Channel Model

In Fig. 3d, we present the evolution of URLLC score as a function of the channel switch probability under the homogeneous setting for 10 users. For this analysis, the switch probabilities are constant across all users and channels i.e. $p_{k,n} = \tilde{p}_{k,n} = p$. Initially, we notice that the GF access algorithm remains constant for all channel switch probabilities. This outcome is expected as this protocol does not incorporate channel conditions into its transmission strategy. Besides, both the iDRQN and MCA-PPO algorithms demonstrate an ability to leverage channel variations to their advantage, with MCA-PPO exhibiting superior performance overall. Indeed, when the entropy is maximal at $p = 0.5$, the learning algorithms exhibit their lowest URLLC score, as there is less information to gain from the channel state. However, their performance peak at $p = 0.2$ and $p = 0.8$ where the predictability of the channel states increases, providing more valuable information for the algorithms to exploit.

VI. Conclusion

References

[1] G. Brown *et al.*, "Ultra-reliable low-latency 5g for industrial automation," *Technol. Rep. Qualcomm*, vol. 2, p. 52065394, 2018.

[2] "Study on scenarios and requirements for next generation access technologies," 3rd Generation Partnership Project (3GPP), TR 38.913. [Online]. Available: http://www.3gpp.org/DynaReport/38913.htm

[3] N.-T. Nguyen *et al.*, "Challenges, designs, and performances of a distributed algorithm for minimum-latency of data-aggregation in multi-channel wsns," *IEEE Transactions on Network and Service Management*, vol. 16, no. 1, pp. 192–205, 2018.

[4] R. Qi, X. Chi, L. Zhao, and W. Yang, "Martingales-based aloha-type grant-free access algorithms for multi-channel networks with mmtc/urllc terminals co-existence," *IEEE access*, vol. 8, pp. 37 608–37 620, 2020.

[5] M. Tan, "Multi-agent reinforcement learning: Independent vs. cooperative agents," in *ICML*, 1993.

[6] C. Zhong *et al.*, "A deep actor-critic reinforcement learning framework for dynamic multichannel access," *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, no. 4, pp. 1125–1139, 2019.

[7] S. Wang *et al.*, "Learning-based multi-channel access in 5g and beyond networks with fast time-varying channels," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 5203–5218, 2020.

[8] M. Sohaib, J. Jeong, and S.-W. Jeon, "Dynamic multichannel access via multi-agent reinforcement learning: Throughput and fairness guarantees," *IEEE Transactions on Wireless Communications*, vol. 21, no. 6, pp. 3994–4008, 2021.

[9] X. Ye *et al.*, "Multi-channel opportunistic access for heterogeneous networks based on deep reinforcement learning," *IEEE Transactions on Wireless Communications*, vol. 21, no. 2, pp. 794–807, 2021.

[10] O. Naparstek and K. Cohen, "Deep multi-user reinforcement learning for distributed dynamic spectrum access," *IEEE Transactions on Wireless Communications*, vol. 18, no. 1, pp. 310–323, 2019.

[11] X. Tan *et al.*, "Cooperative multi-agent reinforcement-learning-based distributed dynamic spectrum access in cognitive radio networks," *IEEE Internet of Things Journal*, vol. 9, no. 19, pp. 19 477–19 488, 2022.

[12] Y. Duan *et al.*, "Benchmarking deep reinforcement learning for continuous control," in *International conference on machine learning*. PMLR, 2016, pp. 1329–1338.

[13] C. S. de Witt *et al.*, "Is independent learning all you need in the starcraft multi-agent challenge?" *arXiv preprint arXiv:2011.09533*, 2020.

[14] C. Yu *et al.*, "The surprising effectiveness of ppo in cooperative multi-agent games," *Advances in Neural Information Processing Systems*, vol. 35, pp. 24 611–24 624, 2022.

[15] J. G. Kuba, R. Chen, M. Wen, Y. Wen, F. Sun, J. Wang, and Y. Yang, "Trust region policy optimisation in multi-agent reinforcement learning," *arXiv preprint arXiv:2109.11251*, 2021.

[16] W. Tang *et al.*, "On channel reciprocity in reconfigurable intelligent surface assisted wireless networks," *IEEE Wireless Communications*, vol. 28, no. 6, pp. 94–101, 2021.

[17] I.-H. Hou and P. R. Kumar, "Packets with deadlines: A framework for real-time wireless networks," *Synth. Lect. Commun*, vol. 6, no. 1, pp. 1–116, 2013.

[18] "Feasibility Study on Licensed-Assisted Access to Unlicensed Spectrum," 3rd Generation Partnership Project (3GPP), TR 36.889. [Online]. Available: http://www.3gpp.org/DynaReport/36889.htm

[19] E. N. Gilbert, "Capacity of a burst-noise channel," *Bell system technical journal*, vol. 39, no. 5, pp. 1253–1265, 1960.

[20] F. A. Oliehoek, *Decentralized POMDPs*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 471–503.

[21] J. Schulman *et al.*, "Trust region policy optimization," in *International conference on machine learning*. PMLR, 2015, pp. 1889–1897.

[22] ——, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[23] ——, "High-dimensional continuous control using generalized advantage estimation," *arXiv preprint arXiv:1506.02438*, 2015.

[24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[25] M. Hausknecht and P. Stone, "Deep Recurrent Q-learning for Partially Observable MDPs," in *AAAI Fall Symposeum*, 2015.

[26] K. Cho *et al.*, "On the properties of neural machine translation: Encoder-decoder approaches," *CoRR*, vol. abs/1409.1259, 2014.

[27] "Study on physical layer enhancements for NR ultra-reliable and low latency case (URLLC)," 3rd Generation Partnership Project (3GPP), TR 38.824. [Online]. Available: http://www.3gpp.org/DynaReport/38824.htm

[28] "NR; Physical channels and modulation," 3rd Generation Partnership Project (3GPP), TS 38.211. [Online]. Available: http://www.3gpp.org/DynaReport/38211.htm