Age of Information-Aware Multi-Tenant Resource Orchestration in Network Slicing

Xianfu Chen*, Celimuge Wu†, Tao Chen*, Nan Wu‡, Honggang Zhang§, and Yusheng Ji¶
*VTT Technical Research Centre of Finland Ltd, Finland
†Graduate School of Informatics and Engineering, University of Electro-Communications, Tokyo, Japan
‡Beijing Institute of Technology, Beijing, China
§College of Information Science and Electronic Engineering, Zhejiang University, China
¶Information Systems Architecture Research Division, National Institute of Informatics, Tokyo, Japan

***Abstract*—To satisfy diverse services from mobile users (MUs) over a common network infrastructure, network slicing is envisioned as a promising technology. This paper considers radio access network (RAN)-only slicing, where the physical RAN is judiciously tailored to accommodate computation and communication functionalities. Multiple service providers (SPs, a.k.a., tenants) compete for a limited number of channels across the discrete scheduling slots in order to serve their respective subscribed MUs. From a MU perspective, the age of information of data packets from traditional mobile services and the energy consumption at mobile device are of practical importance. We characterize the interactions among the SPs via a stochastic game, in which a SP selfishly maximizes its own expected long-term payoff. To approximate the Nash equilibrium solutions, we build an abstract stochastic game exploring the local information of SPs. Furthermore, the decision-making process at a SP can be much simplified by linearly decomposing the per-SP Markov decision process, for which we derive a deep reinforcement learning based scheme to find the optimal abstract control policies. TensorFlow-based experiments validate our studies and show that the proposed scheme outperforms the three baselines and yields the best performance in average utility.**

I. Introduction

To support the ever increasing wireless services, new cell sites are being constantly built, which leads to dense network deployments [1]. In a dense radio access network (RAN), it is expensive to manage the control plane operations. On the other hand, the computation-intensive applications, such as the augmented reality and the interactive online gaming, are gaining popularity [2]. The mobile user (MU)-end terminal devices are in general constrained by the battery capacity and the processing speed of central processing unit (CPU). Hence the tension between computation-intensive applications and resource-constrained mobile devices calls for a revolution in existing computing infrastructures [3]. Mobile-edge computing (MEC) is emerging as a key technology that brings the computing capabilities within the RANs in close proximity to MUs [2]. Offloading a computation task to the MEC server for execution involves wireless data transmissions. How to orchestrate radio resources between MEC and traditional mobile services adds another dimension of complexity to the network management [4]. By abstracting all physical base stations (BSs) in a geographical area as a logical big BS, the software-defined networking (SDN) concept provides not only infrastructure flexibility but also service-oriented customization [5]. In a software-defined RAN, the SDN-orchestrator is responsible for handling all control plane operations.

One key benefit from implementing a software-defined RAN is to facilitate network sharing [6]. As such, a physical RAN is able to host multiple service providers (SPs, a.k.a., tenants), which breaks the traditional single ownership of a network infrastructure [7]. For example, an over-the-top application provider (such as Google [8]) can become 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 3rd Generation Partnership Project Technical Specification Group network sharing paradigm [9], a software-defined RAN and its integration with network function virtualization enable RAN-only slicing that splits the RAN into multiple virtual slices [10]. This paper concentrates on a software-defined RAN where the RAN slices are judiciously tailored to accommodate both computation and communication functionalities [11].

The technical challenges arise from the implementation of RAN-only slicing. Particularly, the mechanisms that efficiently exploit the decoupling of control and data planes in a software-defined RAN must be designed to optimize radio resource utilization. In the considered software-defined RAN, a limited number of channels are auctioned over the discrete time horizon to the SPs, the subscribers of which request MEC and traditional mobile services. The SPs compete to orchestrate the channels for their subscribed MUs in accordance with the network dynamics with the aim of maximizing the expected long-term payoff performance. After collecting the auction bids from all SPs, the SDN-orchestrator allocates channels to the MUs via a Vickrey-Clarke-Groves (VCG) mechanism[1] [12]. For a MU, the "freshness" of data packets from traditional mobile services and the energy consumed by mobile device are of equivalent importance. A relevant metric for quantifying the "freshness" is the notion of age of information (AoI) [13]. To the best of our knowledge, there does not exist a comprehensive study on stochastic AoI-aware resource orchestration in multi-tenancy RAN-only slicing.

II. System Descriptions and Assumptions

In this paper, we consider a system with RAN-only slicing. The infinite time horizon is discretized into scheduling slots, each of which is indexed by an integer $k \in \mathbb{N}_+$ and is assumed to be of equal duration δ (in seconds). The physical RAN is

[1]One main advantage of the VCG mechanism is to ensure truthfulness, efficiency and incentive compatibility.

composed of a set $\mathcal{B}$ of BSs covering a service area, which is represented by a set $\mathcal{L}$ of small locations. A small location can be characterized by the uniform signal propagation conditions [14]. Let $\mathcal{L}_b$ denote the serving area of a BS $b \in B$. We assume for any two BSs b and $b' \in \mathcal{B}$ ($b' \neq b$) that $\mathcal{L}_b \cap \mathcal{L}_{b'} = \emptyset$. The geographical distribution of BSs can be denoted by a topological graph $\mathcal{TG} = \langle \mathcal{B}, \mathcal{E} \rangle$, where $\mathcal{E} = \{e_{b,b'} : b \neq b', b, b' \in \mathcal{B}\}$ with $e_{b,b'} = 1$ if BS b and BS b' are neighbours and otherwise $e_{b,b'} = 0$. In the network, a set $\mathcal{I} = \{1, \cdots, I\}$ of SPs provide both MEC and traditional mobile services to MUs. We assume that a MU can subscribe to only one SP. Let $\mathcal{N}_i$ be the set of MUs of a SP $i \in \mathcal{I}$.

Over the scheduling slots, the MUs dynamically move within $\mathcal{L}$ following a Markov mobility model [15]. We let $\mathcal{N}_{b,i}^k$ be the set of MUs of SP $i \in \mathcal{I}$ appearing in the coverage of a BS $b \in \mathcal{B}$ during a slot k. A MU at a location can only be associated with the BS that covers the location. All MUs share a set $\mathcal{J} = \{1, \cdots, J\}$ of channels with the same bandwidth η (in Hz). The SPs compete for the limited channel access opportunities in order to serve their MUs. At the beginning of each slot k, each SP i submits an auction bid $\boldsymbol{\beta}_i^k = (\nu_i^k, \mathbf{C}_i^k)$, where ν_i^k is the valuation of $\mathbf{C}_i^k = (C_{b,i}^k : b \in \mathcal{B})$ with $C_{b,i}^k$ being the number of requested channels in the service area of a BS b. After receiving $\boldsymbol{\beta}^k = (\boldsymbol{\beta}_i^k : i \in \mathcal{I})$, the SDN-orchestrator performs the centralized channel allocation and calculates the payment τ_i^k for SP i. Let $\boldsymbol{\rho}_n^k = (\rho_{n,j}^k : j \in \mathcal{J})$ 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. The following constraints are taken into account for the channel allocation at the SDN-orchestrator during a scheduling slot,

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

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

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

which ensure that: 1) one channel cannot be allocated to MUs associated with two adjacent BSs in order to avoid interference during data transmissions; and 2) 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. The winner vector from the channel auction at slot k is denoted by $\boldsymbol{\phi}^k = (\phi_i^k : i \in \mathcal{I})$, where $\phi_i^k = 1$ if SP i wins 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 $\boldsymbol{\phi}^k$ according to the VCG mechanism, that is,

$$\boldsymbol{\phi}^k = \arg\max_{\boldsymbol{\phi}} \sum_{i \in \mathcal{I}} \phi_i \cdot \nu_i^k \tag{4a}$$

$$\text{s. t.} \sum_{n \in \mathcal{N}_{b,i}^k} \varphi_n^k = \phi_i \cdot C_{b,i}^k, \forall b \in \mathcal{B}, \forall i \in \mathcal{I}; \tag{4b}$$

$$\text{constraints (1), (2) and (3)}, \tag{4c}$$

where $\varphi_n^k = \sum_{j \in \mathcal{J}} \rho_{n,j}^k$ and $\boldsymbol{\phi} = (\phi_i \in \{0, 1\} : i \in \mathcal{I})$. Then the payment τ_i^k for each SP i can be calculated as $\tau_i^k = \max_{\boldsymbol{\phi}_{-i}} \sum_{i' \in \mathcal{I} \setminus \{i\}} \phi_{i'} \cdot \nu_{i'}^k - \max_{\boldsymbol{\phi}} \sum_{i' \in \mathcal{I} \setminus \{i\}} \phi_{i'} \cdot \nu_{i'}^k$, where $-i = \mathcal{I} \setminus \{i\}$.

Denote by $L_n^k \in \mathcal{L}$ the geographical position of a MU $n \in \mathcal{N}$ during a scheduling slot k. As in [14], we assume that the average channel gain $H_n^k = h(L_n^k)$ of the link between MU n and the associated BS is only determined by the physical distance. At the beginning of each scheduling slot k, MU n independently and randomly generates a number $A_{n,(\mathrm{t})}^k \in \mathcal{A} = \{0, 1, \cdots, A_{(\mathrm{t})}^{(\max)}\}$ of computation tasks[2] according to an unknown Markov process [16]. We use $(\mu_{(\mathrm{t})}, \vartheta)$ to represent a single computation task, where $\mu_{(\mathrm{t})}$ and ϑ are the input data size (in bits) and the number of CPU cycles required to accomplish one input bit of the computation task, respectively. A computation task can be 1) processed locally at the MU or 2) offloaded to the MEC server for execution. The computation offloading decision for MU n at a slot k determines the number $R_{n,(\mathrm{t})}^k$ ($\leq A_{n,(\mathrm{t})}^k$) of tasks to be offloaded to the MEC server. The remaining $A_{n,(\mathrm{t})}^k - \varphi_n^k \cdot R_{n,(\mathrm{t})}^k$ tasks are to be processed locally. Meanwhile, a data queue is equipped at a MU to buffer the packets from the traditional mobile service. For each MU n, let $W_{n,(\mathrm{p})}^k$ be the queue length at the beginning of a scheduling slot k and $A_{n,(\mathrm{p})}$ be the fixed number of new packets arriving evenly during the slot. The arriving time instant of each packet arrival belongs to $\{\delta \cdot (k - 1 + a/A_{n,(\mathrm{p})}) : a \in \{1, \cdots, A_{n,(\mathrm{p})}\}\}$. Let $R_{n,(\mathrm{p})}^k$ be the number of packets that are scheduled for transmission from MU n at slot k. The queue evolution of MU n can be written as the form below,

$$W_n^{k+1} = \max\left\{W_n^k - \varphi_n^k \cdot R_{n,(\mathrm{p})}^k, 0\right\} + A_{n,(\mathrm{p})}, \tag{5}$$

where a large enough queue size is assumed at the MUs to avoid the possibility of packet drops. For traditional mobile service, it is critical to maintain the "freshness" of data packets. The data "freshness" of a MU n at the beginning of scheduling slot k (or equivalently, at the end of scheduling slot $k-1$) can be quantified by the AoI T_n^k [13],

$$T_n^k = (k - 1) \cdot \delta - \lambda_n^k, \tag{6}$$

with λ_n^k being the arriving time instant of the oldest packet from the data queue at the beginning of slot k. The AoI of the data queue evolves as

$$T_n^{k+1} = T_n^k + \delta - \frac{\delta}{A_{n,(\mathrm{p})}} \cdot \varphi_n^k \cdot R_{n,(\mathrm{p})}^k. \tag{7}$$

The energy (in Joules) consumed by a MU $n \in \mathcal{N}$ for the reliable transmissions 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 can be calculated to be

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

[2]For the purpose of making theoretical analysis tractable, we assume that the maximum CPU power at a mobile device matches the maximum computation task arrivals and a MU can process $A_{(\mathrm{t})}^{(\max)}$ tasks within the duration of one scheduling slot.

where σ^2 is the noise power spectral density and $\mu_{(\mathrm{p})}$ is the size of data packets. Let $\Omega^{(\max)}$ be the maximum transmit power for all MUs, namely, $P^k_{n,(\mathrm{tr})} \leq \Omega^{(\max)} \cdot \delta$, $\forall n$ and $\forall k$. For the rest number $A^k_{n,(\mathrm{t})} - \varphi^k_n \cdot R^k_{n,(\mathrm{t})}$ of computation tasks that are processed at the mobile device of MU n, the CPU energy consumption is given by

$$P^k_{n,(\mathrm{CPU})} = \varsigma \cdot \mu_{(\mathrm{t})} \cdot \vartheta \cdot \varrho^2 \cdot \left(A^k_{n,(\mathrm{t})} - \varphi^k_n \cdot R^k_{n,(\mathrm{t})}\right), \quad (9)$$

where ς is the effective switched capacitance [17] and ϱ is the CPU-cycle frequency of the mobile devices.

III. Stochastic Game Formulation

At a scheduling slot k, the local state of a MU $n \in \mathcal{N}$ can be given as $\boldsymbol{\chi}^k_n = (L^k_n, A^k_{n,(\mathrm{t})}, W^k_n, T^k_n) \in \mathcal{X} = \mathcal{L} \times \mathcal{A} \times \mathcal{W} \times \mathcal{T}$, where $\mathcal{W}$ and $\mathcal{T}$ denote the sets of queue and AoI states. $\boldsymbol{\chi}^k = (\boldsymbol{\chi}^k_n : n \in \mathcal{N}) \in \mathcal{X}^{|\mathcal{N}|}$ then characterizes the global network state the with $|\mathcal{N}|$ meaning the cardinality of the set $\mathcal{N}$. For a SP $i \in \mathcal{I}$, we define $\boldsymbol{\pi}_i = (\pi_{i,(\mathrm{c})}, \boldsymbol{\pi}_{i,(\mathrm{t})}, \boldsymbol{\pi}_{i,(\mathrm{p})})$ as a control policy, 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. Note that with the channel allocation results, 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 $\boldsymbol{\pi}_{i,(\mathrm{p})}$ of SP i depend only on $\boldsymbol{\chi}^k_i = (\boldsymbol{\chi}^k_n : n \in \mathcal{N}_i) \in \mathcal{X}_i = \mathcal{X}^{|\mathcal{N}_i|}$. The joint control policy of all SPs in the network is given by $\boldsymbol{\pi} = (\boldsymbol{\pi}_i : i \in \mathcal{I})$. With the observation of global network state $\boldsymbol{\chi}^k$ at the beginning of each scheduling slot k, SP i announces the auction bid $\boldsymbol{\beta}^k_i$ to the SDN-orchestrator and decides the $\mathbf{R}^k_{i,(\mathrm{t})}$ computation tasks as well as $\mathbf{R}^k_{i,(\mathrm{p})}$ data packets to be transmitted following the control policy $\boldsymbol{\pi}_i$. Namely, $\boldsymbol{\pi}_i(\boldsymbol{\chi}^k) = (\pi_{i,(\mathrm{c})}(\boldsymbol{\chi}^k), \boldsymbol{\pi}_{i,(\mathrm{t})}(\boldsymbol{\chi}^k_i), \boldsymbol{\pi}_{i,(\mathrm{p})}(\boldsymbol{\chi}^k_i)) = (\boldsymbol{\beta}^k_i, \mathbf{R}^k_{i,(\mathrm{t})}, \mathbf{R}^k_{i,(\mathrm{p})})$, where $\mathbf{R}^k_{i,(\mathrm{t})} = (R^k_{n,(\mathrm{t})} : n \in \mathcal{N}_i)$ and $\mathbf{R}^k_{i,(\mathrm{p})} = (R^k_{n,(\mathrm{p})} : n \in \mathcal{N}_i)$. Accordingly, SP i achieves an instantaneous payoff at slot k, which is

$$\begin{aligned} &F_i\Big(\boldsymbol{\chi}^k, \boldsymbol{\varphi}^k_i, \mathbf{R}^k_{i,(\mathrm{t})}, \mathbf{R}^k_{i,(\mathrm{p})}\Big) \\ &= \sum_{n \in \mathcal{N}_i} \alpha_n \cdot U_n\Big(\boldsymbol{\chi}^k_n, \varphi^k_n, R^k_{n,(\mathrm{t})}, R^k_{n,(\mathrm{p})}\Big) - \tau^k_i, \end{aligned} \quad (10)$$

with $\boldsymbol{\varphi}^k_i = (\varphi^k_n : n \in \mathcal{N}_i)$ and $\alpha_n \in \mathbb{R}_+$ being the unit price to charge a MU n for realizing utility

$$\begin{aligned} &U_n\Big(\boldsymbol{\chi}^k_n, \varphi^k_n, R^k_{n,(\mathrm{t})}, R^k_{n,(\mathrm{p})}\Big) = \\ &U^{(1)}_n\big(T^k_n\big) + \ell_n \cdot \Big(U^{(2)}_n\Big(P^k_{n,(\mathrm{CPU})}\Big) + U^{(3)}_n\Big(P^k_{n,(\mathrm{tr})}\Big)\Big). \end{aligned} \quad (11)$$

$U^{(1)}_n(\cdot)$, $U^{(2)}_n(\cdot)$ and $U^{(3)}_n(\cdot)$ in (11) are assumed to be the positive and monotonically decreasing functions, while $\ell_n \in \mathbb{R}_+$ is a weighting factor. It can be easily found that the randomness lying in a sequence $\{\boldsymbol{\chi}^k : k \in \mathbb{N}_+\}$ of global network states is Markovian.

Taking expectation with respect to the per-slot instantaneous payoffs across the scheduling slots, the expected long-term payoff of a SP $i \in \mathcal{I}$ for a given initial global network state $\boldsymbol{\chi}^1 = \boldsymbol{\chi} \triangleq (\boldsymbol{\chi}_n = (L_{n,(\mathrm{u})}, L_{(\mathrm{e})}, A_{n,(\mathrm{t})}, W_n) : n \in \mathcal{N})$ can be expressed as in (12) on the top of Page 4, where $\gamma \in [0, 1)$ is a discount factor. We also define $V_i(\boldsymbol{\chi}, \boldsymbol{\pi})$ as the state-value function of SP i. Each SP i aims to solve a best-response control policy $\boldsymbol{\pi}^*_i$ such that $\boldsymbol{\pi}^*_i = \arg\max_{\boldsymbol{\pi}_i} V_i(\boldsymbol{\chi}, \boldsymbol{\pi}_i, \boldsymbol{\pi}_{-i})$, $\forall \boldsymbol{\chi} \in \mathcal{X}^{|\mathcal{N}|}$. Consider the limited number of channels and the stochastic nature in networking environment, the interactions among the non-cooperative SPs over the scheduling slots are formulated as a stochastic game, $\mathcal{SG}$. In the game, the players are I SPs and there are a set $\mathcal{X}^{|\mathcal{N}|}$ of global network states as well as a collection of control policies $\{\boldsymbol{\pi}_i : \forall i \in \mathcal{I}\}$. A Nash equilibrium (NE), namely, a tuple of best-response control policies $\langle \boldsymbol{\pi}^*_i : i \in \mathcal{I} \rangle$, describes the rational behaviours of the non-cooperative SPs in the $\mathcal{SG}$. For the I-player $\mathcal{SG}$ with an expected infinite-horizon discounted payoff criterion, there always exists a NE in stationary control policies [18]. For simplicity, we 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 $\forall \boldsymbol{\chi} \in \mathcal{X}^{|\mathcal{N}|}$.

IV. Stochastic Game Abstraction and A Deep Reinforcement Learning Scheme

It can be observed from (12) that the expected long-term payoff of a SP $i \in \mathcal{I}$ depends on the information of not only the global network states across the scheduling slots but also the joint control policy $\boldsymbol{\pi}$ of all SPs in the network. That is, the decision makings among the non-cooperative SPs are coupled in the $\mathcal{SG}$, which makes it a daunting task to find the NE. In the following, we will elaborate on how the SPs play the $\mathcal{SG}$ only with limited local information.

A. Abstract Stochastic Game Reformulation

To alleviate the coupling of decision makings among the SPs, we abstract $\mathcal{SG}$ as $\mathcal{AG}$ [19], in which a SP $i \in \mathcal{I}$ behaves based on its own local network dynamics and abstractions of states at other SPs. Let $\mathcal{S}_i = \{1, \cdots, S_i\}$ be an abstraction of $\mathcal{X}_{-i}$, where $S_i \in \mathbb{N}_+$ and $S_i \ll |\mathcal{X}_{-i}|$. We observe that the 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 $[0, \Gamma_i]$ into S_i disjoint intervals, i.e., $[0, \Gamma_{i,1}]$, $(\Gamma_{i,1}, \Gamma_{i,2}]$, $(\Gamma_{i,2}, \Gamma_{i,3}]$, …, $(\Gamma_{i,S_i-1}, \Gamma_{i,S_i}]$, where $\Gamma_{i,S_i} = \Gamma_i$ is the maximum payment and we let $\Gamma_{i,1} = 0$ for the case in which SP i wins the channel auction with no payment [20]. With this regard, SP i abstracts $(\boldsymbol{\chi}_i, \boldsymbol{\chi}_{-i}) \in \mathcal{X}^{|\mathcal{N}|}$ as $\tilde{\boldsymbol{\chi}}_i = (\boldsymbol{\chi}_i, s_i) \in \tilde{\mathcal{X}}_i = \mathcal{X}_i \times \mathcal{S}_i$ if the payment in previous slot belongs to $(\Gamma_{i,s_i-1}, \Gamma_{i,s_i}]$. 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 played by SP i in $\mathcal{AG}$, where $\tilde{\pi}_{i,(\mathrm{c})}$ denotes the abstract channel auction policy. Likewise, the abstract state-value function of SP i under $\tilde{\boldsymbol{\pi}} = (\tilde{\boldsymbol{\pi}}_i : i \in \mathcal{I})$ can be defined by (13) on the top of Page 4, $\forall \tilde{\boldsymbol{\chi}}_i \in \tilde{\mathcal{X}}_i$, where $\tilde{\boldsymbol{\chi}}^k = (\tilde{\boldsymbol{\chi}}^k_i = (\boldsymbol{\chi}^k_i, s^k_i) : i \in \mathcal{I})$ with s^k_i being the abstract state at a slot k and $\tilde{F}_i(\tilde{\boldsymbol{\chi}}^k_i, \boldsymbol{\varphi}_i(\tilde{\boldsymbol{\pi}}_{(\mathrm{c})}(\tilde{\boldsymbol{\chi}}^k)), \boldsymbol{\pi}_{i,(\mathrm{t})}(\boldsymbol{\chi}^k_i), \boldsymbol{\pi}_{i,(\mathrm{p})}(\boldsymbol{\chi}^k_i))$ is the immediate payoff with $\tilde{\boldsymbol{\chi}}^k = (\tilde{\boldsymbol{\chi}}^k_i : i \in \mathcal{I})$ and $\tilde{\boldsymbol{\pi}}_{(\mathrm{c})} = (\tilde{\pi}_{i,(\mathrm{c})} : i \in \mathcal{I})$. Our previous work has verified 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 $\mathcal{AG}$ results in a bounded regret [19], where $\tilde{\boldsymbol{\pi}}^*_i = (\tilde{\pi}^*_{i,(\mathrm{c})}, \boldsymbol{\pi}^*_{i,(\mathrm{t})}, \boldsymbol{\pi}^*_{i,(\mathrm{p})})$ denotes the best-response abstract control policy of SP i. Hereinafter, our focus switches 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 $\mathcal{AG}$. We denote $\tilde{\mathbb{V}}_i(\tilde{\boldsymbol{\chi}}_i) = \tilde{V}_i(\tilde{\boldsymbol{\chi}}_i, \tilde{\boldsymbol{\pi}}^*)$.

$$V_i(\boldsymbol{\chi},\boldsymbol{\pi}) = (1-\gamma)\cdot \mathsf{E}_{\boldsymbol{\pi}}\Bigg[\sum_{k=1}^{\infty}(\gamma)^{k-1}\cdot F_i\big(\boldsymbol{\chi}^k, \boldsymbol{\varphi}_i\big(\boldsymbol{\pi}_{(\mathrm{c})}\big(\boldsymbol{\chi}^k\big)\big), \boldsymbol{\pi}_{i,(\mathrm{t})}\big(\boldsymbol{\chi}_i^k\big), \boldsymbol{\pi}_{i,(\mathrm{p})}\big(\boldsymbol{\chi}_i^k\big)\big)\,|\boldsymbol{\chi}^1=\boldsymbol{\chi}\Bigg] \tag{12}$$

$$\tilde{V}_i(\tilde{\boldsymbol{\chi}}_i,\tilde{\boldsymbol{\pi}}) = (1-\gamma)\cdot \mathsf{E}_{\tilde{\boldsymbol{\pi}}}\Bigg[\sum_{k=1}^{\infty}(\gamma)^{k-1}\cdot \tilde{F}_i\big(\tilde{\boldsymbol{\chi}}_i^k, \boldsymbol{\varphi}_i\big(\tilde{\boldsymbol{\pi}}_{(\mathrm{c})}\big(\tilde{\boldsymbol{\chi}}^k\big)\big), \boldsymbol{\pi}_{i,(\mathrm{t})}\big(\boldsymbol{\chi}_i^k\big), \boldsymbol{\pi}_{i,(\mathrm{p})}\big(\boldsymbol{\chi}_i^k\big)\big)\,|\tilde{\boldsymbol{\chi}}_i^1=\tilde{\boldsymbol{\chi}}_i\Bigg] \tag{13}$$

B. Linear Decomposition of Abstract State-Value Function

Two challenges remain in solving the optimal abstract state-value functions for each SP $i \in \mathcal{I}$ when using dynamic programming methods [21]: 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. From previous analysis, 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. Hence we are motivated to decompose the per-SP MDP in the $\mathcal{AG}$ into $|\mathcal{N}_i|+1$ independent MDPs. Specifically, for a SP $i \in \mathcal{I}$, $\tilde{\mathbb{V}}_i(\tilde{\boldsymbol{\chi}}_i)$, $\forall \tilde{\boldsymbol{\chi}}_i \in \tilde{\mathcal{X}}_i$, can be computed as

$$\tilde{\mathbb{V}}_i(\tilde{\boldsymbol{\chi}}_i) = \sum_{n\in\mathcal{N}_i}\alpha_n\cdot\mathbb{U}_n(\boldsymbol{\chi}_n) - \mathbb{U}_i(s_i), \tag{14}$$

where the per-MU $\mathbb{U}_n$ and the $\mathbb{U}_i(s_i)$ of SP i satisfy, respectively, (15) (which is shown on the top of Page 5) and

$$\mathbb{U}_i(s_i) = (1-\gamma)\cdot\tau_i + \gamma\cdot\sum_{s_i'\in\mathcal{S}_i}\mathbb{P}\Big(s_i'|s_i,\phi_i\Big(\tilde{\boldsymbol{\pi}}^*_{(\mathrm{c})}(\tilde{\boldsymbol{\chi}})\Big)\Big)\cdot\mathbb{U}_i(s_i'). \tag{16}$$

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

We are now able to specify the number of needed channels by a SP $i \in \mathcal{I}$ for its subscribed MUs 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

$$\nu_i = \frac{1}{1-\gamma}\cdot\sum_{n\in\mathcal{N}_i}\alpha_n\cdot\mathbb{U}_n(\boldsymbol{\chi}_n) - \frac{\gamma}{1-\gamma}\cdot\sum_{s_i'\in\mathcal{S}_i}\mathbb{P}\Big(s_i'|s_i,\mathbb{1}_{\{\sum_{b\in\mathcal{B}}C_{b,i}>0\}}\Big)\cdot\mathbb{U}_i(s_i'), \tag{17}$$

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

$$z_n = \arg\max_{z\in\{0,1\}}\Bigg\{(1-\gamma)\cdot U_n\Big(\boldsymbol{\chi}_n, z, \pi^*_{n,(\mathrm{t})}(\boldsymbol{\chi}_n), \pi^*_{n,(\mathrm{p})}(\boldsymbol{\chi}_n)\Big) + \gamma\cdot\sum_{\boldsymbol{\chi}_n'\in\mathcal{X}}\mathbb{P}\Big(\boldsymbol{\chi}_n'|\boldsymbol{\chi}_n, z, \pi^*_{n,(\mathrm{t})}(\boldsymbol{\chi}_n), \pi^*_{n,(\mathrm{p})}(\boldsymbol{\chi}_n)\Big)\cdot\mathbb{U}_n(\boldsymbol{\chi}_n')\Bigg\}, \tag{18}$$

while $\mathbb{1}_{\{\Xi\}}$ equals 1 if the condition Ξ is satisfied and 0, otherwise.

C. Learning Optimal Abstract Control Policy

It can be easily observed that at a current scheduling slot, $\boldsymbol{\beta}_i$ of a SP $i \in \mathcal{I}$ needs $(s_i, \mathbb{P}(s'|s,\iota-1))$ as well as $(\mathbb{U}_n(\boldsymbol{\chi}_n), z_n, L_n)$ from each subscribed MU $n \in \mathcal{N}_i$, where $s' \in \mathcal{S}_i$ and $\iota \in \{1,2\}$. In this paper, we propose that SP i maintains over the scheduling slots a three-dimensional table $\mathbf{Y}_i^k$ of size $S_i \cdot S_i \cdot 2$, entry $y^k_{s,s',\iota}$ of which represents the number of happened transitions from $s_i^{k-1} = s$ to $s_i^k = s'$ when $\phi_i^{k-1} = \iota - 1$ up to a scheduling slot k. $\mathbf{Y}_i^k$ is iteratively updated using the channel auction outcomes. Then the abstract network state transition probability at a slot k can be estimated to be

$$\mathbb{P}\big(s_i^k = s'|s_i^{k-1}=s, \phi_i^{k-1}=\iota-1\big) = \frac{y^k_{s,s',\iota}}{\sum\limits_{s''\in\mathcal{S}_i} y^k_{s'',s',\iota}}, \tag{19}$$

based on which $\mathbb{U}_i(s_i)$, $\forall s_i \in \mathcal{S}_i$ is learned via (20) (which is shown on the top of Page 5) with $\zeta^k \in [0,1)$ being the learning rate. The learning process converges if $\sum_{k=1}^{\infty}\zeta^k = \infty$ and $\sum_{k=1}^{\infty}(\zeta^k)^2 < \infty$ [21].

When a priori statistics of MU mobility and computation task arrivals is not available, Q-learning [21] finds $\mathbb{U}_n(\boldsymbol{\chi}_n)$ for each MU $n \in \mathcal{N}$ by defining the right-hand-side of (15) as the optimal state action-value function $Q_n : \mathcal{X}\times\{0,1\}\times\mathcal{A}\times\mathcal{W} \to \mathbb{R}$. Then we attain

$$\mathbb{U}_n(\boldsymbol{\chi}_n) = \max_{\varphi_n, R_{n,(\mathrm{t})}, R_{n,(\mathrm{p})}} Q_n\big(\boldsymbol{\chi}_n, \varphi_n, R_{n,(\mathrm{t})}, R_{n,(\mathrm{p})}\big), \tag{21}$$

where an action $(\varphi_n, R_{n,(\mathrm{t})}, R_{n,(\mathrm{p})})$ under a current local state $\boldsymbol{\chi}_n$ includes the channel allocation, computation offloading and packet scheduling decisions. However, the tabular nature in Q-function values makes the conventional Q-learning not implementable. For the considered problem solving in this paper, the sizes of $\mathcal{X}$ and action space $\{0,1\}\times\mathcal{A}\times\mathcal{W}$ are calculated as $|\mathcal{L}|\cdot(1+A^{(\max)}_{(\mathrm{t})})\cdot(1+|\mathcal{W}|)\cdot(1+|\mathcal{T}|)$ and $2\cdot(1+A^{(\max)}_{(\mathrm{t})})\cdot(1+R^{(\max)}_{(\mathrm{p})})$, resulting in an extremely slow process of Q-learning, where $R^{(\max)}_{(\mathrm{p})}$ is the maximum number of data packets that can be transmitted over a channel.

The success of a deep neural network in approximately modelling the Q-function inspires us to adopt a deep reinforcement learning (DRL) method [22]. We can then approximate the Q-function by a double deep Q-network (DQN) [23]. Mathematically, $Q_n(\boldsymbol{\chi}_n, \varphi_n, R_{n,(\mathrm{t})}, R_{n,(\mathrm{p})}) \approx Q_n(\boldsymbol{\chi}_n, \varphi_n, R_{n,(\mathrm{t})}, R_{n,(\mathrm{p})}; \boldsymbol{\theta}_n)$, $\forall n \in \mathcal{N}$, 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

$$\mathbb{U}_n(\boldsymbol{\chi}_n) = \tag{15}$$
$$\max_{R_{n,(\mathrm{t})}, R_{n,(\mathrm{p})}} \left\{ (1-\gamma)\cdot U_n\Big(\boldsymbol{\chi}_n, \varphi_n\Big(\tilde{\boldsymbol{\pi}}^*_{(\mathrm{c})}(\tilde{\boldsymbol{\chi}})\Big), R_{n,(\mathrm{t})}, R_{n,(\mathrm{p})}\Big) + \gamma \cdot \sum_{\boldsymbol{\chi}'_n \in \mathcal{X}} \mathbb{P}\Big(\boldsymbol{\chi}'_n | \boldsymbol{\chi}_n, \varphi_n\Big(\tilde{\boldsymbol{\pi}}^*_{(\mathrm{c})}(\tilde{\boldsymbol{\chi}})\Big), R_{n,(\mathrm{t})}, R_{n,(\mathrm{p})}\Big) \cdot \mathbb{U}_n(\boldsymbol{\chi}'_n) \right\}$$

$$\mathbb{U}_i^{k+1}(s_i) = \begin{cases} \big(1-\zeta^k\big)\cdot \mathbb{U}_i^k(s_i) + \zeta^k \cdot \left((1-\gamma)\cdot \tau_i^k + \gamma \cdot \sum\limits_{s_i^{k+1} \in \mathcal{S}_i} \mathbb{P}\big(s_i^{k+1} | s_i, \phi_i^k\big) \cdot \mathbb{U}_i^k\big(s_i^{k+1}\big) \right), & \text{if } s_i = s_i^k \\ \mathbb{U}_i^k(s_i), & \text{otherwise} \end{cases} \tag{20}$$

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 $\Omega^{(\max)}$	3 Watts
Weight of energy consumption ℓ_n	$3, \forall n \in \mathcal{N}$
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
Replay memory size M	5000
Mini-batch size $\lvert\mathcal{O}_n^k\rvert$	$200, \forall n \in \mathcal{N}, \forall k$
Activation function	Tanh [27]
Optimizer	Adam [28]

a replay memory of finite size to store the latest M historical experiences, namely, $\mathcal{M}_n^k = \{\mathbf{m}_n^{k-M+1}, \cdots, \mathbf{m}_n^k\}$, where each experience $\mathbf{m}_n^{k'} = (\boldsymbol{\chi}_n^{k'}, (\varphi_n^{k'}, R_{n,(\mathrm{t})}^{k'}, R_{n,(\mathrm{p})}^{k'}), U_n(\boldsymbol{\chi}_n^{k'}, \varphi_n^{k'}, R_{n,(\mathrm{t})}^{k'}, R_{n,(\mathrm{p})}^{k'}), \boldsymbol{\chi}_n^{k'+1})$ happens at the transition between two consecutive scheduling slots k' and $k'+1$. To perform experience replay [24], 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 (22) on the top of Page 6, 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. By differentiating $\mathsf{LOSS}_n(\boldsymbol{\theta}_n^k)$ with respect to $\boldsymbol{\theta}_n^k$, we obtain the gradient as in (23) on the top of Page 6.

V. NUMERICAL EXPERIMENTS

In this section, we conduct numerical experiments based on TensorFlow [25] to quantitatively examine the performance of the derived DRL-based scheme for AoI-aware multi-tenant resource orchestration in a software-defined RAN. We set up an experimental network with 4 BSs being placed at equal distance 1 Km apart in the centre of a 2×2 Km^2 square service area [14]. The entire area is divided into 1600 locations with each of 50×50 m^2. The average channel gain for a MU $n \in \mathcal{N}$ at $L_n^k \in \mathcal{L}_b$ covered by a BS $b \in \mathcal{B}$ during a slot k is given by $h(L_{n,(\mathrm{u})}^k) = H_0 \cdot (\xi_0/\xi_{b,n}^k)^4$, where $H_0 = -40$ dB is the path-loss constant, $\xi_0 = 2$ m is the reference distance, while $\xi_{b,n}^k$ is the distance between MU n and BS b [26]. The mobilities and the computation task arrivals of all MUs are independently and randomly generated. Moreover, we set $A_{n,(\mathrm{p})} = A_{(\mathrm{p})}$, $\forall n \in \mathcal{N}$. For the utility function in (11), we select $U_n^{(1)}(T_n^k) = \exp\{-T_n^k\}$, $U_n^{(2)}(P_{n,(\mathrm{CPU})}^k) = \exp\{-P_{n,(\mathrm{CPU})}^k\}$ and $U_n^{(3)}(P_{n,(\mathrm{tr})}^k) = \exp\{-P_{n,(\mathrm{tr})}^k\}$. 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 the purpose of performance comparisons, we simulate three baseline schemes, which are specified as follows.

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

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

We first demonstrate the average utility performance per MU per scheduling slot achieved from the proposed DRL-based scheme and the three baselines under various traffic loads $A_{(\mathrm{p})}$ of traditional mobile services. In this experiment, we assume that $J = 11$ channels are shared among the MUs in the network. The results are depicted in Fig. 1, from which we can observe that the proposed scheme achieves a significant performance gain. However, the average utility performance deceases as the traffic load of traditional mobile services increases. The reason behind is that more data packet arrivals lead to larger queue length, larger AoI and higher energy consumption across the MUs. Then in Fig. 2, we exhibit the average utility performance versus the number of channels, where the traffic load of traditional mobile services

$$\mathsf{LOSS}_n\left(\boldsymbol{\theta}_n^k\right) = \mathsf{E}_{\left(\boldsymbol{\chi}_n,\left(\varphi_n,R_{n,(\mathrm{t})},R_{n,(\mathrm{p})}\right),U_n\left(\boldsymbol{\chi}_n,\varphi_n,R_{n,(\mathrm{t})},R_{n,(\mathrm{p})}\right),\boldsymbol{\chi}_n'\right)\in\mathcal{O}_n^k}\Bigg[\Bigg(\left(1-\gamma\right)\cdot U_n(\boldsymbol{\chi}_n,\varphi_n,R_{n,(\mathrm{t})},R_{n,(\mathrm{p})}) + \gamma\cdot Q_n\Bigg(\boldsymbol{\chi}_n',\underset{\varphi_n',R_{n,(\mathrm{t})}',R_{n,(\mathrm{p})}'}{\arg\max}\ Q_n\left(\boldsymbol{\chi}_n',\varphi_n',R_{n,(\mathrm{t})}',R_{n,(\mathrm{p})}';\boldsymbol{\theta}_n^k\right);\boldsymbol{\theta}_{n,-}^k\Bigg) - Q_n\left(\boldsymbol{\chi}_n,\varphi_n,R_{n,(\mathrm{t})},R_{n,(\mathrm{p})};\boldsymbol{\theta}_n^k\right)\Bigg)^2\Bigg] \tag{22}$$

$$\nabla_{\boldsymbol{\theta}_n^k}\mathsf{LOSS}_n\left(\boldsymbol{\theta}_n^k\right) = \mathsf{E}_{\left(\boldsymbol{\chi}_n,(\varphi_n,R_{n,(\mathrm{t})},R_{n,(\mathrm{p})}),U_n(\boldsymbol{\chi}_n,\varphi_n,R_{n,(\mathrm{t})},R_{n,(\mathrm{p})}),\boldsymbol{\chi}_n'\right)\in\mathcal{O}_n^k}\Bigg[\Bigg(\left(1-\gamma\right)\cdot U_n(\boldsymbol{\chi}_n,\varphi_n,R_{n,(\mathrm{t})},R_{n,(\mathrm{p})}) + \gamma\cdot Q_n\Bigg(\boldsymbol{\chi}_n',\underset{\varphi_n',R_{n,(\mathrm{t})}',R_{n,(\mathrm{p})}'}{\arg\max}\ Q_n\left(\boldsymbol{\chi}_n',\varphi_n',R_{n,(\mathrm{t})}',R_{n,(\mathrm{p})}';\boldsymbol{\theta}_n^k\right);\boldsymbol{\theta}_{n,-}^k\Bigg) - Q_n\left(\boldsymbol{\chi}_n,\varphi_n,R_{n,(\mathrm{t})},R_{n,(\mathrm{p})};\boldsymbol{\theta}_n^k\right)\Bigg)\cdot \nabla_{\boldsymbol{\theta}_n^k}Q_n\left(\boldsymbol{\chi}_n,\varphi_n,R_{n,(\mathrm{t})},R_{n,(\mathrm{p})};\boldsymbol{\theta}_n^k\right)\Bigg] \tag{23}$$

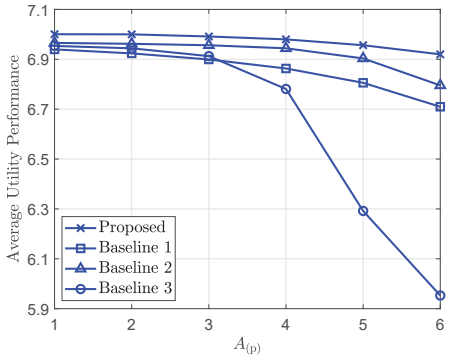

Fig. 1. Average utility performance per MU per scheduling slot versus $A_{(\mathrm{p})}$.

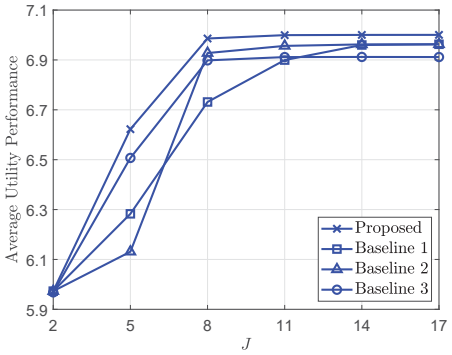

Fig. 2. Average utility performance per MU per scheduling slot versus J.

is fixed to be $A_{(\mathrm{p})} = 3$. 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 performance can be expected for the MUs. When there are a sufficient number of channels in the network, the data transmissions of all MUs 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 AoI-aware resource orchestration among multiple non-cooperative SPs in network slicing, which is formulated as a $\mathcal{SG}$. Without private information exchange among the competing SPs, we reformulate the $\mathcal{SG}$ as a $\mathcal{AG}$. Each SP is hence able to behave independently with the local information only. We further observe that the decisions of the channel auction and the computation offloading as well as packet scheduling are made in sequence, which motivates us to linearly decompose the per-SP single-agent MDP. In this way, the decision making process at a SP is greatly simplified. To address the huge state space, we propose a DRL-based scheme to solve the optimal abstract control policies. Numerical experiments verify our theoretical studies and showcase that the performance achieved from our scheme outperforms the other baselines.

Acknowledgements

The work carried out in this paper was supported by the Academy of Finland under Grant 319759, EU H2020 5G-DRIVE project under Grant 814956, the JSPS KAKENHI under Grant 18KK0279, the JST-Mirai Program under Grant JPMJMI17B3, the Telecommunications Advanced Foundation, the National Key R&D Program of China under Grant 2017YFB1301003, the National Natural Science Foundation of China under Grants 61701439 and 61731002, and the Zhejiang Key Research and Development Plan under Grant 2019C01002.

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] 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.

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

[9] "Telecommunication management; network sharing; concepts and requirements," Rel. 15, 3GPP TS 32.130, Jun. 2018.

[10] 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.

[11] 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.

[12] 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.

[13] S. Kaul *et al.*, "Minimizing age of information in vehicular networks," in *Proc. IEEE SECON*, Salt Lake City, UT, Jun. 2011.

[14] 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.

[15] A. J. Nicholson and B. D. Noble, "BreadCrumbs: Forecasting mobile connectivity," in *Proc. ACM MobiCom*, San Francisco, CA, Sep. 2008.

[16] X. He *et al.*, "Privacy-aware offloading in mobile-edge computing," in *Proc. IEEE GLOBECOM*, Singapore, Dec. 2017.

[17] 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.

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

[19] 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.

[20] J. Jia *et al.*, "Revenue generation for truthful spectrum auction in dynamic spectrum access," in *Proc. ACM MobiHoc*, New Orleans, LA, May 2009.

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

[22] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.

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

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

[25] M. Abadi *et al.*, "Tensorflow: A system for large-scale machine learning," in *Proc. OSDI*, Savannah, GA, Nov. 2016.

[26] 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.

[27] K. Jarrett *et al.*, "What is the best multi-stage architecture for object recognition?" in *Proc. IEEE ICCV*, Kyoto, Japan, Sep.–Oct. 2009.

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