Hybrid radio resource management based on multi-agent reinforcement learning

Ngoc-Lam Dinh, Mickael Maman, Emilio Calvanese Strinati

Hybrid Radio Resource Management for Ultra-Reliable Low Latency Communications based on Multi-Agent Reinforcement Learning

Lam Ngoc Dinh, Mickael Maman and Emilio Calvanese Strinati
CEA-Leti, Université Grenoble Alpes, F-38000 Grenoble, France
{ngoc-lam.dinh, mickael.maman, emilio.calvanese-strinati}@cea.fr

***Abstract*—In this paper, we propose a novel hybrid grant-based and grant-free radio access scheme for Ultra Reliable and Low Latency Communications (URLLC). We provide two multi-agent reinforcement learning algorithms to optimize a global network objective in terms of latency, reliability and network throughput: Multi-Agent Deep Q-Learning (MADQL) and Multi-Agent Deep Deterministic Policy Gradient (MADDPG). In MADQL, each user (agent) learns its optimal action-value function, which is based only on its local observation, and performs an optimal opportunistic action using the shared spectrum. MADDPG involves the attached gNB function as a global observer (critic), which criticizes the action of each associated agent (actor) in the network. By leveraging centralised training and decentralised execution, we achieve a shared goal better than the first algorithm. Then, through a system level simulation where the full protocol stack is considered, we show the gain of our approach to efficiently manage radio resources and guarantee URLLC.**

I. Introduction

Ultra-Reliable Low Latency Communication (URLLC) is one of the use cases of 5G and beyond networks. One limiting factor of URLLC is the deterministic spectrum access scheme and its centralized allocation procedures. Current methods involve the reservation of dedicated radio resources when opportunistic use of the spectrum is possible. Several solutions are proposed to integrate opportunistic approaches: transmission without grant, preemption of radio resources for immediate use (e.g., mini-slot preemption), semi-distributed allocation (e.g., shared resource pool of Device to Device communications), or overlapping transmissions (e.g., Non-orthogonal multiple access). In this paper, we propose to enhance URLLC deterministic protocols by opportunism. This approach will allow to overbook the share resource and will naturally provide heterogeneity management.

An hybrid Grant-Free (GF)/Grant-Based (GB) resource allocation regime was discussed by Zhou et al. [1]. Based on each UE's channel condition and their recorded activities, an optimal resource allocation was proposed to determine the amount of resources for each allocation mode (grant-based and grant-free). However, dynamic resource allocation framework is missing and authors assumed the instantaneous global knowledge of UEs at gNB's side. Nomeir et al. [2] considered an hybrid resource allocation scheme dealing with heterogeneous services (URLLC/eMBB) and proposed combinatorial allocation framework in which eMBB traffic is managed by GB scheduling and GF is adopted for URLLC traffic. Their schemes are less complex compared to state of the art solutions while achieving near-optimal performance. However, the exact delay calculation in a full stack system is omitted and they assumed the perfect knowledge of users at gNB. Huang et al. [3] proposed a Reinforcement Learning (RL) framework to jointly optimize the communication delay and energy consumption under a high URLLC load. They considered both hybrid spectrum access of licensed and unlicensed mmWave band and used the policy gradient method to update the approximate policy to achieve optimal results. RL is employed at the gNB to understand the overall knowledge and behaviours of attaching UEs. Thus the larger the number of UEs, the more complex the centralized decision at gNB and the greater the computational resources required. Liang et al. [4] investigated resource sharing as a Multi-Agent Reinforcement Learning (MARL) problem. In V2X communications, each vehicle (agent) must reuse shared resources in a way that minimizes interference while improving payload delivery rate and network capacity performance. The training procedure relies on deep Q-learning with experience replay to help each agent learn action-value functions and obtain its optimal function. They showed that cooperation between agents is beneficial for efficient decision making on the shared resource pool. Naparstek et al. [5] addressed dynamic, multi-user spectrum access based on the Aloha protocol. The training process is offline and a long short term memory is used to aggregate the observations that are partially observable in each user. This helps each user to have a better estimation of the state over time. Azari et al. [6] proposed an interesting distributed risk-aware ML for the coexistence of scheduled and non-scheduled URLLC traffic. Their ML solution for RRM increased the data throughput for scheduled traffic and reliability is guaranteed for (non-)scheduled URLLC users. They conclude that multiple QoS requirements in URLLC require a novel, scalable and distributed learning approach, and that distributed learning could be a good candidate for dealing with massive cooperation/competitive UL access.

However, there are not many studies that use MARL framework on the hybrid grant-based/grant-free scheduling with the aim of guaranteeing the QoS of massive UL URLLCs such as latency, reliability and throughput. The advantage of GB scheduling is the guaranteed network throughput under a particular scheduling policy at the cost of handshaking latency, but the network throughput and latency under GF is not ensured due to the unexpected collision. Our global optimization is based on distributed decision making. Each agent does not know the behaviour of global network and makes the decision based only on its local observation. Our contributions are (i) the MARL framework on hybrid GB/GF decision, (ii) maximizing the objective function of system (i.e. latency, reliability and throughput), (iii) cooperative distributed training, which means that the training process will take place both at users and gNB and (iv) online decision making. Performance is evaluated through the combination of the MARL framework and full protocols in NS-3 with different traffic profiles (i.e. predictable and unpredictable traffic pattern) that can lead to inefficient learning and thus sub-optimal results.

The remainder of the paper is organized as follows. Section II presents our system model and formulates our problem.

Section III describes the proposed solution, whereas Section IV provides numerical results, demonstrating our proposed algorithm performance. Finally, Section V concludes the paper.

II. System Model

A. System Models

The system consists of a single gNB and N UEs. Let $\mathcal{N}$ be the set of UEs. Each UE_i is d_i away from the gNB and generates traffic which follow Poisson process. $A_i(t), D_{1,i}(t)$ are the total packet size [Byte] arriving or departing from $Q_{1,i}$ of agent i at time slot t, respectively. The total bandwidth is composed of RB_t resource blocks, divided into RB_{GB} for dedicated resources (i.e. formed by T_{GB} OFDM symbols) and RB_{GF} for shared resources (i.e. formed by T_{GF} OFDM symbols). Each UE_i considers 3 queues: $Q_{1,i}$ represents the data packets in the RLC buffer, $Q_{2,i}$, $Q_{3,i}$ show the data packets scheduled for the dedicated allocation and shared allocation, respectively. The dynamic queues in each agent i can be expressed as follows:

$$Q_{1,i}(t+1) = \max\{Q_{1,i}(t) - D_{1,i}(t), 0\} + A_i(t) \quad (1)$$
$$Q_{2,i}(t+1) = \max\{Q_{2,i}(t) - \mathbb{1}_i.D_{2,i}(t), 0\} + A_{2,i}(t) \quad (2)$$
$$Q_{3,i}(t+1) = \max\{Q_{3,i}(t) - \mathbb{1}_i.D_{3,i}(t), 0\} + A_{3,i}(t) \quad (3)$$

where: $A_{2,i}(t) = p_i \times D_{1,i}(t)$, $A_{3,i}(t) = q_i \times D_{1,i}(t)$ are the size of packets in Bytes that start from $Q_{1,i}$ and are sent to $Q_{2,i}$ and $Q_{3,i}$ of agent i, respectively. p_i and q_i (or $p_i^{\pi_i}$ and $q_i^{\pi_i}$) are the probability that the transport blocks are sent to $Q_{2,i}$ and $Q_{3,i}$ according to policy π_i. If the transport blocks are placed in $Q_{2,i}$, the gNB schedules them according to a centralized scheduling policy π_0. Indeed, a quantity equivalent to $D_{2,i}$ (or $D_{2,i}^{\pi_0}$) Bytes will leave $Q_{2,i}$ if agent i is scheduled by gNB. On the other hand, if the transport blocks are queued at $Q_{3,i}$, each agent i will opportunistically perform resource selection from the shared and competitive resource pool and $D_{3,i}(t)$ (or $D_{3,i}^{\pi_i}(t)$) Bytes will be removed from $Q_{3,i}(t)$. In both cases, an indicator function $\mathbb{1}$ shows that the packet transmission was successful and that the receiver was able to transfer it to the upper layer. Thanks to the ACK message, $D_{2,i}$ and $D_{3,i}$ will be deducted from $Q_{2,i}(t)$ and $Q_{3,i}(t)$, respectively.

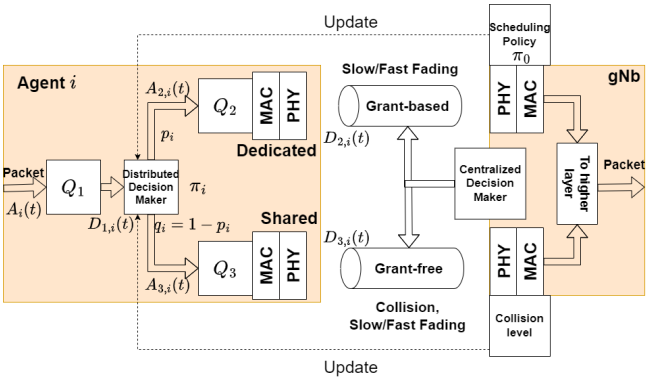


Fig. 1: Hybrid allocation scheme

B. Protocol and Scheduling Policy

In this work, total resources is divided into 2 parts: scheduled resources (grant-based) and shared resources (grant-free). In our hybrid allocation regime, centralized policy π_0 manages scheduled resources in grant-based manner. The partition between scheduled resources and shared resources is also defined by π_0 and is fixed at the beginning. Meanwhile, shared resources are opened for every user such that collisions between users are minimized under decentralized policy π_i. Each agent, who wants to access the scheduled resources, has to perform a 5-step (grant-based) procedure as follows:

- Step 1: Each agent i sends the status of its RLC queue $Q_{1,i}$ in the Scheduling Request (SR)
- Step 2: Upon SR reception, gNB sends Signaling Grant (SG) accompanied with few resources for requested UEs.
- Step 3: After receiving the SG, each agent sends its Buffer Status Report (BSR) to the attached gNB.
- Step 4: At this step, the gNB has a global view of the amount of pending data for each user. According to the predetermined total scheduled resources, resources are allocated to corresponding users following the scheduling policy π_0 (e.g, Round-Robin, Proportional-Fair).
- Step 5: User sends both data and BSR based on allocated resources. The next scheduling phase begins in step 4 until there is no more data queued in the user.

Hence, GB scheduling is centralized and managed by gNB which has global view of user activity. The advantage is the guarantee of allocated resources and collision-free communication. However, the long handshaking procedure (5-step) cause high delay and may not be suitable for URLLC.

In GF, each agent i selects resources from the shared resource pool using slotted ALOHA method and decentralized policy π_i. Each user minimizes the number of collisions (i.e. more than two users choosing the same resource). By doing so, we can overcome the high delay caused by 5-step handshaking. On the other hand, resource allocation will no longer be managed between users and each communication will face a non-negligible collision probability.

C. Objective Function

The average delay is proportional to the average queue length, so we can formulate the minimization of global latency as the maximization of the function $d(t)$.

$$\begin{aligned} d(t) &= \lim_{T\to\infty} \frac{1}{T} \sum_{t}^{T} \sum_{i=1}^{N} -\mathbb{E}_{\pi} \left[Q_{1,i}(t) + Q_{2,i}(t) + Q_{3,i}(t)\right] \\ &= \lim_{T\to\infty} \frac{1}{T} \sum_{t}^{T} \sum_{i=1}^{N} \sum_{j=1}^{3} -\mathbb{E}_{\pi} \left[Q_{j,i}(t)\right] \end{aligned} \quad (4)$$

In this case, $\boldsymbol{\pi} = [\pi_1, .., \pi_N]$ is global policy and it contains the local policy of each agent i, i.e π_i.

The throughput of each agent i is measured by the total time average of transport blocks in bytes that are sent in grant-free and grant-based channels. Next, we defined $r_i = \lim_{T\to\infty} \frac{1}{T} \sum_{t}^{T} \mathbb{E}_{\pi_i}[D_{2,i}(t) + D_{3,i}(t)]$. Maximizing the UL throughput of the network is equivalent to maximizing the total sum-rate $e(t) = \sum_{i=1}^{N} \mathbb{U}_i(r_i)$ where $\mathbb{U}_i(.)$ is a non-decreasing and concave utility function.

To trade-off the maximization of the average delay with the network throughput, we introduce the control parameter $\nu \geq 0$ to build a weighted-sum optimization function. Our global objective function that jointly optimizes latency and throughput is then derived as follows:

$$f_{\pi}(t) = d(t) + \nu e(t) \quad (5)$$

Instead of maximizing the objective function $f_{\pi}(t)$, we will maximize its lower bound function (i.e $g_{\pi}(t) \leq f_{\pi}(t)$). It is achieved based on the fact that the departure rate of the packets from the Q_2 and Q_3 (i.e. r_i) should higher than their arrival rate z_i to stabilize the queue dynamics. ($z_i = \lim_{T\to\infty} \frac{1}{T} \sum_{t}^{T} \mathbb{E}_{\pi_i}[A_{2,i}(t) + A_{3,i}(t)]$)

$$
\begin{aligned}
g_\pi(t) &= d(t) + \nu \sum_{i=1}^{N} \mathbb{U}_i\{z_i\} \\
&= \lim_{T\to\infty} \frac{1}{T} \sum_{t}^{T} \sum_{i=1}^{N} \sum_{j=1}^{3} \nu \mathbb{E}_\pi[\mathbb{U}_i(z_i(t))] - \mathbb{E}_\pi[Q_{j,i}(t)]
\end{aligned} \tag{6}
$$

Without loss of generality, a negative Lyapunov drift-term $\mathbb{E}_\pi[-\nu_1(Q^2_{1,i}(t+1) - Q^2_{1,i}(t)) - \nu_2(\sum_{j=2}^{3} Q^2_{j,i}(t+1) - Q^2_{j,i}(t))]$ is added with no impact on the overall problem because the solved optimal solution pushes the queues to a minimal congested state. Thus, if all the queues are stable, this add-on will converge to 0 as t goes to infinity. Then, without changing the optimal solution, the objective function becomes $h_\pi(t)$.

$$
\begin{aligned}
h_\pi(t) = \lim_{T\to\infty} \frac{1}{T} \sum_{t}^{T} \sum_{i=1}^{N} \Big(& \sum_{j=1}^{3} \mathbb{E}_\pi[-Q_{j,i}(t)] \\
& + \mathbb{E}_\pi \Big[\nu \mathbb{U}_i\left(z_i(t)\right) - \nu_1 \left(Q^2_{1,i}(t+1) - Q^2_{1,i}(t) \right) \\
& - \nu_2 \sum_{j=2}^{3} \left(Q^2_{j,i}(t+1) - Q^2_{j,i}(t) \right) \Big] \Big)
\end{aligned} \tag{7}
$$

Where $\nu, \nu_1, \nu_2 > 0$.

Finally, we transform Equation 7 into a discounted dynamic programming problem with discount factor $0 \leq \alpha \leq 1$. We show that the optimal policy of this problem can be approximated by the policy of original average problem when α is close to 1 [7]. The reward function in the MARL framework is similar to the function to be optimized.

$$
\begin{aligned}
h^\alpha_\pi(t) = \lim_{T\to\infty} \frac{1}{T} \sum_{t}^{T} \alpha^t \sum_{i=1}^{N} \Big(& \sum_{j=1}^{3} \mathbb{E}_\pi[-Q_{j,i}(t)] \\
& + \mathbb{E}_\pi \Big[\nu \mathbb{U}_i\left(z_i(t)\right) - \nu_1 \left(Q^2_{1,i}(t+1) - Q^2_{1,i}(t) \right) \\
& - \nu_2 \sum_{j=2}^{3} \left(Q^2_{j,i}(t+1) - Q^2_{j,i}(t) \right) \Big] \Big)
\end{aligned} \tag{8}
$$

D. Problem Formulation

Our problem can be formulated as follows:

$$
\begin{aligned}
\underset{\pi}{\text{maximize}}\ & h^\alpha_\pi(t) && (\mathcal{P}) \\
\text{s.t.}\ & \nu, \nu_1, \nu_2 \geq 0, && (\mathcal{C}_0) \\
& RB_t = RB^{\rho_{\pi_0}}_{GB} + RB^{1-\rho_{\pi_0}}_{GF} && (\mathcal{C}_1) \\
& 0 \leq \rho_{\pi_0} \leq 1 && (\mathcal{C}_2) \\
& RB_i = \pi_{0,i}(RB^{\rho_{\pi_0}}_{GB}) + \pi_i(RB^{1-\rho_{\pi_0}}_{GF}), \forall i && (\mathcal{C}_3) \\
& \mathbb{P}_{\pi_0}[\gamma^{gb}_i \leq \gamma_t] \leq \epsilon_t, \forall i && (\mathcal{C}_4) \\
& 1 - (1 - \mathbb{P}^{col}_{\pi_i})(1 - \mathbb{P}_{\pi_i}[\gamma^{gf}_i \leq \gamma_t]) \leq \epsilon_t, \forall i && (\mathcal{C}_5) \\
& \overline{A_{j,i}} \leq \overline{D_{j,i}}, \forall j \in \{2,3\} && (\mathcal{C}_6) \\
& \overline{A_{1,i}} \leq \overline{A_{2,i}} + \overline{A_{3,i}}
\end{aligned}
$$

The constraints $(\mathcal{C}_1)$, $(\mathcal{C}_2)$ and $(\mathcal{C}_3)$ limit the number of resource blocks (RB) allocated to each agent i under the policy π_i or π_0. In particular, $(\mathcal{C}_1)$ states that the total number of resource blocks RB_t is partitioned into GB, i.e. $RB^{\rho_{\pi_0}}_{GB}$ and GF, i.e. $RB^{1-\rho_{\pi_0}}_{GF}$. $(\mathcal{C}_2)$ defines this separation, managed by the gNB under policy π_0 with the ratio ρ_{π_0}. The constraint $(\mathcal{C}_3)$ reveals that the resources RB_i of each agent i can be either scheduled by policy π_0 (i.e. $\pi_{0,i}(RB^{\rho_{\pi_0}}_{GB})$) or competed under policy π_i (i.e. $\pi_i(RB^{1-\rho_{\pi_0}}_{GF})$). Then, the constraints $(\mathcal{C}_4)$ and $(\mathcal{C}_5)$ relate to the reliability requirements. The transport blocks will be successfully decoded at the receiver when their SINR γ is above a predefined target γ_t. Regardless of the GF channel (γ^{gf}_i) or the GB channel (γ^{gb}_i), the outage probability must be less than a target ϵ_t. In the GB channel, only fast/slow fading channel is the source of impairment, so $(\mathcal{C}_4)$ guarantees a transmission error below a threshold ϵ_t under the scheduling policy π_0. In the GF channel, the collision due to uncoordinated resource selections between agents, which is characterized by $\mathbb{P}^{col}_{\pi_i}$ is also considered in addition to fast/slow fading. Thus, $(\mathcal{C}_5)$ takes into account both impairments simultaneously. The constraint $(\mathcal{C}_6)$ guarantees the stability of the queues in each agent i under any policy π_i. The operator $\overline{X}$ is $\lim_{T\to\infty} \frac{1}{T} \sum_{t}^{T} \mathbb{E}_{\pi_i}[X(t)]$. According to queue theory, the time average of arriving process should be smaller than or equal to the one departing from the queue. The average amount of departure process at $Q_{2,i}(t)$ depends on the centralized scheduling policy of gNB π_0, while others depend on its decentralized policy π_i.

III. Algorithm Design

In this paper, we design two different algorithms which are based on MARL framework to solve the problem $\mathcal{P}$. Firstly, Multi-agent Deep Q-Learning (MADQL) is proposed for each agent i can learn action-value function (i.e., Q_i function) based only on its local observation and there is no communication between agents for mutual information exchange [8]. Then, we propose another algorithm based on Multi-agent Deep Deterministic Policy Gradient (MADDPG) method. In this approach, a centralised gNB criticizes the actions of all the agents in a particular state based on its accessibility to the global observation. These action values are essential for each agent to update its local policy to the optimal policy. It provides each agent with the advantage of evaluating the state-value function without the need for the nonviable assumption of a global access to the state.

The State/Observation space $\boldsymbol{s}_i$ for each agent i, at time slot t has to consider some information about its queues, $\boldsymbol{Q_i(t)} = [Q_{1,i}(t), Q_{2,i}(t), Q_{3,i}(t)]$, the mean traffic rate λ_i, which will be used to estimate $A_i(t)$ and the scheduling policy π_0 in GB access and occupancy level information $\boldsymbol{occ}_i(t)$ in GF access. Thus, at each time slot t, $\boldsymbol{s_i(t)} = [\boldsymbol{Q_i(t)}, \lambda_i, \pi_0(t), \boldsymbol{occ}_i(t)]$. Afterwards, we define a subset $\mathcal{K} \in \mathcal{N}$ of agents in which each agent $k \in \mathcal{K}$ can observe other agent's states, $\mathcal{K} = \{1, ..., K\}$. If only local observation is permitted (i.e. $K = 1$), each agent observes itself to make decision. If $K = N$, each agent can access to global state. The global state is formed as follows:

$$
\mathcal{S}(t) = \bigcup_{k=1}^{K} \boldsymbol{s_k(t)} \tag{9}
$$

Similarly, we can define the global action of each agent in the network as follows:

$$
\mathcal{A}(t) = \bigcup_{k=1}^{K} \boldsymbol{a_k(t)} \tag{10}
$$

where on each agent i, at time slot t, the one-hot encoding action vector $\boldsymbol{a_i(t)}$ with (1+B_s) elements implies flow control and resource selection. If $a_i[0](t) = 1$ at time slot t, agent i promotes access to GB radio resources with the 5-step

procedure according to the centralised policy π_0. Otherwise, $a_i[j](t) = 1\ \forall j \neq 0$, agent i executes the GF procedure to opportunistically choose resource j from the shared radio resources B_s for its resource block communication. The data flow control is performed as follows: First, from $Q_{1,i}(t)$, agent i transmits a quantity of $D_{1,i}(t)$ bytes to $Q_{2,i}(t)$ if $a_i[0](t) = 1$ and to $Q_{3,i}(t)$, otherwise. Then, the data in $Q_{2,i}(t)$ and $Q_{3,i}(t)$ is framed as transport block for transmission. The transport block size is decided based on the scheduling policy π_0 in GB and the amount of shared resources in GF, respectively. In case of a transport block error, due to fading in GB and collision/fading in GF, a non-acknowledgement is returned to the agent and the resource selection procedures are repeated.

The reward function quantifies how good an action is taken under a particular state. For each agent i, we define the reward under state $\boldsymbol{s_i(t)}$ and action $\boldsymbol{a_i(t)}$, $R_i(\boldsymbol{s_i(t)}, \boldsymbol{a_i(t)}) \in \mathbb{R}$ according to the objective function in Equation 7 as follows:

$$\begin{aligned} R_i(\boldsymbol{s_i(t)}, \boldsymbol{a_i(t)}) = \sum_{j \in 3} & -Q_{j,i}(t+1) + \nu \log(z_i(t)) \\ & - \nu_1 \left(Q^2_{1,i}(t+1) - Q^2_{1,i}(t)\right) \\ & - \nu_2 \sum_{j=2}^{3} \left(Q^2_{j,i}(t+1) - Q^2_{j,i}(t)\right) \end{aligned} \tag{11}$$

Accordingly, the global reward observed by gNB for the set $\mathcal{N}$, i.e $\mathcal{R}(\mathcal{S}(t), \mathcal{A}(t)) \in \mathbb{R}^N$ of agents is derived as follows:

$$\mathcal{R}(\mathcal{S}, \mathcal{A}) = \bigcup_{i=1}^{N} \boldsymbol{R_i}\left(\boldsymbol{s_i(t)}, \boldsymbol{a_i(t)}\right) \tag{12}$$

A. Algorithm based Multi-Agent Deep Q-Learning (MADQL)

The objective of Q-learning is to learn the action-value function Q($S(t), A(t)$), showing the current network state at time t. To avoid confusion with designed queues $Q_{1,i}, Q_{2,i}, Q_{3,i}$ of each agent i, we will hereafter refer to Q-learning as W-learning. Figure 2 depicts the algorithm framework.

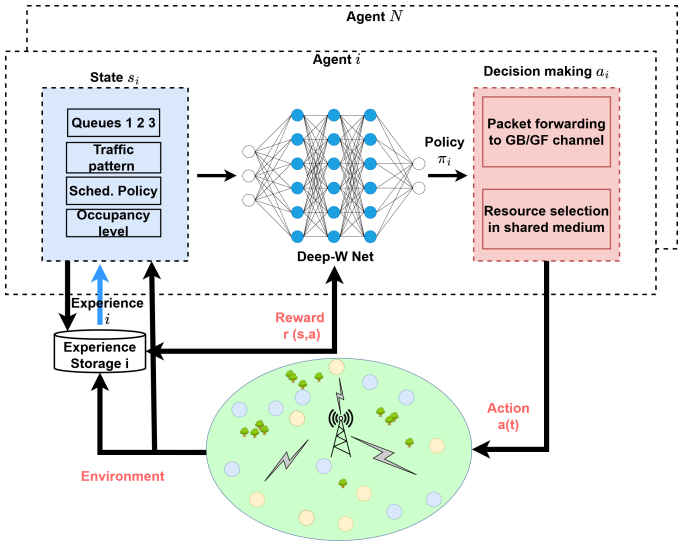


Fig. 2: Multi Agent Deep W(Q)-Learning architecture

In this algorithm, each agent has its own memory play, which is stored locally. Also, we assumed that Partially Observable Markov Decision Process (POMDP) is considered where each agent does not have full observation dynamic [8]. Under local policy π_i, the action value function of agent i, i.e., W_{π_i}-function is defined as: $W_{\pi_i}(s_i, a_i) = R_i(s_i, a_i) + \alpha W_{\pi_i}(s'_i, \pi_i(s'_i))$. Where s_i and a_i are respectively current state and action of agent i which returns a corresponding rewards $R_i(s_i, a_i)$ and turns agent i into new state s_i

By estimating this function, we will guide an agent in selecting the optimal action in a particular state. However, due to the complex dynamic state of the multi-agent system, it is viable to approximate their value function with the parameter ω using a neural network, i.e, $W_{\pi_i}(s_i, a_i, \omega)$. The approximation of W-function will be expressed by the algorithm 1 and the value function will be updated each period T.

Algorithm 1: W(Q)-learning based on W(Q)-value

Initialization Replay memory Mem
Random initialization $W_{\pi_i}(s_i, a_i, \omega)$, $W_{\pi_i}(s_i, a_i, \phi)\ \forall s_i, a_i \in \mathcal{S}, \mathcal{A}$
Define M-batch size
for *time slot* t **do**
  for *each agent* i **do**
    if ***inactive*** **then**
      continue
    end
    Given state $\boldsymbol{s_i(t)} \in S(t)$
    Select $\boldsymbol{a_i(t)}$ as the output of $decay\ \varepsilon - greedy$
    Observe next state $\boldsymbol{s_i(t+1)}$
    Store $(\boldsymbol{s_i(t)}, \boldsymbol{a_i(t)}, \boldsymbol{R_i(t)}, \boldsymbol{s_i(t+1)})$ to Mem.
    if *More than* M *samples are collected* **then**
      Randomly sampling mini-batch M from Mem
      Given $(\boldsymbol{s_i(t)}, \boldsymbol{a_i(t)}, \boldsymbol{s'_i(t+1)}) \in M$
      Calculate $y_i = \max_{a_i}[R_i(s'_i, a_i) + \alpha W_{\pi_i}(s'_i, a_i, \phi)]$
      Perform gradient descent on
      $\sum_{(s,a,s',y) \in M}(y_i - W_{\pi_i}(s_i, a_i, \omega))^2$
      if $mod(t, T)$ == *0* **then**
        $\phi = \omega$
      end
    end
  end
end

In this algorithm, we use a technique called Experience Replay [9] to efficiently utilize the collected samples and eliminate their correlation. Also, two separate networks are used to independently select the action and learn the value function to avoid overestimation [10]. As we used ω-parameterized network to evaluate the action-value function, the network which is responsible for action selection will be parameterized by ϕ. When local action of each agent i is executed at time slot t, each agent will observe the next state at time slot $t+1$ and store sample $(s_i(t), a_i(t), R_i(t), s_i(t+1))$ to the buffer. The reward will be calculated accordingly because it is a function of state. After M samples collected (batch-size), we use fixed target technique which holds the target W-function parameterized with ϕ and updates the target every steps [11].

B. Algorithm based Multi-Agent Deep Policy Gradient (MADDPG) Learning

In this section, we propose another algorithm to solve problem $\mathcal{P}$ which is based on optimal policy learning rather than action-value function learning. It avoids the sensitivity of value function based learning to the high variance of the multi-state environment. Specifically, the use of W(Q)-learning in a multi-agent environment faces a challenge when policy of each agent changes over time and the environment is non stationary. Thus, the convergence of multi-agent algorithms based on W(Q)-learning in dynamic environment is often time consuming. On the other hand, the policy gradient method often requires the coordination of several agents and leads to high training

variance. In this section, we propose an algorithm based on Actor-Critic approach (policy-learning) that directly learns the policy leading to the optimal solution. Both the actor and the critic are approximated by neural networks. The training process is centralized at the gNB, where a centralized critic part learns the global states, actions and policies of all agents. Then, each agent i attached to the gNB can obtain the training knowledge to derive its own policy in the decentralized actor part. Since we need to learn/approximate the individual policy and value function of each agent, we parameterize its policy and value function W as θ_i and w_i, respectively. And let's assume $\boldsymbol{\theta} = \{\theta_1, ..., \theta_N\}$, $\boldsymbol{\pi} = \{\pi_1, ..., \pi_N\}$. The objective of our algorithm is to maximize the reward function $\sum_i^N R_i$. The framework can then be displayed on Figure 3.

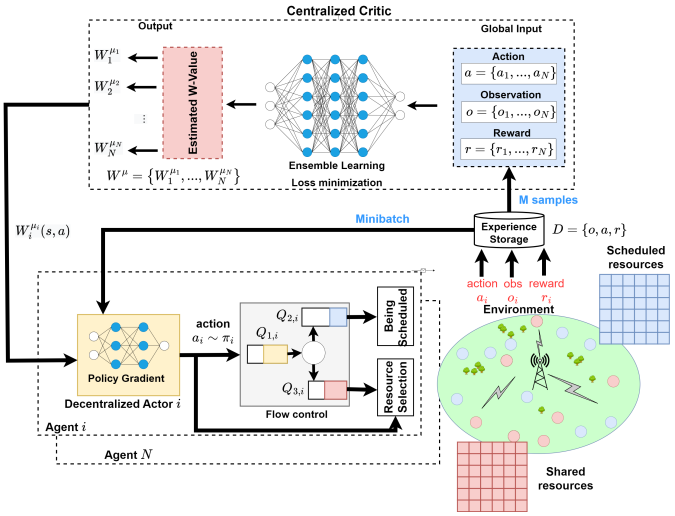


Fig. 3: Multi-Agent Deep Deterministic Policy Gradient Decision architecture

The motivation of this architecture is the separation of centralized critic part (embedded in gNB) which has global observation of all users, and decentralized actor part (embedded in each agent i) which has only local observations. Then, centralized critic will help us to approximate the value function $W_i^{\mu_i}(s_i(t), a_i(t))$ of agent i following its parameterized policy π^{μ_i} taking action $a_i(t)$ at state $s_i(t)$ at time slot t. This information will be sent into agent i and it will use such information to estimate or optimize policy π^{μ_i}. centralized critic improves the estimation of state-action value learning of which decentralized actor use to improves the policy evaluation. To the end, this approach will converge to optimal policy without requiring global observation of agent i in the network. The algorithm can be expressed in Algorithm 2. In this algorithm, $\mu' = \{\mu_{\theta_i'}\}$ is the set of target policies with delayed parameters θ_i'. The approximate policy of each agent i is learned by maximizing the log probability of agent $i's$ actions. τ is a parameter for updating the target network, α is learning parameter (discounted factor).

IV. Results and Discussions

A. Simulation Model

Our network contains a single gNB and $N = 30$ UEs, all placed at the same distance from the gNB ($d = 80m$). Each user's traffic is generated using a Poisson process with a fixed packet size of 20 Bytes. After being encapsulated with a header in internet protocol and packet data convergence protocol

Algorithm 2: Semi-distributed Learning Algorithm

Replay memory Mem
Random action initialization (Exploration)
for *each time slot* t **do**
  gNb observes global action $\mathcal{A}$(t), reward $\mathcal{R}$(t) and state $\mathcal{S}$(t), $\mathcal{S}$(t+1)
  Store $(\mathcal{S}(t), \mathcal{A}(t), \mathcal{S}(t+1), \mathcal{R}(t))$ in Mem
  Set $\mathcal{S}(t) \leftarrow \mathcal{S}(t+1)$
  for *each agent* $i \in \mathcal{N}$ **do**
    if ***inactive*** **then**
      continue
    end
    Perform $a_i = \mu_{\theta_i}(s_i)$ according current policy π_i
    Sample M samples (S^m, A^m, R^m, S'^m) in Mem
    Set target $\boldsymbol{y_i^m} = R_i^m + \alpha W_i^{\mu'}(S'^m, A'^m)$
    gNB updates Critic part (Gradient descent)
    $\mathcal{L}(\theta_i) = \frac{1}{|M|}\sum_m (y_i^m - W_i^{\mu}(S^m, A^m))^2$
    Agent i updates Actor part (Policy gradient)
    $\Delta_{\theta_i} J \approx \frac{1}{|M|}\sum_m \Delta_{\theta_i}\mu_i(s_i^m)\Delta_{a_i} W_i^{\mu}(S^m, A^m)$
    Update target network parameters
    $\theta_i' \leftarrow \tau\theta_i + (1-\tau)\theta_i'$
  end
end

layers, they arrive at the RLC Layer (Q_1) for transmission to the GB channel (Q_2) or the GF channel (Q_3). In this work, we consider a total Bandwidth of $50MHz$ in the Sub-6GHz band and half of it is dedicated to the GB channel ($\rho_{\pi_0} = 0.5$). The remaining BW is divided equally into 15 groups (RBGs) for opportunistic access to resources. In order to model dynamic channel and antenna model, the 3GPP Indoor-Factory scenario [12] is considered and gNB and each of agent are equipped with 4×4, 2×2 planar linear antennas, respectively. The details of simulation parameters are given in Table I.

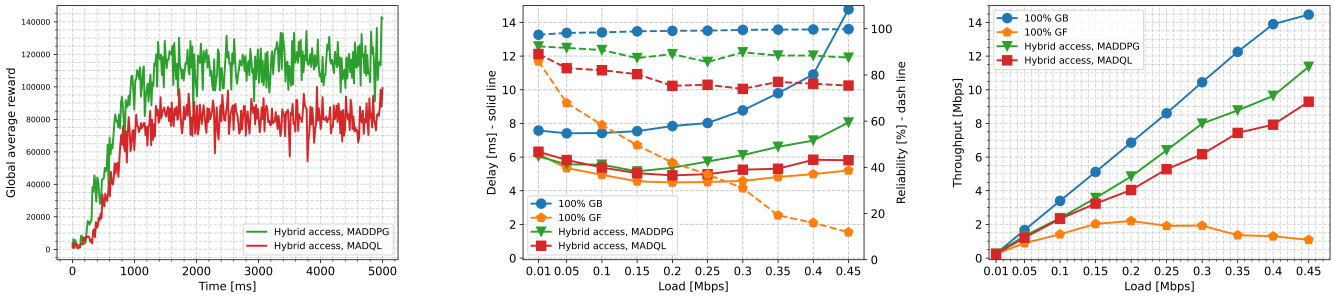
TABLE I: Simulation parameters

Nb of users	30
Channel dynamics	3GPP indoor-factory
Distance	80 m
Tx power	8 dBm
Central frequency	3.61 GHz
Bandwidth	50 MHz
ρ_{π_0}	0.5
Scheduling policy π_0	Round-robin
Shared resources in GF (B_s)	15 RBGs
Reliability target ϵ_t	0.01
Max nb of retransmissions	5

Concerning the hyper-parameters of learning algorithms, Table II shows their numerical values in our simulation.

TABLE II: Hyper-parameters

Hyper-parameters	**MADQL**	**MADDPG**
Actor learning rate α_a	**N/A**	10^{-2}
Critic learning rate α_c	**N/A**	10^{-2}
Delay network update rate τ	**N/A**	0.1
Discount factor	0.99	0.9
Learning rate α	2.10^{-2}	**N/A**
Exploration rate ε	0.99	**N/A**
Decay rate ε_d	5e-4	**N/A**
Minimum exploration rate ε_m	0.01	**N/A**
Batch size	64	
Hidden layers	2	
Dimension of hidden layer	64	
Time period T	0.1 s	**N/A**
ν, ν_1, ν_2	(10000,1000,500)	



(a) Average reward for a 0.1 Mbps traffic load (b) Delay and Reliability performance (c) Throughput performance

Fig. 4: Performance comparison between MADDPG algorithm, MADQL algorithm and GB/GF access.

B. Simulation Results

Figure 4a compares the obtained rewards between MADQL-based and MADDPG-based algorithms when each user has 0.1 Mbps traffic load. It shows the convergence of both algorithms and the best rewards of MADDPG. Figures 4b and 4c display the performance in terms of **latency**, **reliability** and **network throughput** between *(i)* 100 % GB ($\rho_{\pi_0} = 1$), *(ii)* 100 % GF ($\rho_{\pi_0} = 0$), *(iii)* MADQL-based hybrid access scheme ($\rho_{\pi_0} = 0.5$) and *(iv)* MADDPG-based hybrid access scheme ($\rho_{\pi_0} = 0.5$) as a function of the traffic load of each user. As expected, the latency due to the handshaking procedure of 100 % GB is the most important to ensure resource access to all users (i.e. maximum reliability). 100 % GB guarantees radio resources for each user according to the round-robin policy π_0. Thus, the network throughput increases linearly when higher traffic loads are generated for each user, until saturation. On the other hand, the latency of successful communications of 100 % GF is the best, but the shared resources are mismanaged (i.e. worst reliability). When the number of opportunistic accesses becomes high, it leads to a higher probability of collision and accidentally reduces the network throughput at some traffic load generation ($0.15 Mbps$). In a more efficient way, MADQL and MADDPG significantly improve the network throughput as the optimal action-value/policy guides user to select a safer action for opportunistic use of shared resources. MADQL better exploits shared resource to maintain low latency and significantly reduces users collisions (i.e. higher reliability than 100 % GF). Through the coordination of the centralised gNB in estimating each agent's action, MADDPG improves the management of the shared resources (i.e fewer collisions) while maintaining low latency. In particular, with the gNb coordination, MADDPG achieves better performance when traffic load is high.

V. Conclusions

In this paper, we propose two different multi-agent based algorithms for hybrid Grant-Based/Grant-Free access schemes that are capable of empowering shared resources in the GF channel to improve communication latency and network throughput with low impact on reliability. By means of a system-level simulation, where a full protocol stack is considered, it has been demonstrated that the use of semi-distributed approach (MADDPG), with the support of centralized gNb (critic) having the full evaluation of each associated agent (actor), provides better opportunistic access with fast uplink delay and less collision between users. However, the application of the MADQL approach where only local observation is possible should not be discarded when the algorithm favors agents that exploit shared spectrum access to minimise their uplink latency at the cost of collisions and thus reduce transmission reliability and throughput. The performance gains of our proposal are confirmed when compared with the typical centralized, round-robin scheduling policy (100% GB) and the decentralized, slotted ALOHA protocol (100 % GF). Based on the promising findings presented in this paper, in our future work, we will evaluate more complex scenarios with the participation of a larger number of users with heterogeneous traffic and distributed access using collaborative MARLs will be studied.

Acknowledgment

This work was supported by the European Union H2020 / Taiwan Project 5G CONNI [13] under grant N°861459.

References

[1] Z. Zhou, R. Ratasuk, N. Mangalvedhe, and A. Ghosh, "Resource Allocation for Uplink Grant-Free Ultra-Reliable and Low Latency Communications," in *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*, pp. 1–5, June 2018.

[2] M. W. Nomeir, Y. Gadallah, and K. G. Seddik, "Uplink Scheduling for Mixed Grant-Based eMBB and Grant-Free URLLC Traffic in 5G Networks," in *2021 17th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pp. 187–192, Oct. 2021.

[3] Q. Huang, X. Xie, and M. Cheriet, "Reinforcement learning-based hybrid spectrum resource allocation scheme for the high load of URLLC services," *EURASIP Journal on Wireless Communications and Networking*, vol. 2020, p. 250, Dec. 2020.

[4] L. Liang, H. Ye, and G. Y. Li, "Spectrum Sharing in Vehicular Networks Based on Multi-Agent Reinforcement Learning," *IEEE Journal on Selected Areas in Communications*, vol. 37, pp. 2282–2292, Oct. 2019.

[5] O. Naparstek and K. Cohen, "Deep Multi-User Reinforcement Learning for Distributed Dynamic Spectrum Access," *IEEE Transactions on Wireless Communications*, vol. 18, pp. 310–323, Jan. 2019.

[6] A. Azari, M. Ozger, and C. Cavdar, "Risk-Aware Resource Allocation for URLLC: Challenges and Strategies with Machine Learning," 2018.

[7] E. A. Feinberg, P. O. Kasyanov, and N. V. Zadoianchuk, "Average-Cost Markov Decision Processes with Weakly Continuous Transition Probabilities," Feb. 2012.

[8] N. Yang, H. Zhang, and R. Berry, "Partially Observable Multi-Agent Deep Reinforcement Learning for Cognitive Resource Management," in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, pp. 1–6, Dec. 2020.

[9] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. A. Riedmiller, "Playing atari with deep reinforcement learning," *CoRR*, vol. abs/1312.5602, 2013.

[10] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Adaptive Computation and Machine Learning Series, Cambridge, Massachusetts: The MIT Press, second edition ed., 2018.

[11] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, Feb. 2015.

[12] 3GPP, "Study on channel model for frequencies from 0.5 to 100 GHz, Release 16 (v16.1.0)," 2019.

[13] E. Calvanese Strinati, T. Haustein, M. Maman, W. Keusgen, S. Wittig, M. Schmieder, S. Barbarossa, M. Merluzzi, H. Klessig, F. Giust, D. Ronzani, S. P. Liang, J. S. J. Luo, C. Y. Chien, J. C. Huang, J. S. Huang, and T. Y. Wang, "Beyond 5G Private Networks: the 5G CONNI Perspective," in *2020 IEEE Globecom Workshops (GC Wkshps)*, 2020.