Online Multi-Agent Reinforcement Learning for Multiple Access in Wireless Networks

Jianbin Xiao, Zhenyu Chen, *Graduate Student Member, IEEE*, Xinghua Sun, *Member, IEEE*, Wen Zhan, *Member, IEEE*, Xijun Wang, *Member, IEEE*, and Xiang Chen, *Member, IEEE*

***Abstract*— Next-generation wireless networks face a variety of challenges, including fairness problems and high access efficiency demand. The Media Access Control (MAC) layer plays a key role in improving access efficiency and ensure fairness. In this letter, we propose a new MAC protocol that utilizes multi-agent reinforcement learning (MARL) algorithm based on the multi-agent proximal policy optimization (MAPPO) to address these challenges. However, implementing a centralized training with decentralized execution (CTDE) paradigm in a MAC protocol can lead to signaling overhead issues. Therefore, we designed a joint action estimation method and periodic updating parameters scheme that effectively alleviates the communication overhead associated with CTDE. For comparison, we adopt a fully decentralized framework with low signaling overhead based on independent PPO (IPPO) algorithm. The simulation results indicate that our proposed MAPPO-MAC can outperform CSMA/CA and IPPO-MAC in both throughput and fairness with reduced communication overhead.**

***Index Terms*— Multiple access control protocol, online distributed learning, multi-agent deep reinforcement learning.**

I. Introduction

With the emergence of the Internet of Things (IoT), it has become crucial for next-generation wireless networks (NGWNs) to cope with such intensive demands while also improving Quality of Service (QoS). The media access control (MAC) layer is considered as a key layer in boosting access efficiency, ensuring fairness, and avoiding collisions [1]. However, many conventional distributed MAC protocols rely on randomization to mitigate collisions, such as carrier-sense multiple access with collision avoidance (CSMA/CA) used in Wi-Fi [2]. Although there have been numerous studies on the optimization of CSMA/CA and its variants, the lack of cooperation among stations (STAs) often leads to relatively inefficient and severe fairness issues. Therefore, it is essential to develop a new multiple access protocol at MAC layer to meet the growing demands of QoS in NGWNs.

Recently, multi-agent reinforcement learning (MARL) has shown promising results due to its ability to solve high-dimensional decision problems and quickly converge to cooperative policies. Considering a multi-access network where STAs aim to transmit to an access point (AP) independently, the network can be modeled as a multi-agent system where each STA is equipped with an agent to determine whether to access the channel on its own. As such, some MARL-based MAC protocols have been proposed. For instance, QMIX-advanced Listen-Before-Talk (QLBT) proposed in [3] is designed to achieve higher throughput and lower average latency than CSMA/CA. Based on MADDPG [4], the work [5] proposes a MAC protocol for jointly learning channel access policy and signaling policy, which achieves excellent performance in terms of throughput.

The aforementioned works primarily focus on the centralized training with decentralized execution (CTDE) framework, which employs the overall information of all agents during training and ensures that each agent can independently infer the optimal policy based on its local information. However, most CTDE algorithms perform centralized training in a simulated or laboratory setting, and each agent utilizes the trained strategy in a real-world setting, resulting in a fixed distributed policy that cannot adapt to high-dynamic wireless networks. Conversely, in [3] and [6], the AP collects real-time experiences for centralized training and iteratively updates the distributed policy in an online way. However, these methods incur significant signaling overhead, including training data in the uplink and network parameters in the downlink. Thus, frequent signaling exchanges between the AP and STAs can be unrealistic since it would occupy too much channel resources.

In this letter, we design a MARL-based MAC protocol based on the multi-agent proximal policy optimization (MAPPO) algorithm [7], which is referred to as MAPPO-MAC protocol. Our focus is on making the CTDE framework applicable to practical wireless networks by reducing signaling overhead in both the uplink and downlink. The main contributions are summarized as follows:

- We propose MAPPO-MAC protocol to achieve max-min fairness among STAs. It outperforms CSMA/CA and IPPO-MAC which utilizes a fully decentralized framework based on independent PPO (IPPO) algorithm [8], in terms of network throughput and Jain's fairness index.
- We propose methods to alleviate signaling overhead, including the joint action estimation method and periodic updating of policy parameters. The influence of update interval on convergence performance is also discussed.

II. System Model and Problem of Interest

A. System Model

As illustrated in Fig. 1, we consider that N STAs attempt to transmit packets to an AP in a time-slotted wireless network.

Manuscript received 19 July 2023; revised 23 August 2023 and 11 October 2023; accepted 11 October 2023. Date of publication 20 October 2023; date of current version 12 December 2023. This work was supported in part by National Key R&D Program of China 2022YFB2902002, and in part by National Natural Science Foundation of China under Grant 62271513. The associate editor coordinating the review of this letter and approving it for publication was Z. Qin. *(Corresponding author: Xinghua Sun.)*

Jianbin Xiao, Zhenyu Chen, Xinghua Sun, and Wen Zhan are with the School of Electronics and Communication Engineering, Sun Yat-sen University, Shenzhen 518107, China (e-mail: xiaojb7@mail2.sysu.edu.cn; chenzhy89@mail2.sysu.edu.cn; sunxinghua@mail.sysu.edu.cn; zhanw6@mail.sysu.edu.cn).

Xijun Wang and Xiang Chen are with the School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou 510006, China (e-mail: wangxijun@mail.sysu.edu.cn; chenxiang@mail.sysu.edu.cn).

Digital Object Identifier 10.1109/LCOMM.2023.3326267

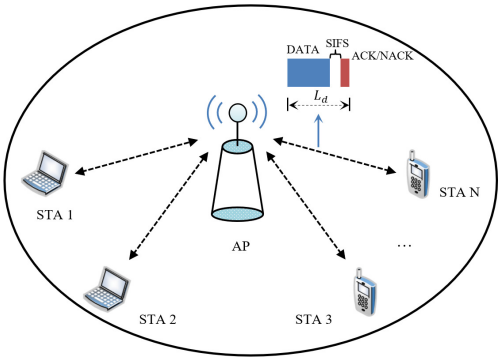

Fig. 1. System model: multiple STAs attempt to transmit packets to AP via a shared channel.

Each STA is assumed to be saturated[1] and all equidistant from the AP. To reduce redundant conflicts, STAs perform carrier sensing before accessing the channel, and transmissions are only permitted if the channel is perceived as idle. We consider the classic collision model [9], where one packet can be successfully decoded only if there are no concurrent transmissions. After each successful transmission, the AP will broadcast an acknowledgment (ACK), which is received by STAs after a short inter-frame space (SIFS). Conversely, if multiple STAs transmit simultaneously, then a NACK will be broadcast to indicate a collision.

B. Problem Formulation

In this letter, we consider the max-min fairness object to maximize the minimum throughput among STAs. By denoting throughput of STA n as Th^n, the optimization problem can be formulated as

$$\max_{\boldsymbol{\pi}} \min_{\forall n \in \{1,2,\ldots,N\}} Th^n$$
$$\text{s.t.} \sum_{n=1}^{N} Th^n \leq 1 \text{ and } Th^n \geq 0, \forall n \in \{1,2,\ldots,N\}, \quad (1)$$

where $\boldsymbol{\pi}$ denotes the access policy of all STAs.

III. MARL Formulation

A. Action

We define a_t^n as an action selected by the STA n at time step t. Here $a_t^n = 1$ means that STA n transmits a packet for L_d slots, and $a_t^n = 0$ means that it performs carrier sensing for one time slot. Furthermore, we denote the joint action as $\boldsymbol{a}_t = [a_t^1, \ldots, a_t^N]$.

B. Local State

We first define the next time step t' as the time slot $t+1$ or $t+L_d$ depending on the joint action $\boldsymbol{a}_t$. After the action a_t^n is executed, STA n can observe $o_{t'}^n \in \{-1, 0, 1\}$ at the next time step t', which represents collision, idleness and successful transmission respectively. $o_{t'}^n$ can be determined by ACK/NACK or the carrier sensing results. $l_{t'}^n = t' - t$ denotes the number of time slots that the action a_t^n lasts. The exponentially weighted average (EWA) throughput of the STA n is defined as:

$$\hat{Th}_{t'}^{\,n} = \begin{cases} (1-\beta^{l_{t'}^n}) \cdot 1 + \beta^{l_{t'}^n} \cdot \hat{Th}_t^{\,n} & \text{if } (a_t^n, o_{t'}^n) == (1,1), \\ \beta^{l_{t'}^n} \cdot \hat{Th}_t^{\,n} & \text{otherwise,} \end{cases} \quad (2)$$

where β is a discount rate. Note that $\hat{Th}_{t'}^{\,n}$ increases only if STA n transmits successfully at time step t. Similarly, EWA throughput of other STAs except n is defined as:

$$\hat{Th}_{t'}^{\,-n} = \begin{cases} (1-\beta^{l_{t'}^n}) \cdot 1 + \beta^{l_{t'}^n} \cdot \hat{Th}_t^{\,-n} & \text{if } (a_t^n, o_{t'}^n) == (0,1), \\ \beta^{l_{t'}^n} \cdot \hat{Th}_t^{\,-n} & \text{otherwise.} \end{cases} \quad (3)$$

The local observation of STA n at next time step t' is defined as $c_{t'}^n \triangleq [o_{t'}^n, l_{t'}^n, \hat{Th}_{t'}^{\,n}, \hat{Th}_{t'}^{\,-n}]$. We obtain the local state s_t^n by concatenating the local observations of past L time steps, i.e., $s_t^n \triangleq [c_{t-L+1}^n, \ldots, c_t^n]$.

C. Global State

The global observation is defined as $z_{t'} = [\boldsymbol{a}_t, o_{t'}, l_{t'}, \hat{\boldsymbol{Th}}_{t'}]$, where $\hat{\boldsymbol{Th}}_{t'} \triangleq [\hat{Th}_{t'}^{\,1}, \ldots, \hat{Th}_{t'}^{\,N}]$. Similar to the definition of local state, the global state at time step t is represented as $S_t \triangleq [z_{t-L+1}, \ldots, z_t]$.

D. Reward Function

Each STA receives a reward after performing an action. To guarantee the fairness, we assume the optimal policy is that the STA with the least throughput occupies the channel and other STAs wait, i.e.,

$$a_t^{n^*} = \begin{cases} 1 & \text{if STA } n == \arg\min \hat{\boldsymbol{Th}}_{\boldsymbol{t}}, \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

We award the actions that are consistent with optimal policy and punish those that are not. Hence the reward is defined as

$$r_{t'}^n = \begin{cases} 1 & \text{if action } a_t^n == a_t^{n^*}, \\ -1 & \text{otherwise.} \end{cases} \quad (5)$$

Therefore, the reward vector is obtained as $\boldsymbol{r}_{t'} \triangleq [r_{t'}^1, \ldots, r_{t'}^N]$.

IV. MARL-Based Multiple Access Algorithm

The overall CTDE framework of MAPPO is shown in Fig. 2. To be specific, an experience memory (EM) with capacity D resides in AP for storing experience tuple $\boldsymbol{e}_t = (S_t, \boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{r}_{t'}, S_{t'}, \boldsymbol{s}_{t'})$, which is obtained according to Algorithm 1. After randomly sampling mini-batch experiences from EM, AP performs centralized training and sends out trained actor network parameters for updating the network in STAs. In decentralized execution, each STA utilizes its actor network to select an action based on the local state. In the rest of this section, we first propose methods to alleviate signaling overhead, followed by an introduction to the neural network architecture and loss function of the proposed algorithm.

[1]The saturated condition is of more interest, with which network throughput is pushed to the limit.

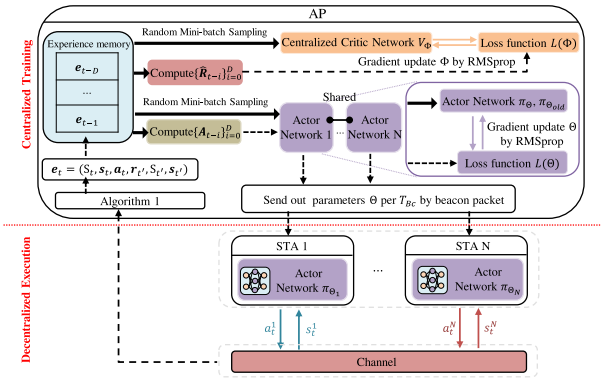

Fig. 2. An overall framework of MAPPO used in multiple access scenario.

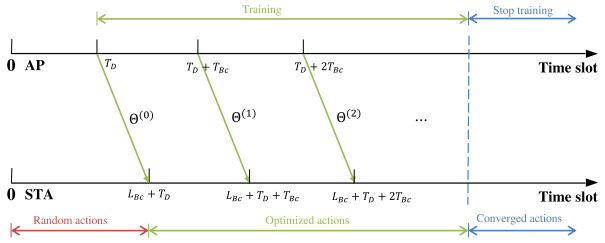

Fig. 3. Diagram of the proposed periodic broadcasting scheme.

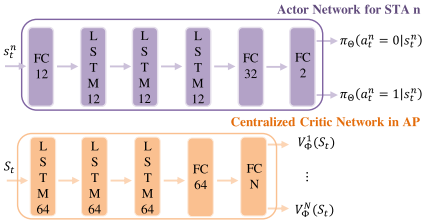

Fig. 4. Architecture of the critic and actor networks in MAPPO.

A. Signaling Overhead Alleviation

1) Uplink Signaling: In order to train the neural networks in AP, the experience tuple $\boldsymbol{e}_t$ needs to be collected at each time step. AP can directly obtain the joint local state $\boldsymbol{s}_{t'}$ and global state $S_{t'}$ by observing the channel state or judging whether a received packet is successful. The calculation of rewards $\boldsymbol{r}_{t'}$ requires joint action. However, upon collision, joint action $\boldsymbol{a}_t$ is agnostic because AP is unable to know which STAs collide with each other. A direct approach is that STAs send messages containing action information to AP. This, nevertheless, causes extra uplink signaling overhead. To avoid this overhead, we propose a method to estimate joint action in the following.

Since AP sends out the network parameters Θ_{old} to STAs for execution, AP knows the transmitting probability of each STA n, i.e., $\pi_{\Theta_{old}}\left(a_t^n = 1 \mid s_t^n\right)$. Therefore, AP can construct a matrix $\boldsymbol{P}_t$ as

$$\boldsymbol{P}_t = \begin{bmatrix} \pi_{\theta_{old}}\left(\hat{a}_t^1 = 0 \mid s_t^1\right) & \pi_{\theta_{old}}\left(\hat{a}_t^1 = 1 \mid s_t^1\right) \\ \vdots & \vdots \\ \pi_{\theta_{old}}\left(\hat{a}_t^N = 0 \mid s_t^N\right) & \pi_{\theta_{old}}\left(\hat{a}_t^N = 1 \mid s_t^N\right) \end{bmatrix}, \quad (6)$$

and then the joint action $\hat{\boldsymbol{a}}_t = [\hat{a}_t^1, \ldots, \hat{a}_t^N]$ can be sampled from $\boldsymbol{P}_t$. In particular, if $\|\hat{\boldsymbol{a}}_t\|_1 \geq 2$, then output $\hat{\boldsymbol{a}}_t$; otherwise resample until $\|\hat{\boldsymbol{a}}_t\|_1 \geq 2$.

2) Downlink Signaling: For reducing the downlink data, the parameters of actor networks Θ are shared among STAs. Even so, different actions can be taken by STAs due to their different local states. To avoid transmitting the actor parameters at each time step, we introduce a periodic broadcasting scheme using beacon packets that contain Θ. As shown in Fig. 3, STAs first use random actions to explore until the experience memory is full which occurs at time step T_D. At this point, AP performs centralized training and updates the neural network parameters. Then, AP transmits a beacon packet to STAs in every T_{Bc} time slots. The latest actor network parameters are included in the beacon packet[2] with the length of L_{Bc} time slots. Once the network performance has converged, the AP will stop centralized training, and does not need to broadcast the actor network parameters, then STAs make access decisions in a distributed manner based on the converged actor networks.

[2]The IEEE 802.11ac [10] defines the beacon frame's structure. Apart from the mandatory fields, there is a free space available for storing weight parameters. Moreover, the beacon interval, i.e., the periodic time to broadcast the beacon frame, can be adjusted through the beacon interval field in the frame body.

Algorithm 1 Experience Tuple Construction Algorithm

Initialization: joint local state $\boldsymbol{s}_t$, global state S_t, time step $t, t' = 0$.
while $t \leq T_{max}$ **do**
  $t' += 1$
  if Transmission finishes or channel is idle **then**
    Update channel observation $o_{t'}$
    $l_{t'} = t' - t$
    if $o_{t'} == 1$ or $o_{t'} == 0$ **then**
      Receive correct joint action $\boldsymbol{a}_t$
    else
      Estimate $\boldsymbol{a}_t$ by probability matrix $\boldsymbol{P}_t$ (6)
    end if
    Extract $\boldsymbol{\hat{T}h}_t$ from z_t in S_t and obtain $\boldsymbol{r}_{t'}$ by (4) $\sim$ (5)
    Calculate $\boldsymbol{\hat{T}h}_{t'}$ by (3), update $z_{t'}$ and $S_{t'}$
    for STA $n = 1$ to N **do**
      Set $o_{t'}^n = o_{t'}$, $l_{t'}^n = l_{t'}$ and $\hat{T}h_{t'}^{-n} = \sum_{j=1, j\neq n}^{N} \hat{T}h_{t'}^j$
      Update $c_{t'}^n$ and $s_{t'}^n$
    end for
    Store experience tuple $\boldsymbol{e}_t = (S_t, \boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{r}_{t'}, S_{t'}, \boldsymbol{s}_{t'})$
    $t \leftarrow t'$, $\boldsymbol{s}_t \leftarrow \boldsymbol{s}_{t'}$, $S_t \leftarrow S_{t'}$
  end if
end while

B. Neural Network Architecture

The architecture of the critic and actor networks is illustrated in Fig. 4. To effectively process a series of observations and extract informative features, we introduce Long Short-Term Memory (LSTM) layers into both networks. Specifically, the actor network of STA n takes the local state s_t^n as input, followed by a series of c_t^n that are fed into a

fully-connected (FC) layer activated by exponential linear unit (ELU) function before being processed by the LSTM layers. The output of the LSTM layers is then passed through two FC layers activated by tanh function, and a softmax function is applied in the final layer to obtain the policy distribution $\pi_\Theta(a_t^n \mid s_t^n)$. The number of actor parameters Θ are about $4,142 \times 4 \approx 16.57$KB if each parameter is stored as 4-byte float values. The critic network takes the global state S_t as input and feeds a series of z_t into LSTM layers. The output of the LSTM layers is then passed through two FC layers activated by tanh and linear function, which yield N value functions that guide the corresponding actor networks.

C. Loss Function

The centralized learning is implemented in AP based on a trajectory $\tau^n = \left(s_{t-D}^n, S_{t-D}, a_{t-D}^n, r_{t-D+1}^n, \ldots, s_t^n, S_t\right)$ for the STA n training, where D is the length of trajectory. The advantages $\{A_{t-i}^n\}_{i=1}^D$ and discounted reward-to-go $\{\hat{R}_{t-i}^n\}_{i=1}^D$ are computed from τ^n. We obtain $\tau_C = \{S_{t-i}, \hat{\boldsymbol{R}}_{t-i}\}_{i=1}^D$ and $\tau_A^n = \{s_{t-i}^n, a_{t-i}^n, A_{t-i}^n\}_{i=1}^D$ for training critic and actor network of STA n, respectively. Then, AP randomly samples mini-batches with size of B from τ_C and $\{\tau_A^n\}_{n=1}^N$.

The parameters of critic network Φ are updated by minimizing the loss function

$$L(\Phi) = \frac{1}{B}\sum_{b=1}^{B} \|\boldsymbol{V}_{\Phi,b} - \hat{\boldsymbol{R}}_b\|^2, \tag{7}$$

where $\boldsymbol{V}_{\Phi,b}$ is the centralized critic network output vector from global state S_b. As for the actor network, we calculate the loss function

$$L(\Theta) = \frac{1}{N \cdot B}\sum_{n=1}^{N}\sum_{b=1}^{B}\Big[\min\left(r_{\Theta,b}A_b^n, clip\left(r_{\Theta,b}, 1-\varepsilon, 1+\varepsilon\right)A_b^n\right) + \sigma\mathcal{S}\left[\pi_\Theta\left(s_b^n\right)\right]\Big], \tag{8}$$

where A_b^n is computed using the generalized advantage estimation (GAE) method [7], i.e., $A_b^n = \hat{A}_{old}^{GAE(\gamma,\lambda)}(s_b^n, a_b^n)$. $r_{\Theta,b}$ is the probability ratio $\frac{\pi_\Theta(a_b^n|s_b^n)}{\pi_{\Theta_{old}}(a_b^n|s_b^n)}$, $clip$ is the clip function which restricts $r_{\Theta,b}$ into the interval $[1-\varepsilon, 1+\varepsilon]$, $\mathcal{S}$ is the policy entropy, and σ is the entropy coefficient. The weights Θ of actor network are updated by gradient decent to minimize the loss function (8).

V. Performance Evaluation

A. Simulation Setup

In the simulations, we refer to IEEE 802.11ac [10], which is a common Wi-Fi standard used in practice. One time slot lasts for 9 μs. The data rate is set to 12.0 megabits per second (Mbps). Hence, the length of the beacon packet is given by $L_{Bc} = 16.57\text{KB} \div 12.0\text{Mbps} \div 9\mu\text{s} \approx 1198$ time slots. The parameters of each layer are initialized using orthogonal, and RMSprop optimizer is used for batch gradient descent. The other parameters of wireless scenario and MAPPO algorithm are summarized in Table I. In order to compare the performance, a series of algorithms are introduced as the baselines.

TABLE I
Parameters of Wireless Scenario and MAPPO Algorithm

Parameters	Value
Frequency band	5GHz
Channel bandwidth	20MHz
Data rate	12Mbps
Slot time	$9\mu s$
History length L	20
Simulation time T_{max}	25s
Data packet length L_d	120 time slots
Sample length D	256
Learning rate	0.002
Entropy coefficient σ	0.015
Batch size B	64
Discount factor γ	0.985
EWA discounted rate β	$0.98^{1/L_d}$
Range of GAE discount factor λ	0.995 to 0.1
Decay rate of λ	0.995
Actor clip factor ε	0.1

1) *Independent PPO (IPPO):* Both actor and critic network are independently deployed in each STA and take local state as input, resulted in a straightforward decentralized algorithm with low signaling overhead.[3] The neural network architecture is the same as that of MAPPO except that the number of outputs in critic is set to 1.
2) *CSMA/CA:* Basic mode: The initial backoff window size is set to 32, the maximum backoff phase is set to 5, and DIFS is set to 4 time slots; RTS/CTS mode: The setting of the basic mode remains unchanged, and the length of RTS and CTS is set to 2 time slots.
3) *Optimal algorithm:* All STAs are able to observe the joint local state $\boldsymbol{s}_t$ and joint action $\boldsymbol{a}_t$. Therefore, all STAs can always select the optimal action at each time step to maximize network throughput while maintaining fairness.

B. Performance Metrics

1) *Short-term throughput:* The short-term throughput of STA n at time slot t (i.e., Th_t^n) is obtained as the ratio of successful transmission time slots over the past $T = 50000$ time slots. We define *network throughput* as sum of all STAs' throughput, i.e., $\sum_{n=1}^N Th_t^n$.
2) *Fairness:* For the measurement of fairness, we introduce the Jain's fairness index (JFI) defined as following:

$$\mathcal{J}(\boldsymbol{Th}_t) = \frac{\left(\sum_{n=1}^N Th_t^n\right)^2}{N \cdot \sum_{n=1}^N \left(Th_t^n\right)^2}, \tag{9}$$

where $\boldsymbol{Th}_t = [Th_1^n, Th_2^n, \ldots, Th_t^N]$. If the throughput difference of each STA is minimum, $\mathcal{J}(\boldsymbol{Th}_t) = 1$. If one STA completely monopolizes the channel resources, $\mathcal{J}(\boldsymbol{Th}_t) = \frac{1}{N}$.
3) *Convergence time:* We define the convergence time as the time taken to converge since initialization. The convergence condition is defined as that the minimum network throughput first exceeds 95% of optimal performance and maintains it until the end of the simulation.

[3]In IPPO, $\hat{\boldsymbol{Th}}_t$ required for reward function cannot be obtained in a fully decentralized framework. Therefore, the AP should broadcast the reward vector $\boldsymbol{r}_{t'}$.

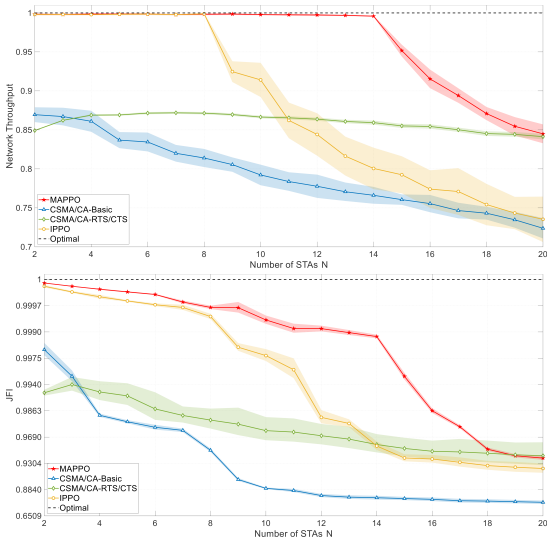

Fig. 5. Network throughput and fairness performance comparison under different values of N among MAPPO, CSMA/CA-Basic, CSMA/CA-RTS/CTS, IPPO and optimal algorithm. Each curve is averaged over ten different runs, and the shaded areas are within the standard deviation.

C. Simulation Results

To examine the performance of the proposed MAPPO-MAC protocol, we compare the network throughput and JFI with IPPO, CSMA/CA, and optimal algorithm under a distinct number of STAs when $T_{Bc} = 12000$. As shown in Fig. 5, in the basic mode of CSMA/CA, since it relies on randomization to mitigate collisions, STAs inevitably waste some time on waiting and collisions, which leads to degradation of throughput and fairness as N increases. Although in the RTS/CTS mode, its performance is substantially enhanced, there is a gap compared to the proposed scheme. As for IPPO, the throughput has a sharp decrease when N exceeds 8. Correspondingly, the JFI value also starts to decrease sharply from $N = 8$. In contrast, MAPPO still performs well thanks to the centralized critic network, which makes full use of global state and enhances policy learning. In particular, with MAPPO, the STA with the lowest EWA throughput would transmit, and thus take turns to occupy the channel, which ensures high efficiency as well as fairness among the STAs.[4]

To investigate the impact of the beacon interval T_{Bc} on the convergence performance, we conduct simulations with varying values of T_{Bc} when $N = 10$. As shown in Fig. 6, the converge time first decreases and then increases as T_{Bc} increases. On one hand, if T_{Bc} is too short, beacon packets are transmitted too frequently, and channel resources are primarily occupied by these packets instead of data packets. Consequently, the AP may not collect sufficient samples to train the actor networks, slowing down the training process and increasing the convergence time. On the other hand, if T_{Bc} is too long, the actor networks can't be updated promptly, causing the difference between $\pi_{\Theta_{old}}$ and π_{Θ} to be larger. As a result, the probability ratio $r_{\Theta,b}$ may be clipped at $1-\varepsilon$ or $1+\varepsilon$, leading to the first term in Eq. (8) making no contribution to the gradient of Θ, which can impact the centralized training process.

[4]It can be seen from Fig. 5 that the performance of MAPPO also deteriorates when $N \geq 14$, it is most likely due to insufficient experience samples. If virtual experiences are used to expand the experience memory [11], then a near-optimal performance can be achieved even when $N = 20$ with the same simulation settings. How to deal with a large number of STAs deserves further study.

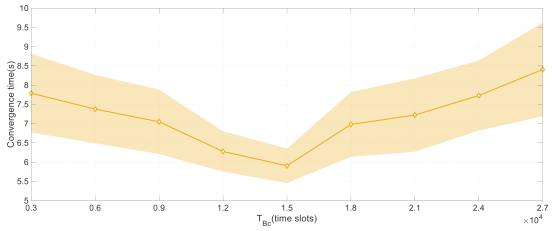

Fig. 6. Convergence time performance with different values of T_{Bc}. The curve is averaged over ten different runs, and the shaded areas are within the standard deviation.

VI. Conclusion

In this letter, we propose an online MARL-based MAC protocol named MAPPO-MAC protocol that efficiently estimates joint actions to reduce uplink signaling while updating actor network parameters of STAs periodically via the downlink beacon frame. We also introduce a novel reward function that ensures both throughput and fairness. Simulation results demonstrate that MAPPO-MAC protocol outperformed classic CSMA/CA and IPPO-MAC in terms of the network throughput and Jain's fairness index. Additionally, we investigate the impact of the beacon interval on convergence performance and find that setting a proper beacon interval is crucial for achieving fast convergence. Overall, our proposed MAPPO-MAC protocol provides a promising solution to the challenges of next-generation wireless networks.

References

[1] H. B. Pasandi and T. Nadeem, "Towards a learning-based framework for self-driving design of networking protocols," *IEEE Access*, vol. 9, pp. 34829–34844, 2021.

[2] A. Colvin, "CSMA with collision avoidance," *Comput. Commun.*, vol. 6, no. 5, pp. 227–235, Oct. 1983.

[3] Z. Guo, Z. Chen, P. Liu, J. Luo, X. Yang, and X. Sun, "Multi-agent reinforcement learning-based distributed channel access for next generation wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 5, pp. 1587–1599, May 2022.

[4] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6379–6390.

[5] M. P. Mota, A. Valcarce, J.-M. Gorce, and J. Hoydis, "The emergence of wireless MAC protocols with multi-agent reinforcement learning," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2021, pp. 1–6.

[6] Z. Chen and X. Sun, "Scalable multi-agent reinforcement learning-based distributed channel access," in *Proc. IEEE Int. Conf. Commun.*, May 2023, pp. 1–6.

[7] C. Yu et al., "The surprising effectiveness of PPO in cooperative, multi-agent games," 2021, *arXiv:2103.01955*.

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

[9] L. Dai and X. Sun, "A unified analysis of IEEE 802.11 DCF networks: Stability, throughput, and delay," *IEEE Trans. Mobile Comput.*, vol. 12, no. 8, pp. 1558–1572, Aug. 2013.

[10] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Enhancements for Very High Throughput for Operation in Bands Below 6GHz*, Standard 802.11ac, Dec. 2013.

[11] L. Zou et al., "Pseudo dyna-Q: A reinforcement learning framework for interactive recommendation," in *Proc. 13th Int. Conf. Web Search Data Mining*, Jan. 2020, pp. 816–824.