This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2020.2971212, IEEE Transactions on Communications

Intelligent Sharing for LTE and WiFi Systems in Unlicensed Bands: A Deep Reinforcement Learning Approach

Junjie Tan, *Student Memeber, IEEE*, Lin Zhang, *Member, IEEE*, Ying-Chang Liang, *Fellow, IEEE*, and Dusit Niyato, *Fellow, IEEE*

***Abstract*—Operating LTE networks in unlicensed bands together with legacy WiFi systems is deemed as a promising technique to support explosively growing mobile traffic. In conventional LTE/WiFi spectrum sharing schemes, LTE systems need to know WiFi traffic demands for optimizing system parameters to protect WiFi systems, for which the two systems are required to cooperate with each other via signallings exchanges. However, it is difficult to establish a dedicated channel among the two independent systems for exchanging signallings. Hence, in this paper, we propose an intelligent duty-cycle medium access control protocol to realize the effective and fair spectrum sharing between LTE and WiFi systems without requiring signalling exchanges. Specifically, we first design a duty-cycle spectrum sharing framework, which allows an LTE system to share the spectrum with a WiFi system by using time sharing. After that, we develop deep reinforcement learning (DRL)-based algorithms to learn WiFi traffic demands by analyzing WiFi channel activity, e.g., the idleness/business of WiFi channels, which can be observed by the LTE system via monitoring WiFi channels. Based on the learnt knowledge, the LTE system can adaptively optimize LTE transmission time to maximize its own throughput and meanwhile to provide sufficient protection to the WiFi system. Simulation results show that, in terms of LTE throughput and WiFi protection, the performance of the proposed intelligent scheme can approach that of the genie-aided exhaustive search algorithm, which needs the perfect knowledge of WiFi traffic demands through massive signalling exchanges and is of high computational complexity.**

***Index Terms*—Long-term evolution (LTE), deep reinforcement learning (DRL), unlicensed band.**

I. Introduction

As smart mobile devices and various emerging mobile applications are becoming popular, mobile traffic is growing explosively, which puts heavy burdens on current mobile cellular networks [2]. Since radio spectrum is quite limited, it is difficult to expand network capacity through using more licensed bands [3]. It has thus been proposed to enable *long term evolution* (LTE) networks to operate in unlicensed bands, in addition to their licensed bands [4]–[6].

When an LTE system shares the same unlicensed band with a legacy WiFi system, the most challenging issue is to prevent the WiFi system from severe performance degradation, while maximizing LTE throughput. The *carrier sense multiple access with collision avoidance* (CSMA/CA) protocol employed by WiFi systems regulates that WiFi *access points* (APs) and users need to listen before transmissions, and only access the channel when the channel is idle [7]. Hence, WiFi systems could be vulnerable if LTE systems operate in the same bands [8]–[10]. To address this issue, there have been proposed two types of LTE/WiFi spectrum sharing methods, namely *listen-before-talk* (LBT) methods and duty-cycle methods. Typically, the LTE system equipped with LBT methods and that with duty-cycle methods are respectively called *LTE-license-assisted access* (LTE-LAA) system and *LTE-unlicensed* (LTE-U) system [5].

For LBT methods, LTE systems perform *clear channel assessment* (CCA) before accessing the channel and transmit only if the channel is idle [11]–[15]. A simplified CSMA/CA protocol with a fixed backoff stage is proposed in [11], [12]. In particular, *contention window* (CW), power allocation, and user association are jointly optimized in [11] to satisfy the *quality of service* (QoS) requirements of LTE users while minimizing the collision probability of WiFi users. [12] maximizes the number of WiFi users offloaded to an LTE system by jointly optimizing CW, subcarrier assignment, power allocation, and user association, under the condition that the QoS requirements of users in both LTE and WiFi systems can be satisfied. Moreover, another novel LBT scheme is studied in [13] and [14], where the sum throughput of unlicensed bands is maximized by the joint optimization of LTE transmission time and user association. In addition, [15] focuses on the hidden terminal problem caused by CCA and handles the problem by the orchestration of both scheduling access and random access.

For duty-cycle methods, LTE systems access channels periodically [16]–[19]. In particular, a small cell scenario is considered in [16], in which LTE small cells transmit *almost blank subframes* (ABSs) to mitigate the interference to a WiFi system. [17] proposes a duty-cycle mechanism to guarantee the proportional fairness between an LTE system and a WiFi system by optimizing channel access probabilities and transmission durations. [18] investigates the user offloading between a WiFi system and an LTE system, and derives

This work was supported in part by the National Natural Science Foundation of China under Grants 61631005, U1801261 and 61801101, and the National Key R&D Program of China under Grant 2018YFB1801105, and the 111 project under Grant B20064.

This paper was partly presented in IEEE International Conference on Communications (ICC) 2019 [1].

J. Tan, L. Zhang, and Y.-C. Liang are with the National Key Laboratory on Communications, and also with the Center for Intelligent Networking and Communications (CINC), University of Electronic Science and Technology of China (UESTC), Chengdu, China (emails: tan@kuspot.com, linzhang1913@gmail.com, and liangyc@ieee.org).

D. Niyato is with the School of Computer Science and Engineering, Nanyang Technological University, Singapore (email: dniyato@ntu.edu.sg).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2020.2971212, IEEE Transactions on Communications

the optimal offloading strategy and duty cycle allocation to maximize the minimum average throughput of each user. In addition, [19] considers the cochannel interference caused by the coexistence of multiple LTE systems, and proposes a duty cycle coordination method to enhance the overall LTE throughput.

A. Motivations

By analyzing existing LTE/WiFi spectrum sharing schemes, we find that most of them assume LTE systems to know perfectly the information about WiFi systems, e.g., WiFi traffic demands or the number of WiFi users. In those schemes, only after receiving the required information from WiFi systems can LTE systems optimize system parameters to protect WiFi systems in terms of collision probability or the throughput of WiFi systems. However, in reality, such information about WiFi systems may not be available to LTE systems due to the difficulty in establishing a dedicated channel among the two independent systems.

In the literature, there have been some studies attempting to address the signalling issue mentioned above, e.g., [11], [17], and [20]. In [11] and [17], it is assumed that the method given in [21] can be used to infer the number of WiFi users by observing the collision probability of a WiFi system, but the results are derived in the WiFi system operating without LTE systems. Hence, this method cannot be applied directly when the WiFi system is sharing the spectrum with an LTE system. [20] proposes another scheme, where an LTE system collects historical traffic data periodically and adopts *deep learning* (DL) to allocate resources in advance. In this scheme, instantaneous signalling exchanges are not required, but it is still expensive for LTE systems to fetch complete historical information about WiFi systems periodically. Moreover, the scheme in [20] cannot respond rapidly to the improper resource allocation caused by prediction errors, which also limits its applications. As such, it is highly desired to develop a novel spectrum sharing scheme that can address the above issues.

B. Our Contributions

This paper considers an LTE system to share the same unlicensed band with a WiFi system in the absence of signalling exchanges. The designed objective is to maximize LTE network capacity and simultaneously protect the WiFi system, i.e., the LTE system should exploit the unlicensed band while fulfilling WiFi traffic demands. However, WiFi traffic demands are typically time-varying and unknown to the LTE system without signalling exchanges. This makes conventional mathematical methods inapplicable.

On the other hand, we notice that the LTE system can monitor WiFi channels to obtain WiFi channel activity, e.g., the number of idle/busy slots. In fact, WiFi channel activity is substantially correlated with WiFi traffic demands, which are also correlated with a hidden WiFi traffic model. Note that the mathematical relationship between WiFi channel activity and WiFi traffic demands is unknown to the LTE system, meanwhile the WiFi traffic model is also unavailable at the LTE system. Hence, we adopt *deep reinforcement learning* (DRL) to learn them from historical data. With the learnt knowledge, the LTE system can infer future WiFi traffic demands based on the observed WiFi channel activity. Eventually, LTE system can automatically optimize its parameters to maximize its network capacity and meanwhile protect the WiFi system without signalling exchanges. Following this idea, in this paper, we develop an intelligent duty-cycle *medium access control* (MAC) protocol, which consists of a duty-cycle spectrum sharing framework and DRL-based algorithms. In particular, the duty-cycle spectrum sharing framework is designed to allow the two systems to share the same unlicensed band orthogonally in time domain. Then, DRL-based algorithms are designed for the LTE system to learn WiFi traffic demands by analyzing observed WiFi channel activity, and accordingly optimize LTE transmission time to achieve the designed goal. To highlight ours contributions, we summarize this paper in the following.

- We design an intelligent duty-cycle MAC protocol to achieve effective and fair spectrum sharing between LTE and WiFi systems in the absence of signalling exchanges. In particular, we develop DRL-based algorithms for the LTE system to adaptively optimize LTE transmission time to exploit the unlicensed band while satisfying WiFi traffic demands.
- We propose to indicate WiFi traffic demands with WiFi channel activity, which has substantial correlations with WiFi traffic demands and can be observed directly by LTE systems via monitoring WiFi channel. As such, with the developed DRL-based algorithms, the LTE system can learn WiFi traffic demands by analyzing the observed WiFi channel activity.
- We design the intelligent duty-cycle MAC protocol by considering that the WiFi system can be either delay-sensitive or delay-tolerant. In particular, delay-tolerant WiFi systems allow delayed transmissions of packets, while delay-sensitive WiFi systems need to transmit packets as soon as possible. In this way, the proposed scheme can be applied widely.
- Simulation results demonstrate that the performance of the developed DRL-based algorithms can approach that of the genie-aided exhaustive search algorithm, which requires the perfect knowledge of WiFi traffic demands and has high computational complexity.

C. Related Work on DRL

As wireless networks are getting more complicated, traditional modelling and optimization methods are becoming less effective and inefficient. On the other hand, in the area of computer science, data-driven methods are attracting great attention due to the great success of computer vision and natural language processing in recent years. Among them, DL is the most important technique [22]. Using *deep neural networks* (DNNs), DL can find the relationship hidden in the training data, and then use it to make accurate predictions even in a unknown situation. Without the needs for modelling, the data-driven nature of DL is an appealing feature in system design, especially for complicate wireless networks

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2020.2971212, IEEE Transactions on Communications

[23], [24]. By now, some work has been done to optimize wireless networks using DL, e.g., transmitter-receiver design [25], channel estimation [26], [27], radio signal classification [28], and spectrum sensing [29].

Although DL is powerful in making predictions, such as classification and regression, it shows weakness in solving decision-making problems. In fact, decision-making problems are specialized in by another important branch of machine learning techniques, called *reinforcement learning* (RL) [30]. RL aims to learn the optimal policy in a dynamic environment by iteratively making decisions and receiving feedbacks for policy improvement. However, the capability of traditional RL techniques is limited by the curse of dimensionality, and it will be inapplicable in large-scale systems [30]. To overcome this problem, DRL has thus been proposed. By integrating DL into RL, DRL uses DNNs to overcome the curse of dimensionality and hence is able to solve large-scale problems effectively [31]. To date, there are some studies applying DRL successfully to wireless networks [32], [33], including *vehicle-to-anything* (V2X) networks [34], [35], *cognitive radio* (CR) [36], [37], and MAC protocol design [38], [39]. In particular, a DRL-based adaptive modulation and coding scheme is developed in [36] for primary users to learn the interference pattern of secondary users. In [37], DRL is adopted for multiple users to access the spectrum in a distributed manner without requiring message exchanges. In [38] and [39], DRL is used to improve the MAC protocol design. Particularly, a DRL-based universal MAC protocol is proposed in [38], which maximizes the spectrum efficiency in a heterogeneous network involving other multiple access schemes. In [39], the conventional CSMA/CA protocol is improved for densely deployed WiFi networks, in which DRL is adopted to learn the optimal CW for each WiFi node to improve overall throughput.

D. Organization of the Paper

In Section II, we provide the system model, which is followed by the design of the spectrum sharing framework in Section III. In Section IV, we develop DRL-based algorithms to achieve intelligent sharing between LTE and WiFi systems. After that, the performance of the proposed scheme is demonstrated through extensive simulation results in Section V, and the conclusions are finally drawn in Section VI.

II. System Model

A. LTE/WiFi Spectrum Sharing System

We consider an LTE/WiFi spectrum sharing system as depicted in Fig. 1, including an LTE system with saturated traffic and a WiFi system with unsaturated traffic. For the LTE system, an LTE BS shares the same unlicensed band with a legacy WiFi system to serve some *LTE user equipments* (L-UEs). For the WiFi system, there exist a WiFi AP and several *WiFi user equipments* (W-UEs) contending to transmit data according to the CSMA/CA protocol [7]. For simplicity, both the WiFi AP and the W-UEs are called *WiFi stations* (W-STAs).

A typical application scenario is the spectrum sharing between an LTE system deployed in a hotspot area and a nearby private WiFi system. In the hotspot deployment [40], the LTE system is saturated due to tremendous traffic demands from lots of L-UEs. Meanwhile, due to private access, the WiFi system has limited active users and unsaturated traffic. In fact, WiFi systems are typically under-utilized in reality [41], and thus leave lots of white spaces in unlicensed bands [42], especially for the 5GHz UNII bands with the mean occupancy below 5% [43]. Hence, the LTE system can exploit the unlicensed band under-utilized by the WiFi system to enhance LTE network capacity.

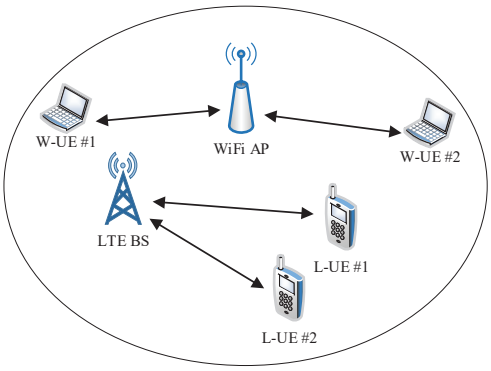


Fig. 1. The considered LTE/WiFi spectrum sharing system.

Similar to [7], the time of the designed system is slotted, and each slot is with σ time units. We denote by T_s (in slots) and T_c (in slots) the channel busy time for a successful transmission and that for a collision, respectively. Since it is difficult to establish an additional control channel among independent systems in practice, we assume that there are no dedicated channels between the LTE and WiFi systems for signalling exchanges.

B. Traffic Model of the WiFi System

We consider the WiFi system to be unsaturated, i.e., it does not always have data packets to transmit, for which every W-STA generates packets following a Poisson process with a rate of λ (packets per T_s slots). Moreover, the W-STAs arrive in and depart from the WiFi system dynamically. Similar to [44], we consider a discrete-time model to describe the changes of the number N of the W-STAs in the WiFi system, and N varies every T_N slots. In general, the dynamics of N can be modeled as a *discrete-state Markov chain* (DSMC) [45]. The number of W-STAs N is assumed to be bounded by the maximum value $N_{\max}$ and the minimum value $N_{\min}$. Hence, the transition matrix of the DSMC can be represented by $\mathbf{P} = [p_{i,j}], i, j \in \{N_{\min}, N_{\min}+1, \ldots, N_{\max}\}$, where $p_{i,j}$, the i-th row and j-th column element of $\mathbf{P}$, denotes the probability that N transits from i to j. The specific values of the elements in $\mathbf{P}$ depend on the pattern that the W-STAs follow in practice. For instance, if the W-STAs depart and arrive following Poisson processes, $\mathbf{P}$ can be determined by a Skellam distribution [44].

III. Spectrum Sharing Framework

In this section, we propose a duty-cycle spectrum sharing framework for the intelligent MAC protocol. The frame structure is illustrated in Fig. 2. As the figure shows, we divide each

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2020.2971212, IEEE Transactions on Communications

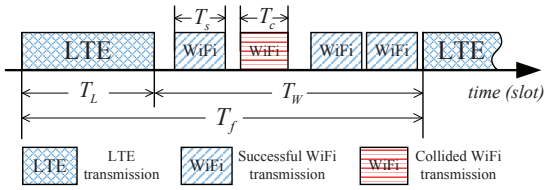


Fig. 2. Frame structure of the duty-cycle spectrum sharing framework.

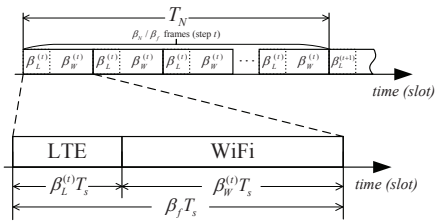


Fig. 3. The relations among steps, frames, and slots.

frame into two parts, including the LTE transmission part and the WiFi transmission part. Denote by T_f the frame length, T_L the LTE transmission time, T_W the WiFi transmission time, respectively, and they are measured in slots. In particular, we design the frame length T_f to be a fixed value. When a frame starts, the LTE system transmits and keeps the channel busy for T_L slots. In the rest of the frame, the LTE system stops transmission, and W-STAs contend to transmit for T_W slots. During the LTE transmission time, the channel will be sensed busy, and consequently W-STAs will freeze the backoff procedures and keep waiting. During the WiFi transmission time, W-STAs contend for transmission if they have packets in the buffers. At the end of the frame, undelivered packets will be dropped by W-STAs, in order to keep the freshness of the packets.

Compared with LBT methods, the designed duty-cycle framework has two primary benefits. Firstly, due to fixed frame length, the start time of LTE transmissions is deterministic, which facilitates the synchronization between L-UEs and the LTE BS. In contrast, the start time of LTE transmissions in LBT methods is determined by the WiFi traffic because LBT methods perform CCA before each LTE transmission. This causes additional synchronization overheads. Secondly, using the designed duty-cycle framework, the LTE system can guarantee different levels of fairness for the WiFi system by simply tuning LTE transmission time T_L. For simplicity, T_f, T_L, T_W, and T_N are normalized by T_s in the sequel, thus we define $\beta_f = T_f/T_s$, $\beta_L = T_L/T_s$, $\beta_W = T_W/T_s$, and $\beta_N = T_N/T_s$. We consider β_f, β_L, β_W, and β_N to be integers, for which T_s can be replaced with a similar value if it cannot divide T_N evenly, and then T_f, T_L, and T_W should be adjusted accordingly.

Within the duty-cycle framework, if β_L is too large, the WiFi system may not have enough time to transmit all the packets, leading to lots of WiFi packets being accumulated and undelivered at the end of a frame. On the other hand, if β_L is too small, the WiFi system may not have enough packets in the buffers to transmit, leading to lots of idle slots and low spectrum utilization efficiency. Ideally, to maximize the LTE network capacity without sacrificing WiFi performance, the LTE system should fully leverage these idle slots caused by the empty buffer. For this goal, if we define LTE throughput as the transmission time ratio of the LTE system, i.e., β_L/β_f, the LTE system is designed to choose an optimal β_L for each frame, which can maximize the LTE throughput by leaving no idle slots caused by the empty buffer and simultaneously protect the WiFi system by allowing all the WiFi packets to be transmitted.

Intuitively, the idle slots caused by the empty buffer will be reduced when there are lots of WiFi packets to be transmitted. Denote by n the traffic demand of the WiFi system, which is defined as the number of the packets that are generated and need to be transmitted by all W-STAs in a frame. Hence, the optimization for β_L should be related to n. According to Section II-B, both the number of W-STAs N and the traffic demand n are time-varying. In particular, n varies randomly in each frame. Unfortunately, in the absence of signalling exchanges, the varying WiFi traffic demands cannot be known to the LTE system. Since conventional optimization techniques require exact information for optimization, they cannot be applied here. Moreover, the randomness caused by the backoff procedure in the CSMA/CA protocol further increases the difficulty in optimizing β_L. Since the backoff counters are chosen randomly in CSMA/CA, even for two frames with the same N, n, and β_L, the experimental results may be totally different, implying that the optimal β_L is random even for the same WiFi traffic demands. Such randomness also makes conventional optimization methods inapplicable.

In the next section, we develop DRL-based algorithms to address the aforementioned challenges and realize intelligent spectrum sharing. Firstly, we focus on average WiFi traffic demands and optimize β_L on long time basis, which can reduce the randomness of a single frame. Secondly, we propose to indicate WiFi traffic demands with WiFi channel activity, which can be observed by the LTE system via observing WiFi channels. After that, DRL is adopted to learn WiFi traffic demands and the WiFi traffic model from the observed WiFi channel activity, and thus to optimize β_L automatically.

IV. Deep Reinforcement Learning Approach

In this section, we develop DRL-based algorithms within the spectrum sharing framework proposed in Section III. In the following, we first illustrate the basic principle of the developed algorithms. After that, we give a brief introduction to *Markov decision process* (MDP), followed by the MDP formulation of the studied problem and the elaboration of the DRL-based algorithms.

A. Basic Principle

Although the LTE system cannot know exact WiFi traffic demands in the absence of signalling exchanges, the LTE system is able to monitor WiFi channels to obtain WiFi channel activity, which is the statistics of the channel occupancy during the WiFi transmission time in each frame, such as the numbers

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2020.2971212, IEEE Transactions on Communications

of idle slots and busy slots. For example, the LTE system can continuously monitor the energy level of the received signals to obtain WiFi channel activity. Alternatively, the LTE system can also decode the MAC headers of WiFi packets to obtain WiFi channel activity because WiFi MAC headers include an unencrypted field called *network allocation vector* (NAV) carrying the information of channel busy time [46]. Naturally, when WiFi traffic demand n is large, the number of idle slots tends to be small. Motivated by this, we propose to use WiFi channel activity to indicate WiFi traffic demands and to optimize β_L.

B. Main Challenges and Solutions

However, the WiFi channel activity in a single frame is random and unpredictable, for the reason that it is affected by the WiFi traffic demand and the CSMA/CA protocol, and both of them are random in each frame. Fortunately, we notice that such randomness of a single frame can be removed by the average operation across multiple frames. For example, the WiFi traffic is bursty, making WiFi traffic demand random in each frame, but the average WiFi traffic demand recorded over a period of time will statistically remain stable. For the traffic model illustrated in Section II-B, the average WiFi traffic demand is $\mathbb{E}[n] = N\lambda\beta_f$, which is only determined by N for given λ and β_f, and N remains constant for T_N slots, i.e., β_N/β_f* frames. Hence, we can focus on the average WiFi channel activity recorded over a period of time and use it to indicate the average WiFi traffic demand (or equivalently the number of W-STAs N) over the period. To make the average WiFi channel activity reflect the same average WiFi traffic demand, we introduce a new time unit called step and define a step as the interval in which N remains constant. In other words, a step contains β_N/β_f frames, and the average WiFi traffic demand remains unchanged in each single step. By letting the LTE system adjust β_L for each step, the average WiFi channel activity of a step is only determined by β_L and the average WiFi traffic demand of the step. We take step t for example and illustrate the relations among steps, frames, and slots in Fig. 3, in which $\beta_L^{(t)}$ and $\beta_W^{(t)}$ represent the β_L and β_W chosen at step t, respectively.

Note that once the average WiFi channel activity is obtained at the end of step t, it is already outdated and contains little information about the average WiFi traffic demand at step $t+1$, since the average WiFi traffic demand is determined by N while N at step $t+1$ may be different from that at step t. Nevertheless, the transition of the average WiFi channel activity follows a certain pattern, including the variation pattern of the average WiFi traffic demands (determined by N) and the impact pattern of β_L on the WiFi system.

To learn and leverage these hidden patterns for optimizing β_L, we propose to apply DRL. In DRL, DNNs are the core of learning and decision-making. By continuously training the DNNs with historical data, the coefficients of the neurons, i.e., weights, of the DNNs can be optimized to establish the mapping relationship between WiFi channel activity and β_L. It should be noted that, the optimal β_L is actually determined by specific WiFi traffic demands, and thus the inference of the WiFi traffic demands becomes an intermediate step contained implicitly in the optimized weights of the DNNs. Eventually, with the optimized weights, DRL can learn the WiFi traffic demand from the WiFi channel activity, and accordingly determine a proper β_L to maximize the LTE throughput while satisfying the traffic demands of the WiFi system.

In fact, the optimization for β_L is a decision-making problem in a dynamic environment. To employ DRL, we need to first model this problem as an MDP. Hence, we next introduce briefly the basic MDP framework before going into the development of DRL-based algorithms.

C. Basic MDP Framework

MDP can be used to model decision-making problems in stochastic and dynamic environments, and each MDP consists of five elements: action space $\mathcal{A}$, state space $\mathcal{S}$, transition probability $P_a(s, s')$, reward $r_a(s, s')$, and discount factor γ. In the terminology of MDP, the decision-maker is called the agent. In the following, we introduce the details of the key elements in MDP.

- Action space: All the available actions that the agent can take constitute the action space, which is denoted by $\mathcal{A}$.
- State space: States include the status of the environment observed by the agent, and they serve as the basis for decision-making. The state space is composed of all of the possible states, and it is denoted by $\mathcal{S}$.
- Transition probability: The transition probability $P_a(s, s') = Pr(s_{t+1} = s' | s_t = s, a_t = a)$ stands for the probability that the state at step $t+1$ is s' after the agent executes action a in state s at step t.
- Reward: After the agent takes an action, the environment will return a value to indicate how much the executed action can help to achieve the designed goal. Such a value is called reward. We define $r_a(s, s')$ as the immediate reward that the agent obtains after it takes the action a and makes the state transit from s to s'.
- Discount factor: Discount factor represents how important the future rewards are when making the present decision, and it is denoted by $\gamma \in [0, 1)$.

To solve an MDP is to find the optimal policy $\pi^*(s)$, which mappings a state to an action, to maximize the long-term weighted cumulative reward

$$R = \sum_{t=0}^{\infty} \gamma^t r_{a_t}(s_t, s_{t+1}), \tag{1}$$

where $a_t = \pi^*(s_t)$. To obtain the solution $\pi^*(s)$, *dynamic programming* (DP) methods can be applied, including value iteration and policy iteration [30]. Since DP methods require to fully know system dynamics, i.e., the transition probability $P_a(s, s')$, they are called model-based RL techniques. Conversely, model-free RL techniques refer to those methods that can solve an MDP without requiring to know the system dynamics.

*For simplicity, we consider that β_N/β_f is an integer, which can be realized by choosing an appropriated β_f.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2020.2971212, IEEE Transactions on Communications

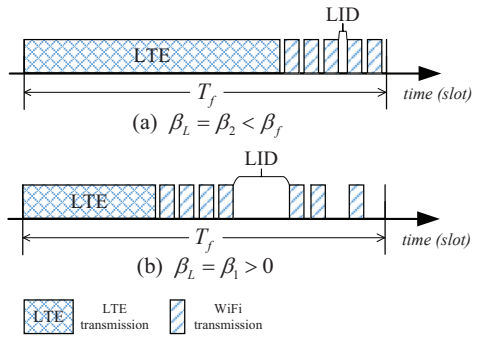


Fig. 4. The illustration of the impacts of β_L on the WiFi system.

D. Type-I DRL-based Algorithm

We first consider a general scenario, where the WiFi system transmits the packets generated in current frames. When the buffers of W-STAs are empty, there may exist some idle slots in which W-STAs are waiting for new packets. Note that each idle slot is produced by either backoff procedure, i.e., all W-STAs are waiting for the counter to be 0, or the empty buffer, i.e., all W-STAs have transmitted all generated packets and wait for a new packet to arrive. Since backoff procedures are inevitable, we aim to enable the LTE system to use these idle slots caused by the empty buffer. Meanwhile, for sufficient protection to the WiFi system, the LTE transmission time should not be larger than those idle time durations caused by the empty buffer.

In practice, the number of idle/busy slots cannot reflect how many idle slots are caused by the empty buffer since idle slots can also be caused by the backoff procedure. Thus, it is challenging for the LTE system to select a β_L to protect the WiFi system appropriately. Fortunately, the *longest idle duration* (LID), which refers to the longest length of the idle durations in a frame and is measured in slots, contains some information about the impacts of β_L on the protection to the WiFi system. To be specific, if β_L is excessive as shown in Fig. 4a, the W-STAs have insufficient transmission time and keep transmitting the packets, leaving a short LID caused only by the backoff procedure. When β_L gets smaller as shown in Fig. 4b, the W-STAs have enough time to transmit the buffered packets, resulting in a long LID that is caused by both the backoff procedure and the empty buffer. Fig. 4 implies that the LID is long if the WiFi system receives sufficient protection, and otherwise is short. Consequently, we can indicate the protection to the WiFi system with the LID.

Based on the observation above, we can formulate the optimization for β_L as an MDP by mapping the key elements from the studied problem. In the terminology of MDP, we refer the agent to as the LTE BS, which decides β_L and monitors WiFi channels to obtain WiFi channel activity.

1) Action space: When a step begins, the agent selects a β_L for the whole step. The selection of β_L is the action of the agent. We use P to discretize the frame length β_f into a finite number of available options for β_L, which constitute the action space $\mathcal{A}$ as follows

$$\mathcal{A} = \{0, P, 2P, \ldots, \beta_f - P\}. \tag{2}$$

Hence, P is determined by both the frame length β_f and the size of the action space $|\mathcal{A}|$, i.e., $P = \beta_f/|\mathcal{A}|$. We denote by $a_t \in \mathcal{A}$ the action that the agent takes at step t.

2) Reward: In order to encourage the LTE system to maximize its own throughput while ensuring the protection to the WiFi system, the LTE system should be rewarded if the WiFi system is well-protected, otherwise the LTE system should be punished. As mentioned above, the LID can be used to indicate the protection to the WiFi system. Since we focus on the average WiFi channel activity of a step, the agent obtains the average LID in a frame of step t, denoted by LID_t, by monitoring the channel and taking the average of the recorded LID for all β_N/β_f frames in that step. By constraining LID_t, we can design the reward to motivate the agent to fulfill the designed objective. In particular, if LID_t is larger or equal to a guard interval, denoted by T_G, the WiFi system is considered to be well-protected, and thus the agent will be rewarded with the achieved LTE throughput. Otherwise, the agent will receive a zero reward. In short, for step t, if the agent takes action a_t, the received reward can be mathematically written as

$$r_{a_t} = \begin{cases} a_t/\beta_f, & LID_t \geq T_G, \\ 0, & LID_t < T_G. \end{cases} \tag{3}$$

When the LTE system keeps silent by letting $\beta_L = 0$, i.e., in the WiFi-only scenario, the observed average LID can be long enough to indicate a well-protected WiFi system. Note the average LID observed in different steps may be different due to the dynamic WiFi traffic demands. In particular, the highest WiFi traffic demands can result in the minimum average LID. However, in the considered unsaturated WiFi system, the average LID still includes the idle slots caused by the empty buffer even when the WiFi system reaches the highest traffic demands. Therefore, a moderate T_G smaller than the minimum average LID should be chosen to simultaneously enhance the LTE throughout meanwhile protect the WiFi system. In summary, T_G can be determined as follows: Firstly, the LTE system keeps silent for certain steps and records all the average LID observed in each step. Secondly, by using the minimum recorded average LID, denoted by $\tilde{T}_{G1}$, to approximate the minimum average LID, T_G can be selected within the range $T_G < \tilde{T}_{G1}$. In fact, T_G can trade off between the LTE throughput and the protection to the WiFi system, which will be elaborated in Section V detailedly.

3) State space: Since states are the basis for the agent to make decisions, they should include enough information about the WiFi system. As explained in Section IV-A, we use WiFi channel activity to indicate WiFi traffic demands. Besides the LID mentioned above, WiFi channel activity includes two other sets of observable data, namely the total numbers of idle slots and busy slots within the WiFi transmission time of a frame. Similar to the acquisition of LID_t, the agent monitors the channel for β_N/β_f frames at step t to obtain the average number of the idle slots in a frame of step t and the average number of the busy slots in a frame of step t, which are denoted by $IDLE_t$ and $BUSY_t$, respectively. In addition, we also put the taken action a_t and the received reward r_t into states, because they can reflect implicitly the rules of how to

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2020.2971212, IEEE Transactions on Communications

evaluate an action. Therefore, at the end of step t, the agent will receive a new state s_{t+1}, which is used to decide a new action for the next step and is given by

$$s_{t+1} = \langle LID_t, IDLE_t, BUSY_t, a_t, r_t \rangle . \quad (4)$$

Note that $LID_t, IDLE_t$, and $BUSY_t$ have at most T_f possible values, while a_t, and r_t respectively have at most β_f/P and β_f possible values. It can be seen that the dimension of the state space is $T_f^3\beta_f^2/P$.

Unfortunately, the transition probability of the MDP cannot be known to the agent because the observed WiFi channel activity is determined by unknown and random WiFi traffic demands. Consequently, model-based RL techniques cannot be adopted to solve the formulated MDP. Moreover, the state-action space has the dimension of $T_f^3\beta_f^3/P^2$, which leads to the curse of dimensionality for traditional model-free RL techniques, e.g., Q-learning and SARSA [30]. Hence, we propose to apply DRL to the problem. As a kind of model-free RL techniques, DRL can solve an MDP without knowing its transition probability. More importantly, by introducing DNNs to evaluate state-action pairs, DRL can overcome the curse of dimensionality when dealing with the MDPs of high-dimensional state-action space.

To make optimal decisions, the agent needs to estimate the expected cumulative reward that it can achieve by taking a specific action at a specific state, and such an estimated value is called Q-value. In DRL, a DNN with weights $\boldsymbol{\theta}$ is built to store the Q-values for all state-action pairs. We denote by $Q(s, a; \boldsymbol{\theta})$ the Q-value of a state-action pair $\langle s, a \rangle$. Thus, this DNN is also called *deep Q-network* (DQN) [47]. If the weights $\boldsymbol{\theta}$ becomes the optimal weights $\boldsymbol{\theta}^*$, the Q-value $Q(s, a; \boldsymbol{\theta}^*)$ can predict accurately the maximum expected cumulative reward for any state-action pair $\langle s, a \rangle$. Then, we have the optimal policy

$$\pi^*(s) = \arg\max_a Q^*(s, a; \boldsymbol{\theta}^*). \quad (5)$$

In order to obtain $\boldsymbol{\theta}^*$ and the corresponding optimal policy, DRL trains the DQN iteratively based on received experiences. The experience received at the end of step t is defined as

$$e_t = \langle s_t, a_t, r_{a_t}, s_{t+1} \rangle . \quad (6)$$

With the experience e_t, the DQN is then trained by minimizing the prediction error, i.e., the loss function given by

$$L(\boldsymbol{\theta}) = \left[y_t^{Tar} - Q(s_t, a_t; \boldsymbol{\theta}) \right]^2, \quad (7)$$

where

$$y_t^{Tar} = r_{a_t} + \gamma \max_a Q(s_{t+1}, a; \boldsymbol{\theta}_t). \quad (8)$$

In other words, when the agent receives experience e_t, the loss function (7) is minimized using gradient descent algorithms, and the weights of the DQN are updated from $\boldsymbol{\theta}_t$ to $\boldsymbol{\theta}_{t+1}$. When the next step $t+1$ begins, the ϵ-greedy policy is adopted to take a new action. Particularly, the ϵ-greedy policy consists of two parts:

- Exploration: the agent selects an action randomly at the probability of ϵ, where $\epsilon \in (0, 1)$;
- Exploitation: the agent selects the action with the highest Q-value, i.e., $\arg\max_a Q(s_{t+1}, a; \boldsymbol{\theta}_{t+1})$, at the probability of $1 - \epsilon$.

Exploration increases the convergence speed of the DQN by helping the agent evaluate each action comprehensively, while exploitation aims to make the best decisions based on the gathered information. They can be traded off by tuning ϵ. By repeatedly taking new actions to generate new experiences and training the DQN with the new experiences, the DQN will converge with $\boldsymbol{\theta}$ being $\boldsymbol{\theta}^*$.

Within the basic DQN framework, we introduce two advanced techniques, including "experience replay" and "quasi-static target network", to further improve its stability [47]. In "experience replay", the agent creates a memory pool $\mathbb{M}$, which is a *first-in-first-out* (FIFO) queue with the size of up to M experiences. Once a new experience is obtained, it will be inserted into the memory pool $\mathbb{M}$. Then, the DQN will be trained in minibatchs. Specifically, a minibatch $\mathbb{B}$ is formed by randomly sampling m experiences from $\mathbb{M}$. In "quasi-static target network", the agent creates a new DQN with the weights $\boldsymbol{\theta}'$ for estimating target values, and thus the new DQN is called the target DQN. For differentiation, we name the former DQN as the trained DQN, which is synchronized with the target DQN every K steps. Denote by $e = \langle s_e, a_e, r_e, s'_e \rangle$ an example of the experience. In addition, $\boldsymbol{\theta}'_t$ and $\mathbb{B}_t$ stand for the weights of the target DQN and the sampled minibatch at step t, respectively. Hence, based on the two advanced techniques, we can respectively rewrite (7) and (8) as

$$L(\boldsymbol{\theta}) = \frac{1}{m} \sum_{e \in \mathbb{B}_t} \left[y_e^{Tar} - Q(s_e, a_e; \boldsymbol{\theta}) \right]^2, \quad (9)$$

and

$$y_e^{Tar} = r_e + \gamma \max_{a'} Q(s'_e, a'; \boldsymbol{\theta}'_t). \quad (10)$$

Through continuous interactions with the WiFi system, the DQNs embedded in the agent are able to learn the pattern of the dynamic WiFi system as well as the relationship between states and actions from past experiences. Eventually, the agent can make proactive adaptations to the dynamic WiFi traffic demands. We summarize the complete algorithm as **Algorithm 1**, of which the flow chart is shown in Fig. 5. In **Algorithm 1**, $t_{\max}$ and $\bmod$ respectively represent the maximum step number and remainder operator; $rand()$ is a function that returns a uniform random number in the interval $[0, 1]$.

We name **Algorithm 1** as the Type-I DRL-based algorithm. In this algorithm, the LID is the key to determine whether the WiFi system gets well-protected. Nevertheless, this indicator is inaccurate to some extent, which may make the LTE system adjust β_L improperly. For further performance enhancement, we then develop the Type-II DRL-based algorithm.

E. Type-II DRL-based Algorithm

The inaccuracy of the indicator used in the Type-I DRL-based algorithm, i.e., LID, can be explained from two aspects. On the one hand, due to the random arrival of WiFi packets, the idle durations shorter than the LID may also include the idle slots caused by the empty buffer. Therefore, the WiFi system may be actually well-protected even if the observed LID is shorter than the threshold. In this case, the LTE system

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2020.2971212, IEEE Transactions on Communications

Algorithm 1 The Type-I DRL-based algorithm

1: Set $\beta_L = 0$, observe the channel for certain steps to obtain $\tilde{T}_{G1}$, and select a T_G within the range $T_G < \tilde{T}_{G1}$.
2: Create a trained DQN with weights $\boldsymbol{\theta}$ and a target DQN with weights $\boldsymbol{\theta}'$, initialize $\boldsymbol{\theta}$ randomly, and let $\boldsymbol{\theta}' = \boldsymbol{\theta}$.
3: Initialize $t = 1$.
4: **repeat**
5: **if** $t < m$ or $rand() < \epsilon$ **then**
6: Randomly select an action $a_t \in \mathcal{A}$.
7: **else**
8: Select the action $a_t = \arg\max_a Q(s_t, a; \boldsymbol{\theta})$.
9: **end if**
10: Monitor the channel for β_N/β_f frames to obtain LID_t, $IDLE_t$, and $BUSY_t$.
11: Obtain r_{a_t}, s_{t+1}, and e_t according to (3), (4), and (6), respectively.
12: Put e_t into $\mathbb{M}$.
13: **if** $t \geq m$ **then**
14: Obtain $\mathbb{B}_t$ by sampling m experiences from $\mathbb{M}$.
15: Update $\boldsymbol{\theta}$ by minimizing (9).
16: **end if**
17: **if** $t \bmod K == 0$ **then**
18: Let $\boldsymbol{\theta}' = \boldsymbol{\theta}$.
19: **end if**
20: Let $t = t + 1$.
21: **until** $t > t_{\max}$

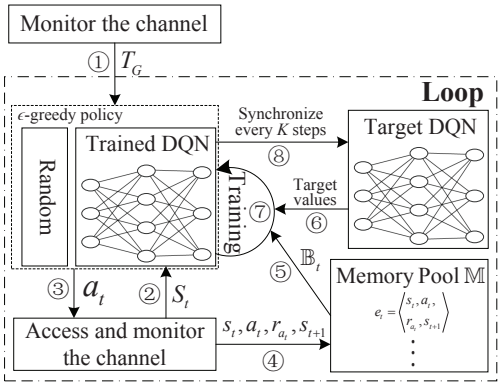


Fig. 5. The action and information flow chart of **Algorithm 1**.

should increase β_L to enhance the LTE throughput, instead of reducing β_L and over-protecting the WiFi system. On the other hand, the random arrival of WiFi packets also makes the LID appear at any time within a frame. Hence, the WiFi system may actually suffer insufficient protection after the appearance of LID even if the observed LID is longer than the threshold. In this case, the LTE system should reduce β_L, instead of increasing β_L and under-protecting the WiFi system.

To address this issue, we propose another algorithm by only allowing the WiFi system to transmit the packets buffered during the previous frame. In this way, the WiFi packets to be transmitted in a frame no longer arrive randomly. To be more specific, the WiFi system has buffered all the WiFi packets at the beginning of the frame and keeps transmitting packets until the buffer is empty. Thus, the idle slots caused by the empty buffer can only appear at the end of a frame. With this property, we can use the length (measured in slots) of the idle slots at the end of a frame, which is called *length of idle ending* (LIE) shown in Fig. 6, as the indicator. LIE can only be caused

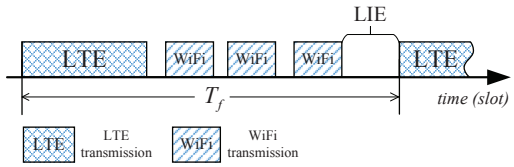


Fig. 6. A sketch for the LIE in a frame.

by either the empty buffer or the backoff procedure, which respectively indicates a well-protected and an under-protected WiFi system. Since the length of the idle slots caused by the backoff procedure is bounded, it is probable that the LIE is caused by the empty buffer rather than the backoff procedure if the LIE is larger than a certain threshold. Consequently, the LIE can indicate the protection of the WiFi system more accurately.

In the end, a new MDP can be formulated, which is similar to the one shown in Section IV-D with the LID substituted for the LIE. In particular, there are no changes in the action space, which stays the same as (2), while the reward and state can be rewritten as follows.

1) Reward: After the agent chooses an action a_t for step t, it observes the LIE of all β_N/β_f frames in the step and calculates their average to obtain LIE_t. Similar to (3), if LIE_t is larger or equal to a guard interval T_G, the agent considers that the WiFi system is well-protected and receives the corresponding throughput as a reward, otherwise, it receives a zero reward. We rewrite (3) as

$$r_{a_t} = \begin{cases} a_t/\beta_f, & LIE_t \geq T_G, \\ 0, & LIE_t < T_G. \end{cases} \tag{11}$$

To differentiate whether LIE_t is caused by the backoff procedure or the empty buffer, the threshold T_G should be larger than the maximum average backoff length. Recall that the idle slots caused by the empty buffer can only appear in the LIE of each frame, and thus the idle slots before the LIE are caused only by the backoff procedure. Hence, T_G can be determined as follows: Firstly, the LTE system keeps silent by setting $\beta_L = 0$ for certain steps and records all the average backoff length observed in each step. Secondly, by using the maximum recorded average LID, denoted by $\tilde{T}_{G2}$, to approximate the maximum average backoff length, T_G can be selected within the range $T_G > \tilde{T}_{G2}$. In fact, adjusting T_G can also achieve different tradeoffs between the LTE throughput and the WiFi protection, which will be revealed in Section V.

2) State space: Similar to LID_t, LIE_t can also be put into the states to indicate WiFi traffic demands. By replacing LID_t in (4) with LIE_t, the new state s_{t+1} observed at the end of step t can be given by

$$s_{t+1} = \langle LIE_t, IDLE_t, BUSY_t, a_t, r_t \rangle . \tag{12}$$

There are T_f possible values for LID_t, which makes the constructed state space also have the dimension of $T_f^3 \beta_f^2 / P$. With the above modifications, we propose the Type-II DRL-based algorithm based on **Algorithm 1**. Specifically, Line 1, Line 10 and Line 11 in **Algorithm 1** are replaced with the following three procedures, respectively:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2020.2971212, IEEE Transactions on Communications

- Set $\beta_L = 0$, observe the channel for certain steps to obtain $\tilde{T}_{G2}$, and select a T_G within the range $T_G > \tilde{T}_{G2}$.
- Monitor the channel for β_N/β_f frames to obtain LIE_t, $IDLE_t$, and $BUSY_t$.
- Obtain r_{a_t}, s_{t+1}, and e_t according to (11), (12), and (6), respectively.

In theory, with a more accurate indicator, the Type-II DRL-based algorithm can optimize the LTE transmission time more precisely than the Type-I DRL-based algorithm can do. However, the Type-II DRL-based algorithm needs the delay transmissions of the WiFi system while the Type-I DRL-based algorithm does not, which makes them suitable to delay-tolerant WiFi systems and delay-sensitive ones, respectively. In the practical deployment, the choice of the algorithms depends on actual situations. For example, if the LTE system has the prior knowledge to know that WiFi users are tolerant to certain delay and the WiFi system can be adjusted to delay transmissions, the Type-II DRL-based algorithm can be adopted to achieve better performance. Otherwise, the LTE system can adopt the Type-I DRL-based algorithm.

V. Simulation Results

In this part, we conduct extensive simulations to examine the proposed DRL-based algorithms. To begin with, we introduce two benchmark algorithms for comparison, which is followed by the simulation setup. After that, we show the impacts of the guard interval T_G on the performance of the proposed DRL-based algorithms. Finally, we compare the performance of the DRL-based algorithms with that of the benchmark algorithms.

A. Benchmark Algorithms

We consider two benchmark algorithms, including the *genie-aided exhaustive search* (GAES) algorithm and the REINFORCE algorithm [30].

1) GAES algorithm: The GAES algorithm assumes that the LTE system can fetch directly the required information from the WiFi system. With the perfect knowledge about the WiFi system, this algorithm aims to maximize the LTE throughput under a guaranteed WiFi packet delivery ratio. However, the resulted WiFi packet delivery ratio in a frame is actually a random variable that is determined jointly by n, β_W, and the CSMA/CA protocol. Hence, the expected WiFi packet delivery ratio is considered. If $\hat{n}$ denotes the number of packets delivered by the end of a frame, we can formulate the following problem.

Problem 1:

$$\max_{\beta_L, \beta_W} \quad \beta_L/\beta_f \tag{13}$$

$$s.t. \quad \beta_L + \beta_W = \beta_f, \tag{14}$$

$$\mathbb{E}\left[\frac{\hat{n}}{n}\right] > \psi, \tag{15}$$

where (15) guarantees the expected packet delivery ratio to be larger than a pre-defined threshold ψ. Nevertheless, the analytical expression of $\mathbb{E}\left[\frac{\hat{n}}{n}\right]$ is difficult to obtain, especially when the WiFi system is unsaturated. As such, we turn to using the arithmetic mean of $\frac{\hat{n}}{n}$ as an estimated value of $\mathbb{E}\left[\frac{\hat{n}}{n}\right]$ by conducting repeated experiments for a given β_W. Finally, exhaustive search can be applied to **Problem 1**, and the GAES algorithm can be described as: the LTE system 1) acquires the exact N and λ; 2) performs exhaustive search over $\mathcal{A}$ and finds the maximum β_L satisfying the constraint (15), where $\mathbb{E}\left[\frac{\hat{n}}{n}\right]$ in (15) is obtained numerically through repeated experiments for each $\beta_W = \beta_f - \beta_L$. Note that the GAES algorithm requires the LTE system to know perfect WiFi traffic demands and to perform exhaustive search, which leads to prohibitive amounts of overheads in the procedure of signalling exchanges and computations. Hence, it is difficult to deploy the GEAS algorithm in practice, and the performance of the GEAS algorithm can only be used as an upper bound.

2) REINFORCE algorithm: Recall that the proposed DRL-based algorithms make decisions based on the estimated Q-values, and thus they are categorized as value-based RL methods. In fact, there is another branch of RL methods, called policy-based methods, in which the policy is parameterized with an approximate function. The action selection and policy improvement in policy-based methods do not consult Q-values, and consequently the curse of dimensionality can be avoided. Here, we consider a typical policy-based method called REINFORCE [30], where the policy π is parameterized by $\boldsymbol{\xi}$, and $\pi(s, a; \boldsymbol{\xi})$ returns the probability that the agent takes action a on the condition of the state being s. In REINFORCE, a finite MDP terminated after ϕ steps is assumed, and its objective function is defined as $\max_{\boldsymbol{\xi}} J(\boldsymbol{\xi}) = \mathbb{E}_\pi\left[\sum_{t=0}^{\phi-1} \gamma^t r_{a_t}(s_t, s_{t+1})\right]$. According to [30], $\boldsymbol{\xi}$ can be optimized iteratively with the gradient ascent with respect to $J(\boldsymbol{\xi})$. For each episode (i.e., termination of the MDP), the update equation of $\boldsymbol{\xi}$ can be given by

$$\boldsymbol{\xi} = \boldsymbol{\xi} + \alpha \sum_{t=0}^{\phi-1} \nabla_{\boldsymbol{\xi}} \log\left[\pi(s, a; \boldsymbol{\xi})\right] \gamma^t r_{a_t}(s_t, s_{t+1}), \tag{16}$$

where α is the learning rate. Since the formulated MDPs in Section IV-C can be infinite, we discretize the MDPs into episodes with 100 steps when applying REINFORCE. Similar to the Type-I and Type-II DRL-based algorithms, we also consider two types of REINFORCE algorithms. In particular, the Type-II REINFORCE algorithm requires a delay-tolerant WiFi system and adopts (11) as the reward function, while the Type-I REINFORCE algorithm does not require this and adopts (3) as the reward function.

B. Simulation Setup

We conduct the simulations using a slotted CSMA/CA simulator. The basic parameters used in the simulations are given in Table I according to [48]. It is worth noting that due to the lack of CCA, the designed duty-cycle MAC protocol may cause collisions to the ongoing WiFi transmissions that just start ahead of new LTE transmissions. To better demonstrate the performance of the proposed schemes, the collisions between WiFi packets and LTE transmissions have been accounted for in simulations. In addition, we adopt the DSMC shown in Fig.7 as the transition pattern of the number of W-STAs, in which N varies every T_N slots. Hence, the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2020.2971212, IEEE Transactions on Communications

TABLE I
BASIC SYSTEM PARAMETERS

Parameters	Value
Slot time, σ	9 μs
Channel busy time caused by a successful transmission, T_s	25σ
Channel busy time caused by a collision	T_c
Initial contention window	16
Maximum backoff stage, m	6
Frame length, T_f	$200T_s$
User transition interval, T_N	$25T_f$
Packet arrival rate, λ	0.05
Maximum number of W-STAs, $N_{\max}$	10
Minimum number of W-STAs, $N_{\min}$	1
Initial number of W-STAs	5

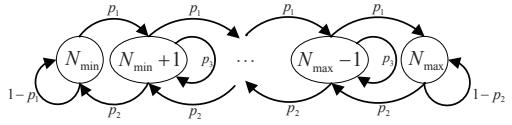


Fig. 7. The DSMC of the transitions of N.

average WiFi traffic demand remains the same for the T_N slots when N does not change. Note that the proposed DRL-based algorithms do not limit or require the LTE system to know the transition pattern. As shown in Fig.7, for every transition of N, N is decreased/increased by one at the probability of p_1 and remains the same at the probability of p_3 if N is not at the boundary values ($N_{\max}$ or $N_{\min}$). We set p_1, p_2, and p_3 to be 0.1, 0.1, and 0.8, respectively. Correspondingly, the elements in the transition matrix $\mathbf{P}$ are given by

$$p_{ij} = \begin{cases} 0.1, \text{ for } j = i \pm 1, \\ 0.8, \text{ for } j = i, i \neq N_{\min} \text{ and } N_{\max}, \\ 0.9, \text{ for } j = i, i = N_{\min} \text{ or } N_{\max}, \end{cases} \tag{17}$$

where $i, j \in \{N_{\min}, N_{\min}+1, \ldots, N_{\max}\}$.

In the DRL-based algorithms, a DNN needs to be built as the DQN. Since the DQN takes a state as the input and then outputs estimated Q-values for each action, the number of neurons in the input layer is the same as the number of elements in a state s, and the number of neurons in the output layer is the same as the number of elements in the action space $\mathcal{A}$. Here, a tradeoff exists in choosing the size of the action space. On the one hand, a larger action space can achieve higher granularity in decision-making. On the other hand, the increase of action space also slows down both the learning speed and the reaction speed [33]. Hence, to balance the two aspects, we choose the size of the action space $|\mathcal{A}|$ to be 50, and thus P is set to be 4. After determining the input and output layers, we also need to select the rest of the hyper-parameters in the architecture of the DNN, including the types of hidden layers, the number hidden layers, and the number of neurons in each hidden layer. These hyper-parameters affect the performance of the DRL-based algorithms. However, nowadays, these hyper-parameters are still mainly selected by experience and trial-and-error procedures. In the next subsection, we will discuss about the selection of DNN architecture for the DQN used in the simulations. Remaining parameters of the DQN are as follows. The activation function employed in each neuron is ReLU. Adam is adopted as the gradient descent algorithm to minimize the loss function (9) at the learning rate of 0.01. The minibatch size m and the memory pool size M are set to be 32 and 2000, respectively. The synchronization period of the target DQN K and the discount factor γ are set to be 100 and 0.5, respectively. In the ϵ-greedy policy, the initial ϵ and the minimum ϵ are chosen to be 0.1 and 0.01, respectively. In addition, we include 50000 steps in an experiment, and thus ϵ is decreased by $\frac{0.1-0.01}{50000}$ for every step.

C. Selection of DNN Architecture

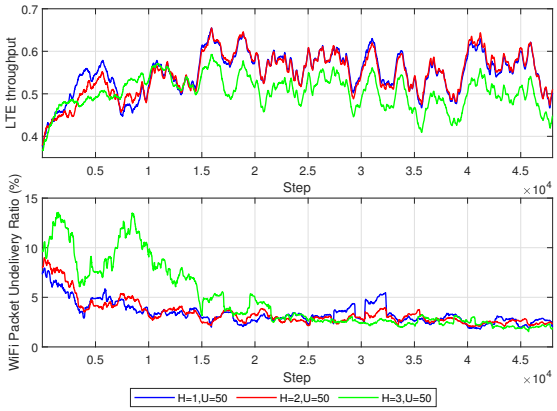
We first discuss about the choice of the types of hidden layers. In DNNs, there are basically four types of layers, including convolutional layers, pooling layers, recurrent layers, and *fully-connected* (FC) layers. In particular, convolutional layers and pooling layers are typically jointly adopted for analyzing image input data, while recurrent layers are typically used to process sequential input data with memory in the time domain. Since the input of the proposed DRL-based algorithms, i.e., the state, does not contain image or sequential data, we do not adopt convolutional layers, pooling layers, or recurrent layers. Meanwhile, FC layers do not have special assumptions on the input data, and thus we choose hidden layers to be FC layers.

Then, similar to [49], we perform extensive simulations to determine the number of hidden layers and the number of neurons in each hidden layer. Denote by H and U the number of hidden layers and the number of neurons in each hidden layer, respectively. We set T_G to be $3T_s$. By fixing $U = 50$, we first compare the performance achieved by the DRL-based algorithms with different values of H, as illustrated in Fig. 8. Each curve in Fig. 8 comes from an experiment with 50000 steps and is smoothed through *moving average* (MA) with the size of 2000, which is the same in the rest of the simulation results. For the Type-I DRL-based algorithm shown in Fig. 8a, it can be seen that $H = 1$ and $H = 2$ achieve similar LTE throughput and they both outperform $H = 3$. However, the WiFi packet undelivery ratio achieved by $H = 2$ is more stable and lower than that by $H = 1$. As for the Type-II DRL-based algorithm shown in Fig. 8b, all of $H = 1$, $H = 2$, and $H = 3$ achieve similar LTE throughput, while $H = 2$ can result in a lower and more stable WiFi packet undelivery ratio than the other two options. Hence, we choose H to be 2. We further show the performance with different values of U in Fig. 9. From the figure, $U = 50$ works the best for both DRL-based algorithms in terms of the LTE throughput, the WiFi packet undelivery ratio, and the convergence rate. Therefore, we set the number of hidden layers and the number of the neurons in each hidden layer to be 2 and 50, respectively.

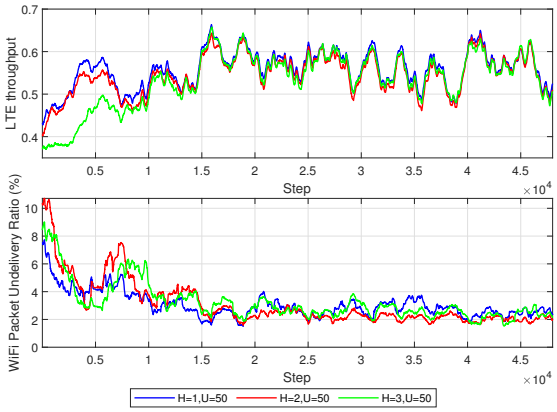
D. Effects of the Guard Interval T_G

We first focus on the Type-I DRL-based algorithm, in which the LID serves as the indicator of the protection to the WiFi system. In this algorithm, T_G can be selected by referring to the average LID observed in the WiFi-only scenario. Fig.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2020.2971212, IEEE Transactions on Communications

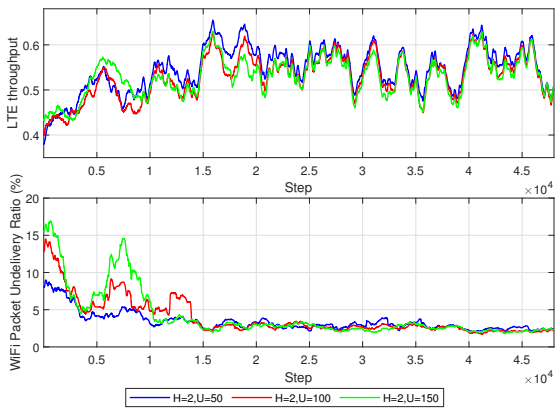


(a) Type-I DRL-based algorithm.

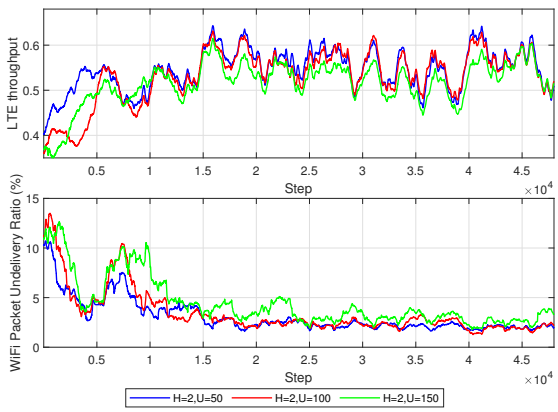


(b) Type-II DRL-based algorithm.

Fig. 8. The performance of the Type-I and Type-II DRL-based algorithms with $U = 50$ and different H: MA=2000.



(a) Type-I DRL-based algorithm.



(b) Type-II DRL-based algorithm.

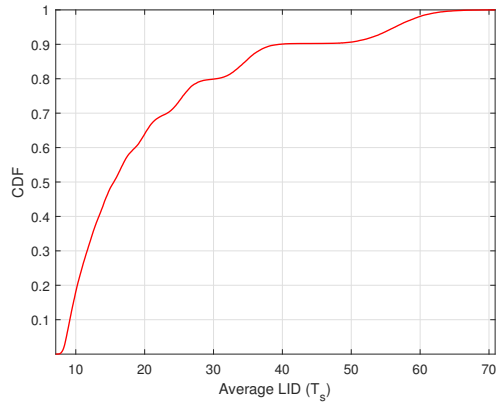
Fig. 9. The performance of the Type-I and Type-II DRL-based algorithms with $H = 2$ and different U: MA=2000.

10a depicts the *cumulative distribution function* (CDF) of the average LID recorded for 50000 steps. From the figure, the minimum recorded average LID, i.e., $\tilde{T}_{G1}$, is around $7T_s$. As explained in Section IV-C, T_G can be selected within the range $T_G < \tilde{T}_{G1}$, and thus we depicts in Fig. 11 the LTE throughput and the WiFi packet undelivery ratio obtained by the Type-I DRL-based algorithm with several typical values of T_G in the given range. Fig. 11 shows that as the step grows, the algorithm can gradually learn to reduce the WiFi packet undelivery ratio, which demonstrates that the WiFi system can get protected by the LTE system with the determined T_G. Meanwhile, the achieved LTE throughput is also increased as the training continues. Both the LTE throughput and the WiFi packet undelivery ratio converge at around step 15000, although there exist some fluctuations. After that, the LTE system can continue to achieve high LTE throughput with a low WiFi packet undelivery ratio. Hence, it can be demonstrated that the proposed algorithm is effective to track the dynamic WiFi traffic demands and select proper LTE transmission time to achieve the designed goal, which implies that WiFi traffic demands can indeed be learned from WiFi channel activity.

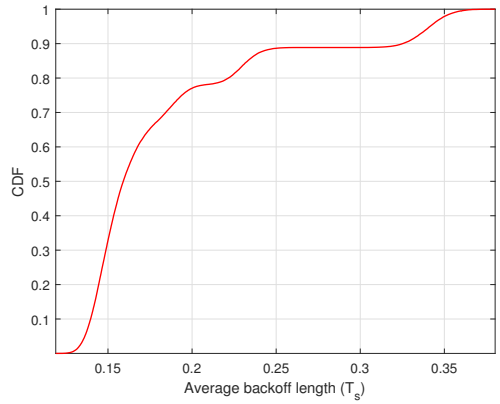
As for the Type-II DRL-based algorithm, T_G can be selected by referring to the average backoff length observed in the WiFi-only scenario. Fig. 10b shows the CDF of the average backoff length recorded for 50000 steps. From the figure, the maximum recorded average backoff length, i.e., $\tilde{T}_{G2}$, is around $0.35T_s$. As explained in Section IV-E, T_G should be larger than $\tilde{T}_{G2}$. Hence, we depict in Fig. 12 the performance achieved by the Type-II DRL-based algorithm with several typical values of T_G larger than $\tilde{T}_{G2}$. From the figure, we can observe a phenomenon similar to that in Fig. 11 and the Type-II DRL-based algorithm also needs around 15000 steps to converge. This demonstrates the effectiveness of the Type-II DRL-based algorithm.

From extensive simulations, we have observed that the proposed DRL-based algorithms can converge within a reasonable period of time, e.g., around 15000 steps to converge as depicted in Fig. 11 and Fig. 12. Once the algorithms converge, they can always track the dynamic WiFi traffic demands and continuously achieve high performance without re-training. Also, the proposed DRL-based algorithms are online training-and-action algorithms, and thus do not have to wait for the convergence before executing. In fact, the proposed algorithms can achieve good performance even before convergence. For instance, from Fig. 11a and Fig. 12a, the proposed algorithms can achieve around 90% of the converged LTE throughput in average during the convergence. Thus, the proposed DRL-based algorithms are suitable for practical implementation.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2020.2971212, IEEE Transactions on Communications

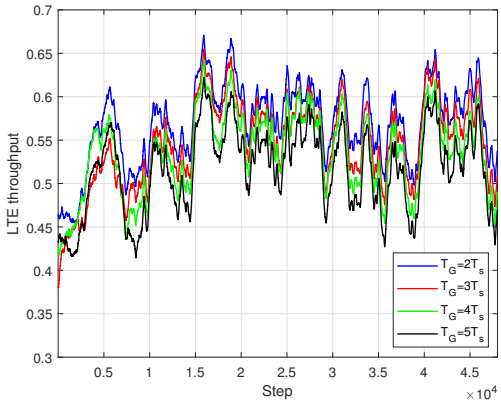


(a) The CDF of the average LID.

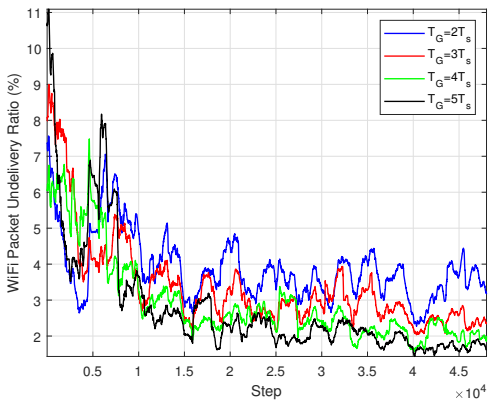


(b) The CDF of the average backoff length.

Fig. 10. The CDF obtained in the WiFi-only scenario.



(a) The evolution of the LTE throughput.



(b) The evolution of the WiFi packet undelivery ratio.

Fig. 11. The performance of the Type-I DRL-based algorithm with different T_G: MA=2000.

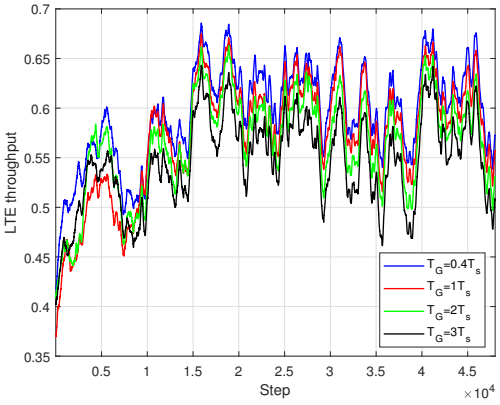
In addition, it can be observed in Fig. 11 and Fig. 12 that a larger T_G can provide more stringent protection to the WiFi system with reduced WiFi packet undelivery ratios but meanwhile leads to lower LTE throughput. According to (3) and (11), the DRL agent is getting easier to receive zero reward as T_G increases, making the LTE system tend to adopt more conservative actions, i.e., smaller β_L. With smaller β_L, the WiFi system has more transmission time and reduces the WiFi packet undelivery ratio. Thus, it can be demonstrated that T_G actually trades off between the LTE throughput and the protection to the WiFi system.

E. Performance Comparison

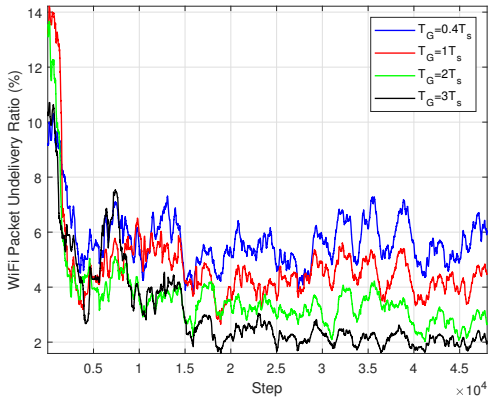
In this subsection, simulation results are provided to compare the performance of the proposed DRL-based algorithms with that of the benchmark algorithms. Note that LID and LIE have different physical meaning, making the Type-I and Type-II DRL-based algorithms achieve different WiFi undelivery ratios and different LTE throughput with the same T_G. Thus, it is unfair to compare their performance with the same T_G. Alternatively, we adopt different T_G for the two algorithms to make them have almost the same WiFi undelivery ratio and then compare the achieved LTE throughput. As shown in Fig. 11b and Fig. 12b, the WiFi packet undelivery ratios of the Type-I and Type-II DRL-based algorithms converge to around 3% with $T_G = 4T_s$ and $T_G = 3T_s$, respectively. For comparison, we put these two curves in Fig. 13 along with the results obtained by the GAES algorithm with $\psi = 97\%$. In the GAES algorithm, repeated experiments containing 10000 frames are conducted to approximate $\mathbb{E}\left[\frac{\hat{n}}{n}\right]$ for each β_L. Meanwhile, we provide the performance of the Type-I and Type-II REINFORCE algorithms with $T_G = 5T_s$ and $T_G = 2T_s$, respectively. To be fair, the approximate function in the REINFORCE algorithms is chosen to be the same DNN used in the DRL-based algorithms, and α is set to be 0.001. Fig. 13a and Fig. 13b share the same legend. For clearly presentation, the legend of Fig. 13a is omitted. In particular, "DRL-I" and "DRL-II" refer to the Type-I and Type-II DRL-based algorithms, respectively, which is similar for "REINFORCE-I" and "REINFORCE-II".

Fig. 13a and Fig. 13b reveal that the proposed DRL-based algorithms can track the dynamics of the WiFi system and approach closely to the performance of the GAES algorithm. In particular, at almost the same converged WiFi packet undelivery ratio, the average LTE throughput achieved by the Typy-I and Type-II DRL-based algorithms reach 89.78% and 91.55% of that achieved by the GAES algorithm, respectively, after the convergence at around step 15000. This result also implies that the LIE used in the Type-II DRL-based algorithm

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2020.2971212, IEEE Transactions on Communications

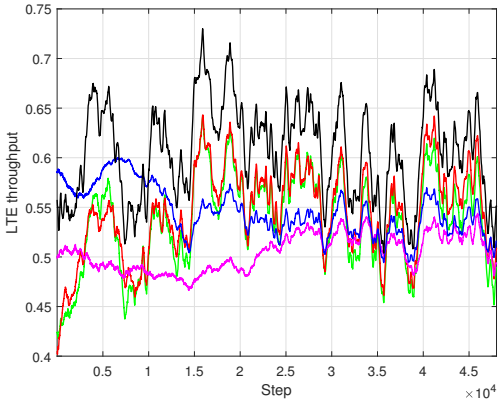


(a) The evolution of the LTE throughput.

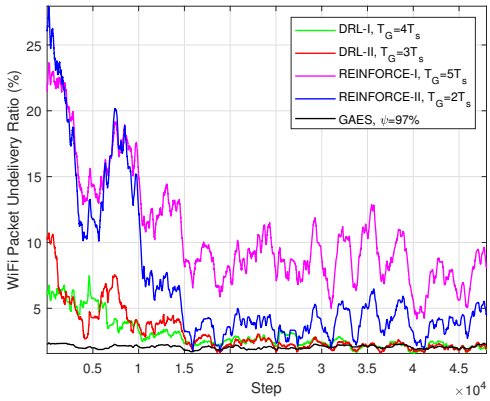


(b) The evolution of the WiFi packet undelivery ratio.

Fig. 12. The performance of the Type-II DRL-based algorithm with different T_G: MA=2000.



(a) The evolution of the LTE throughput.



(b) The evolution of the WiFi packet undelivery ratio.

Fig. 13. The performance of the proposed DRL-based algorithms and benchmark algorithms: MA=2000.

indeed indicates more accurately whether the WiFi system gets well-protected and enables the LTE system to optimize β_L more precisely, achieving more efficient spectrum sharing. Compared with the REINFORCE algorithms, the proposed DRL-based algorithms converge faster, and the converged performance is much better. Particularly, the REINFORCE algorithms achieve lower LTE throughput at higher WiFi packet undelivery ratio than the proposed DRL-based algorithms. In fact, the performance gap between the proposed DRL-based algorithms and the REINFORCE algorithms is caused by the inherent problems of policy-based methods, including slow learning ability and low utilization of data [30]. Hence, the superiority of the proposed DRL-based algorithms can be demonstrated.

F. Extension to Dynamic WiFi Packet Arrival Rates

So far, we have built up the analysis and developed the algorithms based on the WiFi traffic model with a static and identical packet arrival rate λ, as illustrated in Section II-B. Nevertheless, in reality, W-STAs are prone to run different services dynamically, which makes it significant to also consider dynamic packet arrival rates in W-STAs. In fact, the developed DRL-based algorithms does not require to know the WiFi traffic model in implementation, for which reason the proposed schemes can be applied to the scenario with dynamic WiFi packet arrival rates directly without any modification.

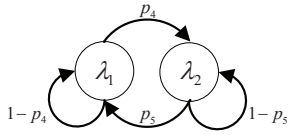
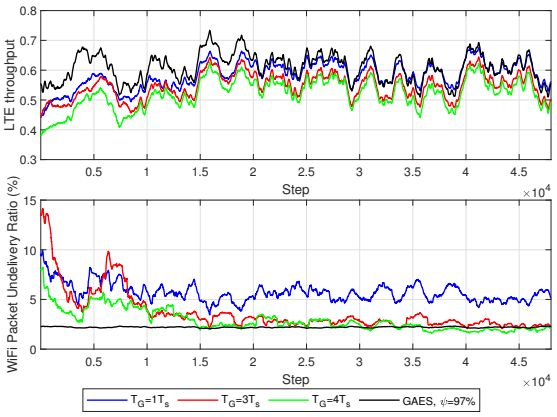


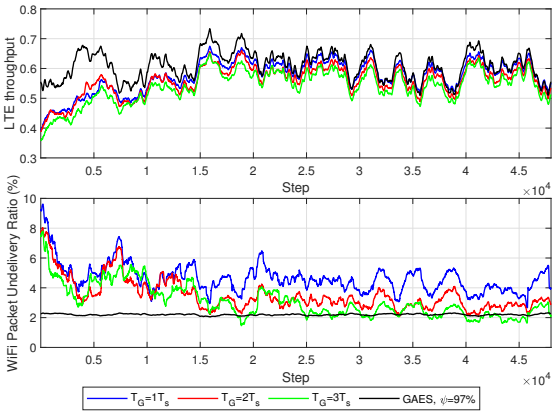
Fig. 14. The DSMC of the transitions of λ.

Next, we evaluate the performance of the proposed Type-I and Type-II DRL-based algorithms with dynamic WiFi packet arrival rates. We assume that each W-STA varies its own packet arrival rates for each step. Similar to Fig. 7, we take a DSMC as the transition pattern of the packet arrival rates, which is shown in Fig. 14. In particular, λ_1, λ_2, p_1, and p_2 are set to be 0.04, 0.06, 0.1, and 0.1, respectively. Then, except λ, we perform simulations with the rest of the parameters used in Section V-D, and depict the results in Fig. 15. It can be seen from the figure that the trends and phenomena are very similar to those observed from the results shown in Section V-D. For example, for both algorithms, they can gradually learn to approach the GAES algorithm in terms of LTE throughput and reduce WiFi packet undelivery ratio as the training continues. In addition, the increase of T_G also reduces

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2020.2971212, IEEE Transactions on Communications



(a) Type-I DRL-based algorithm.



(b) Type-II DRL-based algorithm.

Fig. 15. The performance of the Type-I and Type-II DRL-based algorithms with dynamic WiFi packet arrival rates: MA=2000.

both LTE throughput and WiFi packet undelivery. Hence, it can be demonstrated that the proposed algorithms are also applicable to dynamic WiFi packet arrival rates.

VI. Conclusions

In this work, we have investigated the LTE/WiFi spectrum sharing problem, and developed an intelligent duty-cycle MAC protocol. In particular, we have first proposed a duty-cycle spectrum sharing framework using time sharing. After that, DRL-based algorithms have been proposed for the LTE system to adaptively maximize the LTE throughput while protecting the WiFi system by learning WiFi traffic patterns from the observed WiFi channel activity. As such, the two systems can share the spectrum intelligently without signalling exchanges. Simulation results have demonstrated that even without signalling exchanges, the performance of the proposed DRL-based algorithms can approach that of the genie-aided exhaustive search algorithm, which needs the perfect knowledge of WiFi traffic demands and is of high computational complexity.

References

[1] J. Tan, L. Zhang, Y.-C. Liang, and D. Niyato, "Deep reinforcement learning for the coexistence of LAA-LTE and WiFi systems," in *Proc. IEEE ICC*, 2019, pp. 1–6.

[2] *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016—2021 White Paper*, Mar. 2017.

[3] L. Zhang, M. Xiao, G. Wu, M. Alam, Y.-C. Liang, and S. Li, "A survey of advanced techniques for spectrum sharing in 5G networks," *IEEE Wireless Commun.*, vol. 24, no. 5, pp. 44–51, Oct. 2017.

[4] B. Chen, J. Chen, Y. Gao, and J. Zhang, "Coexistence of LTE-LAA and Wi-Fi on 5 GHz with corresponding deployment scenarios: A survey," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 1, pp. 7–32, Jul. 2017.

[5] X. Wang, S. Mao, and M. X. Gong, "A survey of LTE Wi-Fi coexistence in unlicensed bands," *Mobile Comput. Commun.*, vol. 20, no. 3, pp. 17–23, Jul. 2017.

[6] S.-Y. Lien, C.-C. Chien, H.-L. Tsai, Y.-C. Liang, and D. I. Kim, "Configurable 3GPP licensed assisted access to unlicensed spectrum," *IEEE Wireless Commun.*, vol. 23, no. 6, pp. 32–39, Dec. 2016.

[7] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 535–547, Mar. 2000.

[8] A. K. Bairagi, N. H. Tran, W. Saad, Z. Han, and C. S. Hong, "A game-theoretic approach for fair coexistence between LTE-U and Wi-Fi systems," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 442–455, Jan. 2019.

[9] N. Rupasinghe and İ. Güvenç, "Licensed-assisted access for WiFi-LTE coexistence in the unlicensed spectrum," in *IEEE GLOBECOM Wkshps*, 2014, pp. 894–899.

[10] T. LeAnh, N. H. Tran, D. T. Ngo, Z. Han, and C. S. Hong, "Orchestrating resource management in LTE-unlicensed systems with backhaul link constraints," *IEEE Trans. Commun.*, vol. 18, no. 2, pp. 1360–1375, Jan. 2019.

[11] R. Yin, G. Yu, A. Maaref, and G. Y. Li, "LBT-based adaptive channel access for LTE-U systems," *IEEE Trans. Wireless Commun.*, vol. 15, no. 10, pp. 6585–6597, Oct. 2016.

[12] J. Tan, S. Xiao, S. Han, Y.-C. Liang, and V. C. Leung, "QoS-aware user association and resource allocation in LAA-LTE/WiFi coexistence systems," *IEEE Trans. Wireless Commun.*, vol. 18, no. 4, pp. 2415–2430, Apr. 2019.

[13] S. Han, Y.-C. Liang, Q. Chen, and B. H. Soong, "Licensed-assisted access for LTE in unlicensed spectrum: A MAC protocol design," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 10, pp. 2550–2561, Oct. 2016.

[14] J. Tan, S. Xiao, S. Han, and Y.-C. Liang, "A learning-based coexistence mechanism for LAA-LTE based HetNets," in *Proc. IEEE ICC*, 2018, pp. 1–6.

[15] S.-Y. Lien, J. Lee, and Y.-C. Liang, "Random access or scheduling: Optimum LTE licensed-assisted access to unlicensed spectrum," *IEEE Commun. Lett.*, vol. 20, no. 3, pp. 590–593, Mar. 2016.

[16] H. Zhang, X. Chu, W. Guo, and S. Wang, "Coexistence of Wi-Fi and heterogeneous small cell networks sharing unlicensed spectrum," *IEEE Commun. Mag.*, vol. 53, no. 3, pp. 158–164, Mar. 2015.

[17] C. Cano and D. J. Leith, "Coexistence of WiFi and LTE in unlicensed bands: A proportional fair allocation scheme," in *Proc. IEEE ICCW*, 2015, pp. 2288–2293.

[18] Q. Chen, G. Yu, H. Shan, A. Maaref, G. Y. Li, and A. Huang, "Cellular meets WiFi: Traffic offloading or resource sharing?" *IEEE Trans. Wireless Commun.*, vol. 15, no. 5, pp. 3354–3367, Jul. 2016.

[19] M. R. Khawer, J. Tang, and F. Han, "usICIC-A proactive small cell interference mitigation strategy for improving spectral efficiency of LTE networks in the unlicensed spectrum," *IEEE Trans. Wireless Commun.*, vol. 15, no. 3, pp. 2303–2311, Mar. 2016.

[20] U. Challita, L. Dong, and W. Saad, "Proactive resource management for LTE in unlicensed spectrum: A deep learning perspective," *IEEE Trans. Wireless Commun.*, Jul. 2018.

[21] G. Bianchi and I. Tinnirello, "Kalman filter estimation of the number of competing terminals in an IEEE 802.11 network," in *Proc. IEEE INFOCOM*, vol. 2, 2003, pp. 844–852.

[22] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[23] A. Zappone, M. Di Renzo, and M. Debbah, "Wireless networks design in the era of deep learning: Model-based, AI-based, or both?" *IEEE Trans. Commun.*, vol. 67, no. 10, pp. 7331–7376, Oct. 2019.

[24] A. Zappone, M. Di Renzo, M. Debbah, T. T. Lam, and X. Qian, "Model-aided wireless artificial intelligence: Embedding expert knowledge in deep neural networks for wireless system optimization," *IEEE Veh. Technol. Mag.*, vol. 14, no. 3, pp. 60–69, Sep. 2019.

[25] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. Cogn. Commun. Netw.*, vol. 3, no. 4, pp. 563–575, Dec. 2017.

[26] H. Ye, G. Y. Li, and B.-H. Juang, "Power of deep learning for channel estimation and signal detection in OFDM systems," *IEEE Wireless Commun. Lett.*, vol. 7, no. 1, pp. 114–117, Feb. 2017.

[27] C. Huang, G. C. Alexandropoulos, A. Zappone, C. Yuen, and M. Debbah, "Deep learning for UL/DL channel calibration in generic massive MIMO systems," *arXiv preprint arXiv:1903.02875*, 2019.

[28] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over-the-air deep learning based radio signal classification," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 168–179, Feb. 2018.

[29] C. Liu, J. Wang, X. Liu, and Y.-C. Liang, "Deep CM-CNN for spectrum sensing in cognitive radio," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2306–2321, Oct. 2019.

[30] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[31] N. C. Luong, T. T. Anh, H. T. T. Binh, D. Niyato, D. I. Kim, and Y.-C. Liang, "Joint transaction transmission and channel selection in cognitive radio based blockchain networks: A deep reinforcement learning approach," in *Proc. IEEE ICASSP*, 2019, pp. 8409–8413.

[32] N. C. Luong, D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y.-C. Liang, and D. I. Kim, "Applications of deep reinforcement learning in communications and networking: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3133–3174, May 2019.

[33] Y. S. Nasir and D. Guo, "Multi-agent deep reinforcement learning for dynamic power allocation in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2239–2250, Oct. 2019.

[34] H. Ye and G. Y. Li, "Deep reinforcement learning for resource allocation in V2V communications," in *Proc. IEEE ICC*, 2018, pp. 1–6.

[35] Y. He, N. Zhao, and H. Yin, "Integrated networking, caching, and computing for connected vehicles: A deep reinforcement learning approach," *IEEE Trans. Veh. Technol.*, vol. 67, no. 1, pp. 44–55, Oct. 2018.

[36] L. Zhang, J. Tan, Y.-C. Liang, G. Feng, and D. Niyato, "Deep reinforcement learning based modulation and coding scheme selection in cognitive heterogeneous networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 6, pp. 3281–3294, Jun. 2019.

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

[38] Y. Yu, T. Wang, and S. C. Liew, "Deep-reinforcement learning multiple access for heterogeneous wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1277–1290, Jun. 2019.

[39] R. Ali, N. Shahin, Y. B. Zikria, B.-S. Kim, and S. W. Kim, "Deep reinforcement learning paradigm for performance optimization of channel observation–based MAC protocols in dense WLANs," *IEEE Access*, Dec. 2018.

[40] H.-J. Kwon, J. Jeon, A. Bhorkar, Q. Ye, H. Harada, Y. Jiang, L. Liu, S. Nagata, B. L. Ng, T. Novlan *et al.*, "Licensed-assisted access to unlicensed spectrum in LTE release 13," *IEEE Commun. Mag.*, vol. 55, no. 2, pp. 201–207, Feb. 2017.

[41] R. Raghavendra, J. Padhye, R. Mahajan, and E. Belding, "Wi-Fi networks are underutilized," *Microsoft Res. Tech. Rep.*, 2009.

[42] A. Rajandekar and B. Sikdar, "On the feasibility of using WiFi white spaces for opportunistic M2M communications," *IEEE Wireless Commun.*, vol. 4, no. 6, pp. 681–684, Dec. 2015.

[43] M. Rademacher, K. Jonas, and M. Kretschmer, "Quantifying the spectrum occupancy in an outdoor 5 GHz WiFi network with directional antennas," in *IEEE Proc. WCNC*, 2018, pp. 1–6.

[44] P.-Y. Kong, "Optimal probabilistic policy for dynamic resource activation using Markov decision process in green wireless networks," *IEEE Trans. Mobile Comput.*, vol. 13, no. 10, pp. 2357–2368, Oct. 2014.

[45] B. Li, W. Guo, H. Zhang, C. Zhao, S. Li, and N. Arumugam, "Spectrum detection and link quality assessment for heterogeneous shared access networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1431–1445, Feb. 2019.

[46] Q. Chen, G. Yu, and Z. Ding, "Enhanced LAA for unlicensed LTE deployment based on TXOP contention," *IEEE Trans. Commun.*, vol. 67, no. 1, pp. 417–429, Sep. 2018.

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

[48] D.-J. Deng, C.-H. Ke, H.-H. Chen, and Y.-M. Huang, "Contention window optimization for IEEE 802.11 DCF access control," *IEEE Trans. Wireless Commun.*, vol. 7, no. 12, pp. 5129–5135, Dec. 2008.

[49] N. Zhao, Y.-C. Liang, D. Niyato, Y. Pei, M. Wu, and Y. Jiang, "Deep reinforcement learning for user association and resource allocation in heterogeneous cellular networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 11, pp. 5141–5152, Nov. 2019.