

# Secrecy Preserving in Stochastic Resource Orchestration for Multi-Tenancy Network Slicing

Xianfu Chen\*, Zhifeng Zhao<sup>†</sup>, Celimuge Wu<sup>‡</sup>, Tao Chen\*, Honggang Zhang<sup>†</sup>, and Mehdi Bennis<sup>§</sup>

\*VTT Technical Research Centre of Finland Ltd, Finland

<sup>†</sup>College of Information Science and Electronic Engineering, Zhejiang University, China

<sup>‡</sup>Graduate School of Informatics and Engineering, University of Electro-Communications, Tokyo, Japan

<sup>§</sup>Centre for Wireless Communications, University of Oulu, Finland

**Abstract**—Network slicing is a proposing technology to support diverse services from mobile users (MUs) over a common physical network infrastructure. In this paper, we consider radio access network (RAN)-only slicing, where the physical RAN is tailored to accommodate both computation and communication functionalities. Multiple service providers (SPs, i.e., multiple tenants) compete with each other to bid for a limited number of channels across the scheduling slots, aiming to provide their subscribed MUs the opportunities to access the RAN slices. An eavesdropper overhears data transmissions from the MUs. We model the interactions among the non-cooperative SPs as a stochastic game, in which the objective of a SP is to optimize its own expected long-term payoff performance. To approximate the Nash equilibrium solutions, we first construct an abstract stochastic game using the channel auction outcomes. Then we linearly decompose the per-SP Markov decision process to simplify the decision-makings and derive a deep reinforcement learning based scheme to approach the optimal abstract control policies. TensorFlow-based experiments verify that the proposed scheme outperforms the three baselines and yields the best performance in average utility per MU per scheduling slot.

## I. INTRODUCTION

To keep up with the proliferation of wireless services, new cell sites are being constantly built, eventually leading to dense network deployments [1]. However, it becomes extremely complex to operate the control plane functions in a dense radio access network (RAN). In recent years, the computation-intensive applications (e.g., augmented reality and interactive online gaming) are gaining increasing popularity [2]. The mobile user (MU)-end terminal devices are in general constrained by battery capacity and processing speed of the central processing unit (CPU). The tension between computation-intensive applications and resource-constrained terminal devices calls for a revolution in computing [3]. Mobile-edge computing (MEC) is envisioned as a promising solution, which brings the computing capabilities within the RANs in close proximity to MUs [2]. Offloading a computation task to a MEC server for execution involves data transmissions. How to orchestrate radio resources between MEC and traditional mobile services adds another dimension of complexity to the network operations [4]. By abstracting all physical base stations (BSs) in a geographical area as a logical big BS, the software-defined networking (SDN) concept provides infrastructure flexibility as well as service-oriented customization [5]. In a software-defined RAN, the SDN-orchestrator handles all control plane operations.

One key benefit from a software-defined RAN is to facilitate network sharing [6]. As such, the same physical network is

able to host multiple service providers (SPs, namely, multiple tenants [7]), which breaks the traditional business model regarding the single ownership of a network infrastructure [8]. For example, an over-the-top application provider (e.g., Google [9]) can be a SP so as to lease radio resources from the infrastructure provider to improve the Quality-of-Service and the Quality-of-Experience for its subscribers. Building upon the 3GPP TSG SA 5 network sharing paradigm [10], a software-defined RAN architecture and its integration with network function virtualization enable RAN-only slicing that splits the RAN into multiple virtual slices [11]. This paper is primarily concerned with a software-defined RAN where the RAN slices are specifically tailored to accommodate both computation and communication functionalities [12].

The technical challenges yet remain for the implementation of RAN-only slicing. Particularly, the mechanisms that exploit the decoupling of control and data planes in a software-defined RAN must be developed to optimize radio resource utilization. For the considered software-defined RAN, a limited number of channels are auctioned over the time horizon to the MUs, which request MEC and traditional mobile services. An eavesdropper exists in the network and overhears the MUs during the data transmissions [13]. Multiple SPs compete to orchestrate the channels for their subscribed MUs according to the network dynamics, aiming to maximize the expected long-term payoff performance. Upon receiving the auction bids from all SPs, the SDN-orchestrator allocates channels to the MUs through a Vickrey-Clarke-Groves (VCG) mechanism<sup>1</sup> [14]. To combat the threat from the eavesdropper, each MU then proceeds to use a secrecy-rate [13] to offload computation tasks and schedule packets over the assigned channel. To the best of our knowledge, there does not exist a comprehensive study on stochastic resource orchestration in multi-tenancy RAN-only slicing with secrecy preserving.

## II. SYSTEM MODEL

As shown in Fig. 1, we focus on a system model with RAN-only slicing, where an eavesdropper intentionally overhears the data transmissions of the MUs. The time horizon is divided into discrete scheduling slots, each of which is indexed by an integer  $k \in \mathbb{N}_+$  and is assumed to be of equal duration  $\delta$  (in seconds). The RAN consists of a set  $\mathcal{B}$  of physical BSs covering a service area, which can be represented by a set  $\mathcal{L}$

<sup>1</sup>The VCG mechanism ensures truthfulness, efficiency and incentive compatibility.

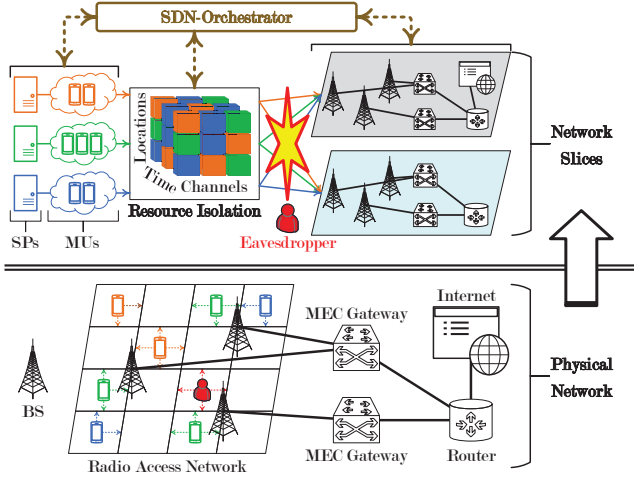


Fig. 1. Illustration of the RAN-only slicing architecture. An eavesdropper overhears the data transmissions from the MUs across the time horizon.

of small locations with each being characterized by uniform signal propagation conditions [15]. We use  $\mathcal{L}_b$  to denote the serving area of a BS  $b \in \mathcal{B}$ . For any two BSs  $b$  and  $b' \in \mathcal{B}$  ( $b' \neq b$ ), we assume that  $\mathcal{L}_b \cap \mathcal{L}_{b'} = \emptyset$ . We denote the geographical distribution of BSs by a topological graph  $\mathcal{T}\mathcal{G} = \langle \mathcal{B}, \mathcal{E} \rangle$ , where  $\mathcal{E} = \{e_{b,b'} : b \neq b', b, b' \in \mathcal{B}\}$  with  $e_{b,b'} = 1$  if BSs  $b$  and  $b'$  are neighbours and otherwise  $e_{b,b'} = 0$ . Suppose that  $I$  SPs provide both MEC and traditional mobile services to MUs while each MU can subscribe to only one SP. Let  $\mathcal{N}_i$  be the set of MUs of a SP  $i \in \mathcal{I} = \{1, \dots, I\}$ .

Across the scheduling slots, the MUs and the eavesdropper move within  $\mathcal{L}$  following a Markov mobility model [16]. Denote by  $\mathcal{N}_{b,i}^k$  the set of MUs of SP  $i \in \mathcal{I}$  moving into the area of a BS  $b \in \mathcal{B}$  during a slot  $k$ . We assume that a MU at a location can only be associated with the BS that covers the location. In the network, all MUs share a set  $\mathcal{J} = \{1, \dots, J\}$  of orthogonal channels with the same bandwidth  $\eta$  (in Hz). The SPs compete for the limited channel access opportunities for their MUs. Specifically, at the beginning of a scheduling slot  $k$ , each SP  $i$  submits an auction bid  $\beta_i^k = (\nu_i^k, \mathbf{C}_i^k)$ , where  $\nu_i^k$  is the valuation over  $\mathbf{C}_i^k = (C_{b,i}^k : b \in \mathcal{B})$  with  $C_{b,i}^k$  being the number of requested channels in the service area of a BS  $b$ . After receiving  $\beta^k = (\beta_i^k : i \in \mathcal{I})$ , the SDN-orchestrator performs channel allocation and calculates payment  $\tau_i^k$  for each SP  $i$ . Let  $\rho_n^k = (\rho_{n,j}^k : j \in \mathcal{J})$  be the channel allocation of a MU  $n \in \mathcal{N} = \cup_{i \in \mathcal{I}} \mathcal{N}_i$ , where  $\rho_{n,j}^k = 1$  if channel  $j$  is allocated to MU  $n \in \mathcal{N}$  during slot  $k$  and  $\rho_{n,j}^k = 0$ , otherwise. We also apply the following constraints for centralized channel allocation at the SDN-orchestrator during a slot,

$$\left( \sum_{i \in \mathcal{I}} \sum_{n \in \mathcal{N}_{b,i}^k} \rho_{n,j}^k \right) \cdot \left( \sum_{i \in \mathcal{I}} \sum_{n \in \mathcal{N}_{b',i}^k} \rho_{n,j}^k \right) = 0, \quad \text{if } e_{b,b'} = 1, \forall e_{b,b'} \in \mathcal{E}, \forall j \in \mathcal{J}; \quad (1)$$

$$\sum_{i \in \mathcal{I}} \sum_{n \in \mathcal{N}_{b,i}^k} \rho_{n,j}^k \leq 1, \forall b \in \mathcal{B}, \forall j \in \mathcal{J}; \quad (2)$$

$$\sum_{j \in \mathcal{J}} \rho_{n,j}^k \leq 1, \forall b \in \mathcal{B}, \forall i \in \mathcal{I}, \forall n \in \mathcal{N}_{b,i}, \quad (3)$$

which ensure that one channel cannot be allocated to MUs associated with two adjacent BSs in order to avoid interference during data transmissions, while in the service area of a BS, one MU can be assigned at most one channel and one channel can be assigned to at most one MU. Denote  $\phi^k = (\phi_i^k : i \in \mathcal{I})$  as the winner vector at the beginning of a scheduling slot  $k$ , where  $\phi_i^k = 1$  if SP  $i$  wins the channel auction and  $\phi_i^k = 0$  indicates that no channel will be allocated to the MUs of SP  $i$  during the slot. The SDN-orchestrator determines  $\phi^k$  via the VCG pricing mechanism, namely,

$$\begin{aligned} \phi^k &= \arg \max_{\phi} \sum_{i \in \mathcal{I}} \phi_i \cdot \nu_i^k \\ \text{s.t.} \quad & \text{constraints (1), (2) and (3);} \\ & \sum_{n \in \mathcal{N}_{b,i}^k} \varphi_n^k = \phi_i \cdot C_{b,i}^k, \forall b \in \mathcal{B}, \forall i \in \mathcal{I}, \end{aligned} \quad (4)$$

where  $\varphi_n^k = \sum_{j \in \mathcal{J}} \rho_{n,j}^k$  and  $\phi = (\phi_i : i \in \mathcal{I})$  with  $\phi_i \in \{0, 1\}$ . The payment  $\tau_i^k$  for each SP  $i$  can be calculated to be  $\tau_i^k = \max_{\phi_{-i}} \sum_{i' \in \mathcal{I} \setminus \{i\}} \phi_{i'} \cdot \nu_{i'}^k - \max_{\phi} \sum_{i' \in \mathcal{I} \setminus \{i\}} \phi_{i'} \cdot \nu_{i'}^k$ , where  $-i$  denotes all the competitors of SP  $i$ .

Let  $L_{n,(u)}^k$  and  $L_{n,(e)}^k \in \mathcal{L}$  be the geographical locations of a MU  $n \in \mathcal{N}$  and the eavesdropper during a scheduling slot  $k$ , respectively. As in [15], we assume that the average channel gains  $H_{n,(u)}^k = h_{(u)}(L_{n,(u)}^k)$  and  $H_{n,(e)}^k = h_{(e)}(L_{n,(u)}^k, L_{n,(e)}^k)$  of links between MU  $n$  and the associated BS as well as the eavesdropper are determined by the respective distances. At the beginning of each scheduling slot  $k$ , MU  $n$  independently generates a random number  $A_{n,(t)}^k \in \mathcal{A} = \{0, 1, \dots, A_{(t)}^{\max}\}$  of computation tasks<sup>2</sup> according to a Markov process [17]. We represent a computation task by  $(\mu_{(t)}, \vartheta)$ , where  $\mu_{(t)}$  and  $\vartheta$  are, respectively, the input data size (in bits) and the number of CPU cycles required to accomplish one input bit of the computation task. For a computation task, two decisions are available: 1) to be processed locally at the MU; or 2) to be offloaded to the MEC server in the computation slice for execution. The computation offloading decision for MU  $n$  at a slot  $k$  specifies the number  $R_{n,(t)}^k$  of tasks to be transmitted to the MEC server. Then the remaining  $A_{n,(t)}^k - \varphi_n^k \cdot R_{n,(t)}^k$  tasks are to be processed locally. Meanwhile, a data queue at a MU buffers the packets from the traditional mobile service. Let  $W_n^k$  and  $A_{n,(p)}^k$  be the queue length and the random new packet arrivals for MU  $n$  at the beginning of a slot  $k$ . We assume that the data packets are of a constant size  $\mu_{(p)}$  (bits) and the packet arrival process is independent among the MUs and identical and independently distributed across time. Let  $R_{n,(p)}^k$  be the number of packets that are scheduled for transmission from MU  $n$  at scheduling slot  $k$ . The queue evolution of MU  $n$  can be written as the form below,

$$W_n^{k+1} = \min \left\{ W_n^k - \varphi_n^k \cdot R_{n,(p)}^k + A_{n,(p)}^k, W_n^{\max} \right\}, \quad (5)$$

where  $W_n^{\max}$  is the queue length limit.

To ensure security, the energy (in Joules) consumed by a MU  $n \in \mathcal{N}$  for transmitting  $\varphi_n^k \cdot R_{n,(t)}^k$  computation tasks and

<sup>2</sup>To ease analysis, we assume that the maximum CPU power at a mobile device matches the maximum computation task arrivals and a MU can process  $A_{(t)}^{\max}$  tasks within one scheduling slot.

$\varphi_n^k \cdot R_{n,(p)}^k$  data packets with a secrecy-rate [13] during a slot  $k$  can be calculated as

$$P_{n,(tr)}^k = \begin{cases} \frac{\delta \cdot \eta \cdot \sigma^2 \cdot \left( 2^{\frac{\varphi_n^k \cdot (\mu_{(t)} \cdot R_{n,(t)}^k + \mu_{(p)} \cdot R_{n,(p)}^k)}{\eta \cdot \delta}} - 1 \right)}{H_{n,(u)}^k - H_{(e)}^k \cdot 2^{\frac{\varphi_n^k \cdot (\mu_{(t)} \cdot R_{n,(t)}^k + \mu_{(p)} \cdot R_{n,(p)}^k)}{\eta \cdot \delta}}}, & \text{if } H_{n,(u)}^k > H_{(e)}^k; \\ 0, & \text{otherwise,} \end{cases} \quad (6)$$

where  $\sigma^2$  is the background noise power spectral density. Let  $\Omega^{(\max)}$  be the maximum transmit power for all MUs, namely,  $P_{n,(tr)}^k \leq \Omega^{(\max)} \cdot \delta$ ,  $\forall n$  and  $\forall k$ . For the remaining  $A_{n,(t)}^k - \varphi_n^k \cdot R_{n,(t)}^k$  computation tasks that are to be locally processed, the CPU energy consumption is

$$P_{n,(CPU)}^k = \varsigma \cdot \mu_{(t)} \cdot \vartheta \cdot \varrho^2 \cdot \left( A_{n,(t)}^k - \varphi_n^k \cdot R_{n,(t)}^k \right), \quad (7)$$

where  $\varsigma$  is the effective switched capacitance [18] and  $\varrho$  is the CPU-cycle frequency of the MU-end devices.

### III. STOCHASTIC GAME FORMULATION

At a scheduling slot  $k$ , the local state of a MU  $n \in \mathcal{N}$  is described as  $\chi_n^k = (L_{n,(u)}^k, L_{(e)}^k, A_{n,(t)}^k, W_n^k) \in \mathcal{X} = \mathcal{L}^2 \times \mathcal{A} \times \mathcal{W}$ , where the SDN-orchestrator broadcasts the information of  $L_{(e)}^k$  to all MUs. Then  $\chi^k = (\chi_n^k : n \in \mathcal{N}) \in \mathcal{X}^{|\mathcal{N}|}$  characterizes the global network state, where  $|\mathcal{N}|$  means the cardinality of the set  $\mathcal{N}$ . Define by  $\pi_i = (\pi_{i,(c)}, \pi_{i,(t)}, \pi_{i,(p)})$  a control policy of a SP  $i \in \mathcal{I}$ , where  $\pi_{i,(c)} = (\pi_{n,(t)} : n \in \mathcal{N}_i)$  and  $\pi_{i,(p)} = (\pi_{n,(p)} : n \in \mathcal{N}_i)$  are the channel auction, the computation offloading and the packet scheduling policies, respectively. The joint control policy of all SPs is given by  $\pi = (\pi_i : i \in \mathcal{I})$ . With the observation of  $\chi^k$  at the beginning of each scheduling slot  $k$ , SP  $i$  announces the auction bid  $\beta_i^k$  to the SDN-orchestrator and decides the  $\mathbf{R}_{i,(t)}^k$  computation tasks as well as  $\mathbf{R}_{i,(p)}^k$  packets to be transmitted following  $\pi_i$ . That is,  $\pi_i(\chi^k) = (\pi_{i,(c)}(\chi^k), \pi_{i,(t)}(\chi^k), \pi_{i,(p)}(\chi^k)) = (\beta_i^k, \mathbf{R}_{i,(t)}^k, \mathbf{R}_{i,(p)}^k)$ , where  $\mathbf{R}_{i,(t)}^k = (R_{n,(t)}^k : n \in \mathcal{N}_i)$  and  $\mathbf{R}_{i,(p)}^k = (R_{n,(p)}^k : n \in \mathcal{N}_i)$ . Accordingly, SP  $i$  realizes an instantaneous payoff

$$F_i(\chi^k, \varphi_i^k, \mathbf{R}_{i,(t)}^k, \mathbf{R}_{i,(p)}^k) = \sum_{n \in \mathcal{N}_i} \alpha_n \cdot U_n(\chi_n^k, \varphi_n^k, R_{n,(t)}^k, R_{n,(p)}^k) - \tau_i^k, \quad (8)$$

where  $\varphi_i^k = (\varphi_n^k : n \in \mathcal{N}_i)$  and  $\alpha_n \in \mathbb{R}_+$  is the unit price to charge a MU  $n$  for achieving utility

$$U_n(\chi_n^k, \varphi_n^k, R_{n,(t)}^k, R_{n,(p)}^k) = U_n^{(1)}(W_n^{k+1}) + U_n^{(2)}(D_n^k) + \ell_n \cdot \left( U_n^{(3)}(P_{n,(CPU)}^k) + U_n^{(4)}(P_{n,(tr)}^k) \right). \quad (9)$$

In (9),  $D_n^k = \max\{W_n^k - \varphi_n^k \cdot R_{n,(p)}^k + A_{n,(p)}^k - W_n^{(\max)}, 0\}$  defines the number of packet drops,  $U_n^{(1)}(\cdot)$ ,  $U_n^{(2)}(\cdot)$ ,  $U_n^{(3)}(\cdot)$  and  $U_n^{(4)}(\cdot)$  are the positive and monotonically decreasing functions, and  $\ell_n \in \mathbb{R}_+$  is a weighting factor. Obviously, the randomness lying in  $\{\chi^k : k \in \mathbb{N}_+\}$  is Markovian.

Taking expectation with respect to the sequence of per-slot instantaneous payoffs, the expected long-term payoff of a SP

$i \in \mathcal{I}$  for a given initial global network state  $\chi^1 = \chi \triangleq (\chi_n = (L_{n,(u)}, L_{(e)}, A_{n,(t)}, W_n) : n \in \mathcal{N})$  can be expressed as in (10), where  $\gamma \in [0, 1]$  is a discount factor.  $V_i(\chi, \pi)$  is also termed as the state-value function of SP  $i$ . The aim of each SP  $i$  is to devise a best-response control policy  $\pi_i^*$  such that  $\pi_i^* = \arg \max_{\pi_i} V_i(\chi, \pi_i, \pi_{-i})$ ,  $\forall \chi \in \mathcal{X}^{|\mathcal{N}|}$ . Due to the limited number of channels and the stochastic nature in networking environment, we formulate the interactions among multiple non-cooperative SPs over the scheduling slots as a stochastic game,  $\mathcal{SG}$ , in which  $I$  SPs are the players and there are a set  $\mathcal{X}^{|\mathcal{N}|}$  of global network states and a collection of control policies  $\{\pi_i : \forall i \in \mathcal{I}\}$ . A Nash equilibrium (NE), which is a tuple of control policies  $\langle \pi_i^* : i \in \mathcal{I} \rangle$ , describes the rational behaviours of the SPs in a  $\mathcal{SG}$ . For the  $I$ -player  $\mathcal{SG}$  with expected infinite-horizon discounted payoffs, there always exists a NE in stationary control policies [19]. Define  $\mathbb{V}_i(\chi) = V_i(\chi, \pi_i^*, \pi_{-i}^*)$  as the optimal state-value function,  $\forall i \in \mathcal{I}$  and  $\forall \chi \in \mathcal{X}^{|\mathcal{N}|}$ .

### IV. ABSTRACT STOCHASTIC GAME REFORMULATION AND DEEP REINFORCEMENT LEARNING

From (10), it can be easily observed that the expected long-term payoff of a SP  $i \in \mathcal{I}$  depends on information of not only the global network state across the scheduling slots but also the joint control policy  $\pi$ . In other words, the decision makings from the non-cooperative SPs are coupled in the  $\mathcal{SG}$ , which makes it a challenging task to find the NE. In this section, we elaborate on how the SPs play the  $\mathcal{SG}$  only with limited local information.

#### A. Stochastic Game Abstraction

To capture the coupling of decision makings among the SPs, we abstract  $\mathcal{SG}$  as  $\mathcal{AG}$  [20], in which a SP  $i \in \mathcal{I}$  behaves based on its own local network dynamics and abstractions of states at other competing SPs. Let  $\mathcal{S}_i = \{1, \dots, S_i\}$  be an abstraction of the state space  $\mathcal{X}_{-i}$ , where  $S_i \in \mathbb{N}_+$  and  $S_i \ll |\mathcal{X}_{-i}|$ . We observe that the behavioural couplings in  $\mathcal{SG}$  exist in the channel auction and the payments of SP  $i$  depend on  $\mathcal{X}_{-i}$ . This allows SP  $i$  to construct  $\mathcal{S}_i$  by classifying the value region  $[0, \Gamma_i]$  of payments into  $S_i$  intervals, i.e.,  $[0, \Gamma_{i,1}]$ ,  $(\Gamma_{i,1}, \Gamma_{i,2}]$ ,  $(\Gamma_{i,2}, \Gamma_{i,3}]$ ,  $\dots$ ,  $(\Gamma_{i,S_i-1}, \Gamma_{i,S_i}]$ , where  $\Gamma_{i,S_i} = \Gamma_i$  is the maximum payment and we let  $\Gamma_{i,1} = 0$  for a special case in which SP  $i$  wins the channel auction with no payment<sup>3</sup>. With this regard, SP  $i$  abstracts  $(\chi_i, \chi_{-i}) \in \mathcal{X}^{|\mathcal{N}|}$  as  $\tilde{\chi}_i = (\chi_i, s_i) \in \tilde{\mathcal{X}}_i = \mathcal{X}_i \times \mathcal{S}_i$  if the payment in previous scheduling slot belongs to  $(\Gamma_{i,s_i-1}, \Gamma_{i,s_i}]$ .

Let  $\tilde{\pi}_i = (\tilde{\pi}_{i,(c)}, \pi_{i,(t)}, \pi_{i,(p)})$  be the abstract control policy in the  $\mathcal{AG}$  played by a SP  $i \in \mathcal{I}$  over  $\tilde{\mathcal{X}}_i$ , where  $\tilde{\pi}_{i,(c)}$  is the abstract channel auction policy. Likewise, the abstract state-value function for SP  $i$  under  $\tilde{\pi} = (\tilde{\pi}_i : i \in \mathcal{I})$  can then be defined as in (11),  $\forall \tilde{\chi}_i \in \tilde{\mathcal{X}}_i$ , where  $\tilde{\chi}^k = (\tilde{\chi}_i^k = (\chi_i^k, s_i^k) : i \in \mathcal{I})$  with  $s_i^k$  being the abstract state at slot  $k$  and  $F_i(\tilde{\chi}_i^k, \varphi_i(\tilde{\pi}_{i,(c)}(\tilde{\chi}_i^k)), \pi_{i,(t)}(\chi_i^k), \pi_{i,(p)}(\chi_i^k))$  is the immediate payoff with  $\tilde{\chi}^k = (\tilde{\chi}_i^k : i \in \mathcal{I})$  and  $\tilde{\pi}_{i,(c)} = (\tilde{\pi}_{i,(c)} : i \in \mathcal{I})$ .

<sup>3</sup>This case happens when there are enough channels to serve all MUs in the network [21].



$$V_i(\boldsymbol{\chi}, \boldsymbol{\pi}) = (1 - \gamma) \cdot \mathbb{E}_{\boldsymbol{\pi}} \left[ \sum_{k=1}^{\infty} (\gamma)^{k-1} \cdot F_i(\boldsymbol{\chi}^k, \varphi_i(\boldsymbol{\pi}_{(c)}(\boldsymbol{\chi}^k)), \boldsymbol{\pi}_{i,(t)}(\boldsymbol{\chi}^k), \boldsymbol{\pi}_{i,(p)}(\boldsymbol{\chi}^k)) \mid \boldsymbol{\chi}^1 = \boldsymbol{\chi} \right] \quad (10)$$

$$\tilde{V}_i(\tilde{\boldsymbol{\chi}}_i, \tilde{\boldsymbol{\pi}}) = (1 - \gamma) \cdot \mathbb{E}_{\tilde{\boldsymbol{\pi}}} \left[ \sum_{k=1}^{\infty} (\gamma)^{k-1} \cdot \tilde{F}_i(\tilde{\boldsymbol{\chi}}_i^k, \varphi_i(\tilde{\boldsymbol{\pi}}_{(c)}(\tilde{\boldsymbol{\chi}}_i^k)), \boldsymbol{\pi}_{i,(t)}(\boldsymbol{\chi}_i^k), \boldsymbol{\pi}_{i,(p)}(\boldsymbol{\chi}_i^k)) \mid \tilde{\boldsymbol{\chi}}_i^1 = \tilde{\boldsymbol{\chi}}_i \right] \quad (11)$$

In our previous work [20], we have proved that instead of playing the original  $\boldsymbol{\pi}^*$  in the  $\mathcal{SG}$ , the NE joint abstract control policy given by  $\tilde{\boldsymbol{\pi}}^* = (\tilde{\boldsymbol{\pi}}_i^* : i \in \mathcal{I})$  in the  $\mathcal{AG}$  leads to a bounded regret, where  $\tilde{\boldsymbol{\pi}}_i^* = (\tilde{\boldsymbol{\pi}}_{i,(c)}^*, \boldsymbol{\pi}_{i,(t)}^*, \boldsymbol{\pi}_{i,(p)}^*)$  denotes the best-response abstract control policy of SP  $i$ . Hereinafter, we switch our focus to the  $\mathcal{AG}$ , in which a SP solves a single-agent Markov decision process (MDP). Suppose all SPs play  $\tilde{\boldsymbol{\pi}}^*$  in the  $\mathcal{AG}$ . Denote  $\tilde{V}_i(\tilde{\boldsymbol{\chi}}_i) = \tilde{V}_i(\tilde{\boldsymbol{\chi}}_i, \tilde{\boldsymbol{\pi}}^*)$ .

### B. Decomposition of Abstract State-Value Function

There remain two challenges involved in solving the optimal abstract state-value functions for each SP  $i \in \mathcal{I}$  using dynamic programming methods [22]: 1) a priori knowledge of the abstract network state transition probability is not feasible; and 2) the size of the decision making space  $\{\tilde{\boldsymbol{\pi}}_i(\tilde{\boldsymbol{\chi}}_i) : \tilde{\boldsymbol{\chi}}_i \in \tilde{\mathcal{X}}_i\}$  grows exponentially as  $|\mathcal{N}_i|$  increases. On the other hand, the channel auction decisions and the computation offloading as well as packet scheduling decisions are made in sequence and are independent across a SP and its subscribed MUs. We are hence motivated to decompose the per-SP MDP in the  $\mathcal{AG}$  into  $|\mathcal{N}_i| + 1$  independent MDPs. More specifically, for a SP  $i \in \mathcal{I}$ ,  $\tilde{V}_i(\tilde{\boldsymbol{\chi}}_i)$ ,  $\forall \tilde{\boldsymbol{\chi}}_i \in \tilde{\mathcal{X}}_i$ , can be computed as

$$\tilde{V}_i(\tilde{\boldsymbol{\chi}}_i) = \sum_{n \in \mathcal{N}_i} \alpha_n \cdot \mathbb{U}_n(\boldsymbol{\chi}_n) - \mathbb{U}_i(s_i), \quad (12)$$

where the per-MU  $\mathbb{U}_n$  and the  $\mathbb{U}_i(s_i)$  of SP  $i$  satisfy, respectively, (13) and

$$\begin{aligned} \mathbb{U}_i(s_i) = & \\ (1 - \gamma) \cdot \tau_i + \gamma \cdot \sum_{s'_i \in \mathcal{S}_i} \mathbb{P}(s'_i | s_i, \phi_i(\tilde{\boldsymbol{\pi}}_{(c)}^*(\tilde{\boldsymbol{\chi}}))) \cdot \mathbb{U}_i(s'_i). & \end{aligned} \quad (14)$$

In the above,  $\tilde{\boldsymbol{\pi}}_{(c)}^*(\tilde{\boldsymbol{\chi}}) = (\tilde{\boldsymbol{\pi}}_{i,(c)}^*(\tilde{\boldsymbol{\chi}}) : i \in \mathcal{I})$ , while  $R_{n,(t)}$  and  $R_{n,(p)}$  are the computation offloading and packet scheduling decisions under  $\boldsymbol{\chi}_n$  of MU  $n \in \mathcal{N}_i$ .

We can now specify the number of needed channels by a SP  $i \in \mathcal{I}$  in the area of a BS  $b \in \mathcal{B}$  as  $C_{b,i} = \sum_{\{n \in \mathcal{N}_i : L_n \in \mathcal{L}_b\}} z_n$  and the valuation of obtaining  $\mathbf{C}_i = (C_{b,i} : b \in \mathcal{B})$  across the whole service area as

$$\begin{aligned} \nu_i = & \frac{1}{1 - \gamma} \cdot \sum_{n \in \mathcal{N}_i} \alpha_n \cdot \mathbb{U}_n(\boldsymbol{\chi}_n) \\ & - \frac{\gamma}{1 - \gamma} \cdot \sum_{s'_i \in \mathcal{S}_i} \mathbb{P}(s'_i | s_i, \mathbb{1}_{\{\sum_{b \in \mathcal{B}} C_{b,i} > 0\}}) \cdot \mathbb{U}_i(s'_i), \end{aligned} \quad (15)$$

which together constitute a bid  $\tilde{\boldsymbol{\pi}}_{i,(c)}^*(\tilde{\boldsymbol{\chi}}_i) = \boldsymbol{\beta}_i \triangleq (\nu_i, \mathbf{C}_i)$  of SP  $i$  in  $\tilde{\boldsymbol{\chi}}_i \in \tilde{\mathcal{X}}_i$ , where  $z_n$  is given by (16) and  $\mathbb{1}_{\{\Xi\}}$  equals 1 if the condition  $\Xi$  is satisfied and 0, otherwise.

### C. Learning Optimal Abstract Control Policy

We can easily find that at a current scheduling slot,  $\boldsymbol{\beta}_i$  of a SP  $i \in \mathcal{I}$  needs  $(s_i, \mathbb{P}(s' | s, \iota - 1))$  and  $(\mathbb{U}_n(\boldsymbol{\chi}_n), z_n, L_n)$  from each subscribed MU  $n \in \mathcal{N}_i$ , where  $s' \in \mathcal{S}_i$  and  $\iota \in \{1, 2\}$ . We

propose that SP  $i$  maintains over the slots a three-dimensional table  $\mathbf{Y}_i^k$  of size  $S_i \cdot S_i \cdot 2$ . Each entry  $y_{s,s',\iota}^k$  in  $\mathbf{Y}_i^k$  represents the number of transitions from  $s_i^{k-1} = s$  to  $s_i^k = s'$  when  $\phi_i^{k-1} = \iota - 1$  up to slot  $k$ .  $\mathbf{Y}_i^k$  is updated using the channel auction outcomes. Then, we estimate the abstract network state transition probability at a slot  $k$  as

$$\mathbb{P}(s_i^k = s' | s_i^{k-1} = s, \phi_i^{k-1} = \iota - 1) = \frac{y_{s,s',\iota}^k}{\sum_{s'' \in \mathcal{S}_i} y_{s'',s',\iota}^k}, \quad (17)$$

based on which  $\mathbb{U}_i(s_i)$ ,  $\forall s_i \in \mathcal{S}_i$  is learned via (18) with  $\zeta^k \in [0, 1]$  being the learning rate. (18) converges if  $\sum_{k=1}^{\infty} \zeta^k = \infty$  and  $\sum_{k=1}^{\infty} (\zeta^k)^2 < \infty$  [22].

Without a priori statistics of MU mobility and computation task as well as packet arrivals,  $Q$ -learning [22] finds  $\mathbb{U}_n(\boldsymbol{\chi}_n)$  for each MU  $n \in \mathcal{N}$  by defining the right-hand-side of (13) as the optimal state action-value function  $Q_n : \mathcal{X} \times \{0, 1\} \times \mathcal{A} \times \mathcal{W} \rightarrow \mathbb{R}$ . In turn, we arrive at

$$\mathbb{U}_n(\boldsymbol{\chi}_n) = \max_{\varphi_n, R_{n,(t)}, R_{n,(p)}} Q_n(\boldsymbol{\chi}_n, \varphi_n, R_{n,(t)}, R_{n,(p)}), \quad (19)$$

where an action  $(\varphi_n, R_{n,(t)}, R_{n,(p)})$  under a current local state  $\boldsymbol{\chi}_n$  consists of the channel allocation, computation offloading and packet scheduling decisions. The tabular nature in representing  $Q$ -function values makes the conventional  $Q$ -learning not readily applicable. In our considered network, the sizes of  $\mathcal{X}$  and action space  $\{0, 1\} \times \mathcal{A} \times \mathcal{W}$  are calculated as  $|\mathcal{L}|^2 \cdot (1 + A_{(t)}^{(\max)}) \cdot (1 + W^{(\max)})$  and  $2 \cdot (1 + A_{(t)}^{(\max)}) \cdot (1 + W^{(\max)})$ , resulting in an extremely slow learning process.

The success of a deep neural network in modelling the  $Q$ -function inspires us to adopt a deep reinforcement learning (DRL) method [23]. We can then approximate the  $Q$ -function by a double deep  $Q$ -network (DQN) [24]. Mathematically,  $Q_n(\boldsymbol{\chi}_n, \varphi_n, R_{n,(t)}, R_{n,(p)}) \approx Q_n(\boldsymbol{\chi}_n, \varphi_n, R_{n,(t)}, R_{n,(p)}; \boldsymbol{\theta}_n)$ ,  $\forall n \in \mathcal{N}$ , where we encapsulate in  $\boldsymbol{\theta}_n$  the set of parameters that are associated with the DQN of a MU  $n$ . During the DRL process, each MU  $n \in \mathcal{N}_i$  of a SP  $i \in \mathcal{I}$  is assumed to be equipped with a finite replay memory to store the latest  $M$  historical experiences, namely,  $\mathcal{M}_n^k = \{\mathbf{m}_n^{k-M+1}, \dots, \mathbf{m}_n^k\}$ , where each experience  $\mathbf{m}_n^{k'} = (\boldsymbol{\chi}_n^{k'}, (\varphi_n^{k'}, R_{n,(t)}^{k'}, R_{n,(p)}^{k'}))$ ,  $U_n(\boldsymbol{\chi}_n^{k'}, \varphi_n^{k'}, R_{n,(t)}^{k'}, R_{n,(p)}^{k'})$ ;  $\boldsymbol{\chi}_n^{k'+1}$ ) happens at the transition between two consecutive scheduling slots  $k'$  and  $k' + 1$ . To perform experience replay [25], MU  $n$  randomly samples a mini-batch  $\mathcal{O}_n^k \subseteq \mathcal{M}_n^k$  to train the DQN parameters using the loss function in (20), where  $\boldsymbol{\theta}_n^k$  and  $\boldsymbol{\theta}_n^{k-}$  are, respectively, the DQN parameters at a scheduling slot  $k$  and a certain previous scheduling slot before slot  $k$ .

## V. NUMERICAL EXPERIMENTS

This section conducts numerical experiments based on TensorFlow [26] to quantify the performance of the derived DRL-based scheme for multi-tenant cross-slice resource orchestration with secrecy preserving in a software-defined RAN. We

$$\mathbb{U}_n(\mathcal{X}_n) = \max_{R_{n,(t)}, R_{n,(p)}} \left\{ (1 - \gamma) \cdot U_n(\mathcal{X}_n, \varphi_n(\tilde{\pi}_{(c)}^*(\tilde{\mathcal{X}})), R_{n,(t)}, R_{n,(p)}) + \gamma \cdot \sum_{\mathcal{X}'_n \in \mathcal{X}} \mathbb{P}(\mathcal{X}'_n | \mathcal{X}_n, \varphi_n(\tilde{\pi}_{(c)}^*(\tilde{\mathcal{X}})), R_{n,(t)}, R_{n,(p)}) \cdot U_n(\mathcal{X}'_n) \right\} \quad (13)$$

$$z_n = \arg \max_{z \in \{0,1\}} \left\{ (1 - \gamma) \cdot U_n(\mathcal{X}_n, z, \pi_{n,(t)}^*(\mathcal{X}_n), \pi_{n,(p)}^*(\mathcal{X}_n)) + \gamma \cdot \sum_{\mathcal{X}'_n \in \mathcal{X}} \mathbb{P}(\mathcal{X}'_n | \mathcal{X}_n, z, \pi_{n,(t)}^*(\mathcal{X}_n), \pi_{n,(p)}^*(\mathcal{X}_n)) \cdot U_n(\mathcal{X}'_n) \right\} \quad (16)$$

$$\mathbb{U}_i^{k+1}(s_i) = \begin{cases} (1 - \zeta^k) \cdot \mathbb{U}_i^k(s_i) + \zeta^k \cdot \left( (1 - \gamma) \cdot \tau_i^k + \gamma \cdot \sum_{s_i^{k+1} \in \mathcal{S}_i} \mathbb{P}(s_i^{k+1} | s_i, \phi_i^k) \cdot \mathbb{U}_i^k(s_i^{k+1}) \right), & \text{if } s_i = s_i^k \\ \mathbb{U}_i^k(s_i), & \text{otherwise} \end{cases} \quad (18)$$

$$\text{LOSS}_n(\theta_n^k) = \mathbb{E}_{(\mathcal{X}_n, (\varphi_n, R_{n,(t)}, R_{n,(p)}), U_n(\mathcal{X}_n, \varphi_n, R_{n,(t)}, R_{n,(p)}), \mathcal{X}'_n) \in \mathcal{O}_n^k} \left[ \left( (1 - \gamma) \cdot U_n(\mathcal{X}_n, \varphi_n, R_{n,(t)}, R_{n,(p)}) + \gamma \cdot Q_n \left( \mathcal{X}'_n, \arg \max_{\varphi'_n, R'_{n,(t)}, R'_{n,(p)}} Q_n(\mathcal{X}'_n, \varphi'_n, R'_{n,(t)}, R'_{n,(p)}; \theta_n^k); \theta_{n,-}^k \right) - Q_n(\mathcal{X}_n, \varphi_n, R_{n,(t)}, R_{n,(p)}; \theta_n^k) \right)^2 \right] \quad (20)$$

set up an experimental network with 4 BSs being placed at equal distance 1 Km apart in the centre of a  $2 \times 2$  Km<sup>2</sup> square service area [15]. The entire area is divided into 1600 locations with each of  $50 \times 50$  m<sup>2</sup>. The average channel gains for a MU  $n \in \mathcal{N}$  at the location  $L_{n,(u)}^k \in \mathcal{L}_b$  covered by a BS  $b \in \mathcal{B}$  during a slot  $k$  are given by  $h_{(u)}(L_{n,(u)}^k) = H_0 \cdot (\xi_0 / \xi_{b,n}^k)^4$  and  $h_{(e)}(L_{n,(u)}^k, L_{(e)}^k) = H_0 \cdot (\xi_0 / \xi_{n,(e)}^k)^4$ , where  $H_0 = -40$  dB is the path-loss constant,  $\xi_0 = 2$  m is the reference distance, while  $\xi_{b,n}^k$  and  $\xi_{n,(e)}^k$  are the distances between MU  $n$  and BS  $b$  as well as the eavesdropper [27]. The mobilities of all MUs as well as the eavesdropper and the computation task arrivals of all MUs are independently and randomly generated. The packet arrivals follow a Poisson arrival process with average rate  $\lambda$  (in packets/slot). For the utility function in (9), we select  $U_n^{(1)}(W_n^{k+1}) = \exp\{-W_n^{k+1}\}$ ,  $U_n^{(2)}(D_n^k) = \exp\{-D_n^k\}$ ,  $U_n^{(3)}(P_{n,(CPU)}^k) = \exp\{-P_{n,(CPU)}^k\}$  and  $U_n^{(4)}(P_{n,(tr)}^k) = \exp\{-P_{n,(tr)}^k\}$ . We design for each MU a DQN with 2 hidden layers with each consisting of 16 neurons. Other parameter values used in the experiments are listed in Table I.

For comparison purpose, three baseline schemes are developed and simulated, namely,

- 1) Channel-aware control policy (Baseline 1) – At the beginning of each slot  $k$ , the need of getting one channel at a MU  $n \in \mathcal{N}$  is evaluated by  $H_{n,(u)}^k - H_{(e)}^k$ ;
- 2) Queue-aware control policy (Baseline 2) – Each MU calculates the preference between having one channel or not using a predefined threshold of the queue length;
- 3) Random control policy (Baseline 3) – This policy randomly generates the value of obtaining one channel for each MU at the beginning of each slot.

With the three baselines, after the centralized channel allocation by the SDN-orchestrator at the beginning of each slot, a MU proceeds to offload a random number of computation tasks and schedule a maximum feasible number of data packets if being assigned a channel.

We first demonstrate the average utility performance per MU per scheduling slot achieved from the proposed DRL-based scheme and the three baselines under different average

TABLE I  
PARAMETER VALUES IN EXPERIMENTS.

Parameter	Value
Set of SPs $\mathcal{I}$	{1, 2, 3}
Set of BSs $\mathcal{B}$	{1, 2, 3, 4}
Number of MUs $ \mathcal{N}_i $	6, $\forall i \in \mathcal{I}$
Channel bandwidth $\eta$	500 KHz
Noise power spectral density $\sigma^2$	-174 dBm/Hz
Scheduling slot duration $\delta$	$10^{-2}$ second
Discount factor $\gamma$	0.9
Utility price $\alpha_n$	1, $\forall n \in \mathcal{N}$
Packet size $\mu_{(p)}$	3000 bits
Maximum transmit power $\Omega^{(\max)}$	3 Watts
Weight of energy consumption $\ell_n$	3, $\forall n \in \mathcal{N}$
Maximum queue length $W^{(\max)}$	10 packets
Maximum task arrivals $A_{(t)}^{(\max)}$	5 tasks
Input data size $\mu_{(t)}$	5000 bits
CPU cycles per bit $\vartheta$	737.5
CPU-cycle frequency $\varrho$	2 GHz
Effective switched capacitance $\varsigma$	$2.5 \cdot 10^{-28}$
Exploration probability $\epsilon$	0.001
Replay memory size $M$	5000
Mini-batch size $ \mathcal{O}_n^k $	200, $\forall n \in \mathcal{N}, \forall k$
Activation function	Tanh [28]
Optimizer	Adam [29]

packet arrival rates. In this experiment, we assume that  $J = 11$  channels are shared among the MUs for the access to the computation and communication slices. The results are depicted in Fig. 2, from which we can observe that the proposed scheme achieves a significant performance gain. However, the average utility performance decreases as the average number of random data packet arrivals increases. The reason behind is that in order to ensure secrecy, more data packet arrivals lead to larger queue length, more packet drops and higher energy consumption across the MUs. Then in Fig. 3, we exhibit the average utility performance versus the number of channels, where the average packet arrival rate is fixed to be  $\lambda = 8$ . More channels available in the system provide more opportunities for the MUs to transmit the data of computation tasks to be offloaded and scheduled packets. Hence better average utility

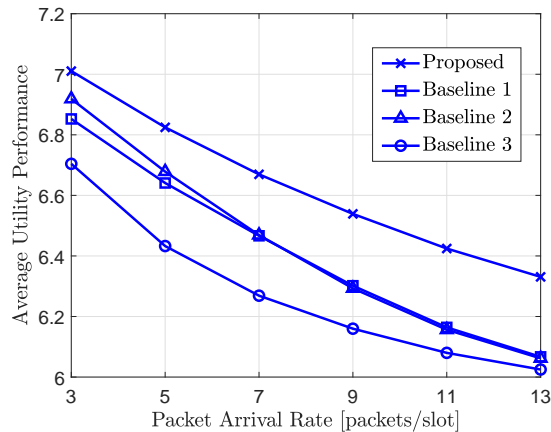


Fig. 2. Average utility performance per MU across the learning procedure versus average packet arrival rates.

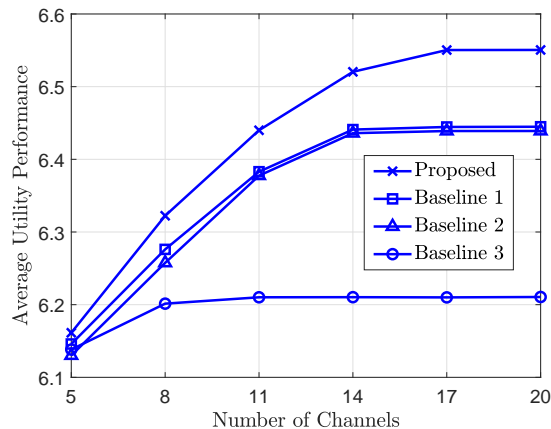


Fig. 3. Average utility performance per MU across the learning procedure versus numbers of channels.

performance can be expected by the MUs. When there are sufficient channels in the network, the data transmissions of all MUs with secrecy preserving can be fully satisfied. Both experiments show that the proposed scheme outperforms the three baselines.

## VI. CONCLUSIONS

In this paper, we investigate the problem of non-cooperative multi-tenant cross-slice resource orchestration with secrecy preserving in a software-defined RAN, which is formulated as a  $SG$ . To alleviate private information exchange among the competing SPs, we approximate the  $SG$  by a  $AG$ . Each SP is thus able to behave independently only with the local information. We observe that the decisions of the channel auction and the computation offloading as well as packet scheduling are sequentially made. This motivates us to linearly decompose the per-SP single-agent MDP, which greatly simplifies the decision making process at a SP. We propose a DRL-based scheme to find the optimal abstract control policies. Numerical experiments showcase that the performance achieved from our scheme outperforms the other baselines.

## REFERENCES

- [1] J. G. Andrews *et al.*, "Femtocells: Past, present, and future," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 3, pp. 497–508, Apr. 2012.
- [2] Y. Mao *et al.*, "A Survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, Q4 2017.
- [3] M. Satyanarayanan, "The emergence of edge computing," *IEEE Comput.*, vol. 50, no. 1, pp. 30–39, Jan. 2017.
- [4] Y. Zhou *et al.*, "Resource allocation for information-centric virtualized heterogeneous networks with in-network caching and mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 66, no. 12, pp. 11339–11351, Dec. 2017.
- [5] A. Gudipati *et al.*, "SoftRAN: Software defined radio access network," in *ACM SIGCOMM HotSDN Workshop*, Hong Kong, China, Aug. 2013.
- [6] C. Liang and F. R. Yu, "Wireless network virtualization: A survey, some research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 358–380, Q1 2015.
- [7] Y. Xiao and M. Krunz, "Dynamic network slicing for scalable fog computing systems with energy harvesting," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 12, pp. 2640–2654, Dec. 2018.
- [8] T. Frisanco *et al.*, "Infrastructure sharing and shared operations for mobile network operators: From a deployment and operations view," in *IEEE NOMS*, Salvador, Bahia, Brazil, Apr. 2008.
- [9] Google, "Project Fi," <https://fi.google.com> [Date Accessed: 12 Dec. 2018].
- [10] "Telecommunication management; network sharing; concepts and requirements," Rel. 15, 3GPP TS 32.130, Jun. 2018.
- [11] O. Sallent *et al.*, "On radio access network slicing from a radio resource management perspective," *IEEE Wireless Commun.*, vol. 24, no. 5, pp. 166–174, Oct. 2017.
- [12] H. Shah-Mansouri, V. W. S. Wong, and R. Schober, "Joint optimal pricing and task scheduling in mobile cloud computing systems," *IEEE Trans. Wireless Commun.*, vol. 16, no. 8, pp. 5218–5232, Aug. 2017.
- [13] Y. Wu *et al.*, "Secrecy-based energy-efficient data offloading via dual connectivity over unlicensed spectrums," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3252–3270, Dec. 2016.
- [14] Z. Ji and K. J. R. Liu, "Dynamic spectrum sharing: A game theoretical overview," *IEEE Commun. Mag.*, vol. 45, no. 5, pp. 88–94, May 2007.
- [15] X. Chen *et al.*, "Energy-efficiency oriented traffic offloading in wireless networks: A brief survey and a learning approach for heterogeneous cellular networks," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 4, pp. 627–640, Apr. 2015.
- [16] A. J. Nicholson and B. D. Noble, "BreadCrumbs: Forecasting mobile connectivity," in *Proc. ACM MobiCom*, San Francisco, CA, Sep. 2008.
- [17] X. He *et al.*, "Privacy-aware offloading in mobile-edge computing," in *Proc. IEEE GLOBECOM*, Singapore, Dec. 2017.
- [18] T. D. Burd and R. W. Brodersen, "Processor design for portable systems," *J. VLSI Signal Process. Syst.*, vol. 13, no. 2–3, pp. 203–221, Aug. 1996.
- [19] A. M. Fink, "Equilibrium in a stochastic  $n$ -person game," *J. Sci. Hiroshima Univ. Ser. A-I*, vol. 28, pp. 89–93, 1964.
- [20] X. Chen *et al.*, "Wireless resource scheduling in virtualized radio access networks using stochastic learning," *IEEE Trans. Mobile Comput.*, vol. 17, no. 4, pp. 961–974, Apr. 2018.
- [21] J. Jia *et al.*, "Revenue generation for truthful spectrum auction in dynamic spectrum access," in *Proc. ACM MobiHoc*, New Orleans, LA, May 2009.
- [22] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.
- [23] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.
- [24] H. van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proc. AAAI*, Phoenix, AZ, Feb. 2016.
- [25] L.-J. Lin, "Reinforcement learning for robots using neural networks," Carnegie Mellon University, 1992.
- [26] M. Abadi *et al.*, "Tensorflow: A system for large-scale machine learning," in *Proc. OSDI*, Savannah, GA, Nov. 2016.
- [27] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3590–3605, Dec. 2016.
- [28] K. Jarrett *et al.*, "What is the best multi-stage architecture for object recognition?" in *Proc. IEEE ICCV*, Kyoto, Japan, Sep.–Oct. 2009.
- [29] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *Proc. ICLR*, San Diego, CA, May 2015.