Access Control for RAN Slicing based on Federated Deep Reinforcement Learning

*Yi-Jing Liu, *Gang Feng, *Jian Wang, †Yao Sun, *Shuang Qin
*National Key Laboratory of Science and Technology on Communications,
and Yangtze Delta Region Institute (Huzhou),
*University of Electronic Science and Technology of China
†James Watt School of Glasgow, University of Glasgow
E-mail:fenggang@uestc.edu.cn

***Abstract*—Network Slicing (NS) has been widely identified as a key architectural technology for 5G-and-beyond systems by supporting divergent requirements sustainably. With the widespread of emerging smart devices, access control becomes an essential yet challenging issue in NS-based wireless networks due to the device-base station (BS)-NS three-layer association relationship. Meanwhile, stringent data security and device privacy concerns are increasing dramatically. In this paper, we propose an efficient access control scheme for radio access network (RAN) slicing by exploiting a federated deep reinforcement learning framework, called FDRL-AC, to improve network throughput and communication efficiency while enforcing the data security and device privacy. Specifically, we use deep reinforcement learning to train local model on devices, where horizontally federated learning (FL) is employed for parameter aggregation on BS, while vertically FL is employed for feature aggregation on the encrypted party. Numerical results show that the proposed FDRL-AC scheme can achieve significant performance gain in terms of network throughput and communication efficiency in comparison with some state-of-art solutions.**

I. Introduction

It is widely acknowledged that network slicing (NS) is one of the most vital architectural technologies for 5G-and-beyond systems. In order to support various applications with diverse quality of service (QoS) requirements in different communication scenarios, multiple isolated network slices (NSs) can be designed, deployed, customized, and optimized on a common physical network infrastructure [1], [2]. The NS-based networks can provide tailored services flexibly to meet the specific needs of applications and corresponding Service Level Agreement. However, driven by the rapidly growing applications with diversified QoS requirements, how to identify and classify service flows for accommodation by appropriate application-specific NS is still a challenging issue, especially in radio access network (RAN) domain [1], [2].

In RAN slicing, access control as well as relevant handoff management are fundamentally distinct from that in conventional mobile networks because of the introduction of NS [2]–[4]. On one hand, NSs are logically virtualized and isolated over shared physical networks, where both physical and virtual resources should be considered to form a function chain for a specific service [2]. On the other hand, for a specific mobile device, it needs to select/re-select a proper NS which may cover multiple base stations (BSs) to guarantee its QoS. Therefore, in NS-based mobile network, access control inherently includes NS selection, BS selection, and handoff management issues, where a joint optimization of NS and BS selection for a device should be addressed [2]–[5]. Most current existing investigations which tackled similar problems applied the static optimization algorithms or heuristic algorithms, such as [3], [4]. However, both the optimization algorithms and heuristic algorithms may not be appreciated for the access control in RAN slicing, as the computational complexity incurred by searching the optimal solution in complex and dynamic scenarios could be too high.

Fortunately, recently emerging deep reinforcement learning (DRL) can be exploited to solve such decision-making problem in the complex and dynamic wireless system with a large size of data [6]. Furthermore, in light of the increasingly stringent data security and device privacy concerns, an emerging approach, federated learning (FL) [7], is introduced, which trains unbalanced data locally at individual devices and exploits the collaboration of the devices. Most of the current work focuses on FL or DRL. Indeed, in order to reduce the amount of required training samples and/or make more precise decisions, combing FL and DRL, called FDRL, is intuitively advantageous [7]

In this paper, we propose an intelligent access control scheme for RAN slicing by combing FL and DRL, called federated deep reinforcement learning for access control (FDRL-AC), to improve network throughput and communication efficiency while enforcing the data security and device privacy. Considering the large state-action space and service diversity, FDRL-AC is designed to consist of two layer model aggregations: 1) Horizontal aggregation: for the same type of services, we aggregate parameters of local DRL model on BSs to share the similar samples; 2) Vertical aggregation: for the services of different types, we aggregate the access features of local DRL model on the third encrypted party [7], [8], where we use Shap-

This work has been supported by the National Key Research Project (Grant 2020YFB1806804), the National Natural Science Foundation of China (Grant 62071091), and ZTE Industry-Academia-Research Cooperation Founds.

ley value [8] to evaluate aggregated access features. Numerical results show that in the typical scenarios, our proposed FDRL-AC scheme significantly outperforms the traditional solutions in terms of network throughput and communication efficiency.

In the rest of this paper, we begin with the system model and problem formulation in Section II and Section III respectively. In Section IV, FDRL-AC is presented to solve the access control problem of RAN slicing. Finally, we present the numerical results in Section V and conclude the paper in Section VI.

II. System Model

A. Network Model

We consider a scenario where the virtualized network is built upon a Software Define Network/Network Function Virtualization -enabled 5G network infrastructure, which is composed of core network (CN) and RAN. As shown in Fig. 1, when the location of a device changes, the initially selected mobility management function (AMF) may be changed to receive services, to enable mobility tracking and enable reachability [9]. In addition, some network functionalities, such as AMF in CN, distributed unit (DU) and radio unit (RU) in RAN, can be shared among multiple slices, while others are slice-specific. More details about network functionalities can be found in [9].

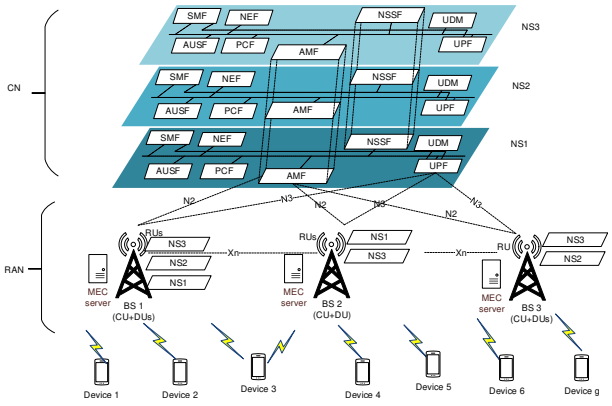

Fig. 1. The NS-based mobile network model.

B. RAN slicing

We consider a multi-NSs and multi-BSs RAN slicing scenario, as shown as Fig.1. When a device accesses the mobile network or experiences a handoff, both BS and NS may need to be selected/reselected for provision seamless service while meeting the QoS requirements. Let $\mathcal{B}$, $\mathcal{N}$, and $\mathcal{D}$ denote the set of BSs, NSs, and devices, respectively. For a specific BS k, we use $\mathcal{N}_k = \{j, ..., g\}$ to represent the set of NSs which are supported by it. For a specific NS j, we use a four-tuple $\left(R_j, T_j, \Omega_j, \vec{W}_j\right)$ to represent the state where R_j and T_j denote the minimal transmission rate and the maximal latency which are provided by NS j to serve devices. Moreover, Ω_j represents the bandwidth allocated to NS j in CN (including transport network), and $\vec{W}_j$ is a vector, which represents the bandwidth allocation of NS j from all BSs. We assume that the kth element in $\vec{W}_j$ is denoted by $b_{j,k}$, which represents the bandwidth resource allocated to NS j by BS k, where $b_{j,k} = 0$ means BS k is not covered by NS j.

C. Service Requirements

Since the services required by devices may vary with time, we assume the time is slotted, where slotted time can be regarded as a sampled version of continuous-time which consists of T time slots. During time slot $t \in [1, T]$, we assume that a device connects only one BS and remains connected to the same NS and BS. Let u be the number of devices in the network. For a specific device $d_i \in \mathcal{D}$, its service quality can be described by two metrics: the minimum transmission rate $\hat{r}_i^t$ and the maximum tolerated latency $\hat{d}_i^t$. Therefore, NS j can accommodate d_i only if $R_j \geq \hat{r}_i^t$ and $T_j \leq \hat{d}_i^t$.

Let $r_{i,t}^{j,k}$ be the transmission rate of d_i which is served by NS j via BS k during time slot t, and $w_{i,t}^{j,k}$ be the wireless bandwidth that BS k allocates to d_i which is served by NS j during time slot t (Here $w_{i,t}^{j,k}$ also called consumed radio resources of d_i during time slot t). In this work, the models may affect the absolute value of communication efficiency, but do not invalidate the relative performance enhancement of our proposed policies. Hence, as we focus the device association in the RAN slicing, we assume the delay in CN (*i.e.*, T_j) is a constant. The similar assumption is widely used in related studies, such as [3], [4]. Therefore, the end-to-end delay can be calculated as $\hat{T}_{i,t}^{j,k} + T_j$, where $\hat{T}_{i,t}^{j,k} = q_i / r_{i,t}^{j,k}$ is the delay in RAN of d_i served by NS j via BS k and q_i is the volume of flow data generated by d_i. Moreover, we use Shannon theory to define the transmission rate (*i.e.*, $r_{i,t}^{j,k} = w_{i,t}^{j,k} log_2\left(1 + \mathit{SINR}_{i,t}^k\right)$), where $\mathit{SINR}_{i,t}^k$ is the signal-to-interference-plus-noise-ratio (SINR) between d_i and BS k during time slot t. Moreover, $\mathit{SINR}_{i,t}^k = p_{i,t}^k \cdot G_{i,t}^k / (\sum_{k \in \mathcal{B}, k' \neq k} p_{i,t}^{k'} G_{i,t}^{k'} + \zeta^2), t \in T$, where $p_{i,t}^k$ represents the transmission power allocated to d_i at BS k, $G_{i,t}^k$ is the channel gain between d_i and BS k, and ζ^2 is the noise power level.

D. Handoff Cost

When the location of a device changes or the service quality of a device cannot be satisfied, a handoff may occur to improve the experience of the user. Traditional reference signal received power (RSRP)-based handoff mechanisms [10] are no longer applicable to RAN slicing. Specifically, a device accesses to an NS via a specific BS, forming a three-layer associate relationship device-BS-NS [11]. Therefore, both the service type of NSs and the RSRP of BSs should be taken into account when a handoff occurs. Therefore, unlike the handoff in traditional mobile networks, there are three types of handoff we need to consider: switching NS only, switching BS only, and switching both NS and BS. Therefore, we define the amount of signaling data for three types of handoff as: 1) q_{NS}, the amount of signaling data needed for switching NS only; 2) q_{BS}, the amount of signaling data needed for switching BS only; 3) q_{N-B}, the amount of signaling data needed for switching both NS and BS; with the relationship $q_{NS} < q_{BS} < q_{N-B}$ [11]. Intuitively, the amount of signaling

data needed incurs corresponding signaling overhead in terms of bandwidth consumption for signaling exchange. Furthermore, due to the bandwidth consumed by service flows and the bandwidth consumed by handoff may not be in the same order of magnitude, we define the handoff cost as follows,

$$\alpha^{\text{HO}} = \begin{cases} q_{NS}/w_{NS}, \text{ if switching NSs only,} \\ q_{BS}/w_{BS}, \text{ if switching BSs only,} \\ q_{N-B}/w_{N-B}, \text{ if switching both NSs and BSs,} \\ 0, \text{ otherwise.} \end{cases} \tag{1}$$

where w_{NS}, w_{BS}, and w_{N-B} represent the bandwidth consumed by switching NS only, switching BS only, switching both BS and NS, respectively.

III. Problem Formulation

A. Problem Statement

Given a set of devices which may require services of different types, we investigate the access control problem under network resource constraints. We define a binary variable $x_{i,t}^{j,k}$ to indicate whether the device d_i is served by NS j via BS k during time slot t or not: $x_{i,t}^{j,k} = 1$ yes and 0 otherwise. Therefore, multiplying the two variables $x_{i,t}^{j,k} x_{i,t-1}^{j',k'}$ in adjacent time slots indicates the handoff decision of d_i from time slot $t-1$ to t, which can be summarized in Table I. Therefore, in order to improve network throughput while reducing handoff cost, we define the communication efficiency of the network during time slot t as follows,

TABLE I
THE RELATIONSHIP BETWEEN HANDOFF AND $x_{i,t}^{j,k} x_{i,t-1}^{j',k'}$

$x_{i,t}^{j,k} x_{i,t-1}^{j',k'}$	NSs	BSs	Switching	α^{HO}
1	$j \neq j'$	$k \neq k'$	both BS and NS	q_{N-B}/w_{N-B}
1	$j = j'$	$k \neq k'$	BS only	q_{BS}/w_{BS}
1	$j \neq j'$	$k = k'$	NS only	q_{NS}/w_{NS}
1	$j = j'$	$k = k'$	no handoff	0

$$e_t = \sum\nolimits_{i \in \mathcal{D}} (\alpha_{i,t}^{\text{flow}} x_{i,t}^{j,k} - \alpha^{\text{HO}} x_{i,t}^{j,k} x_{i,t-1}^{j',k'}), \forall t \in [0,T], \tag{2}$$

where communication efficiency e_t is a bandwidth metric value representing the bandwidth efficiency minus signaling overhead (which is indeed the "handoff cost").

In our model, $x_{i,t}^{j,k}$ is a decision variable. As the access control is indeed a sequential decision problem, we use the long-term communication efficiency of the network as the optimization objective in (3). Therefore, we formulate the access control problem as follows.

$$\max \lim_{T \to +\infty} E[\frac{1}{T} \sum\nolimits_{t=1}^{T} e_t] \tag{3}$$

$$\text{s.t.} \sum\nolimits_{k \in \mathcal{B}} \sum\nolimits_{i \in \mathcal{D}} x_{i,t}^{j,k} r_{i,t}^{j,k} \leq \Omega_j, \forall j \in \mathcal{N}, t \in [0,T] \tag{3.1}$$

$$\sum\nolimits_{i \in \mathcal{D}} x_{i,t}^{j,k} w_{i,t}^{j,k} \leq b_{j,k}, \forall j \in \mathcal{N}, \forall k \in \mathcal{B}, t \in [0,T] \tag{3.2}$$

$$\sum\nolimits_{j \in \mathcal{N}} \sum\nolimits_{k \in \mathcal{B}} x_{i,t}^{j,k} r_{i,t}^{j,k} \geq \hat{r}_i^t, \forall i \in \mathcal{D}, t \in [0,T] \tag{3.3}$$

$$\sum\nolimits_{j \in \mathcal{N}} \sum\nolimits_{k \in \mathcal{B}} x_{i,t}^{j,k} R_j \geq \hat{r}_i^t, \forall i \in \mathcal{D}, t \in [0,T] \tag{3.4}$$

$$\sum\nolimits_{j \in \mathcal{N}} \sum\nolimits_{k \in \mathcal{B}} x_{i,t}^{j,k} (\hat{T}_{i,t}^{j,k} + T_j) \leq \hat{d}_i^t, \forall i \in \mathcal{D}, t \in [0,T] \tag{3.5}$$

$$\sum\nolimits_{j \in \mathcal{N}} \sum\nolimits_{k \in \mathcal{B}} x_{i,t}^{j,k} = 1, \forall i \in \mathcal{D}, t \in [0,T] \tag{3.6}$$

$$x_{i,t}^{j,k} \in \{0,1\}, \forall i \in \mathcal{D}, \forall j \in \mathcal{N}, \forall k \in \mathcal{B}, t \in [0,T] \tag{3.7}$$

In problem (3), constraint (3.1) represents the limitation of wired link resource, where the total transmission rate offered by NS cannot exceed the link resource budget. Constraint (3.2) states the wireless bandwidth limitation, which means that the total wireless bandwidth allocated to devices by NS j via BS k cannot exceed the total bandwidth of NS j allocated from BS k. Constrains (3.3) - (3.5) state that the QoS of devices should be satisfied by its serving BS and NS even the selected NS/BS pair and network environment change. Moreover, constraint (3.6) means that a device can access only one NS via one BS during time slot t. The binary constraint on the decision variable is shown in (3.7). Intuitively, problem (3) can reduce to Multiple Choice Multidimensional Knapsack problem (MMKP). Therefore, problem (3) is NP-hard.

B. Markov Decision Process Modeling for Access Control

As Problem (3) is NP-hard, there is no polynomial-time algorithm for solving it. Meanwhile, in view of the dynamic nature of access conditions, we formulate the access control problem as a Markov Decision Process (MDP) model. An MDP consists of four-tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R)$, where $\mathcal{S}$, $\mathcal{A}$, P, and R respectively represent the state space, the action space, the transition probability between states, and the reward function, which are respectively defined as follows.

State: We assume $\mathcal{S}$ is the set of all network states for all devices, and the number of NSs and BSs are $|\mathcal{N}|$ and $|\mathcal{B}|$ respectively. For a specific device $d_i \in \mathcal{D}$, the state can be represented by $s_t^{\text{i}} = \{I_{\text{i}}, b_{1,1}^t, ..., b_{j,k}^t, ..., b_{|\mathcal{B}|,|\mathcal{N}|}^t\}$, where $s_t^{\text{i}} \in \mathcal{S}$, $I_{\text{i}} = (j,k)$ states the current access scheme of d_i, and $b_{j,k}^t$ represents the available wireless bandwidth allocated to NS j from BS k at the beginning of time slot t.

Action: We assume $\mathcal{A}$ is the set of actions for all devices. For a specific device $d_i \in \mathcal{D}$, let $a_t^{\text{i}} = (j,k,w_{i,t}^{j,k})$ be the action, which means d_i will consume $w_{i,t}^{j,k}$ MHz wireless bandwidth if it accesses to NS j via BS k at time slot t. Here $w_{i,t}^{j,k}$ is selected randomly from $[\hat{r}_i^t / log_2(1 + SINR_{i,t}^k), b_{j,k}^t]$, where $\hat{r}_i^t$ is the minimal transmission rate of d_i at time slot t.

Transition Probability: Let the transition probability of d_i be $P = \left\{ p_{s_t^{\text{i}} s_{t+1}^{\text{i}}}^{a_t^{\text{i}}} | a_t^{\text{i}} \in \mathcal{A}, s_t^{\text{i}}, s_{t+1}^{\text{i}} \in \mathcal{S} \right\}$, which represents the probability that network state of d_i transits from s_t^{i} to s_{t+1}^{i} through action a_t^{i}.

Reward: We define the reward as $r_t = e_t - u \cdot x_t \cdot \alpha_{i,k}^{\text{c}}$, where u is the number of devices, x_t is the number of communication rounds in FL from the first time slot to the tth time slot, and $\alpha_{i,k}^{\text{c}}$ is the communication cost of each round in FL between d_i and BS k. More details about communication round and FL are shown in next section.

In the paradigm of distributed machine learning, federated learning can be exploited to efficiently promote the collaboration between devices while retaining the privacy of local data.

IV. Federated Deep Reinforcement Learning for Access Control

A. Framework of FDRL-AC

By incorporating the DRL into the FL framework, we propose a collaborative federated deep reinforcement learning for access control, called FDRL-AC. Fig. 2 shows the architecture of FDRL-AC, which consists of DRL running on individual devices, and two levels of model aggregation based on DRL: horizontal weights aggregation (called hDRL) and vertical access feature aggregation (called vDRL). Specifically, in hDRL, we exploit hFL for the same type services to aggregate the parameters (*i.e.*, θ_t^{i}) to share the similar data samples, where devices and BSs can be enabled to train a global model (*i.e.*, $g_r(t)$) together without raw data transfer. As the RAN needs to support multiple service types, the selected NS/BS pairs derived from hDRL may be not optimal. Therefore, in vDRL, we exploit vFL to aggregate the access features to form a larger feature space for different types of services (*e.g.*, in the scenario of Fig. 2, there are two service types).

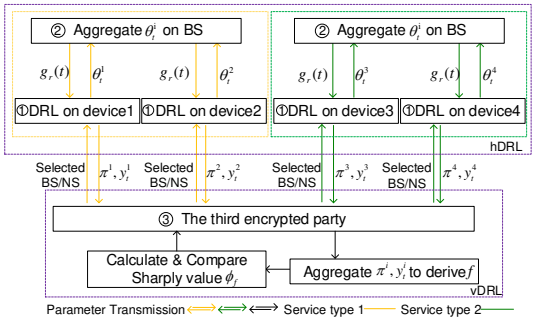

Fig. 2. The federated deep reinforcement learning scheme for access control (FDRL-AC).

DRL on Smart devices: As FL can inherently support privacy protection on private data, the training data should be kept where it is generated. In other words, devices need to train their own data independently. Furthermore, we employ the discrete-action DRL algorithm, double deep Q-Network (DDQN) which can decouple the selection from the evaluation to reduce the correlation between data, to train the local model on individual smart devices. DDQN evaluates the greedy policy according to the Q-network with weights θ and estimates state-action value $Q(\cdot)$ according to the target network $\hat{Q}$ with weights $\hat{\theta}$. The update in DDQN is the same as that in DQN, but the target is replaced by

$$y_t^{\mathrm{i}} = r_{t+1} + \gamma Q(\mathrm{s}_{t+1}^{\mathrm{i}}, \mathrm{argmax}_{\mathrm{a}_t^{\mathrm{i}}} Q(\mathrm{s}_{t+1}^{\mathrm{i}}, \mathrm{a}_t^{\mathrm{i}}; \theta_t^{\mathrm{i}}); \hat{\theta}_t^{\mathrm{i}}), \quad (4)$$

where $\mathrm{argmax}_{\mathrm{a}_t^{\mathrm{i}}} Q(\mathrm{s}_{t+1}^{\mathrm{i}}, \mathrm{a}_t^{\mathrm{i}}; \theta_t^{\mathrm{i}})$ is an ϵ-greedy policy used to select access or handoff actions, and θ_t^{i} is the weight vector of Q-network for device d_i. If d_i satisfies the access condition and takes access/handoff action $\mathrm{a}_t^{\mathrm{i}}$ at the beginning of time slot t, we will obtain the corresponding state-action value, which is given by

$$Q(s_t^{\mathrm{i}}, a_t^{\mathrm{i}}) = \mathbb{E}[\sum_{k=t}^{T} \gamma^k r_t | s_t^{\mathrm{i}}, a_t^{\mathrm{i}}], \quad (5)$$

where $\gamma \in [0, 1]$ is the discount factor representing the discounted impact of the future reward. The objective of DDQN is to minimize the gap between the estimated $Q(\cdot)$ and the target value. Therefore, DDQN running on d_i can be trained by minimizing the loss function, which is given by

$$L(\theta_t^{\mathrm{i}}) = \mathbb{E}[(y_t^{\mathrm{i}} - Q(\mathrm{s}_t^{\mathrm{i}}, \mathrm{a}_t^{\mathrm{i}}; \theta_t^{\mathrm{i}}))^2]. \quad (6)$$

Moreover, the update algorithm in DDQN is given by

$$\theta_{t+1}^{\mathrm{i}} = \theta_t^{\mathrm{i}} + \alpha \left[y_t^{\mathrm{i}} - Q\left(s_t^{\mathrm{i}}, a_t^{\mathrm{i}}; \theta_t^{\mathrm{i}}\right) \right] \nabla Q\left(s_t^{\mathrm{i}}, a_t^{\mathrm{i}}; \theta_t^{\mathrm{i}}\right). \quad (7)$$

After training local data for τ time slots, device d_i will send the local model (*i.e.*, θ_t^{i}) to the BSs to update the global model.

Horizontal Model Aggregation: Once receiving all local models from individual devices, BSs will update the global model as follows,

$$g_r(t) = \frac{\sum_{i=1}^{u_x} K_i \theta_t^{\mathrm{i}}}{K}, \forall 1 \le t \le T, \quad (8)$$

where K_i is the amount of training data of d_i, $K = \sum_{i=1}^{u_x} K_i$ is the total amount of training data of the devices with service of type x, u_x is the number of devices which has the same service type x, and r represents the rth communication round of hDRL. After updating the global model in the rth communication round, BSs will transmit the global mode $g_r(t)$ to all devices with the same type services to update the local DDQN models based on (9).

$$\theta_{t+1}^{\mathrm{i}} = g_r(t) - \frac{\lambda}{K_i} \sum_{i=1}^{u} \nabla L(\theta_t^{\mathrm{i}}), \forall i \in \mathcal{D}, 1 \le t \le T, \quad (9)$$

where λ is the learning rate, and $L(\theta_t^{\mathrm{i}})$ is the loss function of DDQN in (6). After updating the local model, the devices will continue to train their local model. The horizontal model aggregation algorithm is presented as Algorithm 1.

Vertical Model Aggregation: The aforementioned horizontal model aggregation is used for the same type services with similar data samples. As multiple types of services are considered in this paper, vertical model aggregation could be exploited for further improving the network performance, by aggregating local access features incurred from different types of services. Due to the data on each device is private and not visible to other devices, we use a 0-1 matrix to represent a local global decision on NS and BS selection, where we can update global access feature by transforming these 0-1 matrices. In this paper, according to [8], the estimated global target value of a global decision on NS and BS selection is given by

$$\varphi_f = \sum_{i=1}^{u} y_t^{\mathrm{i}} - \mathbb{E}\left[\sum_{i=1}^{u} y_t^{\mathrm{i}}\right], \quad (10)$$

where $f \subseteq \mathcal{X}$ is a specific global access scheme (0-1 matrix), each row vector of f represents a local access scheme. Moreover, y_t^{i} is the target value in (4). In [8], the authors proposed an Monte-Carlo sampling, where the Shapley value is given by

$$\phi_f = \frac{1}{M} \sum_{m=1}^{M} \left(\varphi_{+f} - \varphi_{-f}\right), \quad (11)$$

where M is the number of access feature updates in vDRL. Moreover, ϕ_f is the Shapley value for an access scheme f,

Algorithm 1 Algorithm of Horizon Model Aggregation

Input: s^{i}, a^{i}, α, γ, C, R, K_i, u_x, x, λ, τ
output: NS/BS pair π^{i}, y_t^{i}.

Initialize g_0 and experience relay pool $D_x^{\mathrm{i}}, \forall i \in \mathcal{D}$;
for communication round $r = 1, 2, ..., R$ **do**
  if $r == 1$ **then**
    Initialize θ_0^{i};
  else
    for $i = 1, 2, ..., u_x$ **do**
      $\theta_0^{\mathrm{i}} = g_{r-1}(t) - \frac{\lambda}{K_i} \sum_{i=1}^{u_x} \nabla L(\theta_t^{\mathrm{i}})$.
    end for
  end if
  Let $\hat{\theta}_0^{\mathrm{i}} = \theta_0^{\mathrm{i}}$, $\hat{Q}(\cdot) = \hat{\theta}_0^{\mathrm{i}}$;
  for $t = 1$ to τ **do**
    Receive the initial observed state $s_1^1, s_1^2, ..., s_1^{\mathrm{u}_x}$;
    if $t \leq |D_x^{\mathrm{i}}|$ **then**
      Randomly select $a_t^1, a_t^2, ...$;
    else
      Select $a_t^{\mathrm{i}} = \mathrm{argmax}_a Q(\cdot)$ using ϵ-greedy policy;
      Execute action a_t^{i}, obtain r_t^{i} and s_{t+1}^{i};
      Store $(s_t^{\mathrm{i}}, a_t^{\mathrm{i}}, r_t^{\mathrm{i}}, s_{t+1}^{\mathrm{i}})$ into $D_x^{\mathrm{i}}, \forall i \in \mathcal{D}$;
      Randomly select $\left(s_j^{\mathrm{i}}, a_j^{\mathrm{i}}, r_j^{\mathrm{i}}, s_{j+1}^{\mathrm{i}}\right)$ from D^{i};
      Calculate y_t^{i} according to equation (4);
      Perform gradient descent according to equation (7);
      Update the parameter $\theta_t^{\mathrm{i}}, \forall i \in \mathcal{D}$;
      Every C slots reset $\hat{Q} = Q$;
    end if
  end for
  for $i = 1, 2, ..., u_x$ **do**
    $g_r(t) = \frac{\sum_{i=1}^{u} K_i \theta_\tau^{\mathrm{i}}}{K}$.
  end for
end for
Obtain selected NS/BS pair π^{i}, target value y_t^{i}.

representing the average marginal contribution of f across all possible access feature combinations $\mathcal{X}$. For example, if $\mathcal{X} = \{\mathcal{X}\{1\}, \mathcal{X}\{2\}, \mathcal{X}\{3\}\}$ and $f = \mathcal{X}\{1\}$, we can get the $+f = \mathcal{X}\{1\}$ and $-f$ is randomly chosen in $\{\mathcal{X}\{2\}, \mathcal{X}\{3\}\}$. Therefore, we can obtain the Shapley value ϕ_f through (11) and derive the global optimal 0-1 matrix by comparing the Shapley values. Based on the train Algorithm 1, the FDRL-AC algorithm is presented as Algorithm 2.

V. Numerical Results

In this section, we evaluate the performance of our proposed FDRL-AC scheme through simulation experiments. We employ three reference access control schemes as comparison reference: (1) Greedy Algorithm for access control (GAC): each device chooses NS/BS to access which can provide the maximal available bandwidth. Moreover, the aim of GAC is to find the maximal communication efficiency based on instantaneous network conditions. (2) Centralized DDQN for access control (CAC): all devices transmit data to a controller for centralized training in DDQN. Then the controller makes global decision on NS and BS selection for all devices. (3) Distributed DDQN without data aggregation for access control (DAC): individual devices train their own data through DDQN and make decision on NS and BS selection independently, where no data aggregation of FL is used. Moreover, the reward function in CAC and DAC remains the same as that in FDRL-AC except that the cost of communication round is zero.

Algorithm 2 FDRL-AC Algorithm

Input: M, NS/BS pair π_i, y_t^{i} from **Algorithm 1**, iterations.
output: ϕ_f, f.

Initialize $\phi_{max} = 0$, $f_0 = \varnothing$;
for $m = 1, 2, ..., M$ **do**
  Get π_i, θ_{i}, y_t^{i}, $\mathcal{X}$;
  Remove unfeasible solution in $\mathcal{X}$;
  for $i = 1, 2, ..., |\mathcal{X}|$ **do**
    $f = \mathcal{X}\{i\}$, initial $-f = \varnothing$, and calculate φ_f;
    for iterations$= 1, 2, ...$ **do**
      Choose $-f$ in $\{\mathcal{X} - \mathcal{X}\{i\}\}$;
      if $-f \subseteq F$ **then**
        Continue.
      else
        Calculate φ_{-f}.
      end if
    end for
    Calculate ϕ_f
    if $\phi_{max} \leq \phi_f$ **then**
      $\phi_{max} = \phi_f$, and $f_0 = f$;
    end if
  end for
end for
Obtain ϕ_f, and $f = f_0$.

We consider a network scenario where four BSs are randomly distributed in a square area of 1060×1060 m^2 [4] and five end-to-end slices are deployed in the network. The maximal transmit power and the noise power of BSs are set to 47dBm and -174dBm/Hz respectively, and the path loss for BSs is modeled as $L(d) = 34 + 40\log(d)$ [4]. Furthermore, the wireless bandwidth of a BS is set to 20 MHz. The minimal transmission rate $\hat{r}_i^t$ is randomly generated from [2, 10]Mbps and the delay in CN is randomly generated from [10, 30]ms.

For each device, we consider a three-layer fully connected neural network, including input layer with 12 neurons, hidden layer with 25 neurons, and output layer with 1 neuron. We copy the weights of Q-network θ to the weights of target network $\hat{\theta}$ every 5 training steps. Moreover, the discount factor is set to 0.99, the number of iterations in each communication round is set to 2000, the access feature is updated every 2 communication rounds, and both the learning rate for training and the learning rate for updating local model are set to 0.001.

First, we verify the convergence property of FDRL-AC by depicting its learning curve. We randomly select three corresponding connected weights on three different devices. As

Fig. 3. Convergence of FDRL-AC.

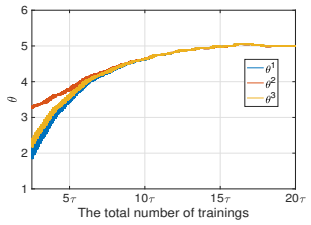

Fig. 4. Partial convergence curve of Fig. 3 within $[5\tau, 20\tau]$.

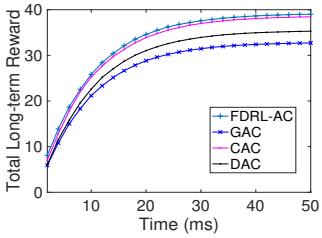

Fig. 5. The performance of the total long-term reward.

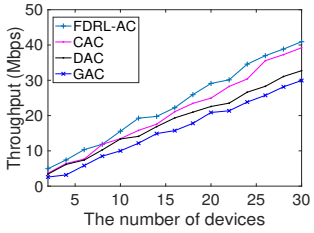

Fig. 6. Comparison of network throughput.

shown in Fig. 3, FDRL-AC converges with the total number of trainings increasing. Furthermore, from Fig. 4, we observe that the three corresponding weights coincide when they tend to be stable, which further illustrates the effectiveness of training a global model with multiple independent devices.

Then, we compare the total long-term reward of the four schemes. Fig. 5 shows the total long-term reward of the four schemes. We can see that FDRL-AC and CAC achieve higher long-term reward than other two schemes. This is because FDRL-AC and CAC aim to find the global optimal access schemes, while DAC and GAC focus on the local access decision.

Next, we examine the performance of the four schemes in terms of network throughput. Fig. 6 shows the network throughput as a function of the number of devices. We can see that FDRL-AC always outperforms CAC, DAC, and GAC on network throughput. This is because that FDRL-AC integrates the similar data samples into a global model before aggregating the access features. Moreover, the access feature aggregation in FDRL-AC takes the global decision into account.

Next, we compare handoff cost of the four schemes. Fig. 7 shows the comparison of handoff cost of the four schemes. We can see that FDRL-AC incurs the highest handoff cost. Moreover, the handoff cost of FDRL-AC and DAC is always higher than that of CAC, this is because FDRL-AC and DAC are based on distributed learning, where smart devices train their own

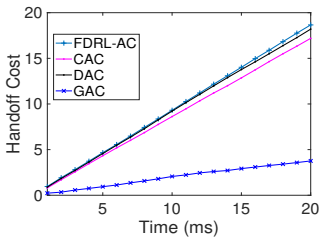

Fig. 7. Comparison of handoff cost.

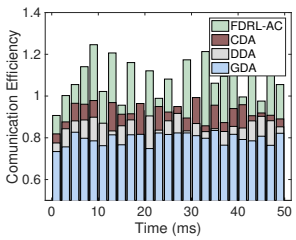

Fig. 8. Comparison of communication efficiency.

data independently. Although FDRL-AC employs two levels of aggregation, training on smart devices independently is not affected.

Finally, we compare the performance of communication efficiency of the four schemes in Fig. 8. We see that FDRL-AC always outperforms DAC, CAC, and GAC in terms of communication efficiency. In particular, numerical results show that FDRL-AC achieves higher communication efficiency by about 14.19%, 20.80%, and 26.60% on average compared with CAC, DAC, and GAC respectively.

VI. Conclusion

In this paper, with the aim to improve network throughput and communication efficiency while enforcing the data security and device privacy, we have modeled the access control problem for RAN slicing as an MDP model and solved it by developing a novel FDRL-AC scheme. In FDRL-AC, we employ two levels of model aggregation based on DRL to promote the collaboration between smart devices. Numerical results show that our proposed FDRL-AC scheme achieves a significant performance improvement in terms of network throughput and communication efficiency when compared with the state-of-the-art algorithms.

References

[1] Ericsson AB, "5G Systems – enabling the transformation of industry and society," *White Paper*, no. January, p. 14.

[2] B. Ojaghi, F. Adelantado, E. Kartsakli, A. Antonopoulos, and C. Verikoukis, "Sliced-ran: Joint slicing and functional split in future 5g radio access networks," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE, 2019, pp. 1–6.

[3] Y. Sun, S. Qin, G. Feng, L. Zhang, and M. Imran, "Service provisioning framework for ran slicing: User admissibility, slice association and bandwidth allocation," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2020.

[4] G. Zhao, S. Qin, G. Feng, and Y. Sun, "Network slice selection in softwarization-based mobile networks," *Transactions on Emerging Telecommunications Technologies*, vol. 31, no. 1, p. e3617, 2020, e3617 ett.3617.

[5] Y. J. Liu, G. Feng, Y. Sun, S. Qin, and Y. C. Liang, "Device association for ran slicing based on hybrid federated deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 15 731–15 745, 2020.

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

[7] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, p. 12, 2019.

[8] G. Wang, C. X. Dang, and Z. Zhou, "Measure contribution of participants in federated learning," *arXiv preprint arXiv:1909.08525*, 2019.

[9] 3GPP, "System architecture for the 5G System (5GS); Stage 2," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 23.501, 03 2020, version 16.4.0.

[10] ETSI, "TS 136 331 - V15.3.0 - LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification." *3GPP TS 36.331 version 13.7.1 Release 13*, vol. 1, pp. 1 – 649, 2018.

[11] Y. Sun, W. Jiang, G. Feng, P. V. Klaine, L. Zhang, M. A. Imran, and Y. C. Liang, "Efficient handover mechanism for radio access network slicing by exploiting distributed learning," *IEEE Transactions on Network and Service Management*, vol. 17, no. 4, pp. 2620–2633, 2020.