Deep-reinforcement learning for fair distributed dynamic spectrum access in priority buffered heterogeneous wireless networks

Siavash Barqi Janiar | Vahid Pourrahmadi

Department of Electrical Engineering, Amir Kabir University of Technology, Tehran, Iran

Abstract

In this paper, distributed heterogeneous wireless networks are studied and a spectrum access scheme that fairly allocates channels among users based on their request levels is investigated. To be closer to what happens in practice, users’ request levels to have different priorities are also considered, i.e. some packets are more important for the user to be transmitted than the others. Due to the distributed nature of the problem and since users are not able to coordinate with each other before transmission, a scheme based on reinforcement learning is proposed in which the network state is estimated based on the previous successful transmissions of packets in the network. The reinforcement learning method’s adaptive nature lets our method work in heterogeneous settings where there exist users with other medium access control protocols. The performance of the proposed scheme is evaluated in different network settings, and it is shown that it tries to implement a fair spectrum access policy (considering the packets’ priorities) while maximising the total throughput of the network.

1 INTRODUCTION

As the number of wireless devices using different channel access protocols increases, fair spectrum access in wireless networks becomes more complicated because of limited frequency resources. A brief survey of the most important spectrum access strategies can be found in [1]. Such network that transmitters use different medium access control (MAC) protocols, e.g. TDMA and slotted-ALOHA, is called a heterogeneous network. Recently, reinforcement learning (RL) [2] as a field of machine learning, had a huge impact on addressing network issues, for example in cognitive radio spectrum access [3–6]. Our proposed methodology utilises deep-RL (DRL) [7] to make a fair spectrum access strategy in a distributed heterogeneous wireless network (DHWN). Users utilising an agent which uses DRL for the decision-making process are called intelligent users in this paper.

Legacy protocols for channel sharing are not efficient enough when a network is as diverse as a heterogeneous network. There are techniques proposed by a few recent papers trying to make a fair model for channel multiple access.

The three schemes introduced in [8] for dynamic spectrum access with RL do a trade-off between throughput and fairness. The fairest scheme suggested by this job is sum-log rate but the problem with it is that implementing this scheme leaves the channel idle for around 40% of the time.

The authors in [9] supposed a heterogeneous network consisting of one agent and some other nodes using different MAC protocols such as slotted-ALOHA and TDMA. Although they have claimed that their algorithm with the \(\alpha\)-fairness objective can make fairness between more than one intelligent user, but we could not see any clue indicating these users’ coexistence with themselves under this objective in this paper.

Although these studies help fairness, all of them have considered that users always have packets to transmit (users are in saturation mode). An important feature of DHWN, in reality, is that users’ request level to transmit differ from each other. We can measure that by users’ packet buffer sizes. Incorporating buffer levels of users, the current RL-based access mechanisms are not applicable to determine which user should transmit in which time slot.

Considering different request level attribute of users necessitates a more complicated model, mostly because of the distributed nature of the wireless networks according to which there is not a specific central node that knows all buffer levels.
instead each user wanted to decide on its own whether to transmit or not. This problem is partially observable and thus for the proposed multi-agent RL method, we have added a few parameters to the observed state trying to make it as complete as possible. To see the behaviour of the system, the proposed method is then applied in a few different network settings which better illustrates the policy that agents are converged to. Users’ different MAC protocols add more complexity to intelligent users’ decision-making process, since their model should be able to estimate the behaviour of other users as well.

Firstly, we consider a network of three intelligent users, with finite number of packets without assigning any priority to them and we prove they can share the channel fair between themselves, without any interaction with each other and without any prior knowledge about the active number of users in the network, their MAC protocols and their buffer sizes. We compare this setting with a genie-aided scheme, which is not distributed and users know features (MAC protocol and buffer sizes) of other active users in the network. Then we compare the setting with a network of three slotted-ALOHA nodes. This is because to prove our users behave intelligent, not random. We compare these two networks from two aspects.

- The two networks having a node with a different packet arriving rate to prove our network can make fair decisions by letting agents with faster data arrival occupy the channel more.
- The two networks having five agents to show that our network prevents packet accumulation more than slotted-ALOHA.

Another important investigation we came up with in this paper was that we considered the priority of packets into our study, thus our proposed model can make desired decisions under this new useful circumstance.

In Section 2, we explain how our RL algorithm works. Then we introduce the structure of our environment in Section 3. Our proposed methodology has been provided in Section 4 and finally, in Section 5, we bring up results of our network and compare them with slotted-ALOHA mode and genie-aided scheme.

2 | REINFORCEMENT LEARNING AND POMDP OVERVIEW

RL [7] is a new way of viewing problems using neural networks to make decisions. In RL, there is an agent that is in interaction with the world such that it observes the universe and estimates the state (s) which is going to be the input of the neural network. Then the agent must decide which action (a) or sequence of actions it has to take considering the output of the policy network (Q-values). After the agent performs the chosen action, it gets a reward (r). Afterward, target values are going to be updated using the Bellman equation for the next iteration of training

\[
Q(s, a) \leftarrow r + \gamma \max_{a' \in A} Q(s', a').
\] (1)

In which \( A \) is the action space and \( \gamma \) is the discount factor and determines the horizon of the agent's insight into the future. In a scenario that states can not be estimated easily, the problem is going to be partially observable and as a result, the RL model is going to be partially observable Markov decision process (POMDP).

POMDP: In a POMDP model, instead of \( s \), we can observe only a part of the state at every time step named \( \tilde{s} \). The most important bottleneck in POMDPs is that there is not any function mapping observations to rewards, because there are cases which for one particular observation, more than one reward can be assumed.

3 | SYSTEM MODEL

DHWN consists of one access point (AP) and a few wireless users who use different MAC protocols (TDMA, slotted-ALOHA and DRL) for channel access (see Figure 1). The transmission frequency of all nodes are the same, so there is one channel shared between all users.

Let \( N \) be the number of users \( (u_i, i = 1, 2, ..., N) \) in the network. \( \tilde{u} \) is the set of all users but \( u \). Let \( a_i(t) \) be the action taken by \( u_i \) at time slot \( t \)

\[
a_i(t) \in \{0, 1\}
\]

where \( a_i(t) = 1 \) means \( u_i \) has attempted to transmit a packet at time slot \( t \) and \( a_i(t) = 0 \) means \( u_i \) was silent at that time slot.

Channel access mechanism: If \( u_i \)'s transmitted signal at time \( t \) be \( s_i(t) \), the signal received by the AP at that time step is going to be

\[
j(t) = \sum_{i=1}^{N} a_i(t)s_i(t) + n
\]
where $n$ is AWGN. We have assumed that the power of noise ($n$) is lower than signals ($s_i$) and so the network operates in an interference limited regime. In such a setting, the packets will be lost only if more than one user tries to transmit at the same time slot. If only one user transmit, the transmission would be successful. Mathematically, the transmission at the time slot $t$ is successful if $\sum_{i=1}^{N} a_i(t) = 1$.

**Packet arrival model:** In previous studies, clients have been assumed to be saturated, i.e. they always have packets to transmit. But in our research, each user gets some packets randomly in time. The time between packet arrivals are determined by exponential random function [10] with $1/\beta = \lambda$ parameter which has been assumed to be fixed during the process. We further assume that every user receives packets with up to three different priorities. The arrival rate of each type is denoted by $\lambda$. Each user stores arrived packets in a buffer associated with their priorities until it gets an opportunity to transmit. $b_i'(t)$ is $u_i$'s buffer size for packets with $j$ priority at time slot $t$ and $\bar{b}_i'(t)$ is the set of all users’ priority $j$ buffer sizes except $u_i$.

**Objective:** The objective of our study is to design a channel access method to keep the latency of packet transmission to its minimum. To satisfy this objective, at every time slot, the user with the largest buffer size should afford to transmit and other users stay silent, i.e. at each time slot the network should try to

$$
\min \left( \max_{\phi \in [1,2,..,N]} b_i'(t) \right).
$$

Equation (2) satisfies both fairness between users and the maximum possible total throughput for the channel. In this paper, we try to reach this objective using DRL and estimating states of it.

## 4 PROPOSED SCHEME

In this network, a specific time frame will be assigned to every TDMA user at the beginning, which determines the user's transmission status at every time slot. Slotted-ALOHA users transmit with a specific probability ($q$) at every time slot. Each intelligent user has been equipped with a DRL-using agent which instructs the user whether to transmit a packet or not. The same policy network has been shared between all agents of the users. As mentioned above, an optimum solution for the objective (2) can be reached if we were able to coordinate users such that at every time slot, only the user with the highest priority and the largest buffer size try to transmit.

In such a network, the best performance would be achieved if all users know the current buffer sizes of all other users. Although not practicable but we use this scheme as the upper bound when we report the performance of our proposed scheme.

In a practicable scheme, since users are not able to know everything about the network and only can observe part of it, in RL terminology, it is a POMDP model. Our contribution to solving this problem is to use additional knowledge that can be achieved from the network. More specifically, every time a user transmits successfully, AP can broadcast that user's buffer length along with its ACK signal. Other users, by recording this information, can enrich their observation, so they can perform better in the POMDP setting.

We consider finite buffer sizes for users. Each user has an initial buffer size at the beginning and some packets will burst into every user's buffer by time. Packets’ arrival times are determined by an exponential random function. We consider three priorities for packets: High, Medium and Low. Vividly, packets with High priority has the highest priority to transmit; every time a user occupies the channel, it transmits the highest available packet in priority.

**The observation vector** for $u_i$ at time $t$, $o_i(t)$, is defined as

$$
o_i(t) = [a_i(t), c, \text{ack}]$$

- $a_i(t)$: $u_i$’s taken action on current time slot
- $c$: the capacity of the channel which can be positive, zero or negative. Capacity is 1 if the channel is idle. If any user tries to transmit, the amount of capacity will decrease by 1.
- $\text{ack}$: This is going to be the acknowledgement (ACK) signal of the transmission for the user. If a user transmits and it is successful (no collision), $\text{ack}$ will be 1. Otherwise, it is 0.

The observation does not tell us anything about the user's buffer size and packet priority status versus other users'. Thus the DNN cannot distinguish between e.g. a state in which a user has a bigger buffer size than others and a state that it has a smaller buffer size than others. So we do not know anything about the state, meaning knowing features of the network such as other users’ buffer sizes and priority of their packets.

To derive the state at a particular time slot, we add memories to every user, so that whenever a user transmits successfully, other users that are listening to the network will memorize that user's buffer size considering the priority of the transmitted packet. Taking $u_i$, its memories, $m_i^j$ ($j = 1, 2, 3$), are the sets of the latest listened buffer sizes of all other users in which High $\equiv 3$, Medium $\equiv 2$ and Low $\equiv 1$. By using this information, we can add another element to the observation vector, $\Delta$, to estimate the states.

Considering maximum priority of $u_i$'s packets to be $\mu_{\eta_i}$ and $\eta$ to be $\max m_i^j$, $\Delta$ can be assigned as follows:

$$
\text{if } \exists \phi \in \{1, 2, 3\} \mid \text{max } m_i^\phi \neq 0, \phi > \mu_{\eta_i} \text{ or } \eta > \bar{b}_i' \text{ then } \Delta \leftarrow 1
$$

else if $\eta = \bar{b}_i' \neq 0$ then
$$
\Delta \leftarrow 2
$$
else if $\eta = \bar{b}_i' = 0$ then
$$
\Delta \leftarrow -1
$$
else
$$
\Delta \leftarrow 0
$$
end if
Finally, \( u_i \)'s estimated state at time slot \( t \) can be written as:

\[
x_i(t) = [a_i(t), c_i, ack_i, \Delta]
\]  

(3)

**Remark:** The proposed solution does not generate states accurately, because \( u_i \) should wait until another user transmits successfully to update its part of the memory associated with that user. As a result, \( u_i \)'s gotten reward at time slot \( t \) is not necessarily equal to \( \bar{r}_i \) at every time step. To improve the convergence speed, we use double deep Q-network (DQN) algorithm here \([7,12]\), according to which \( u_i \)'s transmitted packet priority is not necessarily equal to \( \bar{r}_i \) in which \( j \)'s transmitted packet priority is greater than \( u_i \)'s transmitted packet priority. As a result, \( u_i \) can only attempt to transmit at the beginning of every time slot.

**Policy network** of the agents is a DNN with two residual networks \([11]\) and two fully connected layers. The details of the policy network structure have been presented in Figure 2.

We concatenate the two most recent states as the input of the DNN. It is going to help DNN to sense changes of the state from the previous time slot and hence estimate the Q-values more accurately

\[
S_i(t) = [x_i(t-1), x_i(t)].
\]  

(4)

Thus, the trajectory for replay buffer is going to be

\[
\tau(t) = [S_i(t), a_i(t), r_i(t), S_i(t+1)]
\]  

(5)

where \( r_i(t) \) is \( u_i \)'s gotten reward at time slot \( t \) for taking action \( a_i(t) \).

Our policy is to choose an action with the largest Q-value at every time step. To improve the convergence speed, we use double deep Q-network (DQN) algorithm here \([7,12]\), according to which DNN \( 2 \) (target network) will be updated with lower frequency (after every 200 times that DNN \( 1 \) (current network) has been updated; \( l = 200 \)).

**Reward shaping:** We define \( \xi \) function of \( u_i \) in which if \( u_i \) transmits successfully \( \xi(u_i) = 1 \) and if it transmits but causes a collision, \( \xi(u_i) = 0 \). \( r_i(t) \) as the reward of \( u_i \) at time \( t \), the reward computation can be summarised as

\[
\begin{align*}
\text{if } (\xi(u_i) = 1 \text{ and } b_i^\mu > \eta \text{ and } m_j = 0 \text{ where } j > \mu) \\
\text{or } (\exists \alpha \in \beta_i | \xi(\alpha) = 1 \text{ and } (b_a^\mu > b_j^\mu \text{ or } \mu_a > \mu_j))
\end{align*}
\]

then

\[
r_i(t) \leftarrow 1
\]

else

\[
r_i(t) \leftarrow 0
\]

end if

Our proposed algorithm has been provided in Algorithm 1 where \( \theta \) is the DNN parameters and \( T \) is the number of time slots during the simulation. We assume every time slot is equal to the duration of transmission of a packet and receiving its ACK message\(^1\). Every user can only attempt to transmit at the beginning of every time slot.

Table 1 summarises the symbols used in the paper.

| Symbol | Description |
|--------|-------------|
| \( x_i \) | User index |
| \( c_i \) | User communication capacity |
| \( \Delta \) | Channel uncertainty |
| \( Q_1(S_i(t), a_i(t)) \) | Q-value of DNN1 |
| \( Q_2(S_i(t), a_i(t)) \) | Q-value of DNN2 |
| \( r_i(t) \) | Reward of user \( i \) at time slot \( t \) |
| \( \tau(t) \) | Trajectory for replay buffer |
| \( \Delta \) | Delay due to transmissions |
| \( \theta \) | DNN parameters |
| \( T \) | Number of time slots |

Our proposed algorithm has been provided in Algorithm 1 where \( \theta \) is the DNN parameters and \( T \) is the number of time slots during the simulation. We assume every time slot is equal to the duration of transmission of a packet and receiving its ACK message\(^1\). Every user can only attempt to transmit at the beginning of every time slot.

Table 1 summarises the symbols used in the paper.

**5 | SIMULATION RESULTS**

In our simulation, there are two training and testing phases. In train mode, there is more exploration and in test mode, we do not update the DNN parameters anymore. Let us assume a network of three intelligent users trying to share the channel between themselves. Table 2 lists hyper-parameters of the training.

To examine fairness, among users, at the first study we ignore packeters’ priorities and evaluate the proposed method with the optimum in Section 5.1. Afterward, we compare the behaviour of intelligent users with ALOHA users in Section 5.2. Then in Section 5.3, we test out the coexistence of users with different MAC protocols in a DHWN. After proving the performance of the proposed scheme, we study the behaviour

\(^1\) 2X (processing delay + transmission delay + propagation delay). Source: [13]
TABLE 1 List of symbols

| Symbol | Definition |
|--------|------------|
| \( u_i \) | user \( i \) |
| \( \tilde{u}_i \) | set of all users but \( u_i \) |
| \( b_j(t) \) | \( u_i \)'s buffer size for packets with \( j \) priority at time slot \( t \) |
| \( \tilde{b}_j(t) \) | set of all users' priority \( j \) buffer sizes except \( u_i \) |
| \( m_j \) | the sets of latest listened buffer sizes of all other users |
| \( \eta \) | \( \max m_j \) |
| \( \mu_{u_i} \) | maximum priority of \( u_i \)'s packets |
| \( \lambda = \frac{1}{\beta} \) | exponential random function parameter |
| \( q \) | slotted-ALOHA users' transmission probability |
| \( a(t) \) | the action taken by \( u_i \) at time slot \( t \) |
| \( c \) | capacity of the channel |
| \( \text{ack} \) | acknowledgment (ACK) signal of the transmission |
| \( s_i(t) \) | \( u_i \)'s observation at time slot \( t \) |
| \( r_i(t) \) | \( u_i \)'s state's additional parameter than observation |
| \( r_i(t) \) | \( u_i \)'s estimated state at time slot \( t \) |
| \( S_j(t) \) | input of the Deep-Neural Network (DNN) |
| \( \tau(t) \) | trajectory for replay buffer |
| \( \xi(u_i) \) | \( u_i \)'s transmission function |
| \( \tau_i(t) \) | \( u_i \)'s gotten reward at time slot \( t \) |
| \( Q(s,a) \) | \( s \) can be a single state or a vector of states |
| \( \gamma \) | discount factor |
| \( \epsilon \) | transmission uncertainty |
| \( \epsilon_d \) | \( \epsilon \)'s discount factor |
| \( N \) | number of users |
| \( T \) | testing time |
| \( \Theta \) | DNN parameters |
| \( l \) | update period of the target DNN |

TABLE 2 Hyper-parameters of the simulation

| Hyper-parameter | Assigned value |
|-----------------|----------------|
| \( N \) | 3 |
| \( T \) | 100 000 time slots |
| \( \epsilon \) | 0.6 |
| \( \epsilon_d \) | 0.99999 |
| replay buffer size | 4500 |
| batch size | 96 |
| \( l \) | 200 |
| \( \gamma \) | 0.9 |
| \( \lambda \) | \( 1/300 \) |
| initial buffer size | 60 |
| \# of generated packets each time | 40 |

of our network after considering priorities for packets in Section 5.4.

5.1 Fairness among users

In Figure 3, the bottom and upper sub-plots indicate users’ buffer sizes at every time slot and moving averaged throughput (window = 1000 time slots) for every agent, respectively. The x-axis represents the number of time slots elapsed from the beginning of the simulation. Users in Figure 3(a) are using RL agents for decision-making. As can be seen, the average throughput of all three users is close together which indicates fairness. The buffer size curves [the lower subplot in Figure 3(a)] are non-monotonic which means the proposed algorithm prevents users’ packets to be accumulated. To see how good the performance of the proposed scheme is, Figure 3(b) presents what could be achieved if at all users we had exact information about all other users’ buffer sizes. We refer to this case as the genie-aided scheme which is not a practical scheme due to the high signaling overhead and the delay that it will cause (but it is useful for comparison).

To see how the algorithm adapts itself in time, the sum of users’ throughput for a) the proposed scheme and b) the upper-bound (genie-aided scheme) have been shown in Figure 3(c). By comparing the two curves, it is clear that as time passes, our network gets closer to upper-bound.

Finally in Figure 3(d), what happens in the channel has been shown when we use the proposed scheme and we compared it with the genie-aided case. As expected, in the genie-aided case no collision happens. Other than that we can observe that the performance of the proposed scheme is close to the genie-aided one and more importantly, close together which is expected since packet input rates for all users are the same (\( \lambda \) parameter for all users are the same).

Another comparison with the genie-aided scheme has been shown in Figure 4 by testing the network with different traffic rates, which has been implemented by altering \( \lambda \). As \( \lambda \) increases, packets arrive faster to each user. As a result, individual users’ throughput and overall collision of the DHWN increases (and the idleness of the channel decreases until it gets to 0). As there is no collision in the genie-aided scheme, its throughput saturates when it ends up with 0 idleness (from \( \beta = 100 \) on, in Figure 4). The gap between the two curves is because of the collisions happening in the proposed scheme and it is inevitable. The reason is that in a distributed network there is no way making the number of collisions zero. It can be seen that as long as \( \beta \) is less than about 270, the proposed scheme is working very close to the genie-aided case. For every specific \( \lambda \) in this figure, the network has been run for 20 000-time steps. The reason is that the suitable time that the proposed scheme is close enough to the genie-aided one on average, is around 20 000 as can be seen in Figure 3(c).
To study agents’ buffer sizes, during a specific time frame Figure 5(a) illustrates three important behaviour of the trained network: a) after adding packets, the agent starts sending and no one corrupts it. b) just after the agent’s packets finish, the agent stops sending or significantly decreases the transmission rate. c) right after the agent’s buffer size becomes equal to others’ buffer size, the agents with equal buffer sizes try passing channel to each other in turn. These are all expectations from an intelligent network and can be seen in the genie-aided scheme as well [see Figure 5(b)]

5.2 Comparison with slotted-ALOHA

To better see the performance of the proposed scheme, we have compared its performance with a setting that instead of the RL scheme, users access the channel using the slotted-ALOHA MAC protocol. When there are a few users in the network, maybe slotted-ALOHA be desirable because of its simplicity but there are two crucial drawbacks with slotted ALOHA.

- When the number of users increases, the number of packets accumulated in users’ buffer sizes is significantly numerous in comparison with a network of intelligent users (see Figure 6). It is because there is not any collision avoidance and opportunity finding mechanisms in slotted-ALOHA users.
- When $\lambda$ of users are different from each other, slotted-ALOHA can not guarantee any fairness between users. Figure 7(a) and 7(b) shows a network that $n_0$’s $\lambda$ parameter is different from other users in slotted-ALOHA and proposed scheme, respectively. As it has been compared in Figure 7(c), $n_0$ in slotted-ALOHA protocol tries to accesses the channel as much as other agents, although its $\lambda$ is lower than other users. But in the proposed scheme, it affords less than other users to access the channel due to its lower demand rate.
Now we want to test out the proposed scheme in a more complex situation where besides intelligent and slotted-ALOHA users, there are TDMA users as well. The purpose of such a setting is to examine the performance of the proposed methodology under more sophisticated circumstances. The DHWN consisting of a slotted-ALOHA, a TDMA user and an intelligent user has been simulated and the performance of it can be analysed with the help of Figure 8(a). As can be seen in the picture, all of the users accomplished to send their packets during the test time. Even when TDMA has a lower buffer size, the intelligent user lets it to transmit because it has learned the TDMA user’s transmission pattern [Figure 8(b), have been observed around 80% of the Monte-Carlo experiments]. Although the intelligent user has the highest buffer size, in t = 1 and 7 it does not try to transmit because of the TDMA user.

To see another example of DHWN, we tested out a network including 2 intelligent users, 2 slotted-ALOHA, and 1 TDMA node. Figure 9 shows these users’ behaviour in the network.

### 5.4 Priority

Adding priority to packets does not change the principles of the proposed scheme, it just necessitates more conditions to be considered by the agent. Let us take a network of three intelligent users once again. The behaviour of this network considering priorities for packets have been studied in Figure 10. As can be seen, the user with the highest priority and the largest buffer size usually transmits before other users, take a look at the highlighted parts in Figure 10.

Finally, a DHWN with all assumptions has been simulated and the average throughputs of users have been illustrated in Figure 11. A DHWN consisting of a slotted-ALOHA, TDMA, and an intelligent user, everyone with three different packet priorities. As can be seen in Figure 11, all users were able to transmit all the three different packets arrived in their buffers and
the throughput curves of users show how fair channel has been shared between users at every time step.

6 | CONCLUSION

In this paper, we studied the fairness and throughput of the channel in a DHWN by considering users’ limited buffer sizes and different priorities between packets. Since methodologies proposed by previous studies are not efficient in this setting, we took advantage of information that the AP provides for the users of the network. So by using the simplest DRL algorithm (value iteration), we proposed a scheme that behaves near-optimal. By using our methodology, the wireless network
is going to behave fairly regarding users’ buffer size and avoid packet over-accumulation in the users’ buffers.

REFERENCES

1. Zhao, Q., Sadler, B.: A survey of dynamic spectrum access. IEEE Signal Process. Mag. 24(3), 79–89 (2007)
2. Sutton, R., Barto, A.: Reinforcement learning: An introduction. MIT Press, Cambridge (1998)
3. Li, H.: Multiagent learning for aloha-like spectrum access in cognitive radio systems. J. Wireless Com Netw. 2010(1), 876216 (2010)
4. Yau, K-L., et al.: Enhancing network performance in distributed cognitive radio networks using single-agent and multi-agent reinforcement learning. In: IEEE Local Computer Network Conference pp. 152–159. IEEE, (2010)
5. Wu, C., et al.: Spectrum management of cognitive radio using multi-agent reinforcement learning. In: Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: Industry track. International Foundation for Autonomous Agents and Multiagent Systems pp. 1705–1712. (2010)
6. Bkassiny, M., et al.: Distributed reinforcement learning based MAC protocols for autonomous cognitive secondary users. In: 2011 20th Annual Wireless and Optical Communications Conference (WOCC) pp. 1–6. IEEE, (2011)
7. Levine, S.: Deep reinforcement learning, CS 294-112, UC Berkeley. (2018). http://rail.eecs.berkeley.edu/deeprlcourse-fa18/
8. Naparstek, O., Cohen, K.: Deep multi-user reinforcement learning for dynamic spectrum access in multichannel wireless networks. IEEE Global Commun. Conf. 247, 529–551 (2017)
9. Yu, Y., et al.: Deep-reinforcement learning multiple access for heterogeneous wireless networks. IEEE Int. Conf. Commun. 247, 529–551 (2018)
10. Jensen, J., et al.: Harmonic exponential modeling of transitional speech segments. In: 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.00CH37100) IEEE, (2000)
11. He, K., et al.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778. IEEE, (2016)
12. Hasselt, H., et al.: Deep reinforcement learning with double q-learning. In: Proceedings of the 30th AAAI Conference on Artificial Intelligence, pp. 2094–2100. (2016)
13. Kurose, J., Ross, K.: Computer networking: a top-down approach featuring the internet. Addison-Wesley, Reading (2001)

How to cite this article: Barqi Janiar S, Pourahmadi V. Deep-reinforcement learning for fair distributed dynamic spectrum access in priority buffered heterogeneous wireless networks. IET Commun. 2021;15:674–682. https://doi.org/10.1049/cm2.12098