Multi-Tenant Cross-Slice Resource Orchestration: A Deep Reinforcement Learning Approach

Xianfu Chen, Zhifeng Zhao, Celimuge Wu, Mehdi Bennis, Hang Liu,
Yusheng Ji, and Honggang Zhang

Abstract

In a software-defined radio access network (RAN), a major challenge lies in how to support diverse services for mobile users (MUs) over a common physical network infrastructure. Network slicing is a promising solution to tailor the network to match such service requests. This paper considers a software-defined RAN, where a limited number of channels are auctioned across scheduling slots to MUs of multiple service providers (SPs) (i.e., the tenants). Each SP behaves selfishly to maximize the expected long-term payoff from competition with other SPs for the orchestration of channel access opportunities over its MUs, which request both mobile-edge computing and traditional cellular services in the slices. This problem is modelled as a stochastic game, in which the decision makings of a SP depend on the network dynamics as well as the control policies of its competitors. We propose an abstract stochastic game to approximate the Nash equilibrium. The selfish behaviours of a SP can then be characterized by a single-agent Markov decision process (MDP). To simplify decision makings, we linearly decompose the per-SP MDP and derive an online scheme based on deep reinforcement learning to approach the optimal abstract control policies. Numerical experiments show significant performance gains from our scheme.

Index Terms

X. Chen is with the VTT Technical Research Centre of Finland, Oulu, Finland (e-mail: xianfu.chen@vtt.fi). Z. Zhao and H. Zhang are with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China (e-mail: {zhaozf, honggangzhang}@zju.edu.cn). C. Wu is with the Graduate School of Informatics and Engineering, University of Electro-Communications, Tokyo, Japan (email: clmg@is.uec.ac.jp). M. Bennis is with the Centre for Wireless Communications, University of Oulu, Finland (email: mehdi.bennis@oulu.fi). H. Liu is with the Department of Electrical Engineering and Computer Science, the Catholic University of America, USA (e-mail: liuh@cua.edu). Y. Ji is with the Information Systems Architecture Research Division, National Institute of Informatics, Tokyo, Japan (e-mail: kei@nii.ac.jp).

Network slicing, radio access networks, mobile-edge computing, packet scheduling, Markov decision process, deep reinforcement learning.

I. Introduction

With the proliferation of smart mobile devices, a multitude of emerging broadband applications are driving up the demands of wireless mobile services [1]. To keep up with the demands, new cellular network infrastructures, such as small cells, are being constantly deployed [2]. An advanced dense network infrastructure can reach much higher network capacity due to a shorter transmission range and a smaller number of mobile users (MUs) per cell site. However, the coordinated control plane decisions in a dense radio access network (RAN) makes it expensive to deploy and extremely complex to manage [3]. By abstracting the independent base stations (BSs) as a centralized network controller (CNC), the software-defined networking (SDN) concept simplifies the management of a very dense RAN [4], [5]. In a software-defined RAN, the CNC instead is responsible for making all control plane decisions.

One of the key benefits of the software-defined RAN is to facilitate full network sharing, where the single-ownership of a physical network infrastructure is separated from the wireless mobile services [6]. As such, the same network infrastructure is able to host, on a network-as-a-service basis, multiple tenants, namely, multiple service providers (SPs). For example, an application provider (e.g., Netflix and Google [7]) can become a SP so as to lease wireless radio resources from the infrastructure provider to improve the Quality-of-Service (QoS) and the Quality-of-Experience (QoE) for its subscribed MUs. In addition, a software-defined RAN enables network slicing, which is a network virtualization technique that splits the physical network infrastructure into multiple virtual networks/slices. With the help of network slicing, the slices can be tailored for diverse service requests with various QoS and QoE requirements from MUs.

In spite of the advantages brought by a software-defined networking architecture, technical challenges remain. Particularly, mechanisms that efficiently exploit the decoupling of control plane and data plane in a software-defined RAN must be developed to achieve optimized wireless resource utilization. In the literature, there exist a number of related works on resource orchestration in a software-defined RAN. In [8], Petrov et al. introduced a softwarized fifth-generation (5G) architecture to orchestrate resources for end-to-end reliability of the mission-critical traffic. In [9], Caballero er al. analyzed a “share-constrained proportional allocation” mechanism for resource sharing to realize network slicing, which falls into a network slicing game framework. In our previous work [10], a stochastic game was adopted to model the non-cooperative behaviours of SPs during the competition for limited wireless radio resource. However, the efforts in these works concentrate on the traditional cellular services.

In recent years, the computation-intensive applications, e.g., location-based virtual/augmented reality and online gaming, are gaining increasing popularity. Mobile devices of MUs are in general resource-constrained in battery capacity and processing speed of central processing unit (CPU). The tension between computation-intensive applications and resource-constrained mobile devices calls for a revolution in computing infrastructure [11]. Mobile-edge computing (MEC), which provides computing capabilities within the RANs in close proximity to MUs, is envisioned as a promising paradigm [12]. In this context, we consider a software-defined RAN where the network slices accommodate both the MEC and the traditional cellular services [13]. Offloading a computation task to a MEC server for execution involves wireless transmissions. Hence how to orchestrate radio resources among the network slices must be carefully designed and adds another dimension of technical challenges [14], [15].

This paper is primarily concerned with a software-defined RAN, where the CNC manages a limited number of channels and multiple SPs compete to orchestrate channel access opportunities for their subscribed MUs requesting MEC and traditional cellular services in the slices in accordance with the network dynamics. Network dynamics originates from the mobilities as well as the random computation task and data packet arrivals of MUs. Upon receiving the auction bids from all SPs, the CNC allocates channels to MUs through a Vickrey-Clarke-Groves (VCG) pricing mechanism[1] [16]. Each MU then proceeds to offload computation tasks and scheduled queued packets over the assigned channel with the objective of optimizing the expected long-term performance. The non-cooperative behaviours of SPs are modelled as a stochastic game. The main contributions from this paper are listed as follows.

- To the best knowledge of the authors, this is the first work to investigate multi-tenant cross-slice wireless radio resource orchestration in a software-defined RAN, which is formulated as a stochastic game. In the non-cooperative game, each SP aims to maximize its own expected long-term payoff.
- Without any information exchange among the SPs, we transform the stochastic game into

[1] One major advantage of the VCG mechanism is that the dominant auction policy for a SP is to bid the true values for the channels.

an abstract stochastic game with bounded performance regret. The decision making process at each SP then falls into a single-agent Markov decision process (MDP).

- We propose a linear decomposition approach to solve a per-SP MDP, leading to simplified decision makings. Such a linear decomposition approach allows each MU to compute locally the state-value functions.
- To deal with the huge state space faced by a MU, we leverage a deep reinforcement learning (DRL) algorithm [17] to learn the optimal computation offloading and packet scheduling policies without any a priori statistical knowledge of network dynamics.
- Numerical experiments using TensorFlow [18] are carried out to verify the theoretical studies in this paper, showing that our proposed scheme outperforms three state-of-the-art baseline schemes.

In the next section, we describe the considered system model and the assumptions used throughout this paper. In Section III, we formulate the problem of non-cooperative multi-tenant cross-slice resource orchestration as a stochastic game and discuss a best-response solution. In Section IV, we propose to approximate the stochastic game by an abstract stochastic game and derive an online learning scheme to solve the problem. In Section V, we provide numerical experiments under various settings to evaluate the performance from our scheme. Finally, we draw the conclusions in Section VI.

II. System Descriptions and Assumptions

As being illustrated in Fig. 1, we focus in this paper on a software-defined RAN, where the physical network infrastructure is split into multiple virtual networks to support heterogeneous mobile service requests that are basically categorized into the MEC service slice and the traditional cellular service slice. The shared RAN, which consists of a set $\mathcal{B}$ of BSs, covers a service region with a set $\mathcal{L}$ of locations or small areas, each being characterized by uniform signal propagation conditions [19], [20]. We choose $\mathcal{L}_b$ to designate the set of locations covered by a BS $b \in \mathcal{B}$. For any two BSs in the RAN, we assume that $\mathcal{L}_b \cap \mathcal{L}_{b'} = \emptyset$, where $b' \in \mathcal{B}$ and $b' \neq b$. We represent the geographical distribution of BSs by a topological graph $\mathcal{TG} = \langle \mathcal{B}, \mathcal{E} \rangle$, where $\mathcal{E} = \{e_{b,b'} : b \neq b', b, b' \in \mathcal{B}\}$ represents the relative locations between the BSs with

$$e_{b,b'} = \begin{cases} 1, \text{ if BSs } b \text{ and } b' \text{ are neighbours;} \\ 0, \text{ otherwise.} \end{cases} \tag{1}$$

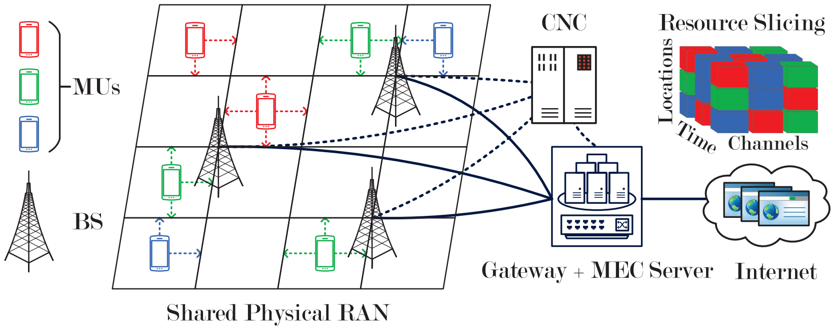

Fig. 1. Architecture of a software-defined radio access network (RAN) (BS: base station; CNC: centralized network controller.). Multiple service providers (SPs), namely, the tenants, provide both mobile-edge computing (MEC) and traditional cellular services. The mobile users (MUs) of SPs, which are shown in the different colors, move cross the service region. Over the time horizon, the CNC allocates the limited wireless radio resource to MUs based on the bids that are submitted by their respective subscribing SPs.

Different SPs provide different mobile services, and each MU can subscribe to only one SP $i \in \mathcal{I} = \{1, \cdots, I\}$. Let $\mathcal{N}_i$ be the set of MUs of SP i, then $\mathcal{N} = \cup_{i \in \mathcal{I}} \mathcal{N}_i$ denotes the set of all MUs across the network.

A. *Inter-Tenant Channel Auction*

We consider a system with a set $\mathcal{J} = \{1, \cdots, J\}$ of non-overlapping orthogonal channels with the same bandwidth η (in Hz). The whole system operates across discrete scheduling slots, each of which is indexed by an integer $k \in \mathbb{N}_+$ and is assumed to be of equal time duration δ (in seconds). Over the time horizon, the MUs move in the service region $\mathcal{L}$ following a Markov mobility model. Such a mobility model is widely used in the literature [21], [22]. Let $\mathcal{N}_{b,i}^k$ be the set of MUs appearing in the coverage of a BS $b \in \mathcal{B}$ at a scheduling slot $k \in \mathbb{N}_+$ that are subscribed to SP $i \in \mathcal{I}$, then $\mathcal{N}_i = \cup_{b \in \mathcal{B}} \mathcal{N}_{b,i}^k$, $\forall k \in \mathbb{N}_+$. During a scheduling slot, a MU at a location can only be associated with the BS that covers the location. The SPs compete for the limited number of channels in order to provide satisfactory mobile services in the slices to their MUs. Specifically, at the beginning of each scheduling slot k, each SP i submits to the CNC a bid given by $\hat{\boldsymbol{\beta}}_i^k = (\hat{\nu}_i^k, \hat{\mathbf{C}}_i^k)$, which is not necessarily equal to $\boldsymbol{\beta}_i^k = (\nu_i^k, \mathbf{C}_i^k)$. Herein, $\mathbf{C}_i^k = (C_{b,i}^k : b \in \mathcal{B})$ with $C_{b,i}^k$ being the number of potentially needed channels within the coverage of a BS b and ν_i^k is the true value over $\mathbf{C}_i^k$. Upon receiving the auction bids $\hat{\boldsymbol{\beta}}^k = (\hat{\boldsymbol{\beta}}_i^k : i \in \mathcal{I})$ from all SPs, the CNC proceeds to allocate channels to MUs and computes payment τ_i^k for each SP i. Let $\boldsymbol{\rho}_n^k = (\rho_{n,j}^k : j \in \mathcal{J})$ be the channel allocation vector for a MU $n \in \mathcal{N}$, where

$$\rho_{n,j}^k = \begin{cases} 1, \text{ if channel } j \text{ is allocated to} \\ \quad \text{MU } n \in \mathcal{N} \text{ at scheduling slot } k; \\ 0, \text{ otherwise.} \end{cases} \tag{2}$$

We apply the following constraints for the centralized channel allocation at the CNC during a single slot,

$$\left(\sum_{i \in \mathcal{I}} \sum_{n \in \mathcal{N}_{b,i}^k} \rho_{n,j}^k\right) \cdot \left(\sum_{i \in \mathcal{I}} \sum_{n \in \mathcal{N}_{b',i}^k} \rho_{n,j}^k\right) = 0, \text{if } e_{b,b'} = 1, \forall e_{b,b'} \in \mathcal{E}, \forall j \in \mathcal{J}; \tag{3}$$

$$\sum_{i \in \mathcal{I}} \sum_{n \in \mathcal{N}_{b,i}^k} \rho_{n,j}^k \leq 1, \forall b \in \mathcal{B}, \forall j \in \mathcal{J}, \tag{4}$$

to ensure that a channel cannot be allocated to the coverage areas of two adjacent BSs in order to avoid interference in data transmissions, and in the coverage of a BS, a MU can be assigned at most one channel and a channel can be assigned to at most one MU.

We denote by $\boldsymbol{\phi}^k = (\phi_i^k : i \in \mathcal{I})$ the winner determination in the channel auction at a scheduling slot k, where $\phi_i^k = 1$ if SP i wins the channel auction while $\phi_i^k = 0$ means no channel is allocated to the MUs of SP i during the slot. The CNC calculates $\boldsymbol{\phi}^k$ through the VCG mechanism that maximizes the "social welfare"[2]

$$\begin{aligned} & \max_{\boldsymbol{\phi}} \sum_{i \in \mathcal{I}} \phi_i \cdot \hat{\nu}_i^k \\ \text{s.t.} \quad & \text{constraints (3) and (4);} \\ & \sum_{n \in \mathcal{N}_{b,i}^k} \varphi_n^k = \phi_i \cdot C_{b,i}^k, \forall b \in \mathcal{B}, \forall i \in \mathcal{I}, \end{aligned} \tag{5}$$

where $\boldsymbol{\phi} = (\phi_i \in \{0, 1\} : i \in \mathcal{I})$ and $\varphi_n^k = \sum_{j \in \mathcal{J}} \rho_{n,j}^k$ is a channel allocation variable that equals 1 if MU n is assigned a channel and 0 otherwise. And the payment for each SP i is

$$\tau_i^k = \max_{\boldsymbol{\phi}_{-i}} \sum_{i' \in \mathcal{I} \setminus \{i\}} \phi_{i'} \cdot \hat{\nu}_{i'}^k - \max_{\boldsymbol{\phi}} \sum_{i' \in \mathcal{I} \setminus \{i\}} \phi_{i'} \cdot \hat{\nu}_{i'}^k, \tag{6}$$

where $-i$ denotes all the other SPs in set $\mathcal{I}$ without the presence of SP i. In particular, the VCG-based channel auction at a slot k possesses the following economic properties:

[2] Other fairness rules (e.g., [23]) can also be implemented and do not affect the proposed resource orchestration scheme in this paper.

- *Efficiency* – When all SPs announce their true auction bids, the CNC allocates channels to maximize the sum of values, resulting in efficient channel utilization.
- *Individual Rationality* – Each SP i can expect a nonnegative payoff $\hat{\nu}_i^k - \tau_i^k$ at any scheduling slot k.
- *Truthfulness* – No SP can improve its payoff by bidding different from its true value, which implies that the optimal bid at any slot k is $\hat{\boldsymbol{\beta}}_i^k = \boldsymbol{\beta}_i^k$, $\forall i \in \mathcal{I}$.

B. Communication and Computation Models

Let $L_n^k \in \mathcal{L}$ be the location of a MU $n \in \mathcal{N}$ during a scheduling slot k, and the average channel gain $H_n^k = h(L_n^k)$ experienced by MU n during the slot is determined by the physical distance between the MU and the associated BS[3] [19], [20]. At the beginning of each scheduling slot k, each MU n independently generates a number $A_{n,(\mathrm{t})}^k \in \mathcal{A} = \{0, 1, \cdots, A_{(\mathrm{t})}^{(\max)}\}$ of computation tasks. We represent a computation task by $(\mu_{(\mathrm{t})}, \vartheta)$ with $\mu_{(\mathrm{t})}$ and ϑ being, respectively, the input data size (in bits) and the number of CPU cycles required to accomplish one input bit of the computation task. It's assumed in this work that the sequence $\{A_{n,(\mathrm{t})}^k : k \in \mathbb{N}_+\}$ follows a Markov process [24]. Two options are available for each computation task[4]: 1) being processed locally at the MU; and 2) being offloaded to the MEC server. In other words, the arriving computation tasks should be executed during the scheduling slot[5]. The computation offloading decision for MU n at a slot k specifies the number $R_{n,(\mathrm{t})}^k$ of tasks to be transmitted to the MEC server. The final number of tasks to be processed by the mobile device hence is $A_{n,(\mathrm{t})}^k - \varphi_n^k \cdot R_{n,(\mathrm{t})}^k$. Meanwhile, a data queue is maintained at each MU to buffer the packets coming from the traditional cellular service. The arriving packets get queued until transmissions and we assume that every data packet has a constant size of $\mu_{(\mathrm{p})}$ (bits). Let W_n^k and $A_{n,(\mathrm{p})}^k$ be, respectively, the queue length and the random new packet arrivals for MU n at the beginning of slot k. The packet arrival process is assumed to be independent among the MUs and i.i.d. across the scheduling slots. Let $R_{n,(\mathrm{p})}^k$ be the decision of the number of data packets that are to be removed from the queue of

[3] It's straightforward that given the mobility model, the average overhead of a MU incurred during inter-BS handovers is fixed.

[4] The kind of computation tasks that can only be processed at the mobile devices [24] does not affect the optimizing goal and hence is neglected.

[5] For simplicity, we assume that the CPU power at a mobile device matches the maximum computation task arrivals and a MU can hence process $A_{(\mathrm{t})}^{(\max)}$ tasks within one scheduling slot.

MU n at slot k. The number of packets that are eventually transmitted then is $\varphi_n^k \cdot R_{n,(\mathrm{p})}^k$, and the queue evolution of MU n can be written as

$$W_n^{k+1} = \min\left\{W_n^k - \varphi_n^k \cdot R_{n,(\mathrm{p})}^k + A_{n,(\mathrm{p})}^k, W^{(\max)}\right\}, \tag{7}$$

where $W^{(\max)}$ is the maximum buffer size that restricts $W_n^k \in \mathcal{W} = \{0, \cdots, W^{(\max)}\}$.

Following the discussions in [25], the energy (in Joules) consumed by a MU $n \in \mathcal{N}$ for reliably transmitting input data of $\varphi_n^k \cdot R_{n,(\mathrm{t})}^k$ computation tasks and $\varphi_n^k \cdot R_{n,(\mathrm{p})}^k$ packets during a scheduling slot $k \in \mathbb{N}_+$ can be computed as

$$P_{n,(\mathrm{tr})}^k = \frac{\delta \cdot \eta \cdot \sigma^2}{H_n^k} \cdot \left(2^{\frac{\varphi_n^k \cdot \left(\mu_{(\mathrm{t})} \cdot R_{n,(\mathrm{t})}^k + \mu_{(\mathrm{p})} \cdot R_{n,(\mathrm{p})}^k\right)}{\eta \cdot \delta}} - 1\right), \tag{8}$$

where σ^2 is the noise power spectral density. Let $P^{(\max)}$ be the maximum transmit power for all MUs, namely, $P_{n,(\mathrm{tr})}^k \leq P^{(\max)}$, $\forall n$ and $\forall k$. For the rest number $A_{n,(\mathrm{t})}^k - \varphi_n^k \cdot R_{n,(\mathrm{t})}^k$ of computation tasks that are processed at the mobile device of MU n, the CPU energy consumption can be calculated as

$$P_{n,(\mathrm{CPU})}^k = \varsigma \cdot \mu_{(\mathrm{t})} \cdot \vartheta \cdot \varrho^2 \cdot \left(A_{n,(\mathrm{t})}^k - \varphi_n^k \cdot R_{n,(\mathrm{t})}^k\right), \tag{9}$$

where ς is the effective switched capacitance that depends on chip architecture of the mobile device [26] and ϱ is the CPU-cycle frequency at a mobile device.

C. Control Policy

We denote $\boldsymbol{\chi}_n^k = (L_n^k, A_{n,(\mathrm{t})}^k, W_n^k) \in \mathcal{X} = \mathcal{L} \times \mathcal{A} \times \mathcal{W}$ as the local network state at a MU $n \in \mathcal{N}$. Thus $\boldsymbol{\chi}^k = (\boldsymbol{\chi}_n^k : n \in \mathcal{N}) \in \mathcal{X}^{|\mathcal{N}|}$ characterizes the global network state, where $|\mathcal{N}|$ means the cardinality of the set $\mathcal{N}$. Each SP $i \in \mathcal{I}$ aims to design a control policy $\boldsymbol{\pi}_i = (\pi_{i,(\mathrm{c})}, \boldsymbol{\pi}_{i,(\mathrm{t})}, \boldsymbol{\pi}_{i,(\mathrm{p})})$, where $\pi_{i,(\mathrm{c})}$, $\boldsymbol{\pi}_{i,(\mathrm{t})} = (\pi_{n,(\mathrm{t})} : n \in \mathcal{N}_i)$ and $\boldsymbol{\pi}_{i,(\mathrm{p})} = (\pi_{n,(\mathrm{p})} : n \in \mathcal{N}_i)$ are the channel auction, the computation offloading and the packet scheduling policies, respectively. Note that the computation offloading policy $\pi_{n,(\mathrm{t})}$ as well as the packet scheduling policy $\pi_{n,(\mathrm{p})}$ are MU-specified, hence both $\boldsymbol{\pi}_{i,(\mathrm{t})}$ and $\boldsymbol{\pi}_{i,(\mathrm{p})}$ depend only on $\boldsymbol{\chi}_i^k = (\boldsymbol{\chi}_n^k : n \in \mathcal{N}_i) \in \mathcal{X}_i = \mathcal{X}^{|\mathcal{N}_i|}$. The joint control policy of all SPs is given by $\boldsymbol{\pi} = (\boldsymbol{\pi}_i : i \in \mathcal{I})$. With the observation of $\boldsymbol{\chi}^k$ at the beginning of each scheduling slot k, SP i announces the auction bid $\boldsymbol{\beta}_i^k$ to the CNC for channel allocation and decides the numbers of computation tasks $\mathbf{R}_{i,(\mathrm{t})}^k$ to be offloaded and packets $\mathbf{R}_{i,(\mathrm{p})}^k$ to be transmitted following $\boldsymbol{\pi}_i$, that is, $\boldsymbol{\pi}_i(\boldsymbol{\chi}^k) = (\pi_{i,(\mathrm{c})}(\boldsymbol{\chi}^k), \boldsymbol{\pi}_{i,(\mathrm{t})}(\boldsymbol{\chi}_i^k), \boldsymbol{\pi}_{i,(\mathrm{p})}(\boldsymbol{\chi}_i^k)) =$

$(\boldsymbol{\beta}_i^k, \mathbf{R}_{i,(\mathrm{t})}^k, \mathbf{R}_{i,(\mathrm{p})}^k)$, where $\mathbf{R}_{i,(\mathrm{t})}^k = (R_{n,(\mathrm{t})}^k : n \in \mathcal{N}_i)$ and $\mathbf{R}_{i,(\mathrm{p})}^k = (R_{n,(\mathrm{p})}^k : n \in \mathcal{N}_i)$. We define an instantaneous payoff function for SP $i \in \mathcal{I}$ at a slot k as below,

$$F_i\big(\boldsymbol{\chi}^k, \boldsymbol{\varphi}_i^k, \mathbf{R}_{i,(\mathrm{t})}^k, \mathbf{R}_{i,(\mathrm{p})}^k\big) = \sum_{n \in \mathcal{N}_i} \alpha_n \cdot U_n\big(\boldsymbol{\chi}_n^k, \varphi_n^k, R_{n,(\mathrm{t})}^k, R_{n,(\mathrm{p})}^k\big) - \tau_i^k, \tag{10}$$

where $\boldsymbol{\varphi}_i^k = (\varphi_n^k : n \in \mathcal{N}_i)$ and $\alpha_n \in \mathbb{R}_+$ can be treated herein as the unit price to charge a MU n for realizing utility $U_n(\boldsymbol{\chi}_n^k, \varphi_n^k, R_{n,(\mathrm{t})}^k, R_{n,(\mathrm{p})}^k)$ from consuming power to process the arriving computation tasks and transmit the queued packets to avoid the packet overflows, which is chosen to be

$$\begin{aligned} U_n\big(\boldsymbol{\chi}_n^k, \varphi_n^k, R_{n,(\mathrm{t})}^k, R_{n,(\mathrm{p})}^k\big) &= U_n^{(1)}\big(W_n^{k+1}\big) + U_n^{(2)}\big(D_n^k\big) \\ &\quad + \ell_n \cdot \big(U_n^{(3)}\big(P_{n,(\mathrm{tr})}^k\big) + U_n^{(4)}\big(P_{n,(\mathrm{CPU})}^k\big)\big), \end{aligned} \tag{11}$$

where $D_n^k = \max\{W_n^k - \varphi_n^k \cdot R_{n,(\mathrm{p})}^k + A_{n,(\mathrm{p})}^k - W^{(\max)}, 0\}$ defines the number of packet drops that occur when the queue vacancy is less than the number of arriving packets, and $\ell_n \in \mathbb{R}_+$ is a constant weighting factor that balances the importance of the energy consumption within a scheduling slot. We assume the functions $U_n^{(1)}(\cdot)$, $U_n^{(2)}(\cdot)$, $U_n^{(3)}(\cdot)$ and $U_n^{(4)}(\cdot)$ are positive monotonically decreasing.

III. Problem Statement and Game-Theoretic Solution

In this section, we first formulate the problem of cross-slice resource orchestration among the non-cooperative SPs (namely, the competitive channel auction, computation offloading and packet scheduling) across the time horizon as a stochastic game and then discuss the best-response solution from a game-theoretic perspective.

A. Stochastic Game Formulation

Due to the limited radio resource and the stochastic nature in networking environment, we therefore formulate the problem of cross-slice resource orchestration among multiple non-cooperative SPs over the time horizon 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 $\{\boldsymbol{\pi}_i : \forall i \in \mathcal{I}\}$. The joint control policy $\boldsymbol{\pi}$ induces a probability distribution over the sequence of global network states $\{\boldsymbol{\chi}^k : k \in \mathbb{N}_+\}$ and the sequences of per-slot instantaneous payoffs $\{F_i(\boldsymbol{\chi}^k, \boldsymbol{\varphi}_i^k, \mathbf{R}_{i,(\mathrm{t})}^k, \mathbf{R}_{i,(\mathrm{p})}^k) : k \in \mathbb{N}_+\}$, $\forall i \in \mathcal{I}$. From assumptions on the mobility of a MU and the random computation task and data packet arrivals, the randomness lying in $\{\boldsymbol{\chi}^k : k \in \mathbb{N}_+\}$ is hence Markovian with the following state transition probability

$$\begin{aligned}
&\mathbb{P}\big(\boldsymbol{\chi}^{k+1}|\boldsymbol{\chi}^k, \boldsymbol{\varphi}\big(\boldsymbol{\pi}_{(\mathrm{c})}\big(\boldsymbol{\chi}^k\big)\big), \boldsymbol{\pi}_{(\mathrm{t})}\big(\boldsymbol{\chi}^k\big), \boldsymbol{\pi}_{(\mathrm{p})}\big(\boldsymbol{\chi}^k\big)\big) \\
&= \prod_{n\in\mathcal{N}} \mathbb{P}\big(L_n^{k+1}|L_n^k\big) \cdot \mathbb{P}\Big(A_{n,(\mathrm{t})}^{k+1}|A_{n,(\mathrm{t})}^k\Big) \cdot \mathbb{P}\big(W_n^{k+1}|W_n^k, \varphi_n\big(\boldsymbol{\pi}_{(\mathrm{c})}\big(\boldsymbol{\chi}^k\big)\big), \pi_{n,(\mathrm{t})}\big(\boldsymbol{\chi}_n^k\big), \pi_{n,(\mathrm{p})}\big(\boldsymbol{\chi}_n^k\big)\big),
\end{aligned} \tag{12}$$

where $\mathbb{P}(\cdot)$ denotes the probability of an event, $\boldsymbol{\varphi} = (\boldsymbol{\varphi}_i : i \in \mathcal{I})$ is the global channel allocation by the CNC, while $\boldsymbol{\pi}_{(\mathrm{c})} = (\pi_{i,(\mathrm{c})} : i \in \mathcal{I})$, $\boldsymbol{\pi}_{(\mathrm{t})} = (\boldsymbol{\pi}_{i,(\mathrm{t})} : i \in \mathcal{I})$ and $\boldsymbol{\pi}_{(\mathrm{p})} = (\boldsymbol{\pi}_{i,(\mathrm{p})} : i \in \mathcal{I})$ are the joint channel auction, the joint computation offloading and the joint packet scheduling policies.

Taking expectation with respect to the sequence of per-slot instantaneous payoffs, the expected long-term payoff[6] of a SP $i \in \mathcal{I}$ for a given initial global network state $\boldsymbol{\chi}^1 = \boldsymbol{\chi} \triangleq (\boldsymbol{\chi}_n = (L_n, A_{n,(\mathrm{t})}, W_n) : n \in \mathcal{N})$ can be expressed as

$$\begin{aligned}
&V_i(\boldsymbol{\chi}, \boldsymbol{\pi}) \\
&= (1-\gamma) \cdot \mathsf{E}_{\boldsymbol{\pi}}\left[\sum_{k=1}^{\infty} (\gamma)^{k-1} \cdot F_i\big(\boldsymbol{\chi}^k, \boldsymbol{\varphi}_i\big(\boldsymbol{\pi}_{(\mathrm{c})}\big(\boldsymbol{\chi}^k\big)\big), \boldsymbol{\pi}_{i,(\mathrm{t})}\big(\boldsymbol{\chi}_i^k\big), \boldsymbol{\pi}_{i,(\mathrm{p})}\big(\boldsymbol{\chi}_i^k\big)\big) | \boldsymbol{\chi}^1 = \boldsymbol{\chi}\right],
\end{aligned} \tag{13}$$

where $\gamma \in [0, 1)$ is a discount factor and $(\gamma)^{k-1}$ denotes the discount factor to the $(k-1)$-th power. $V_i(\boldsymbol{\chi}, \boldsymbol{\pi})$ is also termed as the state value function of SP i in a global network state $\boldsymbol{\chi}$ under a joint control policy $\boldsymbol{\pi}$. The aim of each SP i is to device a best-response control policy $\boldsymbol{\pi}_i^*$ that maximizes $V_i(\boldsymbol{\chi}, \boldsymbol{\pi}_i, \boldsymbol{\pi}_{-i})$ for any given initial network state $\boldsymbol{\chi}$, which can be formally formulated as

$$\boldsymbol{\pi}_i^* = \arg\max_{\boldsymbol{\pi}_i} V_i(\boldsymbol{\chi}, \boldsymbol{\pi}_i, \boldsymbol{\pi}_{-i}), \forall \boldsymbol{\chi} \in \mathcal{X}^{|\mathcal{N}|}. \tag{14}$$

A Nash equilibrium (NE) describes the rational behaviours of the SPs in a stochastic game.

Definition 1. In our formulated stochastic game, $\mathcal{SG}$, a NE is a tuple of control policies $\langle \boldsymbol{\pi}_i^* : i \in \mathcal{I} \rangle$, where each $\boldsymbol{\pi}_i^*$ of a SP i is the best response to the other SPs' $\boldsymbol{\pi}_{-i}^*$.

For the I-player stochastic game $\mathcal{SG}$ with expected infinite-horizon discounted payoffs, there always exists a NE in stationary control policies [27]. Define $\mathbb{V}_i(\boldsymbol{\chi}) = V_i(\boldsymbol{\chi}, \boldsymbol{\pi}_i^*, \boldsymbol{\pi}_{-i}^*)$ as the optimal state value function, $\forall i \in \mathcal{I}$ and $\forall \boldsymbol{\chi} \in \mathcal{X}^{|\mathcal{N}|}$.

[6] Due to the non-cooperative behaviours among SPs, the control policies, $\boldsymbol{\pi}_i$, $\forall i \in \mathcal{I}$, are not unichain. Therefore, our considered Markovian system is non-ergodic, for which we define an expected infinite-horizon discounted payoff function as the optimization goal of a SP.

Remark 1: From (13), we can easily observe that the expected long-term payoff of a SP $i \in \mathcal{I}$ depends on information of not only the global network state across time horizon but the joint control policy $\boldsymbol{\pi}$. In other words, the decision makings from all SPs are tightly coupled in $\mathcal{SG}$.

B. Best-Response Approach

Suppose that in the $\mathcal{SG}$, the global network state information is known and all SPs play the NE control policies $\boldsymbol{\pi}^*$, the best-response of a SP $i \in \mathcal{I}$ under $\boldsymbol{\chi} \in \mathcal{X}^{|\mathcal{N}|}$ can then be obtained as

$$\mathbb{V}_i(\boldsymbol{\chi}) = \max_{\boldsymbol{\pi}_i(\boldsymbol{\chi})} \Bigg\{ (1-\gamma) \cdot F_i\big(\boldsymbol{\chi}, \boldsymbol{\varphi}_i\big(\pi_{i,(\mathrm{c})}(\boldsymbol{\chi}), \boldsymbol{\pi}^*_{-i,(\mathrm{c})}(\boldsymbol{\chi})\big), \boldsymbol{\pi}_{i,(\mathrm{t})}(\boldsymbol{\chi}_i), \boldsymbol{\pi}_{i,(\mathrm{p})}(\boldsymbol{\chi}_i)\big) + \\ \gamma \cdot \sum_{\boldsymbol{\chi}' \in \mathcal{X}^{|\mathcal{N}|}} \mathbb{P}\big(\boldsymbol{\chi}' | \boldsymbol{\chi}, \boldsymbol{\varphi}\big(\pi_{i,(\mathrm{c})}(\boldsymbol{\chi}), \boldsymbol{\pi}^*_{-i,(\mathrm{c})}(\boldsymbol{\chi})\big), \big(\boldsymbol{\pi}_{i,(\mathrm{t})}(\boldsymbol{\chi}_i), \boldsymbol{\pi}^*_{-i,(\mathrm{t})}(\boldsymbol{\chi}_{-i})\big), \big(\boldsymbol{\pi}_{i,(\mathrm{p})}(\boldsymbol{\chi}_i), \boldsymbol{\pi}^*_{-i,(\mathrm{p})}(\boldsymbol{\chi}_{-i})\big)\big) \cdot \\ \mathbb{V}_i(\boldsymbol{\chi}') \Bigg\}, \tag{15}$$

where $\boldsymbol{\chi}_i = (\boldsymbol{\chi}_n : n \in \mathcal{N}_i)$ and $\boldsymbol{\chi}' = (\boldsymbol{\chi}'_n = (L'_n, A'_{n,(\mathrm{t})}, W'_n) : n \in \mathcal{N})$ is the next global network state.

Remark 2: It is a challenging task to find the NE for the $\mathcal{SG}$. In order to operate the NE, all SPs have to know the global network dynamics, which is prohibited in our non-cooperative networking environment.

IV. Abstract Stochastic Game Reformulation and Deep Reinforcement Learning

In this section, we elaborate on how SPs play the cross-slice resource orchestration stochastic game with limited information. By linearly decomposing the abstract state-value functions of a SP, we derive an online learning scheme to approximate the optimal control policies.

A. Stochastic Game Abstraction via Conjectures

To capture the coupling of decision makings among the competing SPs, we abstract the stochastic game $\mathcal{SG}$ as $\mathcal{AG}$ [10], [28]. In the abstract stochastic game $\mathcal{AG}$, a SP $i \in \mathcal{I}$ behaves based on its own local network dynamics and abstractions of states at other competing SPs. Let $\mathcal{S}_i = \{1, \cdots, S_i\}$ be the abstraction of state space $\mathcal{X}_{-i}$, where $S_i \in \mathbb{N}_+$. The state-of-the-art mechanisms for state abstraction are NP-complete [29] and require full network state information sharing among SPs. On the other hand, we note that the behavioural couplings in $\mathcal{SG}$ exist in the channel auction. We allow each SP i in $\mathcal{AG}$ to construct $\mathcal{S}_i$ by classifying the value region $[0, \Gamma_i]$[7] of payments into S_i intervals, namely, $[0, \Gamma_{i,1}]$, $(\Gamma_{i,1}, \Gamma_{i,2}]$, $(\Gamma_{i,2}, \Gamma_{i,3}]$, …, $(\Gamma_{i,S_i-1}, \Gamma_{i,S_i}]$, where Γ_i is the maximum payment value and we let $\Gamma_{i,1} = 0$ for a special case in which SP i wins the auction but pays nothing to the CNC[8]. With this regard, a global network state $(\boldsymbol{\chi}_i, \boldsymbol{\chi}_{-i}) \in \mathcal{X}^{|\mathcal{N}|}$, in which SP i receives a payment τ_i within $(\Gamma_{i,s_i-1}, \Gamma_{i,s_i}]$, is conjectured as $\tilde{\boldsymbol{\chi}}_i = (\boldsymbol{\chi}_i, s_i) \in \tilde{\mathcal{X}}_i$, where $\tilde{\mathcal{X}}_i = \mathcal{X}_i \times \mathcal{S}_i$ and $s_i \in \mathcal{S}_i$. To ease the theoretical analysis in the following, we mathematically represent the conjecture by a surjective mapping function $g_i : \mathcal{X}_{-i} \rightarrow \mathcal{S}_i$. Hence $\mathcal{S}_i$ can be treated as an approximation of $\mathcal{X}_{-i}$ but with $S_i \ll |\mathcal{X}_{-i}|$.

Let $\tilde{\boldsymbol{\pi}}_i = (\tilde{\pi}_{i,(\mathrm{c})}, \boldsymbol{\pi}_{i,(\mathrm{t})}, \boldsymbol{\pi}_{i,(\mathrm{p})})$ be the abstract control policy in the abstract stochastic game $\mathcal{AG}$ played by a SP $i \in \mathcal{I}$ over the abstract network state space $\tilde{\mathcal{X}}_i$, where $\tilde{\pi}_{i,(\mathrm{c})}$ is the abstract channel auction policy. In the abstraction from stochastic game $\mathcal{SG}$ to $\mathcal{AG}$ for SP i, we have

$$\begin{aligned}&\Gamma_{i,s_i} - \Gamma_{i,s_i-1} \geq \max_{\{\boldsymbol{\chi}: g_i(\boldsymbol{\chi}_{-i}) = s_i\}}\\&\left| F_i\big(\boldsymbol{\chi}, \boldsymbol{\varphi}_i\big(\boldsymbol{\pi}_{(\mathrm{c})}(\boldsymbol{\chi})\big), \boldsymbol{\pi}_{i,(\mathrm{t})}(\boldsymbol{\chi}_i), \boldsymbol{\pi}_{i,(\mathrm{p})}(\boldsymbol{\chi}_i)\big) - \tilde{F}_i\big(\tilde{\boldsymbol{\chi}}_i, \boldsymbol{\varphi}_i\big(\tilde{\boldsymbol{\pi}}_{(\mathrm{c})}(\tilde{\boldsymbol{\chi}})\big), \boldsymbol{\pi}_{i,(\mathrm{t})}(\boldsymbol{\chi}_i), \boldsymbol{\pi}_{i,(\mathrm{p})}(\boldsymbol{\chi}_i)\big) \right|,\end{aligned} \tag{16}$$

where $\tilde{F}_i(\tilde{\boldsymbol{\chi}}_i, \boldsymbol{\varphi}_i(\tilde{\boldsymbol{\pi}}_{(\mathrm{c})}(\tilde{\boldsymbol{\chi}})), \boldsymbol{\pi}_{i,(\mathrm{t})}(\boldsymbol{\chi}_i), \boldsymbol{\pi}_{i,(\mathrm{p})}(\boldsymbol{\chi}_i))$ is the payoff of SP i in $\tilde{\boldsymbol{\chi}}_i \in \tilde{\mathcal{X}}_i$ under $\tilde{\boldsymbol{\pi}}_i$, $\tilde{\boldsymbol{\chi}} = (\tilde{\boldsymbol{\chi}}_i : i \in \mathcal{I})$, $\tilde{\boldsymbol{\pi}}_{(\mathrm{c})} = (\tilde{\pi}_{i,(\mathrm{c})} : i \in \mathcal{I})$, and $\boldsymbol{\pi}_{(\mathrm{c})}$ is the original joint channel auction policy in $\mathcal{SG}$. Likewise, the abstract state value function for SP i under $\tilde{\boldsymbol{\pi}} = (\tilde{\boldsymbol{\pi}}_i : i \in \mathcal{I})$ can be defined as (17), $\forall \tilde{\boldsymbol{\chi}}_i \in \tilde{\mathcal{X}}_i$,

$$\begin{aligned}&\tilde{V}_i(\tilde{\boldsymbol{\chi}}_i, \tilde{\boldsymbol{\pi}})\\&= (1-\gamma) \cdot \mathsf{E}_{\tilde{\boldsymbol{\pi}}}\left[\sum_{k=1}^{\infty} (\gamma)^{k-1} \cdot \tilde{F}_i\big(\tilde{\boldsymbol{\chi}}_i^k, \boldsymbol{\varphi}_i\big(\tilde{\boldsymbol{\pi}}_{(\mathrm{c})}\big(\tilde{\boldsymbol{\chi}}^k\big)\big), \boldsymbol{\pi}_{i,(\mathrm{t})}\big(\boldsymbol{\chi}_i^k\big), \boldsymbol{\pi}_{i,(\mathrm{p})}\big(\boldsymbol{\chi}_i^k\big)\big) | \tilde{\boldsymbol{\chi}}_i^1 = \tilde{\boldsymbol{\chi}}_i\right],\end{aligned} \tag{17}$$

where $\tilde{\boldsymbol{\chi}}^k = (\tilde{\boldsymbol{\chi}}_i^k = (\boldsymbol{\chi}_i^k, s_i^k) : i \in \mathcal{I})$ with s_i^k being the abstract state at slot k. We will see the expected long-term payoff achieved by SP i from the $\tilde{\boldsymbol{\pi}}_i$ in $\mathcal{AG}$ is not far from that from the original $\boldsymbol{\pi}_i$ in $\mathcal{SG}$. Let $\Upsilon_i = \max_{s_i \in \mathcal{S}_i} (\Gamma_{i,s_i} - \Gamma_{i,s_i-1})$.

Lemma 1: For an original control policy $\boldsymbol{\pi}$ and the corresponding abstract policy $\tilde{\boldsymbol{\pi}}$ in games $\mathcal{SG}$ and $\mathcal{AG}$, we have

$$\left| V_i(\boldsymbol{\chi}, \boldsymbol{\pi}) - \tilde{V}_i(\tilde{\boldsymbol{\chi}}_i, \tilde{\boldsymbol{\pi}}) \right| \leq \Upsilon_i, \forall i \in \mathcal{I}, \tag{18}$$

[7] From the analysis in previous sections, a payment by the SP i to the CNC is of finite value.

[8] With the VCG mechanism, such a case exists when the number of channels in the network is sufficiently large [23].

$\forall \boldsymbol{\chi} \in \mathcal{X}^{|\mathcal{N}|}$, where $\tilde{\boldsymbol{\chi}}_i = (\boldsymbol{\chi}_i, s_i)$ with $s_i = g_i(\boldsymbol{\chi}_{-i})$.

Proof: The proof proceeds similar to [10], [29]. □

Instead of playing the original joint control policy $\boldsymbol{\pi}^*$ in the stochastic game $\mathcal{SG}$, the NE joint abstract control policy given by $\tilde{\boldsymbol{\pi}}^* = (\tilde{\boldsymbol{\pi}}_i^* : i \in \mathcal{I})$ in the abstract stochastic game $\mathcal{AG}$ leads to a bounded regret, where $\tilde{\boldsymbol{\pi}}_i^* = (\tilde{\pi}_{i,(\mathrm{c})}^*, \boldsymbol{\pi}_{i,(\mathrm{t})}^*, \boldsymbol{\pi}_{i,(\mathrm{p})}^*)$ is the best-response abstract control policy of SP $i \in \mathcal{I}$.

Theorem 1. For a SP $i \in \mathcal{I}$, let $\boldsymbol{\pi}_i$ be the original control policy of an abstract control policy $\tilde{\boldsymbol{\pi}}_i$. The original joint control policy $\boldsymbol{\pi}^*$ of a joint abstract control policy $\tilde{\boldsymbol{\pi}}^*$ satisfies

$$V_i\big(\boldsymbol{\chi}, (\boldsymbol{\pi}_i, \boldsymbol{\pi}_{-i}^*)\big) \leq \mathbb{V}_i(\boldsymbol{\chi}) + 2 \cdot \Upsilon_i, \forall \boldsymbol{\chi} \in \mathcal{X}^{|\mathcal{N}|}, \tag{19}$$

where $(\boldsymbol{\pi}_i, \boldsymbol{\pi}_{-i}^*)$ is the joint control policy that results from SP i unilaterally deviating from $\boldsymbol{\pi}_i^*$ to $\boldsymbol{\pi}_i$ in original stochastic game $\mathcal{SG}$.

Proof: The proof follows a contradiction, which assumes that for a SP $i \in \mathcal{I}$, there exists a certain global network state $\boldsymbol{\chi} \in \mathcal{X}^{|\mathcal{N}|}$ such that $V_i(\boldsymbol{\chi}, (\boldsymbol{\pi}_i, \boldsymbol{\pi}_{-i}^*)) > \mathbb{V}_i(\boldsymbol{\chi}) + 2 \cdot \Upsilon_i$, where $\boldsymbol{\pi}_i$ is the original control policy corresponding to a non-best-response abstract control policy $\tilde{\boldsymbol{\pi}}_i$. Using the result from Lemma 1, we arrive at

$$\begin{aligned}
\tilde{V}_i\big(\tilde{\boldsymbol{\chi}}_i, (\tilde{\boldsymbol{\pi}}_i, \tilde{\boldsymbol{\pi}}_{-i}^*)\big) &\geq V_i\big(\boldsymbol{\chi}, (\boldsymbol{\pi}_i, \boldsymbol{\pi}_{-i}^*)\big) - \Upsilon_i \\
&> (\mathbb{V}_i(\boldsymbol{\chi}) + 2 \cdot \Upsilon_i) - \Upsilon_i \\
&\geq \Big(\Big(\tilde{V}_i(\tilde{\boldsymbol{\chi}}_i, \tilde{\boldsymbol{\pi}}^*) - \Upsilon_i\Big) + 2 \cdot \Upsilon_i\Big) - \Upsilon_i = \tilde{V}_i(\tilde{\boldsymbol{\chi}}_i, \tilde{\boldsymbol{\pi}}^*),
\end{aligned} \tag{20}$$

which is contradicted by the definition of a NE in the abstract stochastic game $\mathcal{AG}$. This concludes the proof. □

Hereinafter, we switch our focus from the stochastic game $\mathcal{SG}$ to the abstract stochastic game $\mathcal{AG}$. Suppose all SPs play the NE joint abstract control policy $\tilde{\boldsymbol{\pi}}^*$ in the abstract stochastic game $\mathcal{AG}$. Denote $\tilde{\mathbb{V}}_i(\tilde{\boldsymbol{\chi}}_i) = \tilde{V}_i(\tilde{\boldsymbol{\chi}}_i, \tilde{\boldsymbol{\pi}}^*)$, $\forall \tilde{\boldsymbol{\chi}}_i \in \tilde{\mathcal{X}}_i$ and $\forall i \in \mathcal{I}$. The best-response abstract control policy of a SP i can be computed as

$$\begin{aligned}
\tilde{\mathbb{V}}_i(\tilde{\boldsymbol{\chi}}_i) = \max_{\tilde{\boldsymbol{\pi}}_i(\tilde{\boldsymbol{\chi}}_i)} \Bigg\{ &(1-\gamma)\tilde{F}_i\big(\tilde{\boldsymbol{\chi}}_i, \boldsymbol{\varphi}_i\big(\tilde{\pi}_{i,(\mathrm{c})}(\tilde{\boldsymbol{\chi}}_i), \tilde{\boldsymbol{\pi}}_{-i,(\mathrm{c})}^*(\tilde{\boldsymbol{\chi}}_{-i})\big), \boldsymbol{\pi}_{i,(\mathrm{t})}(\boldsymbol{\chi}_i), \boldsymbol{\pi}_{i,(\mathrm{p})}(\boldsymbol{\chi}_i)\big) \\
&+ \gamma \sum_{\tilde{\boldsymbol{\chi}}_i' \in \tilde{\mathcal{X}}_i} \mathbb{P}\big(\tilde{\boldsymbol{\chi}}_i' | \tilde{\boldsymbol{\chi}}_i, \boldsymbol{\varphi}_i\big(\tilde{\pi}_{i,(\mathrm{c})}(\tilde{\boldsymbol{\chi}}_i), \tilde{\boldsymbol{\pi}}_{-i,(\mathrm{c})}^*(\tilde{\boldsymbol{\chi}}_{-i})\big), \boldsymbol{\pi}_{i,(\mathrm{t})}(\boldsymbol{\chi}_i), \boldsymbol{\pi}_{i,(\mathrm{p})}(\boldsymbol{\chi}_i)\big) \tilde{\mathbb{V}}_i(\tilde{\boldsymbol{\chi}}_i') \Bigg\},
\end{aligned} \tag{21}$$

$\forall \tilde{\boldsymbol{\chi}}_i \in \tilde{\mathcal{X}}_i$, which is based on only the local information.

Remark 3: There remain two challenges involved in solving (21) for each SP $i \in \mathcal{I}$: 1) a priori knowledge of the abstract network state transition probability, which incorporates the statistics of MU mobilities, the computation task and packet arrivals and the conjectures of other competing SPs' local network information (i.e., the statistics of $\mathcal{S}_i$), is not feasible; and 2) given a specific classification of the payment values, 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.

B. Decomposition of Abstract State-Value Function

From two facts: 1) the channel auction decision as well as the computation offloading and packet scheduling decisions are made in sequence and are independent across a SP and its subscribed MUs; and 2) the per-slot instantaneous payoff function (10) of a SP is of an additive nature, we are hence motivated to decompose the per-SP MDP described by (21) into $|\mathcal{N}_i| + 1$ independent single-agent MDPs. More specifically, for a SP $i \in \mathcal{I}$, the abstract state value function $\tilde{\mathbb{V}}_i(\tilde{\boldsymbol{\chi}}_i)$, $\forall \tilde{\boldsymbol{\chi}}_i \in \tilde{\mathcal{X}}_i$, can be calculated as

$$\tilde{\mathbb{V}}_i(\tilde{\boldsymbol{\chi}}_i) = \sum_{n \in \mathcal{N}_i} \alpha_n \cdot \mathbb{U}_n(\boldsymbol{\chi}_n) - \mathbb{U}_i(s_i), \tag{22}$$

where the per-MU expected long-term utility $\mathbb{U}_n$ and the expected long-term payment $\mathbb{U}_i(s_i)$ of SP i satisfy, respectively,

$$\mathbb{U}_n(\boldsymbol{\chi}_n) = \max_{R_{n,(\mathrm{t})}, R_{n,(\mathrm{p})}} \Bigg\{ (1-\gamma) \cdot U_n\big(\boldsymbol{\chi}_n, \varphi_n\big(\tilde{\boldsymbol{\pi}}^*_{(\mathrm{c})}(\tilde{\boldsymbol{\chi}})\big), R_{n,(\mathrm{t})}, R_{n,(\mathrm{p})}\big) + \gamma \cdot \sum_{\boldsymbol{\chi}'_n \in \mathcal{X}} \mathbb{P}\big(\boldsymbol{\chi}'_n | \boldsymbol{\chi}_n, \varphi_n\big(\tilde{\boldsymbol{\pi}}^*_{(\mathrm{c})}(\tilde{\boldsymbol{\chi}})\big), R_{n,(\mathrm{t})}, R_{n,(\mathrm{p})}\big) \cdot \mathbb{U}_n(\boldsymbol{\chi}'_n) \Bigg\}, \tag{23}$$

and

$$\mathbb{U}_i(s_i) = (1-\gamma) \cdot \tau_i + \gamma \cdot \sum_{s'_i \in \mathcal{S}_i} \mathbb{P}\big(s'_i | s_i, \phi_i\big(\tilde{\boldsymbol{\pi}}^*_{(\mathrm{c})}(\tilde{\boldsymbol{\chi}})\big)\big) \cdot \mathbb{U}_i(s'_i), \tag{24}$$

with $\tilde{\boldsymbol{\pi}}^*_{(\mathrm{c})}(\tilde{\boldsymbol{\chi}}) = (\tilde{\pi}^*_{i,(\mathrm{c})}(\tilde{\boldsymbol{\chi}}_i) : i \in \mathcal{I})$ and $R_{n,(\mathrm{t})}$ and $R_{n,(\mathrm{p})}$ being the computation offloading and packet scheduling decisions under a current local network state $\boldsymbol{\chi}_n$ of MU $n \in \mathcal{N}_i$. It is worth to note that the winner determination and payment calculation from the VCG auction outcomes at the CNC leads to (24).

Remark 4: We highlight below two key advantages of the linear decomposition approach in (22).

1) Simplified decision makings: The linear decomposition motivates a SP $i \in \mathcal{I}$ to let the MUs locally make the computation offloading and packet scheduling decisions, which reduces the $(\mathcal{A} \times \mathcal{W})^{|\mathcal{N}_i|}$ at SP i to $|\mathcal{N}_i|$ local spaces $\mathcal{A} \times \mathcal{W}$ at the MUs.
2) Near optimality: The linear decomposition approach, which can be viewed as a special case of the feature-based decomposition method [30], provides an accuracy guarantee of the approximation of the abstract state value function [31].

With the decomposition of the abstract state value function as in (22), we can now define the number of requested channels by a SP $i \in \mathcal{I}$ in the coverage of a BS $b \in \mathcal{B}$ as

$$C_{b,i} = \sum_{\{n \in \mathcal{N}_i : L_n \in \mathcal{L}_b\}} z_n, \tag{25}$$

and the true value of obtaining $\mathbf{C}_i = (C_{b,i} : b \in \mathcal{B})$ across the service region as

$$\nu_i = \frac{1}{1-\gamma} \cdot \sum_{n \in \mathcal{N}_i} \alpha_n \cdot \mathbb{U}_n(\boldsymbol{\chi}_n) - \frac{\gamma}{1-\gamma} \cdot \sum_{s_i' \in \mathcal{S}_i} \mathbb{P}\Big(s_i' | s_i, \mathbb{1}_{\left\{\sum_{b \in \mathcal{B}} C_{b,i} > 0\right\}}\Big) \cdot \mathbb{U}_i(s_i'), \tag{26}$$

which together constitute the optimal bid $\tilde{\pi}^*_{i,(\mathrm{c})}(\tilde{\boldsymbol{\chi}}_i) = \boldsymbol{\beta}_i \triangleq (\nu_i, \mathbf{C}_i)$ of SP i under a current abstract network state $\tilde{\boldsymbol{\chi}}_i \in \tilde{\mathcal{X}}_i$, where z_n given by

$$\begin{aligned} z_n = \arg\max_{z \in \{0,1\}} \Bigg\{ & (1-\gamma) \cdot U_n\big(\boldsymbol{\chi}_n, z, \pi^*_{n,(\mathrm{t})}(\boldsymbol{\chi}_n), \pi^*_{n,(\mathrm{p})}(\boldsymbol{\chi}_n)\big) \\ & + \gamma \cdot \sum_{\boldsymbol{\chi}_n' \in \mathcal{X}} \mathbb{P}\big(\boldsymbol{\chi}_n' | \boldsymbol{\chi}_n, z, \pi^*_{n,(\mathrm{t})}(\boldsymbol{\chi}_n), \pi^*_{n,(\mathrm{p})}(\boldsymbol{\chi}_n)\big) \cdot \mathbb{U}_n(\boldsymbol{\chi}_n') \Bigg\}, \end{aligned} \tag{27}$$

indicates the preference of a MU $n \in \mathcal{N}_i$ between obtaining one channel or not, and $\mathbb{1}_{\{\Xi\}}$ is an indicator function that equals 1 if the condition Ξ is satisfied and 0 otherwise. Note that the calculation of the optimal bid $\boldsymbol{\beta}_i$ at SP i needs the private information of $(\mathbb{U}_n(\boldsymbol{\chi}_n), z_n, L_n)$ from each subscribed MU $n \in \mathcal{N}_i$.

C. Learning Optimal Abstract Control Policy

In the calculation of true value as in (26) for a SP $i \in \mathcal{I}$ at the beginning of each scheduling slot k, the abstract network state transition probability $\mathbb{P}(s'|s, \iota - 1)$ (where s', $s \in \mathcal{S}_i$ and $\iota \in \{1, 2\}$), which is necessary for the prediction of the value of expected future payments, is unknown. We propose that SP i maintains over the scheduling slots a three-dimensional table $\mathbf{Y}_i^k$ of size $S_i \cdot S_i \cdot 2$. Each entry $y^k_{s,s',\iota}$ in table $\mathbf{Y}_i^k$ 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 scheduling slot k. $\mathbf{Y}_i^k$ is updated using the channel auction outcomes from the CNC. Then, the abstract network state transition probability at scheduling slot k can be estimated to be[9]

$$\mathbb{P}\big(s_i^k = s' | s_i^{k-1} = s, \phi_i^{k-1} = \iota - 1\big) = \frac{y_{s,s',\iota}^k}{\sum\limits_{s'' \in \mathcal{S}_i} y_{s'',s',\iota}^k}. \tag{28}$$

Applying the union bound and the weak law of large numbers [32],

$$\lim_{k \to \infty} \mathbb{P}\Big(\big|\mathbb{P}\big(s_i^{k+1} = s' | s_i^k = s, \phi_i^k = \iota - 1\big) - \mathbb{P}\big(s_i^k = s' | s_i^{k-1} = s, \phi_i^{k-1} = \iota - 1\big)\big| > \omega\Big) = 0, \tag{29}$$

establishes, for an arbitrarily small constant $\omega \in \mathbb{R}_+$, $\forall s$, $s' \in \mathcal{S}_i$ and $\forall \iota \in \{1, 2\}$. The state value function $\mathbb{U}_i(s_i)$, $\forall s_i \in \mathcal{S}_i$, is learned according to

$$\mathbb{U}_i^{k+1}(s_i) = \tag{30}$$
$$\begin{cases} \big(1 - \zeta^k\big) \cdot \mathbb{U}_i^k(s_i) + \zeta^k \cdot \left((1 - \gamma) \cdot \tau_i^k + \gamma \cdot \sum\limits_{s_i^{k+1} \in \mathcal{S}_i} \mathbb{P}\big(s_i^{k+1} | s_i, \phi_i^k\big) \cdot \mathbb{U}_i^k\big(s_i^{k+1}\big)\right), & \text{if } s_i = s_i^k; \\ \mathbb{U}_i^k(s_i), & \text{otherwise}, \end{cases}$$

based on ϕ_i^k and τ_i^k from the channel auction, where $\zeta^k \in [0, 1)$ is the learning rate. The convergence of (30) is guaranteed by $\sum_{k=1}^{\infty} \zeta^k = \infty$ and $\sum_{k=1}^{\infty} (\zeta^k)^2 < \infty$ [31].

Given that all SPs deploy the best-response channel auction policies, the well-known value iteration [31] can be used by the MUs to find the optimal per-MU state value functions (23). However, this method requires full knowledge of the local network state transition probabilities, which is challenging without a priori statistical information of MU mobility, computation task arrivals and packet arrivals.

1) Conventional Q-learning: One attractiveness of the Q-learning is that it assumes no a priori knowledge of the local network state transition statistics. Combining (23) and (27), we define for each MU $n \in \mathcal{N}$ the optimal state-action value function $Q_n : \mathcal{X} \times \{0, 1\} \times \mathcal{A} \times \mathcal{W} \to \mathbb{R}$,

$$\begin{aligned} Q_n\big(\boldsymbol{\chi}_n, \varphi_n, R_{n,(\mathrm{t})}, R_{n,(\mathrm{p})}\big) &= (1 - \gamma) \cdot U_n\big(\boldsymbol{\chi}_n, \varphi_n, R_{n,(\mathrm{t})}, R_{n,(\mathrm{p})}\big) \\ &\quad + \gamma \cdot \sum_{\boldsymbol{\chi}'_n \in \mathcal{X}} \mathbb{P}\big(\boldsymbol{\chi}'_n | \boldsymbol{\chi}_n, \varphi_n, R_{n,(\mathrm{t})}, R_{n,(\mathrm{p})}\big) \cdot \mathbb{U}_n(\boldsymbol{\chi}'_n), \end{aligned} \tag{31}$$

[9] To ensure that division by zero is not possible, each entry in a table $\mathbf{Y}_i^1$, $\forall i \in \mathcal{I}$, needs to be initialized, for example, to 1 as in numerical experiments.

where an action $(\varphi_n, R_{n,(\mathrm{t})}, R_{n,(\mathrm{p})})$ under a current local network state $\boldsymbol{\chi}_n$ consists of the channel allocation, computation offloading and packet scheduling decisions. The optimal state value function $\mathbb{U}_n(\boldsymbol{\chi}_n)$ can be hence derived from

$$\mathbb{U}_n(\boldsymbol{\chi}_n) = \max_{\varphi_n, R_{n,(\mathrm{t})}, R_{n,(\mathrm{p})}} Q_n\big(\boldsymbol{\chi}_n, \varphi_n, R_{n,(\mathrm{t})}, R_{n,(\mathrm{p})}\big) . \tag{32}$$

By substituting (32) into (31), we get

$$\begin{aligned} &Q_n\big(\boldsymbol{\chi}_n, \varphi_n, R_{n,(\mathrm{t})}, R_{n,(\mathrm{p})}\big) = (1-\gamma) \cdot U_n\big(\boldsymbol{\chi}_n, \varphi_n, R_{n,(\mathrm{t})}, R_{n,(\mathrm{p})}\big) + \\ &\gamma \cdot \sum_{\boldsymbol{\chi}_n' \in \mathcal{X}} \mathbb{P}\big(\boldsymbol{\chi}_n' | \boldsymbol{\chi}_n, \varphi_n, R_{n,(\mathrm{t})}, R_{n,(\mathrm{p})}\big) \cdot \max_{\varphi_n', R_{n,(\mathrm{t})}', R_{n,(\mathrm{p})}'} Q_n\big(\boldsymbol{\chi}_n', \varphi_n', R_{n,(\mathrm{t})}', R_{n,(\mathrm{p})}'\big) , \end{aligned} \tag{33}$$

where $(\varphi_n', R_{n,(\mathrm{t})}', R_{n,(\mathrm{p})}')$ is an action under $\boldsymbol{\chi}_n'$. Using Q-learning, the MU finds $Q_n(\boldsymbol{\chi}_n, \varphi_n, R_{n,(\mathrm{t})}, R_{n,(\mathrm{p})})$ iteratively using observations of the local network state $\boldsymbol{\chi}_n = \boldsymbol{\chi}_n^k$ at a current scheduling slot k, the action $(\varphi_n, R_{n,(\mathrm{t})}, R_{n,(\mathrm{p})}) = (\varphi_n^k, R_{n,(\mathrm{t})}^k, R_{n,(\mathrm{p})}^k)$, the achieved utility $U_n(\boldsymbol{\chi}_n, \varphi_n, R_{n,(\mathrm{t})}, R_{n,(\mathrm{p})})$ and the resulting local network state $\boldsymbol{\chi}_n' = \boldsymbol{\chi}_n^{k+1}$ at the next slot $k+1$. The learning rule is given by

$$\begin{aligned} &Q_n^{k+1}\big(\boldsymbol{\chi}_n, \varphi_n, R_{n,(\mathrm{t})}, R_{n,(\mathrm{p})}\big) = Q_n^k\big(\boldsymbol{\chi}_n, \varphi_n, R_{n,(\mathrm{t})}, R_{n,(\mathrm{p})}\big) + \\ &\zeta^k \cdot \Bigg((1-\gamma) U_n\big(\boldsymbol{\chi}_n, \varphi_n, R_{n,(\mathrm{t})}, R_{n,(\mathrm{p})}\big) + \gamma \cdot \max_{\varphi_n', R_{n,(\mathrm{t})}', R_{n,(\mathrm{p})}'} Q_n^k\big(\boldsymbol{\chi}_n', \varphi_n', R_{n,(\mathrm{t})}', R_{n,(\mathrm{p})}'\big) - \\ &Q_n^k\big(\boldsymbol{\chi}_n, \varphi_n, R_{n,(\mathrm{t})}, R_{n,(\mathrm{p})}\big)\Bigg), \end{aligned} \tag{34}$$

which converges to the optimal control policy if: a) the local network state transition probability is stationary; and b) all state-action pairs are visited infinitely often [33]. Condition b) can be satisfied when the probability of choosing any action in any local network state is non-zero (i.e., *exploration*). Meanwhile, in order to behave well, a MU has to exploit the most recently learned Q-function (i.e., *exploitation*). A classical way to balance *exploration* and *exploitation* is the ϵ-greedy strategy [31].

Remark 5: The tabular nature in representing Q-function values makes the conventional Q-learning not readily applicable to high-dimensional scenarios with huge state space, where the learning process can be extremely slow. In our system, the sizes of local network state space $\mathcal{X}$ and action space $\{0,1\} \times \mathcal{A} \times \mathcal{W}$ are calculated as $|\mathcal{L}| \cdot (1 + A_{(\mathrm{t})}^{(\max)}) \cdot (1 + W^{(\max)})$ and $2 \cdot (1 + A_{(\mathrm{t})}^{(\max)}) \cdot (1 + W^{(\max)})$. Consider a service region with $1.6 \cdot 10^3$ locations (as in [20] and numerical experiments), $A_{(\mathrm{t})}^{(\max)} = 5$ and $W^{(\max)} = 10$, the MU has to update totally $1.39392 \cdot 10^7$ Q-function values, which is impossible for the conventional Q-learning process to converge within limited number of scheduling slots.

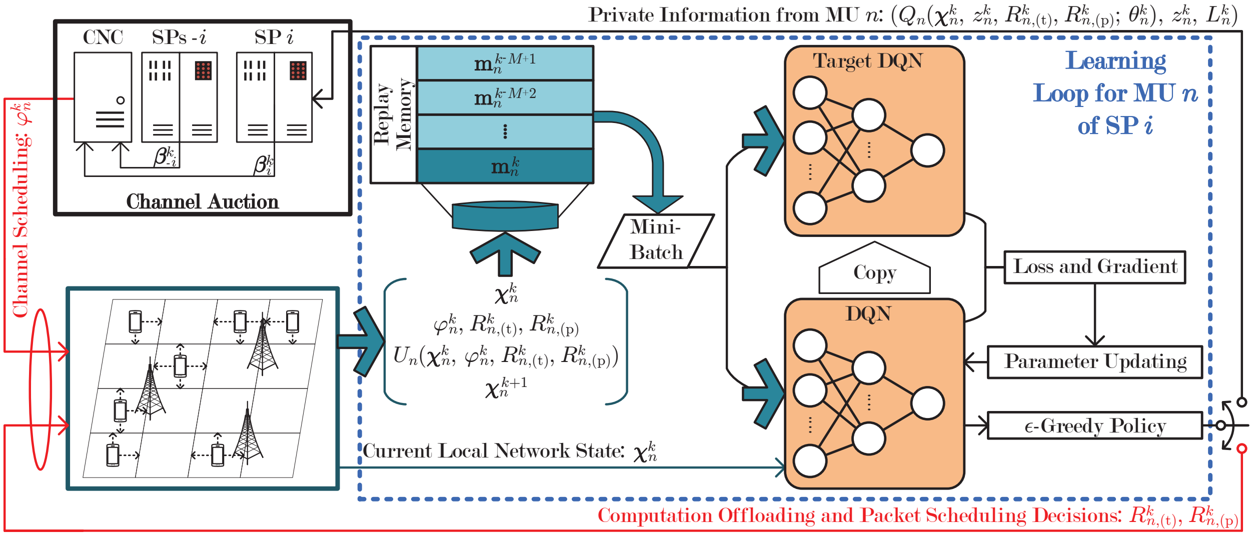
Private Information from MU n: $(Q_n(\chi_n^k, z_n^k, R_{n,(\mathrm{t})}^k, R_{n,(\mathrm{p})}^k; \theta_n^k), z_n^k, L_n^k)$
CNC
SPs -i
SP i
β_{-i}^k
β_i^k
Channel Auction
Channel Scheduling: φ_n^k
Replay Memory
$\mathbf{m}_n^{k-M+1}$
$\mathbf{m}_n^{k-M+2}$
$\vdots$
$\mathbf{m}_n^k$
Mini-Batch
Target DQN
Learning Loop for MU n of SP i
Copy
Loss and Gradient
DQN
Parameter Updating
ϵ-Greedy Policy
χ_n^k
φ_n^k, $R_{n,(\mathrm{t})}^k$, $R_{n,(\mathrm{p})}^k$
$U_n(\chi_n^k, \varphi_n^k, R_{n,(\mathrm{t})}^k, R_{n,(\mathrm{p})}^k)$
χ_n^{k+1}
Current Local Network State: χ_n^k
Computation Offloading and Packet Scheduling Decisions: $R_{n,(\mathrm{t})}^k$, $R_{n,(\mathrm{p})}^k$

Fig. 2. Application of a double deep Q-network (DQN) to approximate the optimal state-action value Q-function of a mobile user (MU) $n \in \mathcal{N}_i$ subscribed to service provider (SP) $i \in \mathcal{I}$ (CNC: centralized network controller.).

2) *Deep Reinforcement Learning:* The advances in neural networks [34] and the success of a deep neural network in modelling an optimal state-action Q-function [35] inspire us to resort to a double deep Q-network (DQN) to address the massive local network state space $\mathcal{X}$ at each MU $n \in \mathcal{N}$ in the network [17]. That is, the Q-function in (33) can be approximated by $Q_n(\boldsymbol{\chi}_n, \varphi_n, R_{n,(\mathrm{t})}, R_{n,(\mathrm{p})}) \approx Q_n(\boldsymbol{\chi}_n, \varphi_n, R_{n,(\mathrm{t})}, R_{n,(\mathrm{p})}; \boldsymbol{\theta}_n)$, where $\boldsymbol{\theta}_n$ denotes a vector of parameters associated with the DQN of MU n. The implementation of such a DRL algorithm for finding the approximated Q-function of MU n is illustrated in Fig. 2.

More specifically, each MU $n \in \mathcal{N}$ is equipped with a replay memory of a finite size M to store the experience $\mathbf{m}_n^k = (\boldsymbol{\chi}_n^k, (\varphi_n^k, R_{n,(\mathrm{t})}^k, R_{n,(\mathrm{p})}^k), U_n(\boldsymbol{\chi}_n^k, \varphi_n^k, R_{n,(\mathrm{t})}^k, R_{n,(\mathrm{p})}^k), \boldsymbol{\chi}_n^{k+1})$, which is happened at the transition of two consecutive scheduling slots k and $k+1$, during the process of DRL. The memory of experiences can be represented as $\mathcal{M}_n^k = \{\mathbf{m}_n^{k-M+1}, \cdots, \mathbf{m}_n^k\}$. Each MU n maintains a DQN as well as a target DQN, namely, $Q_n(\boldsymbol{\chi}_n, \varphi_n, R_{n,(\mathrm{t})}, R_{n,(\mathrm{p})}; \boldsymbol{\theta}_n^k)$ and $Q_n(\boldsymbol{\chi}_n, \varphi_n, R_{n,(\mathrm{t})}, R_{n,(\mathrm{p})}; \boldsymbol{\theta}_{n,-}^k)$, with $\boldsymbol{\theta}_n^k$ and $\boldsymbol{\theta}_{n,-}^k$ being the associated parameters at a current scheduling slot k and a certain previous scheduling slot before slot k, respectively. According to the experience replay technique [36], at each slot k, MU n randomly samples a mini-batch $\mathcal{O}_n^k \subseteq \mathcal{M}_n^k$ of size $O < M$ from $\mathcal{M}_n^k$ to train the DQN. The training objective is to update the parameters $\boldsymbol{\theta}_n^k$ in the direction of minimizing the loss function given by

$$\begin{aligned}
&\mathsf{LOSS}_n\big(\boldsymbol{\theta}_n^k\big) = \qquad (35)\\
&\mathsf{E}_{\left(\boldsymbol{\chi}_n,(\varphi_n,R_{n,(\mathrm{t})},R_{n,(\mathrm{p})}),U_n(\boldsymbol{\chi}_n,\varphi_n,R_{n,(\mathrm{t})},R_{n,(\mathrm{p})}),\boldsymbol{\chi}'_n\right)\in\mathcal{O}_n^k}\Bigg[\Bigg((1-\gamma)\cdot U_n\big(\boldsymbol{\chi}_n,\varphi_n,R_{n,(\mathrm{t})},R_{n,(\mathrm{p})}\big)\ +\gamma\cdot\\
&Q_n\Bigg(\boldsymbol{\chi}'_n, \underset{\varphi'_n,R'_{n,(\mathrm{t})},R'_{n,(\mathrm{p})}}{\arg\max}\ Q_n\big(\boldsymbol{\chi}'_n,\varphi'_n,R'_{n,(\mathrm{t})},R'_{n,(\mathrm{p})};\boldsymbol{\theta}_n^k\big);\boldsymbol{\theta}_{n,-}^k\Bigg) - Q_n\big(\boldsymbol{\chi}_n,\varphi_n,R_{n,(\mathrm{t})},R_{n,(\mathrm{p})};\boldsymbol{\theta}_n^k\big)\Bigg)^2\Bigg],
\end{aligned}$$

which is a mean-squared measure of the Bellman equation error at a scheduling slot k. By differentiating $\mathsf{LOSS}_n(\boldsymbol{\theta}_n^k)$ with respect to $\boldsymbol{\theta}_n^k$, we obtain the gradient as

$$\begin{aligned}
&\nabla_{\boldsymbol{\theta}_n^k}\mathsf{LOSS}_n\big(\boldsymbol{\theta}_n^k\big) =\\
&\mathsf{E}_{\left(\boldsymbol{\chi}_n,(\varphi_n,R_{n,(\mathrm{t})},R_{n,(\mathrm{p})}),U_n(\boldsymbol{\chi}_n,\varphi_n,R_{n,(\mathrm{t})},R_{n,(\mathrm{p})}),\boldsymbol{\chi}'_n\right)\in\mathcal{O}_n^k}\Bigg[\Bigg((1-\gamma)\cdot U_n\big(\boldsymbol{\chi}_n,\varphi_n,R_{n,(\mathrm{t})},R_{n,(\mathrm{p})}\big) +\\
&\gamma\cdot Q_n\Bigg(\boldsymbol{\chi}'_n, \underset{\varphi'_n,R'_{n,(\mathrm{t})},R'_{n,(\mathrm{p})}}{\arg\max}\ Q_n\big(\boldsymbol{\chi}'_n,\varphi'_n,R'_{n,(\mathrm{t})},R'_{n,(\mathrm{p})};\boldsymbol{\theta}_n^k\big);\boldsymbol{\theta}_{n,-}^k\Bigg) -\\
&Q_n\big(\boldsymbol{\chi}_n,\varphi_n,R_{n,(\mathrm{t})},R_{n,(\mathrm{p})};\boldsymbol{\theta}_n^k\big)\Bigg)\cdot\nabla_{\boldsymbol{\theta}_n^k}Q_n\big(\boldsymbol{\chi}_n,\varphi_n,R_{n,(\mathrm{t})},R_{n,(\mathrm{p})};\boldsymbol{\theta}_n^k\big)\Bigg]. \qquad (36)
\end{aligned}$$

Algorithm 1 details the online training procedure of MU n.

V. Numerical Experiments

In order to quantify the performance gain from the proposed DRL-based online learning scheme for multi-tenant cross-slice resource orchestration in a software-defined RAN, numerical experiments based on TensorFlow [18] are conducted.

A. Parameter Settings

For experimental purpose, we build up a RAN, which is composed of 4 BSs in a 2×2 Km^2 square area. Fig. 3 shows the layout of the RAN. The BSs are placed at equal distance apart. The entire service region is divided into 1600 locations with each representing a small area of

Algorithm 1 Online DRL for Approximating Optimal State-Action Value Q-Functions of a MU $n \in \mathcal{N}_i$ of a SP $i \in \mathcal{I}$

```
initialize the replay memory $\mathcal{M}_n^k$ of size $M \in \mathbb{N}_+$, the mini-batch $\mathcal{O}_n^k$ of size $O < M$, a DQN and a target DQN with two sets $\boldsymbol{\theta}_n^k$ and $\boldsymbol{\theta}_{n,-}^k$ of parameters, and the local network state $\boldsymbol{\chi}_n^k$, for $k = 1$.
repeat
    At the beginning of scheduling slot $k$, the MU observes the packet arrivals $A_{n,(\mathrm{p})}^k$, takes $\boldsymbol{\chi}_n^k$ as an input to the DQN with parameters $\boldsymbol{\theta}_n^k$, and then selects a random action $(z_n^k, R_{n,(\mathrm{t})}^k, R_{n,(\mathrm{p})}^k)$ with probability $\epsilon$ or with probability $1 - \epsilon$, an action $(z_n^k, R_{n,(\mathrm{t})}^k, R_{n,(\mathrm{p})}^k)$ that is with maximum value $Q_n(\boldsymbol{\chi}_n^k, z_n^k, R_{n,(\mathrm{t})}^k, R_{n,(\mathrm{p})}^k; \boldsymbol{\theta}_n^k)$.
    MU $n$ sends $[Q_n(\boldsymbol{\chi}_n^k, z_n^k, R_{n,(\mathrm{t})}^k, R_{n,(\mathrm{p})}^k; \boldsymbol{\theta}_n^k), z_n^k, L_n^k)$ to the subscribing SP $i$. SP $i$ submits its bidding vector $\boldsymbol{\beta}_i = (\nu_i, \mathbf{C}_i)$ to the CNC, where $\nu_i$ is given by (26) and $\mathbf{C}_i = (C_{b,i} : b \in \mathcal{B})$ with each $C_{b,i}$ given by (25).
    With bids from all SPs, the CNC determines the auction winners $\boldsymbol{\phi}^k$ and channel allocation $\boldsymbol{\rho}_i^k = (\boldsymbol{\rho}_n^k : n \in \mathcal{N}_i)$ according to (5), and calculates the payments $\tau_i^k$ according to (6) for SP $i$.
    With the channel allocation $\boldsymbol{\rho}_n^k$, winner determination $\phi_i^k$ and payment $\tau_i^k$, SP $i$ updates $\mathbf{Y}_i^k$ and $\mathbb{U}_i^{k+1}(s_i^k)$ according to (30), and MU $n$ makes computation offloading $\varphi_n^k R_{n,(\mathrm{t})}^k$ and packet scheduling $\varphi_n^k R_{n,(\mathrm{p})}^k$.
    MU $n$ achieves utility $U_n(\boldsymbol{\chi}_n^k, \varphi_n^k, R_{n,(\mathrm{t})}^k, R_{n,(\mathrm{p})}^k)$ and observes $\boldsymbol{\chi}_n^{k+1}$ at the next slot $k+1$.
    MU $n$ updates the $\mathcal{M}_n^k$ with $\mathbf{m}_n^k = (\boldsymbol{\chi}_n^k, (\varphi_n^k, R_{n,(\mathrm{t})}^k, R_{n,(\mathrm{p})}^k), U_n(\boldsymbol{\chi}_n^k, \varphi_n^k, R_{n,(\mathrm{t})}^k, R_{n,(\mathrm{p})}^k), \boldsymbol{\chi}_n^{k+1})$.
    With a randomly sampled $\mathcal{O}_n^k$ from $\mathcal{M}_n^k$, MU $n$ updates the DQN parameters $\boldsymbol{\theta}_n^k$ with the gradient in (36).
    MU $n$ regularly reset the target DQN parameters with $\boldsymbol{\theta}_{n,-}^{k+1} = \boldsymbol{\theta}_n^k$, and otherwise $\boldsymbol{\theta}_{n,-}^{k+1} = \boldsymbol{\theta}_{n,-}^k$.
    The scheduling slot index is updated by $k \leftarrow k + 1$.
until A predefined stopping condition is satisfied.
```

50×50 m^2. In other words, each BS covers 400 locations[10]. The channel gain experienced by a

[10] The numerical experiments can be readily extended to more practical network layouts, such as [37].

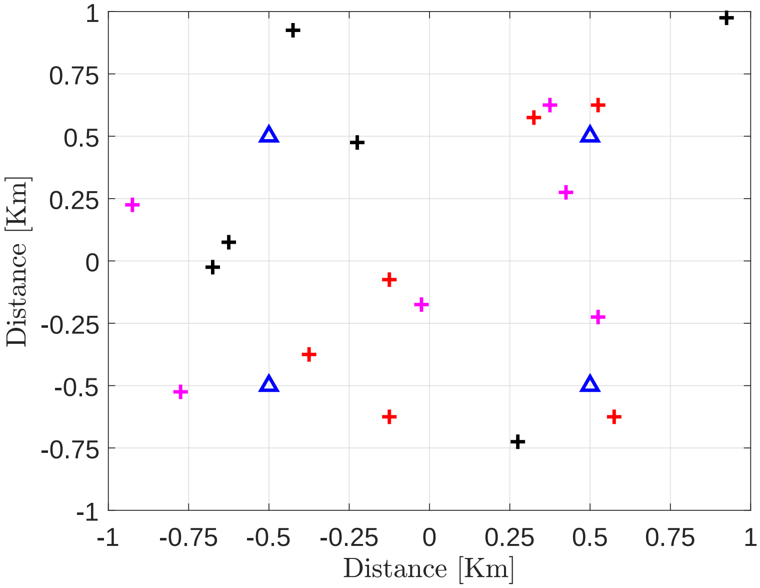

Fig. 3. A snapshot of the RAN: BSs, MUs of SPs 1, 2 and 3 are shown in blue triangles, red, black and magenta pluses.

MU $n \in \mathcal{N}$ in the coverage of a BS $b \in \mathcal{B}$ at location $L_n^k \in \mathcal{L}_b$ during a slot k is given as

$$h(L_n^k) = H_0 \cdot \left(\frac{\xi_0}{\xi_{b,n}^k}\right)^4, \tag{37}$$

where $H_0 = -40$ dB is the path-loss constant, $\xi_0 = 2$ m is the reference distance and $\xi_{b,n}^k$ is the physical distance between MU n and BS b. The state transition probability matrices for the Markov processes of mobilities and computation task arrivals of all MUs are independently and randomly generated. The packet arrivals follow a Poisson arrival process with average rate λ (in packets/slot). $U_n^{(1)}(\cdot)$, $U_n^{(2)}(\cdot)$, $U_n^{(3)}(\cdot)$ and $U_n^{(4)}(\cdot)$ in (11) are chosen to be

$$U_n^{(1)}\left(W_n^{k+1}\right) = \exp\left\{-W_n^{k+1}\right\}, \tag{38}$$

$$U_n^{(2)}\left(D_n^{k}\right) = \exp\left\{-D_n^{k}\right\}, \tag{39}$$

$$U_n^{(3)}\left(P_{n,(\mathrm{tr})}^{k}\right) = \exp\left\{-P_{n,(\mathrm{tr})}^{k}\right\}, \tag{40}$$

$$U_n^{(4)}\left(P_{n,(\mathrm{CPU})}^{k}\right) = \exp\left\{-P_{n,(\mathrm{CPU})}^{k}\right\}. \tag{41}$$

For a MU, we design a DQN with 2 hidden layers with each consisting of 16 neurons[11]. Tanh is selected as the activation function [39] and Adam as the optimizer [40]. With the consideration

[11] The tradeoff between the time spent during the training process and a performance improvement with a deeper and/or wider neural network is still an open problem [38].

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 $\lvert\mathcal{N}_i\rvert$	6, $\forall i \in \mathcal{I}$
Channel bandwidth η	500 KHz
Noise power spectral density σ^2	-174 dBm/Hz
Scheduling slot duration δ	10^{-2} second
Discount factor γ	0.9
Utility price α_n	1, $\forall n \in \mathcal{N}$
Packet size $\mu_{(\mathrm{p})}$	3000 bits
Maximum transmit power $P^{(\max)}$	3 Watts
Weight of transmit power ℓ_n	3, $\forall n \in \mathcal{N}$
Maximum queue length $W^{(\max)}$	10 packets
Maximum task arrivals $A_{(\mathrm{t})}^{(\max)}$	5 tasks
Input data size $\mu_{(\mathrm{t})}$	5000 bits
CPU cycles per bit ϑ	737.5
CPU-cycle frequency ϱ	2 GHz
Effective switched capacitance ς	$2.5 \cdot 10^{-28}$
Exploration probability ϵ	0.001

of limited memory capacity of mobile devices, we set the replay size as $M = 5000$. Other parameter values used in the experiments are listed in Table I.

For performance comparisons, three baseline schemes are simulated, namely,

1) Channel-aware control policy (Baseline 1) – At the beginning of each slot, the need of getting one channel at a MU is evaluated by the average channel gain;
2) Queue-aware control policy (Baseline 2) – Each MU calculates the preference between having one channel or not using a predefined threshold of the data queue length;
3) Random control policy (Baseline 3) – This policy randomly generates the value of obtaining one channel for each MU at each scheduling slot.

During the implementation of the above three baselines, after the centralized channel allocation at the CNC, each MU proceeds to select a random number of computation tasks for offloading and decides a maximum feasible number of packets for transmission.

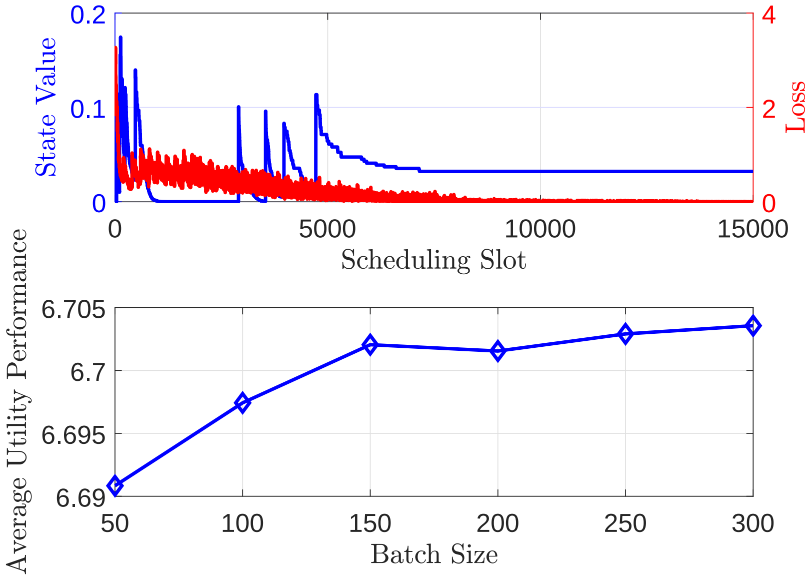

Fig. 4. Illustration for convergence speed of the proposed online resource orchestration scheme based on DRL (upper) and average utility performance per MU across the learning procedure versus batch sizes (lower).

B. Experiment Results

1) Experiment 1 – Convergence performance: Our goal in this experiment is to validate if the considered software-defined RAN remains stable by implementing our proposed online learning scheme for multi-tenant cross-slice resource orchestration. We fix the average packet arrival rate and the number of channels to be $\lambda = 6$ packets per slot and $J = 9$, respectively. The batch size is set as $O = 200$. In all experiments, we use $S_i = 36$, $\forall i \in \{1, 2, 3\}$. Without loss of the generality, we plot the variations in $\mathbb{U}_1(2)$ of SP 1 and the loss function $\mathsf{LOSS}_1(\boldsymbol{\theta}_1^k)$ of a MU $1 \in \mathcal{N}_1 = \{1, 2, \cdots, 6\}$ versus the scheduling slots in the upper subplot in Fig. 4, which validates the convergence behaviour of our scheme. The learning scheme converges within 10^4 scheduling slots. In the lower subplot in Fig. 4, we plot the average utility performance per MU with different choices of batch size. It is obvious from (36) that for a MU, a larger batch size results in a more stable gradient estimate, i.e., a smaller variance, hence a better average utility performance across the learning procedure. Given the replay memory capacity, the utility performance improvement, however, saturates, when the batch size exceeds 200. Hence we continue to use $O = 200$ in Experiments 2 and 3 to strike a balance between performance improvement and computation overhead.

2) Experiment 2 – Performance under various λ: This experiment primarily aims to demonstrate the average performance per scheduling slot in terms of the average queue length, the average packet drops, the average transmit energy consumption, the average CPU energy consumption and the average utility under different packet arrival rates. We choose the number of channels that can be used across the MUs in the network as $J = 11$. The results are exhibited in Figs. 5, 6 and 7. Fig. 5 illustrates the average queue length and the average packet drops per MU. Fig. 6 illustrates the average transmit energy consumption and the average CPU energy consumption per MU. Fig. 7 illustrates the average utility per MU.

Each plot compares the performance of the proposed scheme to the three baseline multi-tenant cross-slice resource orchestration schemes. From Fig. 7, it can be observed that the proposed scheme achieves a significant gain in average utility per MU. Similar observations can be made from the curves in Fig. 5, which shows that minimum queue length and packet drops can be realized from the proposed scheme. As the packet arrival rate increases, each MU consumes more transmit energy for the delivery of incoming data packets. However, implementing Baselines 1 and 3, the average CPU energy consumption per MU keeps constant due to the fact that the opportunities of winning the channel auction do not change. On the other hand, the average CPU energy consumption per MU from Baseline 2 decreases since a larger queue length indicates a bigger chance of getting one channel and hence a higher probability of offloading the computation task. In contrast to Baseline 2, the proposed scheme transmits more data packets to avoid packet drops by leaving more computation tasks processed at the mobile devices, leading to increased average CPU energy consumption.

3) Experiment 3 – Performance with different J: In the last experiment, we simulate the average resource orchestration performance per scheduling slot achieved from the proposed scheme and other three baselines versus the numbers of channels. The packet arrival rate in this experiment is selected as $\lambda = 8$. The average queue length, average packet drops, average transmit energy consumption, average CPU energy consumption and average utility per MU across the entire learning period are depicted in Figs. 8, 9 and 10. It can be easily observed from Figs. 8 and 10 that as the number of available channels increases, the average queue length and the average packet drops decrease, while the average utility per MU improves. As the number of channels that can be allocated to the MUs increases, it becomes more likely for a MU to obtain one channel. Therefore, the MU is able to offload more computation tasks and transmit more data packets, and at the same time, the average transmit energy consumption surely

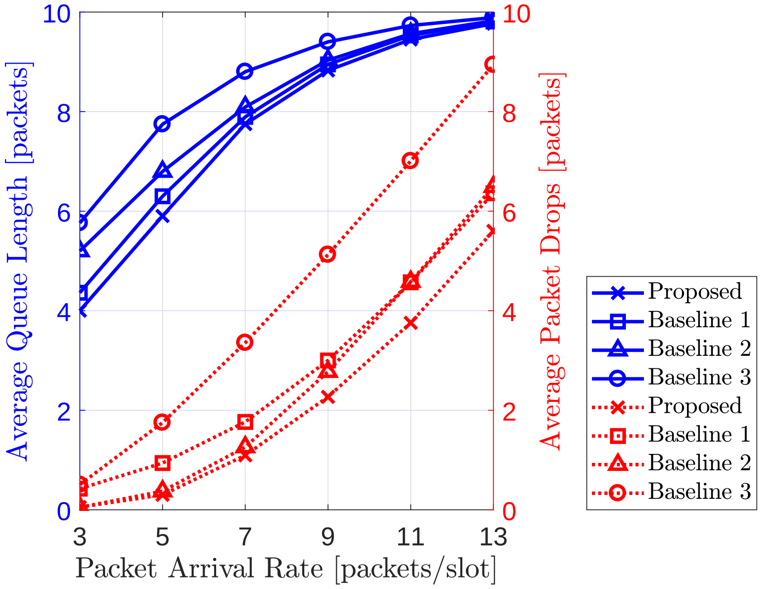

Fig. 5. Average queue length and packet drops per MU across the learning procedure versus average packet arrival rates.

increases, as shown in Fig. 9. From both Experiments 2 and 3, the proposed online learning scheme outperforms the three baselines.

VI. Conclusions

We study in this paper the multi-tenant cross-slice resource orchestration in a software-defined RAN. Over the scheduling slots, the SPs bid to orchestrate the limited channel access opportunities over their MUs with MEC and traditional cellular service requests in the slices. The CNC regulates the channel auction through a VCG pricing mechanism at the beginning of each slot. We formulate the non-cooperative problem as a stochastic game, in which each SP aims to maximize its own expected long-term payoff and the channel auction, computation offloading and packet scheduling decisions of a SP require complete information of the network dynamics as well as the control policies of other SPs. To solve the problem, we approximate the interactions among the competing SPs by an abstract stochastic game. In the abstract stochastic game, a SP is thus able to behave independently with the conjectures of other SPs' behaviours. We observe that the channel auction decision and the computation offloading and packet scheduling decisions are sequentially made. This motivates us to linearly decompose the per-SP MDP, which simplifies the decision making process of a SP. An online scheme based on DRL is proposed to find the

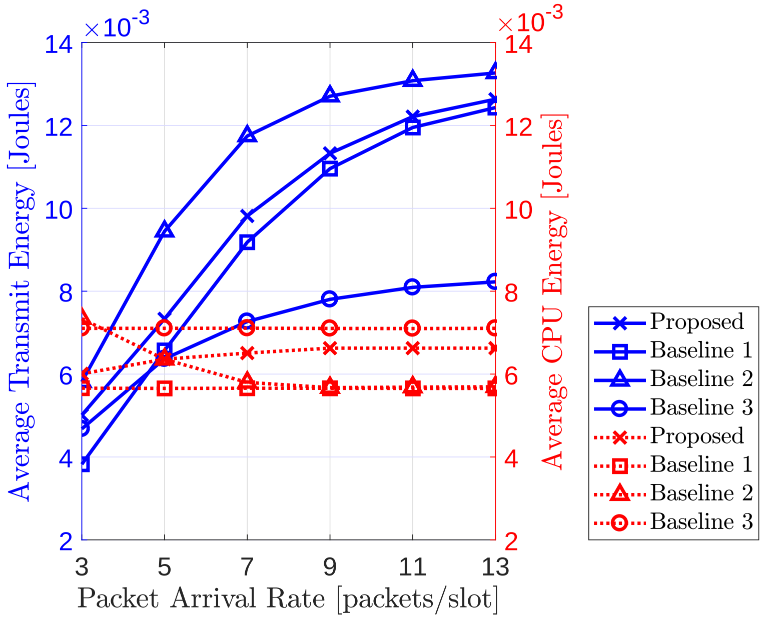

Fig. 6. Average transmit energy and CPU energy consumptions per MU across the learning procedure versus average packet arrival rates.

optimal abstract control policies. Numerical experiments showcase that significant performance gains can be achieved from our scheme.

References

[1] "Cisco visual networking index: Global mobile data traffic forecast update, 2016–2021," White Paper, Cisco, Feb. 2017.

[2] J. G. Andrews, H. Claussen, M. Dohler, S. Rangan, and M. C. Reed, "Femtocells: Past, present, and future," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 3, pp. 497–508, Apr. 2012.

[3] A. Gudipati, D. Perry, L. E. Li, and S. Katti, "SoftRAN: Software defined radio access network," in *ACM SIGCOMM HotSDN Workshop*, Hong Kong, China, Aug. 2013.

[4] T. Chen, H. Zhang, X. Chen, and O. Tirkkonen, "SoftMobile: Control evolution for future heterogeneous mobile networks," *IEEE Wireless Commun.*, vol. 21, no. 6, pp. 70–78, Dec. 2014.

[5] T. Chen, M. Matinmikko, X. Chen, X. Zhou, and P. Ahokangas, "Software defined mobile networks: Concept, survey and research directions," *IEEE Commun. Mag.*, vol. 53, no. 11, pp. 126–133, Nov. 2015.

[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] Google, "Project Fi," https://fi.google.com [Date Accessed: 12 Jul. 2018].

[8] V. Petrov, M. A. Lema, M. Gapeyenko, K. Antonakoglou, D. Moltchanov, F. Sardis, A. Samuylov, S. Andreev, Y. Koucheryavy, and M. Dohler, "Achieving end-to-end reliability of mission-critical traffic in softwarized 5G networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 485–501, Mar. 2018.

[9] P. Caballero, A. Banchs, G. de Veciana, and X. Costa-Pérez, "Network slicing games: Enabling customization in multi-tenant networks," in *Proc. IEEE INFOCOM*, Atlanta, GA, May 2017.

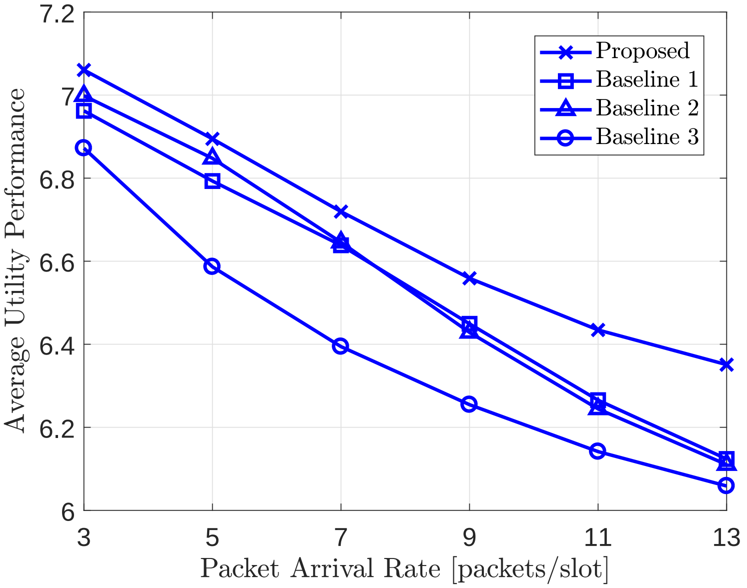

Fig. 7. Average utility performance per MU across the learning procedure versus average packet arrival rates.

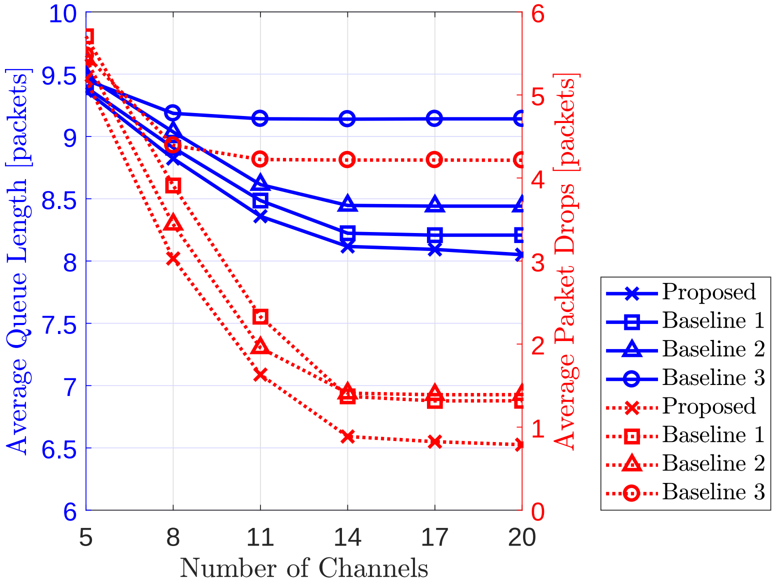

Fig. 8. Average queue length and packet drops per MU across the learning procedure versus numbers of channels.

[10] X. Chen, Z. Han, H. Zhang, G. Xue, Y. Xiao, and M. Bennis, "Wireless resource scheduling in virtualized radio access networks using stochastic learning," *IEEE Trans. Mobile Comput.*, vol. 17, no. 4, pp. 961–974, Apr. 2018.

[11] M. Satyanarayanan, "The emergence of edge computing," *IEEE Comput.*, vol. 50, no. 1, pp. 30–39, Jan. 2017.

[12] Y. Mao, C. You, J. Zhang, K. Huang and K. B. Letaief, "A Survey on mobile edge computing: The communication

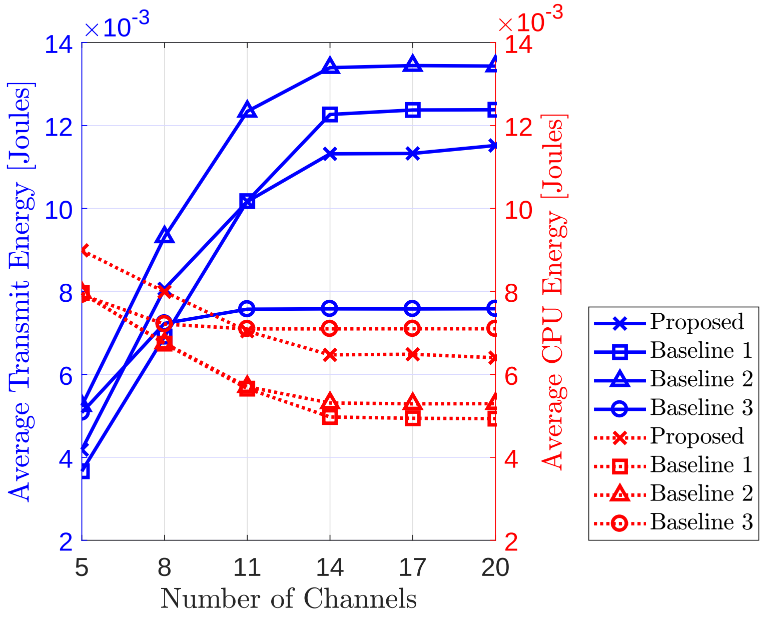

Fig. 9. Average transmit energy and CPU energy consumptions per MU across the learning procedure versus numbers of channels.

perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, Q4 2017.

[13] D. T. Hoang, D. Niyato, P. Wang, A. D. Domenico, and E. C. Strinati, "Optimal cross slice orchestration for 5G mobile services," in *Proc. IEEE VTC*, Chicago, IL, Aug. 2018.

[14] P. Zhao, H. Tian, S. Fan, and A. Paulraj, "Information prediction and dynamic programming based RAN slicing for mobile edge computing," *IEEE Wireless Commun. Lett.*, Early Access Article, 2018.

[15] Y. Zhou, F. R. Yu, J. Chen, and Y. Kuo, "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.

[16] 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.

[17] H. van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proc. AAAI*, Phoenix, AZ, Feb. 2016.

[18] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: A system for large-scale machine learning," in *Proc. OSDI*, Savannah, GA, Nov. 2016.

[19] C. Ho, D. Yuan, and S. Sun, "Data offloading in load coupled networks: A utility maximization framework," *IEEE Trans. Wireless Commun.*, vol. 13, no. 4, pp. 1921–1931, Apr. 2014.

[20] X. Chen, J. Wu, Y. Cai, H. Zhang, and T. Chen, "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.

[21] A. J. Nicholson and B. D. Noble, "BreadCrumbs: Forecasting mobile connectivity," in *Proc. ACM MobiCom*, San Francisco,

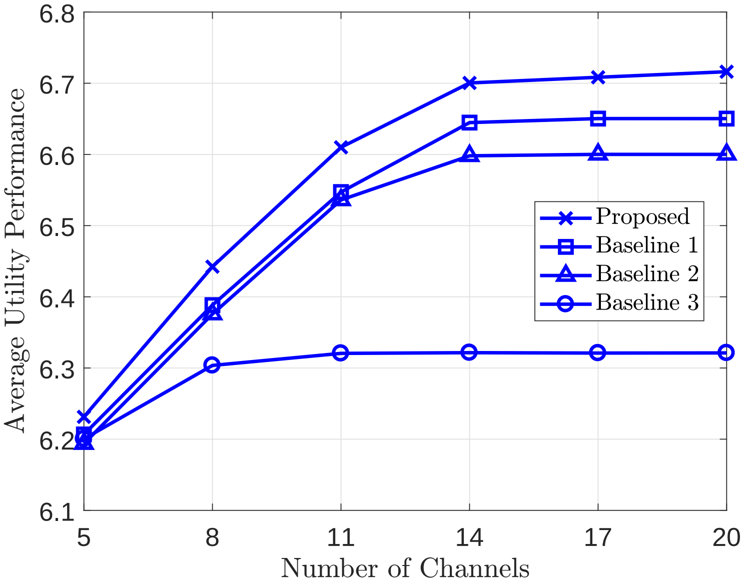

Fig. 10. Average utility performance per MU across the learning procedure versus numbers of channels.

CA, Sep. 2008.

[22] M. H. Cheung and J. Huang, “DAWN: Delay-aware Wi-Fi offloading and network selection,” *IEEE J. Sel. Areas Commun.*, vol. 33, no. 6, pp. 1214–1223, Jun. 2015.

[23] J. Jia, Q. Zhang, Q. Zhang, and M. Liu, “Revenue generation for truthful spectrum auction in dynamic spectrum access,” in *Proc. ACM MobiHoc*, New Orleans, LA, May 2009.

[24] X. He, J. Liu, R. Jin, and H. Dai, “Privacy-aware offloading in mobile-edge computing,” in *Proc. IEEE GLOBECOM*, Singapore, Dec. 2017.

[25] R. A. Berry and R. G. Gallager, “Communication over fading channels with delay constraints,” *IEEE Trans. Inf. Theory*, vol. 48, no. 5, pp. 1135–1149, May 2002.

[26] 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.

[27] A. M. Fink, “Equilibrium in a stochastic n-person game,” *J. Sci. Hiroshima Univ. Ser. A-I*, vol. 28, pp. 89–93, 1964.

[28] C. Kroer and T. Sandholm, “Imperfect-recall abstractions with bounds in games,” in *Proc. ACM EC*, Maastricht, the Netherlands, Jul. 2016.

[29] D. Abel, D. Hershkowitz, and M. Littman, “Near optimal behavior via approximate state abstraction,” in *Proc. ICML*, New York, NY, Jun. 2016.

[30] J. N. Tsitsiklis and B. van Roy, “Feature-based methods for large scale dynamic programming,” *Mach. Learn.*, vol. 22, no. 1-3, pp. 59–94, Jan. 1996.

[31] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.

[32] M. Loève, *Probability Theory I*. Berlin, Germany: Springer-Verlag, 1977.

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

[34] Apple, “The future is here: iPhone X,” https://www.apple.com/newsroom/2017/09/the-future-is-here-iphone-x/ [Date

Accessed: 16 Jul. 2018].

[35] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.

[36] L.-J. Lin, "Reinforcement learning for robots using neural networks," Carnegie Mellon University, 1992.

[37] I. Siomina and D. Yuan, "Analysis of Cell Load Coupling for LTE Network Planning and Optimization," *IEEE Trans. Wireless Commun.*, vol. 11, no. 6, pp. 2287–2297, Jun. 2012.

[38] C. L. P. Chen and Z. Liu, "Broad learning system: An effective and efficient incremental learning system without the need for deep architecture," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 1, pp. 10–24, Jan. 2018.

[39] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?" in *Proc. IEEE ICCV*, Kyoto, Japan, Sep.–Oct. 2009.

[40] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *Proc. ICLR*, San Diego, CA, May 2015.