Deep-Reinforcement-Learning-based Hierarchical Time-Division-Duplexing Control for Dense Wireless and Mobile Networks

Van Dat Tuong, Nhu-Ngoc Dao, Sungrae Cho, and Wonjong Noh

***Abstract*—Future wireless and mobile network services require to accommodate highly dynamic downlink (DL) and uplink (UL) traffic asymmetry. To this requirement, the third generation partnership project (3GPP) introduced the enhanced interference mitigation and traffic adaptation (eIMTA) strategy, which enables flexible allocation of subframes for DL and UL traffic in the time-division-duplexing scheme. However, the service may experience severe intercell cross-link interference, especially in a dense environment. In this study, we developed an optimal duplexing framework that serves various traffic demands effectively. In the proposed framework, deep-reinforcement-learning processes are implemented at base stations (BSs) for obtaining the optimal radio frame configuration (RFC) policies that maximize the long-term system utility in terms of interference reduction, DL and UL rates. The training process at each BS considers the traffic demand state and previous RFCs of different BSs. Training processes were coordinated in a single-leader multi-follower Stackelberg game that guarantees convergence. We obtained the Stackelberg equilibrium as system utility, i.e., the data rate is maximized, and accordingly, interference is minimized, with the optimal RFC setup. Extensive simulations illustrate that our proposed framework outperforms the existing schemes in terms of data-rate improvement and interference reduction.**

***Index Terms*—Duplexing control, Radio frame configuration, Intercell interference, Reinforcement learning, Deep Q-learning, Stackelberg game.**

I. Introduction

ACCORDING to the latest Cisco visual networking index report [1], global mobile data traffic was forecast to grow seven-fold from 2017 to 2022 because of the massive explosion of connecting devices. To manage this sudden surge, small-cell-based wireless and mobile networks (WMNs) have drastically increased [2]. The dense deployments of small cells can provide higher downlink (DL) rate, uplink (UL) rate, and lower latency by shortening the distance between user equipment (UE) and base stations (BSs) [2].

In addition, the vast diversity of user services, which requires various traffic demands from the network, leads to the inevitable highly dynamic DL and UL traffic. For instance, user-behavior measurements from a live mobile network in the city of Vienna, Austria, have been presented that the user traffic demand varies significantly over different time of a day as well as different network cells [3]. In reality, many people desire to watch live broadcast or share user created streaming of exciting events such as live performances, football matches, cultural festivals, or exhibitions, over social media channels. Responsively, network providers must accommodate this dynamic nature of user traffic with a practical duplexing strategy that can configure radio frames dynamically to serve various user demands in real-time and in the best manner. Since the deployment of LTE-Advanced Release 12 (LTE Rel-12) [4], the enhanced interference mitigation and traffic adaptation (eIMTA) strategy has been introduced to offer flexible DL and UL traffic adaptations for TDD systems [4]. The eIMTA provides seven radio frame configuration (RFC) patterns with different allocations of 10 successive radio subframes. Accordingly, each BS can dynamically adapt RFC based on a link-direction selection criteria, e.g., aggregated traffic demand ratio of DL over UL.

Despite these advantages, the coexistence of different link directions over the same frequency resources in adjacent BSs causes severe intercell interference. Typical intercell interference schemes in TDD WMN systems are shown in Fig. 1, which includes downlink-to-downlink (DL–DL), uplink-to-uplink (UL–UL), downlink-to-uplink (DL–UL), and uplink-to-downlink (UL–DL) interferences. In macro deployment schemes, the cross-link interference, especially the DL–UL interference, is a critical problem due to the conflict in the direction and imbalance between DL and UL transmission powers. Consequently, the gain of the adaptive TDD RFC may completely vanish owing to the appearance of the intercell interference. Thus, the effective management of the trade-off between the fulfillment of traffic demand and mitigation of intercell interference is an emerging challenge for network providers.

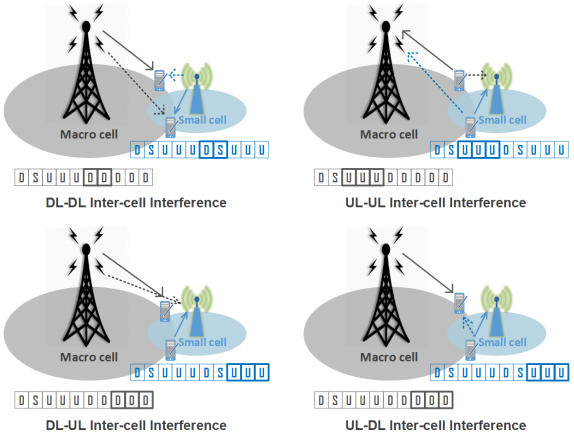
In this paper, we propose an optimal duplexing framework based on the reinforcement learning (RL) algorithm and Stackelberg game theory. The distinct contributions of this study are as follows.

1) Based on the standard TDD RFCs specified in LTE Rel-12 [4], we modeled the network environment with various traffic demands and interference penalties into realizable states. By using the state realizations of the traffic demand model and interference penalty model, we utilized deep RL algorithms for obtaining the optimal TDD RFC policies at every BS. Each BS aims to maximize the gain in terms of achievable data rate and interference reduction.

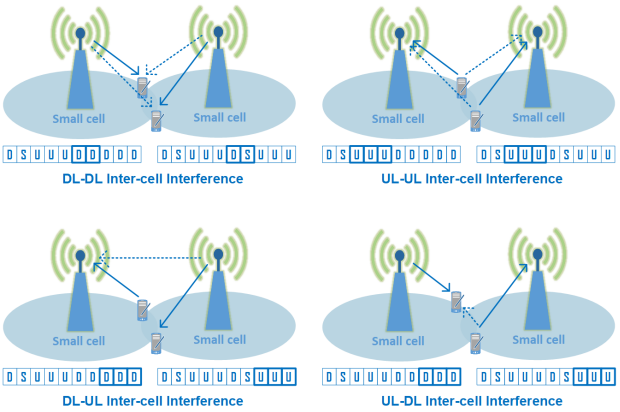
V. D. Tuong, and S. Cho are with the School of Computer Science and Engineering, Chung-Ang University, Seoul 06974, Republic of Korea. *(email: vdtuong@uclab.re.kr, srcho@cau.ac.kr)*

N.-N. Dao is with the Institute of Computer Science, University of Bern, Switzerland. *(email: nhungoc.dao@inf.unibe.ch)*

W. Noh is with the School of Software, Hallym University, Chuncheon 24252, Republic of Korea. *(email: wonjong.noh@hallym.ac.kr)*



(a) Macro cell versus small cell schemes



(b) Small cell versus small cell schemes

Fig. 1. Intercell interference schemes in TDD WMN systems.

2) To achieve a rapid system convergence, we further enhanced the learning processes within a coordination scheme. In this scheme, a single-leader multi-follower Stackelberg game [5] was modeled, in which the leader is the macro BS (MBS) and followers are small BSs (SBSs). The equilibriums of the Stackelberg game were obtained through the deep RL processes, which decide the optimal TDD RFCs for maximum data-rate and interference-reduction objectives.
3) Finally, extensive simulations show the convergence and effectiveness of the proposed framework. Numerical results illustrate that the proposed framework significantly improves the system performance up to approximately 30% in terms of the long-term average data rate and interference reduction, compared to the existing schemes.

The remainder of the paper is organized as follows. Section II gives a brief review of related work. In Section III, the system model is described. Section IV briefly presents the RL algorithm and then describes the training algorithms at the BSs. In Section V, the coordination scheme between the training processes is formulated as a Stackelberg game. Simulation results are discussed in Section VI. Finally, Section VII concludes the paper and outlines future work.

II. Related Work

Extensive studies have been conducted using numerous approaches to address the duplexing control problem for WMN systems [6]–[23]. We can divide these into two main approaches: optimization [6]–[17] and machine learning [18]–[23].

As a mainstream approach, the first approach formulates an optimization problem and then obtains an optimal or suboptimal solution depending on the performance-complexity trade-off. The dynamic adaptation of TDD RFCs in outdoor picocell systems has been analyzed to significantly increase the data rate than that by the conventional synchronous networks [6]. A more specific dynamic TDD scheme with the enhanced local-area small BSs has been investigated using the stochastic geometry algorithm [7]. The decentralized cooperative scheme for DL/UL adaptation is feasible in small-cell networks, relying on the exchange process of low-rate signaling among BSs [8]. Flexible duplexing schemes have been investigated more deeply in an advanced case of a multichannel DL and UL transmission [9]. Here, the non-contiguous transmission strategy is efficiently applied to guarantee that the maximum interference does not exceed an allowable level. Based on a general intercell interference model, the joint DL/UL resource allocation and power control problem in flexible duplexing schemes is tackled to maximize the minimum level of quality-of-service (QoS) satisfaction per link [10], [13]. However, the cross-link interference is assumed to be merely proportional to the traffic load. This assumption is valid only when each resource unit has the same chance to be allocated to DL or UL. The dynamic TDD RFC in multicell scenarios can be optimized by recasting the max–min fairness problem into a fixed point framework [11].

Moreover, another method has been investigated using stochastic geometry analysis [12], [14], in which pairs of DL and UL users who use the same radio resource simultaneously [12] are determined to enhance data rate as minimizing inter-user interference in the dynamic duplexing schemes. Bai et al. [15] introduced a system design and performance aspects for full-duplexing in 5G small-cell networks. An efficient reference-signal design, low-overhead channel state information feedback, and signaling mechanisms have been proposed to enable full-duplexing with considerable data rate gains and significant transmission latency reduction. The intercell radio frame coordination schemes can also be optimized through strategic algorithms as in [16], [17], by using a sliding code book or a cyclic-offset code book.

In practice, WMNs are highly dynamic, where the channels are frequently changed, leading to time-varying solutions. As a result, the optimization solutions must be recomputed every time the system model changes, thus incurring huge network overhead. The overhead grows dramatically in dense networks. Fortunately, machine learning techniques can serve as an effective alternative solution. A multi-agent Q-learning solution has been proposed to obtain the optimized DL/UL switching points for TDD femtocell networks [21], in which

each femto BS is an agent, learning the optimal DL/UL switching policy through trial-and-error search. All agents are non-cooperative; they consider the local states of traffic demand and interference, thus each agent guaranteeing a convergence. However, a system convergence is not guaranteed in which all agents converge simultaneously. In addition, such a machine-learning system fails to consider both the dynamic TDD RFC scheme of the macro BS and the TDD RFCs defined by 3GPP. Another machine-learning framework, called *Downlink to Uplink Ratio Determination* (DIANA), was developed to adjust the TDD RFCs for the hybrid optical-wireless networks [22]. This framework succeeds in suitably changing TDD RFC by sensing the traffic changes in the network based on software-defined networking (SDN) controller knowledge.

Specifically, game theory has been incorporated with RL to address the problem of self organization in small-cell networks [18]. The formulated game model considers each BS as one player, locally learning to optimize transmission configuration while mitigating the interference. The proposed model shows a convergence to an epsilon Nash equilibrium when all small BSs share the same interest. A non-cooperative game, an extension of that in [18], was formulated, in which the small BSs are players, and each player learns from its local traffic load, interference to update DL/UL switching point [19], or TDD RFC [20]. Similar to [21], none of these studies considered the dynamic TDD RFC scheme of the macro BS. Recently, the TDD reconfiguration schemes in TDD indoor small-cell networks were considered as a multi-agent Q-learning process, in which the objective is to maximize the quality-of-experience (QoE) for UEs [23]. However, this method does not guarantee convergence.

While some excellent works have been proposed to address the TDD control problem, they are under recognizable restrictions. Methods following the optimization approach hardly adapt to changes in the network model owing to their heavy computations. In contrast, the methods following the machine-learning approach can adapt gradually to the network changes; however, none of the existing studies has considered the dynamic TDD RFC scheme of the macro BS in addition to guaranteeing system convergence. Therefore, we are motivated to research and develop an optimal duplexing solution for WMN systems.

III. System Model

In this section, we first analyze the network model, and then discuss the traffic demand, interference penalty, and spectral efficiency.

A. Network Model

A typical TDD WMN model consists of three major entities: MBSs, SBSs, and UEs. Each SBS connects directly to one local MBS and communicates with the local MBS via the *Xn* interface. SBSs do not communicate with each other directly; however, they can communicate via connections through the MBS. For simplicity, all BSs are assumed to operate on a single channel band.

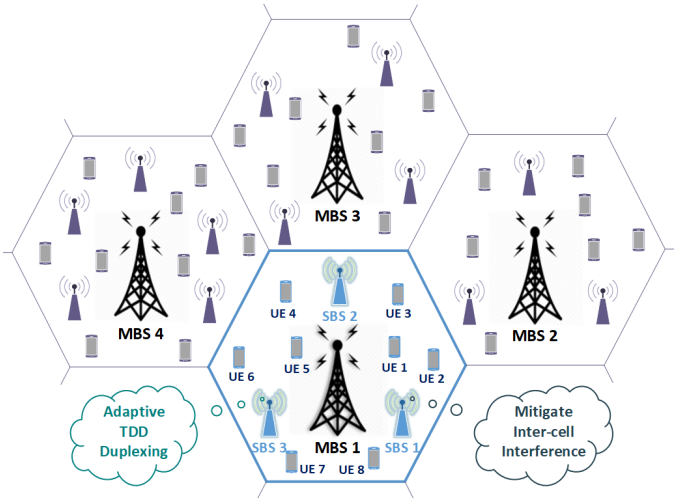


Fig. 2. Wireless and mobile networks from a system-model perspective.

Fig. 2 illustrates an WMN system model, where each hexagon is one MBS coverage area, comprising one MBS, K connected SBSs, and U served UEs, operating in the TDD mode. We assume that all SBSs work in a synchronized manner, i.e., they update TDD RFCs simultaneously. Moreover, the updating timescale of each BS is assumed to be one radio frame. Let $\mathcal{K} = \{k \mid k = 0, 1, \ldots, K\}$ denote the set of indices of $K + 1$ BSs in one MBS coverage area, where $k = 0$ indicates the MBS and $k > 0$ indicates the SBS k. Let $\mathcal{U} = \{u \mid u = 1, 2, \ldots, U\}$ denote the set of indices of U served UEs, where U is the number of subscribed UEs. The UEs scan to associate to a BS, which has the maximal received signal strength indication (RSSI). Let $\phi_{u,k}$ denote the association between UE u and BS k, where $\phi_{u,k} = 1$ indicates that UE u associates with BS k; otherwise $\phi_{u,k} = 0$. We assume that all BSs operate in the closed-access mode, implying that each BS has its own subscribed UEs, and no handover is considered.

In the network, each BS selects a TDD configuration from a set of 7 LTE TDD RFCs [4], as shown in Fig. 3. We assume that there are T time slots, each of which is denoted by t, with $t = 0, 1, \ldots, T-1$. Let $C = \{0, 1, \ldots, 6\}$ denote the set of 7 TDD RFCs specified in [4], and $c_k(t) \in C$ is the TDD RFC of BS k at time slot t. Note that different TDD RFCs direct the BS to serve DL and UL traffic at different subframes, leading to the ability to adapt to various traffic demands.

B. Traffic Demand Model

The network model supports a wide diversity of user services and various traffic demands. Each BS maintains DL buffer partitions, where each partition stores the DL traffic demand of one UE [21]. In addition, each UE has a UL buffer to store its UL traffic demand, and all buffers are assumed to be sufficient for this purpose. We denote $q_u^{DL}(t)$ and $q_u^{UL}(t)$ as the DL and UL queue lengths of UE u at time slot t, respectively. The DL and UL residual bit numbers at time slot t of BS k are $Q_k^{DL}(t)$ and $Q_k^{UL}(t)$, respectively. Then, the value of DL/UL

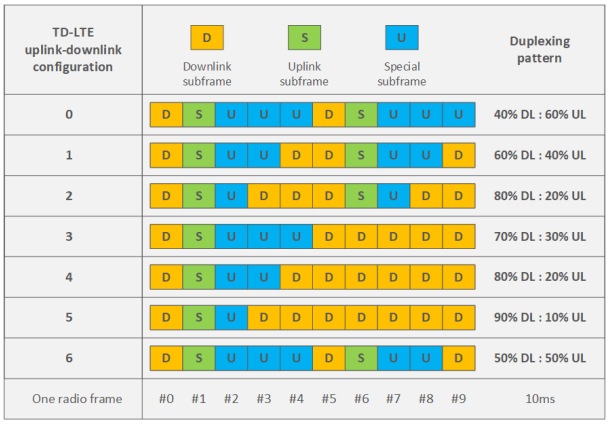


Fig. 3. TD-LTE uplink–downlink configuration patterns.

traffic demand ratio (TDR) at time slot t for BS k is given by

$$\psi_k(t) = \frac{Q_k^{DL}(t)}{Q_k^{UL}(t)} = \frac{\sum_{u\in U} q_u^{DL}(t)}{\sum_{u\in U} q_u^{UL}(t)}, \ \phi_{u,k} = 1. \tag{1}$$

For any BS k, the traffic arrivals follow the Poisson arrival process with aggregated DL rate λ_k^{DL} and aggregated UL rate λ_k^{UL}. The DL and UL packet sizes in bits are fixed, and can be denoted as δ_k^{DL} and δ_k^{UL}, respectively. We respectively denote $P_m(k)$ and $P_n(k)$ as the probabilities of m arrived DL packets and n arrived UL packets for BS k in time interval T as

$$P_m(k) = \frac{(\lambda_k^{DL}T)^m}{m!}\exp(-\lambda_k^{DL}T), \tag{2}$$

$$P_n(k) = \frac{(\lambda_k^{UL}T)^n}{n!}\exp(-\lambda_k^{UL}T). \tag{3}$$

The TDRs of all BSs are realistic time-varying ratios; hence, we can model them as finite-state Markov chains (FSMC) [24]. Because seven TDD RFCs serve six DL/UL TDRs from 40%/60% to 90%/10%, we can partition and quantize the value of random variable ψ_k into six discrete levels [1, 2, 3, 4, 5, 6], i.e., TDR = 1, 2, ..., 6 if $\psi_k \in$ (0, 45%/55%], (45%/55%, 55%/45%], ..., (85%/15%, ∞), respectively. Each level corresponds to a state of the Markov chain, and thus it forms six-element state-space $\mathcal{D} = \{1, 2, \ldots, 6\}$. The TDR state realization of ψ_k at time slot t can be denoted as $d_k(t)$. According to certain transition probabilities, the received TDR $d_k(t)$ varies from one state to another when one time-slot elapses. The transition probability that $d_k(t)$ jumps from one state x_k to another state y_k at time slot t can be denoted as $\Theta_{x_k,y_k}(t)$. The DL/UL TDR state transition probability matrix, $\Psi_k(t)$, is defined as

$$\Psi_k(t) = [\Theta_{x_k,y_k}(t)]_{6\times 6}, \tag{4}$$

where $\Theta_{x_k,y_k}(t) = \Pr(d_k(t+1) = y_k \mid d_k(t) = x_k)$ and $x_k, y_k \in \mathcal{D}$.

C. Interference Penalty Model

One of the main objectives of this work is to mitigate interference between BSs. In the network model, interference consists of intercell and intracell interferences, where intercell interference considers the forward-link interference (FLI) and cross-link interference (CLI). As all BSs operate on a single-channel band, the intracell interference is well-mitigated. Hence, we modeled an interference penalty (IP) according to the level of the intercell interference only. As a result, the objective turns into the minimizing of the IP of the BSs.

Each TDD RFC is constructed from 10 subframes, as shown in Fig. 3, where each subframe indicates resource allocation for a DL or UL traffic direction, depending on its predefined character. In one frame, character U indicates an allocation for UL traffic, D indicates an allocation for DL traffic, and S indicates a switch from DL to UL traffic. Specifically, switching subframe S is divided into three consecutive portions. The first is one small DL portion, the second is the gap portion, and last is one small UL portion. There are nine sub-configurations of the special subframe, in which the DL portion occupies almost the entire subframe. Thus, for simplicity, we assume that S also indicates an allocation for DL traffic.

At a subframe time slot, if two adjacent BSs are under a traffic asymmetry, e.g., BS k_1 serves subframe D while BS k_2 serves subframe U, a severe CLI is formulated between them. In contrast, if two adjacent BSs are under a traffic symmetry, e.g., both BS k_1 and BS k_2 serve subframe D, an FLI is formulated between them. We observed that the number of similar and different subframes between TDD RFCs can present a level of intercell interference. We propose a novel algorithm, namely IP of Configurations (IPC), to estimate the IP of BSs based on their TDD RFCs. We denote $\tilde{I}(c_{k_1}, c_{k_2})$ as the IP of BS k_1 during one frame, considering intercell interference from neighboring BS k_2, where c_{k_1} and c_{k_2} are their TDD RFCs. The value of $\tilde{I}(c_{k_1}, c_{k_2})$ can be calculated as

$$\tilde{I}(c_{k_1}, c_{k_2}) = w_F F(c_{k_1}, c_{k_2}) + w_C C(c_{k_1}, c_{k_2}), \tag{5}$$

where $F(c_{k_1}, c_{k_2})$ and $C(c_{k_1}, c_{k_2})$ represent the numbers of similar and different subframes between c_{k_1} and c_{k_2}, respectively, and w_F and w_C specify the weights of the penalty for FLI and CLI, respectively. In the dynamic TDD schemes, the value of w_C should be larger than that of w_F because cross-link channels are highly contradictory to each other, while forward-link channels are nearly in the same direction.

Based on the predefined LTE TDD RFCs, we formulated the values of $F(c_{k_1}, c_{k_2})$ and $C(c_{k_1}, c_{k_2})$ as listed in Tab. I.

D. Spectral Efficiency

Given the realization of DL/UL TDR and TDD RFC states of other BSs, each BS decides a TDD RFC. Then, it observes a return in terms of the DL and UL rates. We adopted a block fading model, which includes one each of large-scale and small-scale fading components [25], [26]. The channel gain at time slot t is obtained as follows.

- For the DL from BS k to UE u,

$$g_{k,u}(t) = |h_{k,u}(t)|^2 \alpha_{k,u} \tag{6}$$

- For the UL from UE u to BS k,

$$g_{u,k}(t) = |h_{u,k}(t)|^2 \alpha_{u,k}, \tag{7}$$

TABLE I
NUMBER OF SIMILAR AND DIFFERENT SUBFRAMES BETWEEN TDD CONFIGURATIONS

RFC No. (FLI|CLI)	0 (F|C)	1 (F|C)	2 (F|C)	3 (F|C)	4 (F|C)	5 (F|C)	6 (F|C)
0	10| 0	8 | 2	6 | 4	7 | 3	6 | 4	5 | 5	9 | 1
1	8 | 2	10| 0	8 | 2	7 | 3	8 | 2	7 | 3	9 | 1
2	6 | 4	8 | 2	10| 0	7 | 3	8 | 2	9 | 1	7 | 3
3	7 | 3	7 | 3	7 | 3	10| 0	9 | 1	8 | 2	8 | 2
4	6 | 4	8 | 2	8 | 2	9 | 1	10| 0	9 | 1	7 | 3
5	5 | 5	7 | 3	9 | 1	8 | 2	9 | 1	10| 0	6 | 4
6	9 | 1	9 | 1	7 | 3	8 | 2	7 | 3	6 | 4	10| 0

where $h_{k,u}(t)$ and $h_{u,k}(t)$ are the small-scale fading components at time slot t; $\alpha_{k,u}$ and $\alpha_{u,k}$ are the large-scale fading components, which do not change over the time slots.

In this study, we assume that all BSs do not change DL transmission power, and all UEs have the same UL transmission power. Let P_k denote the DL transmission power of BS k, and p denote the UL transmission power of UEs. The signal-to-interference-plus-noise ratio (SINR) of DL and UL traffic at time slot t are calculated with both FLI and CLI as follows:

- For the DL from BS k to UE u,

$$\Upsilon_{k,u}(t) = \frac{g_{k,u}(t)P_k}{\sum_{l\in\mathcal{K}\setminus\{k\}} g_{l,u}(t)P_l + \sum_{v\in\mathcal{U}\setminus\{u\}} g_{v,u}(t)p + \sigma^2} \quad (8)$$

- For the UL from UE u to BS k,

$$\Upsilon_{u,k}(t) = \frac{g_{u,k}(t)p}{\sum_{l\in\mathcal{K}\setminus\{k\}} g_{l,k}(t)P_l + \sum_{v\in\mathcal{U}\setminus\{u\}} g_{v,k}(t)p + \sigma^2}, \quad (9)$$

where σ^2 is the additive white Gaussian noise power spectral density, which is assumed to be the same at all receivers.

During one radio frame, each BS was assumed to serve one UE with its single channel band. Then, system rate ($S_k(F)$) of BS k during one radio frame F can be expressed as a weighted sum of DL rate ($S_{k,u}(t)$) within DL subframes and UL rate ($S_{u,k}(t)$) within UL subframes as follows.

$$\begin{aligned} S_k(F) &= w_D S_{k,u}(t) + w_U S_{u,k}(t) \\ &= w_D \sum_{F_D} W_k log_2(1+\Upsilon_{k,u}(t)) \\ &\qquad + w_U \sum_{F_U} W_k log_2(1+\Upsilon_{u,k}(t)), \end{aligned} \quad (10)$$

where w_D and w_U are the weights of the DL and UL, and they specify how much DL and UL rates contribute to the system rate; F_D and F_U are the sets of DL and UL subframes in frame F; W_k is the single channel bandwidth; and u is the served UE of BS k.

IV. DEEP REINFORCEMENT LEARNING FOR DUPLEXING CONTROL

In this section, we first introduce RL and deep Q-learning algorithms and then describe the learning algorithms of the MBS and SBSs.

A. Overview of RL and Deep Q-learning

RL is a robust machine learning technique, which aims to maximize long-term rewards [27]. An RL agent learns to take action by interacting with the environment. The outstanding features of RL are trial-and-error search and delayed reward. Trial-and-error search presents a trade-off between exploration and exploitation based on a specific probability, while delayed reward indicates that the agent can consider either an immediate or cumulative rewards in the long run in the value function. This flexibility can be achieved by setting discounting factors to cumulative rewards [28]. In RL, the environment can be described as a Markov decision process (MDP), in which the state space, state transition probability, and reward function are not necessarily required [27]. We can classify RL into model-based and model-free based on the existence of environmental state-transition probability. Model-based RL must perform supervised learning with an inherent model [29], [30]. In contrast, model-free RL can learn parameters from zeros. Some recent studies have shown that model-free reinforcement can handle deep neural networks effectively [30]–[32]. The raw state representations of complex systems can be inputted directly to the neural networks for training.

Q-learning is a typical model-free RL algorithm [33], which stores Q-value for each state-action pair. The Q-value can be implemented using a look-up table or evaluated by a nonlinear approximator, i.e., a deep neural network. Deep Q-learning is introduced initially to teach machines to play games without human control [34]. It uses neural networks to process the raw state representation input directly. The key idea of the deep Q-learning algorithm is to approximate the Q-value by using a deep neural network, so-called the deep Q-network (DQN). Given neural network parameters θ, the Q-value function can be represented by $Q(s,a;\theta)$, in which s and a are the state vector and action, respectively. The neural network is trained by updating θ to approximate the Q-value based on the interacting experiences of the agent. Mnih et al. [30] proved that deep Q-learning is more advantageous than conventional Q-learning with higher performance and faster convergence. However, the performance of the deep Q-learning algorithm might not be stable owing to the use of a nonlinear approximator. Therefore, an advanced version of deep Q-learning, namely deep double Q-learning [35], was developed to address this issue, and it shows the following three improvements.

1) *Feature set:* We determined the state features to feed into the multilayer deep convolution networks, which utilize hierarchical layers of tiled convolution filters, to exploit the local spatial correlations and make it possible to extract high-level features from raw input data [27], [36]. As such, all features of each state are trained in the deep convolution neural network.

2) *Experience replay mechanism:* The algorithm stores interaction experience tuples, $\text{ex}(t) = \langle s_t, a_{s_t}, r_t, s_{t+1} \rangle$, into a replay memory pool, $M(t) = \{\text{ex}(1), \ldots, \text{ex}(t)\}$. The learning process was performed using random samples from the memory pool rather than directly using the consecutive samples as in Q-learning. This allows the network to learn efficiently by randomly considering any experience instead of focusing on the immediate experience. The algorithm also breaks down the correlations between observations to achieve better stability.
3) *Target Q-network:* We adopted a second neural network for updating the target Q values. In the training process, value estimations can be out of control if one network is used for both estimated and target Q values. Thus, another target network was set to reduce the correlations between the target and estimated Q-values, and it can improve the stability of the algorithm.

In the training phase of deep double Q-learning algorithm, multiple episodes are implemented. In each episode, a state is observed, and then the agent selects an action based on the ϵ-greedy strategy, which ensures both exploration and exploitation. The algorithm prefers exploration at the beginning with a reasonably randomized policy and later slowly moves toward exploiting a deterministic policy. Next, the system performs the selected action and observes a reward and next state. The experience tuple is then saved to the replay memory for the training process at later steps. Random batches of experience are sampled from the replay memory and fed into the neural networks for training. A loss function is formulated between the estimated and target Q-values. The algorithm then updates network parameters by minimizing the loss function at each iteration. Loss function was minimized by mini-batch Stochastic Gradient Descent (SGD) algorithm, which has the benefits of computation cost and training speed. Loss function $L(\theta)$ can be presented as follows:

$$L(\theta) = \mathop{\mathbb{E}}_{\langle s, a_s, r, s' \rangle} \left[\left(\bar{y} - Q(s, a_s; \theta) \right)^2 \right], \tag{11}$$

where $\bar{y}$ is the target Q-value of the target Q-network and θ is the parameters of the training Q-network. The target Q-value is calculated as follows:

$$\bar{y} = r + \gamma Q\big(s', \operatorname*{argmax}_{a_{s'}} Q(s', a_{s'}; \theta); \bar{\theta}\big), \tag{12}$$

where γ is the discounting factor and $\bar{\theta}$ is the parameters of the target Q-network. Here, $\bar{\theta}$ can be updated every G steps.

In this work, we considered the WMN model under realistic scenarios, in which the traffic demand of each BS dynamically changes. Moreover, the duplexing strategies of the neighboring BSs are strictly related as they can either mutually reduce or enlarge the interference. As a result, the optimal duplexing strategy of one BS must consider both the various traffic demands and duplexing strategies of the other BSs. This leads to a large number of system states for any BS, especially when the number of SBSs increases. Moreover, the advantages of deep double Q-learning algorithm can help solving the large state-space problems effectively. Therefore, we propose to use the deep double Q-learning algorithm for training TDD RFC policies of the BSs.

Algorithm 1 MBS deep double Q-learning algorithm

Initialize experience replay buffer.
Initialize training Q-network Q with parameters θ_M.
Initialize target Q-network Q with parameters $\bar{\theta}_M = \theta_M$.
% Training
for episode $n = 1, \ldots, N$ **do**
    Observe the environment and formulate the MBS beginning state $\Xi_0(1)$, including
        DL/UL TDR of MBS.
        Previous TDD RFCs of all SBSs.
    for $t = 1, 2, \ldots, T$ **do**
        Choose an action $a_0(t)$ based on ϵ-greedy strategy.
        Perform action $a_0(t)$
        Observe reward $R_0(t)$ and next state $\Xi_0(t+1)$.
        Store experience tuple $\langle \Xi_0(t), a_0(t), R_0(t), \Xi_0(t+1) \rangle$ into the replay buffer.
        Sample a random batch of M experience tuples.
        Calculate target Q-value $\bar{y}$ of the target Q-network.
        Calculate the loss $L(\theta_M)$.
        Perform SGD on $L(\theta_M)$ with respect to θ_M.
        Update the training Q-network parameters θ_M.
        Every G steps, update the target Q-network parameters with rate σ.
        $\bar{\theta}_M = \sigma \theta_M + (1 - \sigma) \bar{\theta}_M$.
    end for
end for

B. MBS Deep Double Q-learning Algorithm

We consider the time scale over radio frames for MBS training, in which one time-slot is one radio frame. To obtain the optimal RFC policy for the MBS, it is necessary to identify the states, actions, and reward function, as described in the following subsections.

1) States: The state of the MBS at time frame t is determined by the realization of state $d_0(t)$ of random variable ψ_0 and the realization of previous TDD RFC states $\{c_k(t-1) \mid k = 1, 2, \ldots, K\}$ of all K SBSs. Consequently, the state vector of the MBS at time frame t can be described as follows:

$$\Xi_0(t) = \left[d_0(t), \; c_1(t-1), \; \ldots, \; c_K(t-1) \right]. \tag{13}$$

2) Actions: The MBS selects a random or specific TDD RFC for exploration or maximizing the long-term return, respectively. We denote the action of the MBS as

$$a_0(t) \in C = \{0, \; 1, \; 2, \; \ldots, \; 6\}. \tag{14}$$

3) Reward function: In this work, we consider the weighted sum of DL and UL rates to be the reward of the MBS. Additionally, the reward must have a penalty for raising intercell interference in the network. Therefore, we define the reward of the MBS as follows:

$$R_0(t) = S_0(t) - \sum_{k \in \mathcal{K} \setminus \{0\}} \tilde{I}(a_0(t), c_k(t-1)). \tag{15}$$

At time frame t, the MBS achieves reward $R_0(t)$ when action $a_0(t)$ is performed with observed state $\Xi_0(t)$. The

Algorithm 2 SBS deep double Q-learning algorithm

1: Initialize experience replay buffer.
2: Initialize training Q-network Q with parameters θ_S.
3: Initialize target Q-network Q with parameters $\bar{\theta}_S = \theta_S$.
% Training
4: **for** episode $n = 1, \ldots, N$ **do**
5: Observe the environment and formulate the SBS beginning state $\Xi_k(1)$, including
6: DL/UL TDR of SBS k.
7: Previous TDD RFCs of MBS, neighboring SBSs.
8: **for** $t = 1, 2, \ldots, T$ **do**
9: Choose an action $a_k(t)$ based on ϵ-greedy strategy.
10: Perform action $a_k(t)$
11: Observe reward $R_k(t)$ and next state $\Xi_k(t+1)$.
12: Store experience tuple $\langle \Xi_k(t), a_k(t), R_k(t), \Xi_k(t+1) \rangle$ into the replay buffer.
13: Sample a random batch of M experience tuples.
14: Calculate target Q-value $\bar{y}$ of the target Q-network.
15: Calculate the loss $L(\theta_S)$.
16: Perform SGD on $L(\theta_S)$ with respect to θ_S.
17: Update the training Q-network parameters θ_S.
18: Every G steps, update the target Q-network parameters with rate σ.
19: $\bar{\theta}_S = \sigma\theta_S + (1-\sigma)\bar{\theta}_S$.
20: **end for**
21: **end for**

goal is to find the optimal RFC policy for the MBS to maximize the long-term return. Thus, the duplexing control problem of the MBS can be expressed as follows:

$$R_0^{long} = \max_{a_0(t)} \mathbb{E}\Big[\sum_{t=0}^{T-1} \xi^t R_0(t) \Big], \tag{16}$$

where ξ^t approaches zero when t is large enough.

The training algorithm of the MBS is described in Alg. 1. Here, the MBS collects the status of the current traffic demand and previous TDD RFCs of SBSs. Then, it assembles the whole information into a system state and processes it to obtain a TDD RFC action based on the ϵ-greedy strategy (see lines 5–9 in Alg. 1). The network model returns a reward in terms of achievable system rate and interference penalty, and the experience is saved to the replay memory (see lines 10–12 in Alg. 1). DQN parameters θ is updated after performing gradient descent on the loss of training Q-values with mini-batch samples of experience (see the last lines in Alg. 1).

C. SBS Deep Double Q-learning Algorithm

Each SBS is deployed a deep double Q-learning algorithm to learn its optimal RFC policy. We consider that an SBS has some neighboring SBSs that significantly generate intercell interference. Then, we set an interference threshold to specify the neighboring SBSs. As the positions of SBSs in the network are not frequently changed, we assume that during T time frames, the neighboring SBSs of any SBS also do not change. Let $\mathcal{Z}_k = \{z_1, \ldots, z_Z\}$ denote the set of Z neighboring SBSs of SBS k.

Similar to the MBS training, we consider the time scale over radio frames for SBS training, where one time-slot is one radio frame. To obtain the optimal TDD RFC policies for SBSs, it is necessary to identify the states, actions, and reward function, described as follows.

1) States: The state of SBS k at time frame t is determined by realizing state $d_k(t)$ of random variable ψ_k and the previous TDD RFC states $\{c_0(t-1),\ c_{z_1}(t-1),\ \ldots,\ c_{z_Z}(t-1)\}$ of the MBS and Z neighboring SBSs. Consequently, the state vector of SBS k at time frame t can be described as follows:

$$\Xi_k(t) = \left[d_k(t),\ c_0(t-1),\ c_{z_1}(t-1),\ \ldots,\ c_{z_Z}(t-1) \right]. \tag{17}$$

2) Actions: Each SBS selects a random or specific TDD RFC for exploration or maximizing the long-term return, respectively. We denote the action of SBS k as follows:

$$a_k(t) \in C = \{0,\ 1,\ 2,\ \ldots,\ 6\}. \tag{18}$$

3) Reward function: Similar to the MBS, each SBS receives a comprehensive reward, which is the weighted sum of DL and UL rates, in addition to a negative penalty owing to intercell interference. Therefore, we can define the reward of SBS k as follows:

$$\begin{aligned} R_k(t) =& S_k(t) - \tilde{I}(a_k(t), c_0(t-1)) \\ & - \sum_{z \in \mathcal{Z}_k} \tilde{I}(a_k(t), c_z(t-1)). \end{aligned} \tag{19}$$

At any time frame t, SBS k achieves reward $R_k(t)$ when action $a_k(t)$ is performed with observed state $\Xi_k(t)$. The goal is to find the optimal TDD RFC policy for SBS k to maximize the long-term return. Thus, the duplexing control problem of SBS k can be expressed as follows:

$$R_k^{long} = \max_{a_k(t)} \mathbb{E}\Big[\sum_{t=0}^{T-1} \xi^t R_k(t) \Big], \tag{20}$$

where ξ^t approaches zero when t is large enough.

Alg. 2 is the SBS training algorithm, where each SBS collects the status of the current traffic demand and previous TDD RFCs of the MBS and neighboring SBSs. Then, it assembles the whole information into a system state and processes it to obtain a TDD RFC action based on the ϵ-greedy strategy (see lines 5–9 in Alg. 2). The network model returns a reward in terms of achivable system rate and interference penalty, and the experience is saved to the replay memory (see lines 10–12 in Alg. 2). DQN parameters θ is updated after performing gradient descent on the loss of training Q-values with mini-batch samples of experience (see the last lines in Alg. 2).

V. Hierarchical Stackelberg Game

Next, we propose a coordination scheme for training the MBS and SBSs as a single-leader multi-follower Stackelberg game [5], in which the leader is the MBS and followers are SBSs.

The relationship between the MBS leader and its SBS followers is illustrated in Fig. 4. The RFC policy training

processes of the BSs are adjusted to follow the Stackelberg game's hierarchical architecture. The leader and followers sequentially select their TDD RFCs, which aim to maximize the long-term return in terms of system rate while mitigating intercell interference. We consider that each game turn is scaled to one radio frame. For example, during frame F, the MBS leader observes traffic demand and TDD RFCs of all SBSs, and then selects a TDD RFC $c_0(F+1)$ based on Alg. 1 at the beginning of frame $F+1$. The SBS followers observe their traffic demands during frame $F+1$ as well as the RFC of the MBS leader at this frame along with RFCs of other SBSs at frame F; they then simultaneously select their TDD RFCs at the beginning of frame $F+2$ based on Alg. 2. These TDD RFCs are transferred back to the MBS leader for playing a new turn at frame $F+3$. The system will iterate game turns at the next frames until achieving the game equilibrium.

Generally, a Stackelberg game framework is optimized by seeking equilibrium, which includes the Nash equilibrium (NE) among the followers and the Stackelberg–Nash equilibrium (SNE) between the leader and followers [5]. Let $f_k(c_0, c_1, \ldots, c_k, \ldots, c_K)$ with $k \in 1, \ldots, K$ and $F(c_0, c_1, \ldots, c_K)$ denote the utility functions of SBS follower k and MBS leader, respectively. We can define NE and SNE for our Stackelberg game as follows.

Definition 1. *The NE among K SBS followers is a feasible RFC policy vector* $(c_1^*, \ldots, c_k^*, \ldots, c_K^*)$*, which satisfies*

$$f_k(c_0, c_1^*, \ldots, c_k^*, \ldots, c_K^*) \geqslant f_k(c_0, c_1^*, \ldots, c_k, \ldots, c_K^*), \quad (21)$$

where $k \in 1, \ldots, K$ *and* c_0 *is a given TDD RFC of the MBS leader.*

Definition 2. *The SNE between the MBS leader and SBS followers is a feasible RFC policy vector* $(c_0^*, c_1^{**}, \ldots, c_K^{**})$*, which satisfies*

$$F(c_0^*, c_1^{**}, \ldots, c_K^{**}) \geqslant F(c_0, c_1^*, \ldots, c_K^*), \quad (22)$$

where $(c_1^{**}, \ldots, c_K^{**})$ *and* $(c_1^*, \ldots, c_K^*)$ *are the NEs with respect to* c_0^* *and* c_0*, respectively.*

Then, we can formulate a general problem to find the game equilibrium (NE and SNE) as follows:

$$\max_{c_0} \quad F(c_0, c_1^*, \ldots, c_K^*), \quad (23)$$

$$\text{s.t.} \quad c_k^* = \underset{c_k}{\operatorname{argmax}}\, f_k(c_0, c_1^*, \ldots, c_k, \ldots, c_K^*). \quad (24)$$

In our Stackelberg game, the subgame of each SBS follower (Eq. (20)) optimizes its TDD RFC, considering the TDD RFCs of the MBS leader and Z neighboring SBS followers. As the effect of non-neighboring SBS followers on the utility of one SBS follower is small, we can apply the solution of the followers in (20) to the game equilibrium problem in (24). Furthermore, the subgame of the MBS leader (Eq. (16)) optimizes its TDD RFC by considering the TDD RFCs of all SBS followers. Therefore, the solution of the leader in (16) achieves the perfect Stackelberg game equilibrium in (23).

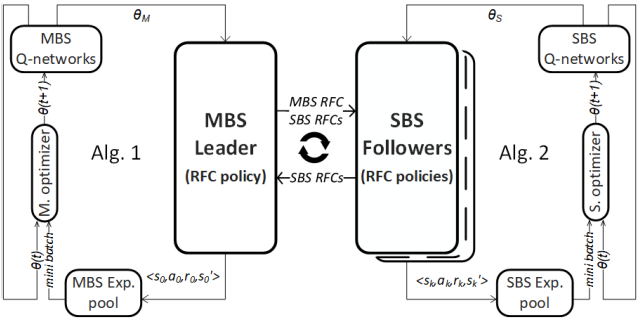


Fig. 4. Hierarchical relationship between the MBS and SBSs in the Stackelberg game.

VI. Performance Evaluation

In this section, simulations were conducted to illustrate the performance of the proposed duplexing framework under different parameter settings and the results were compared with those of the existing duplexing schemes.

A. Simulation Settings

For the simulation environment, we set up a GPU-based server empowered by Nvidia GPU version GTX 1050 Ti. In addition, the server has a CPU Intel Core i5-8500 with 250-GB memory. We implemented our simulations on Matlab R2019a [37] with the Reinforcement Learning Toolbox, Python 3.7, and Windows 10 Education. It provides functions and blocks for training policies using RL algorithms. The system also supports the effective training of policies using deep neural networks such as deep double Q-learning. We extended the built-in features of the Reinforcement Learning Toolbox to implement the MBS and SBS deep double Q-learning algorithms.

To evaluate the advantages of the proposed framework, we additionally consider the following three popular schemes for a comprehensive comparison:

1) *Fixed RFC strategy*, wherein all BSs maintain their fixed RFCs. This scheme installs a TDD RFC for each BS, and the TDD RFC does not change regardless of the changes of traffic demand in the network.
2) *Random RFC strategy*, wherein all BSs arbitrarily select their own TDD RFC among standard RFCs [4] at the beginning of every frame.
3) *Traffic-matched RFC strategy*, wherein the network selects RFCs that best match each traffic demand (Tab. II). In this scheme, each BS computes the remaining sum of DL and UL traffic from the associated UEs. Next, BS chooses the RFC closest to the DL/UL TDR.

We set up a network model, in which all SBSs are uniformly distributed around the MBS, while UEs are randomly distributed within the MBS coverage area. We assume that the bandwidth of each BS is normalized. The various DL/UL TDR states, which we defined in section III-B, follow the Markov model. In our simulations, we used different Poisson DL and UL arrival rates for investigating the efficiency of the proposed framework in various scenarios. Based on the generated arrival-rate datasets, we obtained the transition probability

TABLE II
RECONFIGURATION STRATEGY OF THE TRAFFIC-MATCHED SCHEME

Condition	Selected configuration
DL/UL TDR ≤ 45%/55%	0
DL/UL TDR ≤ 55%/45%	6
DL/UL TDR ≤ 65%/35%	1
DL/UL TDR ≤ 75%/25%	3
DL/UL TDR ≤ 85%/15%	2 or 4, randomly
Otherwise	5

between six DL/UL TDR states in three different scenarios, as follows.

- *Network system mostly maintains a DL/UL TDR (Scenario Ψ_1)*, e.g., the networks in smart factories, smart buildings, and smart offices. Here, the highest probability remains at the previous DL/UL TDR state.

$$\Psi_1 = \begin{bmatrix} 0.4 & 0.25 & 0.15 & 0.1 & 0.06 & 0.04 \\ 0.2 & 0.4 & 0.2 & 0.1 & 0.06 & 0.04 \\ 0.08 & 0.2 & 0.4 & 0.2 & 0.08 & 0.04 \\ 0.04 & 0.08 & 0.2 & 0.4 & 0.2 & 0.08 \\ 0.04 & 0.06 & 0.1 & 0.2 & 0.4 & 0.2 \\ 0.04 & 0.06 & 0.1 & 0.15 & 0.25 & 0.4 \end{bmatrix} \quad (25)$$

- *Network system tends to change DL/UL TDR by a small amount to that the level of a neighbor (Scenario Ψ_2)*, e.g., the networks of institutions, schools, and campuses. The highest probability is obtained in the cases of changing DL/UL TDR state to a neighbor level.

$$\Psi_2 = \begin{bmatrix} 0.2 & 0.4 & 0.2 & 0.1 & 0.06 & 0.04 \\ 0.35 & 0.15 & 0.35 & 0.07 & 0.05 & 0.03 \\ 0.08 & 0.35 & 0.1 & 0.35 & 0.08 & 0.04 \\ 0.04 & 0.08 & 0.35 & 0.1 & 0.35 & 0.08 \\ 0.03 & 0.05 & 0.07 & 0.35 & 0.15 & 0.35 \\ 0.04 & 0.06 & 0.1 & 0.2 & 0.4 & 0.2 \end{bmatrix} \quad (26)$$

- *Network system tends to change DL/UL TDR by a large amount to a different level (Scenario Ψ_3)*, e.g., the networks in specific places such as stadiums, parks, exhibitions, and convention centers. Here, the highest probability is the change of the DL/UL TDR state to a distinct level.

$$\Psi_3 = \begin{bmatrix} 0.04 & 0.06 & 0.1 & 0.15 & 0.25 & 0.4 \\ 0.08 & 0.04 & 0.08 & 0.15 & 0.25 & 0.4 \\ 0.2 & 0.08 & 0.04 & 0.08 & 0.2 & 0.4 \\ 0.4 & 0.2 & 0.08 & 0.04 & 0.08 & 0.2 \\ 0.4 & 0.25 & 0.15 & 0.08 & 0.04 & 0.08 \\ 0.4 & 0.25 & 0.15 & 0.1 & 0.06 & 0.04 \end{bmatrix} \quad (27)$$

The simulation parameters are summarized in Tab. III.

B. Simulation Results

First, we investigate the effects of the DL and UL weights on the convergence of the proposed framework, followed by the performance comparisons with the existing duplexing strategies in terms of system-rate achievement and intercell interference reduction. The former TDR transition probability matrix, in which the network system mostly maintains a DL/UL TDR, is selected as the baseline. Then, we finally prove the outstanding performance of the proposed framework over various TDR transition probabilities.

TABLE III
SIMULATION PARAMETERS

Parameter	Value	Description
$Sect$	3	The number of sectors of the MBS
K	$\{3, 6, 9, 12\}$	The number of SBSs
W_k	10 MHz	The bandwidth of BS k allocated to UEs
P_0	46 dBm	The transmit power of the MBS
P_k	30 dBm	The transmit power of SBS k
p	20 dBm	The uploading power of UEs
w_F	0.03	The weight of forward-link interference
w_C	0.07	The weight of cross-link interference
T	200000	The number of training time frames
γ	0.7	The discounting factor
α	0.9	The learning rate, $alphaDecay$ = 0.9998
ϵ	0.7	The exploration probability, $Decay$ = 0.9994

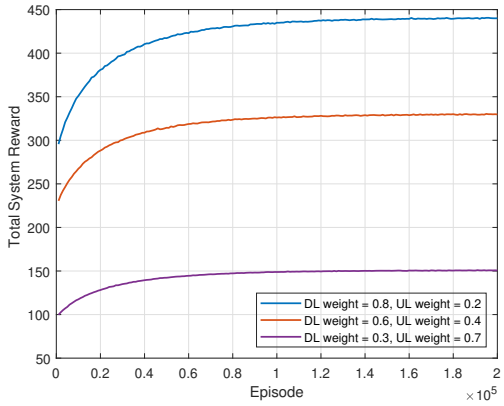


Fig. 5. Convergence performance of the proposed framework with different parameter settings of the DL and UL rates, for an MBS coverage area with 12 SBSs.

Fig. 5 shows the convergence of the proposed framework under different scenarios with different weights of DL and UL rates, i.e., $w_D = 0.8, w_U = 0.2$; $w_D = 0.6, w_U = 0.4$; and $w_D = 0.3, w_U = 0.7$. In the first scenario, the network provider desires to increase DL traffic service to the users. The second scenario involves a near balance between serving DL and UL traffic. The last scenario investigates the case when the network provider wants to earn more profit from serving UL traffic. In all three scenarios, when the number of episodes increases, we observe that the total system reward increases until it reaches a relatively stable value. The system reward converges to around 440 in the case of the highest DL weight, while it converges to around 150 in the case of the lowest DL weight. In the case of balancing DL and UL weights, the reward converges to around 330. The results show that our proposed framework can not only converge with

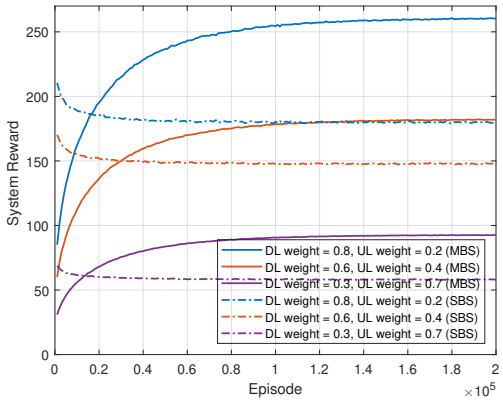


Fig. 6. Convergence performance of the deep double Q-learning algorithms of the MBS and SBSs with different parameter settings of the DL and UL rates, for an MBS coverage area with 12 SBSs.

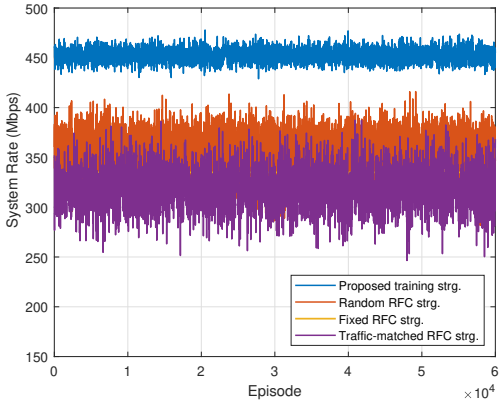


Fig. 7. Total achievable system rate of different schemes for an MBS coverage area with 12 SBSs. ($w_D = 0.8, w_U = 0.2, \epsilon = 0, T = 60000$ frames)

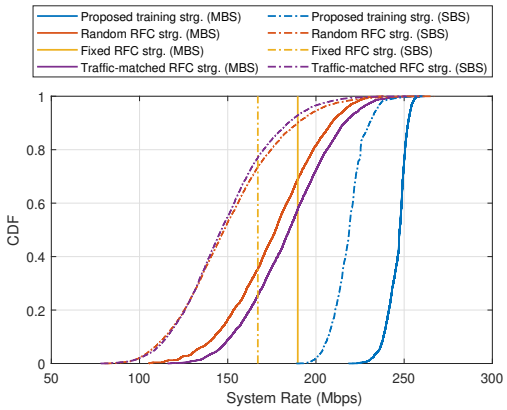


Fig. 8. CDFs of the MBS system rate and SBS average system rate for an MBS coverage area with 12 SBSs. ($w_D = 0.8, w_U = 0.2, \epsilon = 0, T = 60000$ frames)

different weights of DL and UL rates but also support the network provider to analyze a profiting strategy as relatively serving biased traffic requests. In particular, the total system reward reaches the highest in the first scenario; hence, we use the first settings $w_D = 0.8, w_U = 0.2$ for later performance comparisons.

Fig. 6 presents the convergence performance of the deep double Q-learning algorithms of the MBS and SBSs. Here, similar to the convergence analysis of the whole framework, we considered different scenarios with different weights of DL and UL rates, $w_D = 0.8, w_U = 0.2$; $w_D = 0.6, w_U = 0.4$; and $w_D = 0.3, w_U = 0.7$. Fig. 6 shows that the system reward of the MBS is meager at the beginning of the training process. When the number of episodes increases, the reward increases and converges to a relatively stable value. The figure also shows a similar convergence performance of the SBS deep double Q-learning algorithm. However, unlike the system reward of the MBS, the SBS average system reward is higher at the beginning of the training process. It gradually reduces as the number of episodes increases. However, the system rate of the SBSs in the proposed framework is still better than in the existing duplexing schemes, as discussed later.

Fig. 7 shows the efficiency of our proposed framework compared to the existing methods such as the fixed TDD RFC strategy, random TDD RFC strategy, and traffic-matched TDD RFC strategy. The figure shows that the overall system rate in our proposed scheme is higher than in the existing schemes. Specifically, our proposed framework achieves a system rate of around 450, while the random and fixed TDD RFC schemes achieve a system rate of around 350. The system rate is lowest at around 330 in the traffic-matched TDD RFC scheme.

Fig. 8 illustrates the efficiencies of the training algorithms of the MBS and SBSs over the existing strategies. During the training processes, we set a decay value for the exploration probability to improve the convergence rate. Then, for checking the efficiency of the training processes, we selected the final exploration probability of zero. The results were extracted from last 6000 time frames. Fig. 8 shows that the MBS system rate after training with the proposed MBS deep double Q-learning algorithm is more stable than that obtained by the existing strategies. In addition, the MBS system rate obtained using our proposed MBS training algorithm is mostly higher than that obtained using the random TDD RFCs, fixed TDD RFCs, and the TDD RFCs matching the traffic demand. Similarly, Fig. 8 shows that the average system rate of the SBSs obtained using our proposed SBS deep double Q-learning algorithm is also more stable and significantly improved as compared with those obtained using the existing strategies.

In Fig. 9, the mean of the boxplots shows that the overall intercell interference is reduced more in our proposed framework than when using the existing schemes. The intercell interference is proportional to the network topology scale. In particular, the interference is relatively small in the network topologies with 3 and 6 SBSs, while it dramatically increases in topologies with 9 and 12 SBSs. Moreover, the ranging interval of intercell interference in our proposed framework is much smaller with least outliers than those in the random TDD RFC, and traffic-matched TDD RFC schemes. In addition, the interval is comparable to the fixed TDD RFC scheme. In other words, our proposed framework provides a stable interference

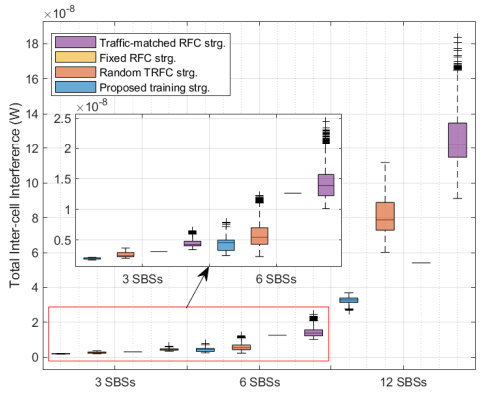


Fig. 9. Comparison of the aggregated intercell interference-reduction performances. ($w_D = 0.8, w_U = 0.2, \epsilon = 0, T = 60000$ frames)

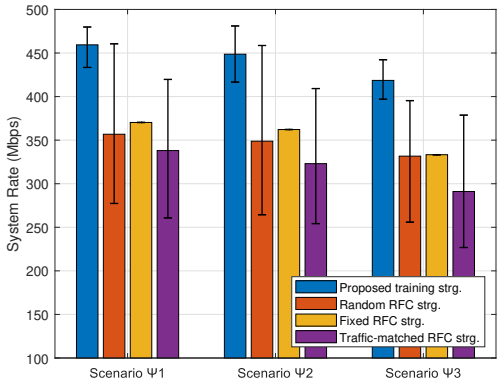


Fig. 10. Overall system-performance comparison between our proposed framework and the existing schemes, considering different transition probabilities of the DL/UL traffic demand ratio. ($w_D = 0.8, w_U = 0.2, \epsilon = 0, T = 60000$ frames)

control while the environment is dynamic.

Fig. 10 shows the comparison between the system performances of our proposed framework and the existing schemes, considering the training processes under different scenarios, such as the traffic demand mostly maintaining the DL/UL TDR, changing a small amount to a near DL/UL TDR, and tending to change a large amount to a distinct DL/UL TDR. The result is computed as the average between 10 simulations. The figure shows that the total system rate in our proposed scheme is higher than in the random, traffic-matched, and fixed TDD RFC schemes. The variance of the total system rate in our proposed scheme is smaller than those in the random and traffic-matched TDD RFC schemes. This proves the stability achievement of our proposed framework.

VII. Conclusions

In this study, we developed the optimal duplexing framework, which allows the network to adapt radio frames to various traffic demands while mitigating intercell interference between the BSs. To deal with the dynamic nature and uncertainty of the DL and UL traffic asymmetry, we utilized the deep Q-learning algorithm at each BS for learning its own optimal TDD radio frame configuration. We further accelerated the learning processes by implementing a single-leader multi-follower Stackelberg game, wherein the leader is the MBS and the followers are the SBSs. Extensive simulations showed that the proposed framework significantly improves the system utility by approximately 30% in terms of the long-term average system rate and interference reduction, compared to that obtained by the existing algorithms. Future work will involve adopting the proposed framework into 5G TDD systems, which utilize the 5G new radio Slot Format [38].

References

[1] "Cisco visual networking index: Global mobile data traffic forecast update, 2017–2022," White Paper, Cisco, Feb. 2019. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-738429.html

[2] "Small cell backhaul requirements," White Paper, NGMN Alliance, Jun. 2012. [Online]. Available: https://www.ngmn.org/publications/ngmn-whitepaper-small-cell-backhaul-requirements.html

[3] M. Laner, P. Svoboda, S. Schwarz, and M. Rupp, "Users in cells: A data traffic analysis," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Paris, France, 2012, pp. 3063–3068.

[4] "3GPP TR 36.828: Further enhancements to LTE time division duplex (TDD) for downlink-uplink (DL-UL) interference management and traffic adaptation," 3GPP Standard, 3GPP, 2012.

[5] D. Fudenberg and J. Tirole, *Game theory*. Cambridge, Massachusetts, 1991, vol. 393, no. 12.

[6] A. Khoryaev, A. Chervyakov, M. Shilov, S. Panteleev, and A. Lomayev, "Performance analysis of dynamic adjustment of TDD uplink-downlink configurations in outdoor picocell LTE networks," in *Proc. 4th Int. Congr. Ultra Modern Telecommun. Control Syst. Workshops*, St. Petersburg, Russia, 2012, pp. 914–921.

[7] B. Yu, S. Mukherjee, H. Ishii, and L. Yang, "Dynamic TDD support in the LTE-B enhanced local area architecture," in *Proc. IEEE GLOBECOM Workshops*, Anaheim, CA, USA, 2012, pp. 585–591.

[8] A. A. Dowhuszko, O. Tirkkonen, J. Karjalainen, T. Henttonen, and J. Pirskanen, "A decentralized cooperative uplink/downlink adaptation scheme for TDD small cell networks," in *Proc. IEEE 24th Int. Symp. Pers. Indoor Mobile Radio Commun. (PIMRC)*, London, UK, 2013, pp. 1682–1687.

[9] A. Kliks and P. Kryszkiewicz, "Multichannel simultaneous uplink and downlink transmission scheme for flexible duplexing," *EURASIP J. Wireless Commun. Netw.*, vol. 2017, no. 1, p. 111, 2017.

[10] M. S. Elbamby, M. Bennis, W. Saad, M. Debbah, and M. Latva-Aho, "Resource optimization and power allocation in in-band full duplex-enabled non-orthogonal multiple access networks," *IEEE J. on Sel. Areas in Commun.*, vol. 35, no. 12, pp. 2860–2873, 2017.

[11] Q. Liao, "Dynamic uplink/downlink resource management in flexible duplex-enabled wireless networks," in *Proc. ICC Workshops*, Paris, France, 2017, pp. 625–631.

[12] W. Noh, W. Shin, and H.-H. Choi, "Performance analysis of user pairing algorithm in full-duplex cellular networks," *Mobile Inf. Syst.*, vol. 2017, p. 8182150, 2017.

[13] Y. Lin, Y. Gao, Y. Li, X. Zhang, and D. Yang, "Qos aware dynamic uplink-downlink reconfiguration algorithm in TD-LTE hetnet," in *Proc. IEEE GLOBECOM Workshops*, Atlanta, GA, USA, 2013, pp. 708–713.

[14] B. Yu, L. Yang, H. Ishii, and S. Mukherjee, "Dynamic TDD support in macrocell-assisted small cell architecture," *IEEE J. on Sel. Areas in Commun.*, vol. 33, no. 6, pp. 1201–1213, 2015.

[15] J. Bai, S.-p. Yeh, F. Xue, Y.-s. Choi, P. Wang, and S. Talwar, "Full-duplex in 5G small cell access: System design and performance aspects," *arXiv preprint arXiv:1903.09893*, 2019.

[16] A. A. Esswie and K. I. Pedersen, "Inter-cell radio frame coordination scheme based on sliding codebook for 5G TDD systems," in *Proc. IEEE VTC-spring*, Kuala Lumpur, Malaysia, 2019, pp. 1–6.

[17] A. A. Esswie, K. I. Pedersen, and P. E. Mogensen, "Quasi-dynamic frame coordination for ultra-reliability and low-latency in 5G TDD systems," *arXiv preprint arXiv:1903.09363*, 2019.

[18] M. Bennis, S. M. Perlaza, P. Blasco, Z. Han, and H. V. Poor, "Self-organization in small cell networks: A reinforcement learning approach," *IEEE Trans. Wireless Commun.*, vol. 12, no. 7, pp. 3202–3212, 2013.

[19] M. S. ElBamby, M. Bennis, W. Saad, and M. Latva-Aho, "Dynamic uplink-downlink optimization in TDD-based small cell networks," in *Proc. 11th ISWCS*, Barcelona, Spain, 2014, pp. 939–944.

[20] Z. Yu, K. Wang, H. Ji, X. Li, and H. Zhang, "Dynamic frame configuration in TD-LTE heterogeneous cellular networks," in *Proc. IEEE GLOBECOM Workshops*, San Diego, CA, USA, 2015, pp. 1–6.

[21] Y. Wang and M. Tao, "Dynamic uplink/downlink configuration using q-learning in femtocell networks," in *Proc. IEEE/CIC ICCC*, Shanghai, China, 2014, pp. 53–58.

[22] P. Sarigiannidis, A. Sarigiannidis, I. Moscholios, and P. Zwierzykowski, "DIANA: A machine learning mechanism for adjusting the TDD uplink-downlink configuration in XG-PON-LTE systems," *Mobile Inf. Syst.*, vol. 2017, p. 8198017, 2017.

[23] C.-H. Tsai, K.-H. Lin, H.-Y. Wei, and F.-M. Yeh, "QoE-aware Q-learning based approach to dynamic TDD uplink-downlink reconfiguration in indoor small cell networks," *Wireless Networks*, vol. 25, no. 6, pp. 3467–3479, 2019.

[24] L. Kleinrock, *Queueing systems. Volume I: theory*. Wiley, New York, 1975.

[25] L. Liang, J. Kim, S. C. Jha, K. Sivanesan, and G. Y. Li, "Spectrum and power allocation for vehicular communications with delayed CSI feedback," *IEEE Wireless Commun. Letters*, vol. 6, no. 4, pp. 458–461, 2017.

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

[27] H. Y. Ong, K. Chavez, and A. Hong, "Distributed deep q-learning," *arXiv preprint arXiv:1508.04186*, 2015.

[28] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT Press, Cambridge, MA, 2018.

[29] M. Deisenroth and C. E. Rasmussen, "PILCO: A model-based and data-efficient approach to policy search," in *Proc. 28th Int. Conf. Mach. Learn.*, Bellevue, WA, USA, 2011, pp. 465–472.

[30] S. Gu, T. Lillicrap, I. Sutskever, and S. Levine, "Continuous deep Q-learning with model-based acceleration," in *Proc. 33rd Int. Conf. Mach. Learn.*, New York, NY, USA, 2016, pp. 2829–2838.

[31] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[32] M. Hausknecht and P. Stone, "Deep reinforcement learning in parameterized action space," *arXiv preprint arXiv:1511.04143*, 2015.

[33] C. J. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, no. 3-4, pp. 279–292, 1992.

[34] 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, pp. 529–533, 2015.

[35] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proc. 30th AAAI Conf. Artificial Intell.*, Phoenix, Arizona, USA, 2016, pp. 2094–2100.

[36] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[37] Matlab R2019a. [Online]. Available: http://www.mathworks.com/

[38] "3GPP TS 38.300: NR and NG-RAN overall description," 3GPP Standard, 3GPP, 2017.