Mobile Load Balancing for 5G RAN Slicing in Mobile Network

Hong Xu[1], Liushan Zhou[2,3], Hong Shen[1], and Tiankui Zhang[2(✉)]

[1] China Telecom Co., Ltd., BeiJing Branch, Beijing 100010, China
{xuhong.bj,shenhong.bj}@chiantelecom.cn
[2] Beijing University of Posts and Telecommunications, Beijing 100876, China
zhouliushan@bupt.edu.cn
[3] China Mobile Communications Co., Ltd., HeNan Branch, Henan 450008, China

Abstract. With the rapid development of mobile Internet, network slicing is defined as one of key technologies to deal with the issue of differentiated requirements of diversified services. The introduction of network slicing brings many challenges to the implementation of Radio Access Network (RAN). Considering Mobile Load Balancing (MLB) for RAN slicing, a mobile load balancing algorithm based on Deep Reinforcement Learning (DRL) is proposed. First of all, we propose a system utility model to measure the satisfaction of the system. And a mobile load balancing strategy for RAN slicing is proposed, including slice-level load control realized by adjusting the proportion of radio resource allocated to slice by Radio Remote Unit (RRU), and cell-level load balancing based on handoff. In order to improve the system satisfaction and maximize the system utility, the joint optimization problem of system satisfaction and utility function is proposed. The DRL is used to solve these optimization problems. The simulation results show that the proposed algorithm can effectively reduce the total number of handoffs in the system and bring less balancing overhead. In addition, the proposed algorithm can effectively reduce the number of unsatisfied users and achieve higher system satisfaction.

Keywords: RAN · Network slicing · SLA · Mobile load balancing

1 Introduction

Network slicing has been identified as one of the enabling technologies of 5G, and it will play an important role in future mobile networks [1]. A network slice can be regarded as an independent virtual network to provide services, which meets the diversified communication requirements of users by providing flexible and on-demand network services [2]. By exploiting network slicing technology, the network capabilities in terms of capacity, delay, transmission rate, etc., could be dramatically improved due to the high flexibility and efficiency of resource allocation [3].

In addition to these significant benefits, the introduction of network slicing also brings many challenges to RAN, including network function virtualization,

X. Wang et al. (Eds.): AICON 2021, LNICST 396, pp. 235–250, 2021.
https://doi.org/10.1007/978-3-030-90196-7_21

network resource allocation, and mobility management [3,4]. In particular, MLB is very important to guarantee the service requirements of users while communication environment changes (e.g. channel conditions of different access points, slice resources allocation). It is not only related to ensuring Service Level Agreement (SLA) requirements of users, but also has a significant impact on both slice deployment and radio resource management [5]. Compared with traditional networks, mobile load balancing for RAN slicing is more complicated, which faces the following challenges:

Network Architecture Changes. Specifically, User Equipment (UE) should be accessed with an RAN slicing via a specific RRU, forming a UE-RRU-RAN Slicing three-layer association architecture. Therefore, the load of cell and slice, service type and Reference Signal Receiving Power (RSRP) should be considered simultaneously to guarantee slices' SLA and improve user's service experience.

The Granularity of Load Information Changes. In related specification of MLB enhancement, some slice-related content has been introduced. For example, a slicing load control mechanism is proposed to help to guarantee slicing SLA by using Network Data Analysis Function (NWDAF) for slicing level analysis [6]. At the same time, it is clear that the influence of network slicing should be considered in the mechanism of MLB enhancement [7].

Mobile Load Balancing Strategy Changes. To solve MLB for RAN slicing, cell-level load balancing and slice-level load control need to be considered at the same time. Different from the traditional mechanism, there are several types of balancing strategies, for example, only adjusting the resources of slices, switching between RRUs, or switching between slices and RRUs, or even applying for deploying a new slice. Different types of balancing strategies may lead to different balancing costs. For example, adjusting resource of slices only needs to exchange signaling in the same RRU, which means lower balancing cost, while switching both slice and RRU needs higher cost.

With the development of technology, Reinforcement learning (RL), which is one class of machine learning methods that can adapt to unknown environments by learning from the environmental feedbacks, has already been applied in load balancing [8–10]. These successes envision a bright future to exploit DRL in realizing adaptive and autonomous large-scale load balancing under complex ultradense networks. However, this paradigm still remains to be explored. Some studies have been carried out on the handoff mechanism for RAN slicing, Y. Sun et al. [11] proposed a multi-agent reinforcement learning based smart handoff policy with data Sharing, to reduce handoff cost while maintaining user QoS requirements in RAN slicing. However, many existing researches have not considered how to achieve mobile load balancing based on Network slicing architecture.

Considering mobile network architecture for RAN slicing, we propose a mobile load balancing algorithm based on DRL. First of all, a system satisfaction model is proposed to measure the satisfaction of UEs' requirements. At the same time, the MLB strategy for RAN slicing is proposed, including slice-level load control realized by adjusting the proportion of radio resource allocated to slice

by RRU, and cell-level load balancing based on handoff. In order to improve the system satisfaction and maximize the system utility, the joint optimization problem of system satisfaction and user utility function is proposed in slice-level load control and cell-level load balancing respectively. The Deep Q Network (DQN) is used to solve these optimization problems.

2 System Model

We consider a mobile network architecture based on slicing shown in Fig. 1, which consists of multiple network slices, RRUs and UEs. These slices share the physical resources in the Core Network (CN) and the RAN [12]. Each slice is composed of different network function modules (e.g. mobility management, access control, security, etc.) to provide secure and differentiated services for UE.

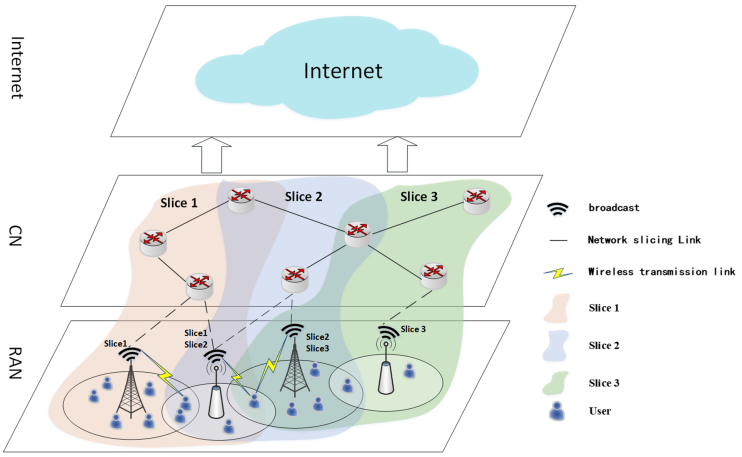

Fig. 1. RAN model based on network slicing.

2.1 Radio Access Network Model

We consider RAN model with multiple RRUs and slices shown in Fig. 1. Let $\mathcal{J} = \{1 \ldots, J\}$, $\mathcal{N} = \{1, \ldots, N\}$ and $\mathcal{U} = \{1, \ldots, U\}$ be the set of RRUs, RAN slices and UEs. N RAN slices are deployed on demand in a mobile network consisting of J RRUs. Multiple RAN slices are deployed on a unified physical infrastructure. Radio resources are allocated according to the requirements of slices. Let $R_u^{\min}$ be the UE's minimum service rate, which represents SLA requirements of UE. Let $\mathcal{T} = \{T_1, T_2, \ldots, T_m\}$ be the set of all service types, and $\varphi_u \in \mathcal{T}$ be the service type of UE u. When the service type can meet the SLA requirements of UE, $\varphi_u = T_m$.

Let $c_{j,n}$ denote the proportion of radio resources allocated to slice by RRU, leading to the constraint

$$\sum_{n\in\mathcal{N}} c_{j,n} = 1 \tag{1}$$

When the RRU j is not in the coverage area of slice n, $c_{j,n}$=0.

Two elements (T_n, $\mathbf{C}_n$) are used to determine a specific slice. For slice n, T_n is the type of service that slice n can provide, and $\mathbf{C}_n$ is a vector representing the proportion of radio resources allocation of slice n from all RRUs, where $c_{j,n}$ is the j element of vector $\mathbf{C}_n$. In Fig. 1, UE can access slice 1 via RRU 1 and RRU 2. The connection indication variable is defined as $x^u_{j,n}$, when UE accesses the network through slice n of RRU j, $x^u_{j,n}$= 1, otherwise, $x^u_{j,n}$= 0. Each UE can only access a slice through one RRU, leading to the constraint

$$\sum_{n\in\mathcal{N}}\sum_{j\in\mathcal{J}} x^u_{j,n} = 1 \tag{2}$$

Let L^u_j be the pathloss between RRU j and UE u, h^u_j and P^u_j are the channel gain and allocated power of the link from RRU j to UE u, respectively. σ^2 is Additive White Gaussian Noise (AWGN) power. Express Signal to Interference plus Noise Ratio (SINR) experienced by UE u of RRU j as

$$\mathrm{SINR}^u_{j,n} = \frac{P^u_j L^u_j \left|h^u_j\right|^2}{\sigma^2 + \sum\limits_{i\in\mathcal{J},i\neq j} P^u_i L^u_i \left|h^u_i\right|^2} \tag{3}$$

According to Shannon formula and (3), the spectrum efficiency can be calculated as

$$e^u_{j,n} = \log_2\left(1 + \mathrm{SINR}^u_{j,n}\right) \tag{4}$$

If $x^u_{j,n}$= 1, radio resources will be allocated to UE u. According to UE's minimum service rate requirement $R^{\min}_u$, the radio resource allocated to UE u on RRU j can be calculated as

$$w^u_{j,n} = \frac{R^{\min}_u}{B \times e^u_{j,n}} \tag{5}$$

where B is the bandwidth of the subcarrier.

Assuming that each RRU has the same total amount of radio resources, Let W_{total} be the maximum amount of radio resources that can be provided to UE, leading to the constraint

$$\sum_{n\in\mathcal{N}}\sum_{u\in\mathcal{U}} x^u_{j,n} w^u_{j,n} \leq W_{total} \tag{6}$$

Moreover, the radio resource utilization of RRU j can be expressed as

$$\Omega_j = \frac{\sum\limits_{n\in\mathcal{N}}\sum\limits_{u\in\mathcal{U}} x^u_{j,n} w^u_{j,n}}{W_{total}}, \forall j \in \mathcal{J} \tag{7}$$

2.2 System Satisfaction Model

In order to measure the system satisfaction, Let Sat_u be the satisfaction of UE u with the slice's SLA requirement. The satisfaction on rate of UE u is calculated with sigmoid function as follows

$$\mathrm{Sat}_u = \frac{1}{1+e^{-\varphi(\Gamma_u - R_u^{\min})}} \tag{8}$$

where Γ_u is the achievable data rate of UE u. If the value of Sat_u is greater than 0.5, UEs are said to be satisfied, and not satisfied otherwise. Based on satisfaction of UE, satisfaction of slice can be calculated as

$$\lambda_n = \frac{\sum_{u=1}^{|U_n|} \mathrm{Sat}_u}{|U_n|} \tag{9}$$

where U_n is the total number of UEs accessing the network via slice n. We consider the worst result achieved by slices for system satisfaction, the system satisfaction is defined as

$$\lambda = \min\{\lambda_n\} \tag{10}$$

3 Mobile Load Balancing Strategy for RAN Slicing

Due to the different arrival locations of users, some RRU will be overloaded, while some RRU will be lightly loaded, resulting in radio resources can not be used effectively, and the satisfaction of UEs can not be well guaranteed. Therefore, it is necessary to judge whether mobile load balancing is necessary according to the current network state and the coverage range of RAN slices, so as to ensure the service experience and SLA requirements of UEs.

Ensuring the SLA requirements of users and realizing the load balancing of the whole network are the key standards to verify the MLB algorithm, and MLB will also affect radio resource allocation and slice deployment scheme in mobile network. Therefore, this paper defines a MLB strategy for RAN slicing, which includes both cell-level load balancing and slice-level load control.

First of all, slice-level load control and cell-level load balancing are performed by monitoring system satisfaction. For slice-level load control, the proportion of radio resources allocated by RRU to the slice needs to be adjusted by analyzing the radio resource utilization of slice from each RRU. For cell-level load balancing, once the overload threshold is met, UE selects the target RRU and slice to switch by analyzing the load of neighbor cells and slice coverage. Slice-level load control and cell-level load balancing will be introduced in detail.

3.1 Slice-Level Load Control

When $c_{j,n} \neq 0$, $\rho_{j,n}$ is used to represent the load of slice n on RRU j, which is defined as

$$\rho_{j,n} = \frac{\sum_{u \in \mathcal{U}} x_{j,n}^u w_{j,n}^u}{c_{j,n} W_{total}} \tag{11}$$

(11) is expressed as the ratio of the sum of radio resources required by UEs to access network via slice n on RRU j to the sum of radio resources allocated to slice n by RRU j. When $\rho_{j,n} < 1$, the radio resources allocated to slice n by RRU j have not been fully utilized; when $\rho_{j,n} = 1$, radio resources allocated to slice n by RRU j are exactly able to meet the SLA requirements of UEs; when $\rho_{j,n} > 1$, it indicates that radio resources allocated by RRU j to slice n are insufficient, some UEs' SLA requirements will not be met.

Therefore, the load of slice can be measured by the number of UEs whose SLA requirements are not met, and satisfaction of slice can be calculated by (9). According to the achievable data rate of UE can, we can also calculate the satisfaction of UE by (8).

If RRU j is not overloaded, but radio resources allocated to slice n are insufficient, the additional radio resources that slice n needs to obtain from RRU j can be expressed as

$$c_{j,n}^{addition} = \rho_{j,n} - 1 \tag{12}$$

The RRU j dynamically updates the proportion of radio resources allocated to each slice by analyzing its own radio resource utilization Ω_j. According to (11), the radio resources utilization of slices on RRU j is analyzed, radio resources allocated to slice with low radio resource utilization are reduced, and the proportion of radio resources allocated to slice n is increased.

Therefore, we define the balancing strategy for slice-level load control as M_{NS}: adjusting the proportion of radio resources $c_{j,n}$.

3.2 Cell-Level Load Balancing

Similar to the method of calculating slice-level load, the load of RRU can be defined by the radio resource utilization of the RRU j, and the load of the RRU j is defined as

$$\rho_j = \Omega_j = \frac{\sum_{n \in \mathcal{N}} \sum_{u \in \mathcal{U}} x_{j,n}^u w_{j,n}^u}{W_{total}}, \forall j \in \mathcal{J} \tag{13}$$

When $\rho_j < 1$, the sum of radio resources required by all UEs accessed to RRU j is less than total radio resources that RRU j can provide, RRU j is not overloaded; when $\rho_j = 1$, RRU j is just fully loaded; when $\rho_j > 1$, RRU j is overloaded. Based on the above analysis, the load that RRU j needs to transfer can be calculated as

$$\rho_z = \rho_j - \varsigma \tag{14}$$

where ς is overload threshold.

The load that each neighbor cell can accept can be calculated as

$$\rho_i = \varsigma - \rho_j, \forall i \in \mathcal{J}, i \neq j \tag{15}$$

For cell-level load balancing, once the overload threshold is met, UE selects target RRU and slice to switch by analyzing the load of neighbor cells and slice coverage. Therefore, we define three types of balancing strategies for cell-level load:

1) M_{BS}: switching between RRUs, UE switches between RRUs under the same slice coverage;
2) M_{NS-BS}: switching both slice and RRU, slices' SLA need to be considered;
3) M_{New}: deploying a new slicing on the light load RRU, which is a special handoff for RAN slicing.

4 Problem Formulation and Algorithm

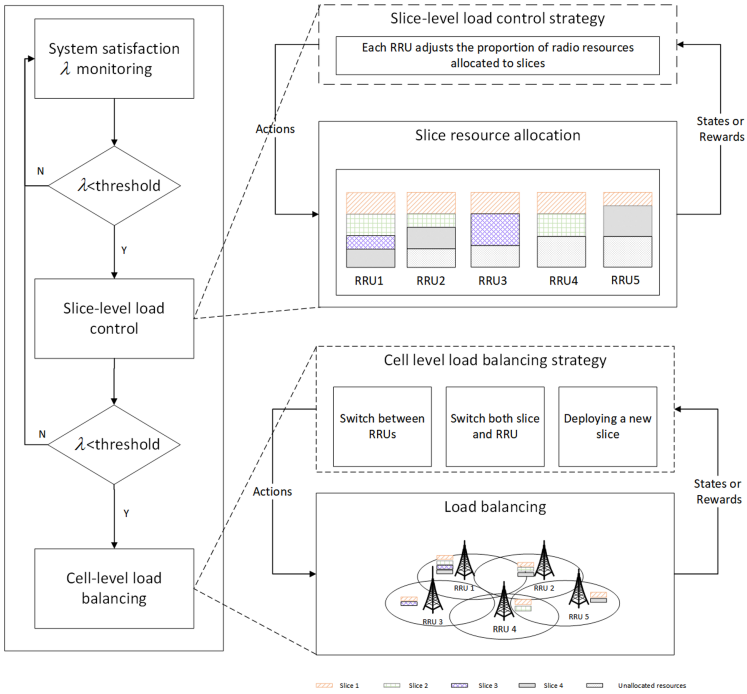

Fig. 2. RAN model based on network slicing.

Based on proposed MLB strategy for RAN slicing, a solution of mobile load balancing for 5G RAN slicing is proposed, as shown in Fig. 2. By monitoring the system satisfaction, the slice-level load control is performed first if system satisfaction does not meet the requirements, the proportion of radio resources allocated by each RRU to slices is adjusted to balance the radio resource utilization and satisfaction of each slice, so as to improve the satisfaction of the system. If the system satisfaction is effectively improved after slice-level load control, the system satisfaction will be continuously monitored; otherwise, cell-level load balancing will be performed, edge users of overloaded RRUs select target RRU and slice to switch to ensure slice performance and users' SLA requirements, and improve the system satisfaction.

4.1 Problem Formulation

In the process of MLB for RAN slicing, it's necessary to consider the system overhead, such as signaling exchange and delay. Therefore, the objective of this paper is to optimize network through slice-level load control and cell-level load balancing, so as to ensure users' SLA requirements and slices' performance, as well as reduce the balancing overhead.

The utility of UE adopting the balancing strategy is composed of achievable data rate and overhead caused by balancing strategy. The data rate of UE is related to the system state, and the gain of achievable data rate can be expressed as

$$g^u = \delta \times \Gamma^u_{j,n} \tag{16}$$

where δ represents the unit price of service rate.

In addition, RRU and slice associated with UE will change after adopting the corresponding balancing strategy, which resulting in the balancing cost. According to the analysis of mobile load balancing strategy in Sect. 3, the cost function generated by balancing can be expressed as

$$\eta^u = \begin{cases} 0, x_{j',n'} = x_{j,n} \\ \beta_{NS} \cdot e^{\rho_j}, x_{j',n'} = x_{j,n}, C'_n \neq C_n \\ \beta_{BS} \cdot e^{\rho_{j'}}, x_{j',n'} \neq x_{j,n}, j' \neq j, n' = n \\ \beta_{NS-BS} \cdot e^{\rho_{j'}}, x_{j',n'} \neq x_{j,n}, j' \neq j, n' \neq n \\ \beta_{New} \cdot e^{\rho_{j'}}, n' \notin \mathcal{N} \end{cases} \tag{17}$$

where, β_{NS}, β_{BS}, β_{NS-BS} and β_{New} are the unit cost of M_{NS}, M_{BS}, M_{NS-BS} and M_{New} respectively. Then the utility function can be expressed as

$$r^u = g^u - \eta^u \tag{18}$$

In this paper, we expect to improve system satisfaction and maximize system utility through mobile load balancing. So we combine system satisfaction and system utility as objective function, which can be expressed as

$$\max_{x,c,\lambda} \lambda \times \sum_{u \in \mathcal{U}} r^u \tag{19}$$

$$s.t. \sum_{n \in \mathcal{N}} \sum_{j \in \mathcal{J}} x^u_{j,n} = 1, \forall u \in \mathcal{U} \tag{19a}$$

$$\sum_{n \in \mathcal{N}} c_{j,n} \leq 1, \forall j \in \mathcal{J} \tag{19b}$$

$$\sum_{n \in \mathcal{N}} \sum_{u \in \mathcal{U}} x^u_{j,n} w^u_{j,n} \leq W_{total}, \forall j \in \mathcal{J} \tag{19c}$$

$$\rho_j < \varsigma, \forall j \in \mathcal{J} \tag{19d}$$

$$\rho_{j,n} \leq 1, \forall j \in \mathcal{J}, \forall n \in \mathcal{N} \tag{19e}$$

Constraint (19a) ensures that each UE can only access a slice via one RRU. Constraint (19b) indicates that the sum of the proportion of radio resource allocated to each slice by this RRU cannot be greater than 1. The total radio resource of each RRU is limited in constraint (19c). Constraint (19d) implies that the load of each RRU cannot exceed overload threshold. Constraint (19e) indicates that the SLA requirements of UEs associated with slice must be met.

Considering the aforementioned challenges including the three layer architecture, SLA guaranteeing as well as different balancing costs, DRL that incorporate information on surrounding environment could be used to design a smart MLB mechanism dedicated for RAN slicing.

4.2 Deep Reinforcement Learning

In this article, We solve mobile load balancing for RAN slicing with DRL. Specifically, DRL aims at maximizing a cumulative reward by selecting a sequence of optimal actions under different system states in a stochastic unknown environment. We denote the state-value function of an arbitrary policy at the time t as follows:

$$V_{\pi}(s) = E_{\pi}\left[\sum_{t=0}^{\infty} \gamma^{t} r(s,a)\right] \tag{20}$$

where s represents the state, a represents the action, π represents the policy, $r(s,a)$ is the reward function, which reflects the learning goal, and $\gamma \in [0,1]$ is the discount factor, indicating the attenuation degree of the reward. According to the dynamic programming equation, there is at least one optimal strategy π^{*} that makes the following equation

$$V_{\pi}^{*}(s) = \max_{a}\left(r(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^{a} V_{\pi}^{*}(s')\right) \tag{21}$$

where s' is the next state when the environment state changes, $\mathcal{P}_{ss'}^{a}$ is the a stationary transition probability.

In DRL-based algorithm, the objective of the agent is to find the optimal policy π^{*} that maximizes the Q-function $Q(s,a)$. The optimal Q-function is used to measure the performance of the agent. Based on the Bellman equation, the optimal policy can be selected according to the following equation

$$Q(s,a) = \omega + \gamma \max_{a'} Q(s',a') \tag{22}$$

where ω is the weight parameter of Q-Network.

4.3 Slice-Level Load Control Based on DQN

For slice-level load control, the system satisfaction will be improved by adjusting the proportion of radio resources allocated to each slice by RRU. When system

satisfaction is decrease, the load control decision is made according to the state. For slice-level load control, the optimization problem reduce in (19) to

$$\max_{c} \lambda \times \sum_{u \in \mathcal{U}} r^u s.t.\,(19a)\,(19b)\,(19e) \tag{23}$$

At each step, the agent will adjust the proportion of radio resource as the conditions of the slices keep changing. We begin to formulate the Markov decision process (MDP) by defining states, actions, reward, and next state as follows.

State: $s^n \in \mathcal{S}^{\mathcal{N}}$ represents the network slicing state ($\mathcal{S}$ is a set of all states). We define the state s^n of slice n, as a tuple $s^n = \{\lambda_n, c_{1,n}, \ldots, c_{j,n}, \rho_{1,n}, \ldots, \rho_{j,n}\}$, where λ_n is satisfaction of slice n, which can be calculated by (9); $c_{j,n}$ is the proportion of radio resources allocated to slice n by RRU j; $\rho_{j,n}$ is the load of slice n on RRU j, which is calculated by (11).

Action: Based on states, the agent learns an optimal radio resource allocation for each slice by selecting an action a^n based on state s^n. The action set of a slice is defined as the proportion of radio resources allocated to slice by RRU, $a^n = \{-0.6, -0.4, -0.2, 0, 0.2, 0.4, 0.6\}$. If the selected action is negative, it means that radio resources of slice should be reduced by this percentage, otherwise, it means that an increase proportion of radio resource of this slice, and a value of zero means the radio resource of slice remains the same.

Reward: When actions is selected from the action set, the reward function is also needed to judge the merits of the selected actions. The objective of slice-level load control should take into consideration several variables as the reward for the learning. We define the reward function as the product of the system satisfaction and the sum of the system utility. If the selected action does not satisfy the constraints (19a), (19b) and (19e), the reward function is set to -1 .

$$r^n = \begin{cases} \lambda \times \sum\limits_{u \in \mathcal{U}} r^u, \mathrm{C}\,(19\mathrm{a})\,, \mathrm{C}\,(19b)\,\mathrm{C}\,(19\mathrm{e}) \\ -1, \text{otherwise} \end{cases} \tag{24}$$

Next State: After receiving the reward for the selected action, it will enter into the next state. The proportion of radio resource allocated to slice, the satisfaction of slice, and the load of slice will be changed. The next state parameters are affected by the action taken on the slice. After updating the parameters, they are stored in memory and used to predict possible actions during training.

The proportion of radio resource of slice will be updated at the decision step. It can be calculate as

$$c'_{j,n} = \begin{cases} c_{j,n}, \text{if} a_j = 0 \\ (1 + a^n)\, c_{j,n}, \text{otherwise} \end{cases} \tag{25}$$

Where a^n is the action taken on the radio resource of slice. It should be noted that these values are updated at each decision step in order to update the radio resources of slice. Finally, the slice allocates the resource to its users. The whole process of the DQN algorithm is given in Algorithm 1.

Algorithm 1. DQN Based Slice-Level Load Control

Input: A replay memory D, action-valuefunction Q, The status information of RAN slices, $s^n = \{\lambda_n, c_{1,n}, \ldots, c_{j,n}, \rho_{1,n}, \ldots, \rho_{j,n}\}$.

Output: Action a^n

while $s = s^n$ **do**
  Select action at satisfying $a = \arg\max_{a \in \mathcal{A}} Q_\pi^*(s, a)$
  Update $c_{j,n}$ by (25) and observe reward by (24) and new state
  Update the proportion of radio resource allocation of slices according to the selected action
  Store experience $\{s, a, r, s'\}$ in the replay memory D
  Calculate output Q-values
  Update target Q-value of action by (22)
  Train Q Network using a loss function as the mean squared error of output and target
  Update $t = t + 1$
end while

4.4 Cell-Level Load Balancing Based on DQN

Similarly, the DRL is also used to solve cell-level load balancing. Target RRU and slice are selected to switch by observing the surrounding environment. We aim to achieve load balancing in a long time and improve the system capacity. For cell-level load balancing, the optimization problem can be expressed as

$$\max_x \lambda \times \sum_{u \in \mathcal{U}} r^u, \text{s.t.} : (19a)\,(19c)\,(19d) \tag{26}$$

Its agent, state, action and reward function are defined as follows. Specifically, the whole process of algorithm is given in Algorithm 2.

State: $s \in \mathcal{S}^{\mathcal{N}}$ represents state of network ($\mathcal{S}$ is a set of all states). The state is composed of the load of RRU j and the fraction of the edge users, as a tuple $s_t = [\rho_1, \ldots, \rho_j, \ldots, \rho_J, E_1, \ldots, E_j, \ldots, E_J]$. Where ρ_j is the load of the RRU, which can be calculated by (13), E_j is the fraction of the edge users. Here, we categorize the edge users according to their downlink SINRs to the corresponding serving RRU and the neighboring RRUs.

Action: An action means selecting both target RRU and RAN slicing when a handoff occurs. In detail, we denote the action taken by UE to access slice via RRU can be expressed as $a = (j, n)$, where j and n is the target RRU and RAN slicing respectively.

Reward: The achievable data rate of UE is affected by balancing strategy. Therefore, the reward function is defined as the joint optimization of system satisfaction and utility function, where the utility function is composed of the

achievable data rate of UE after taking action and balancing overhead. Similarly, if the selected action does not satisfy the constraints (19a), (19c) and (19d), the reward function is set to –1.

$$r = \begin{cases} \sum\limits_{u \in \mathcal{U}} r^u, \mathrm{C}\,(19\mathrm{a})\,, \mathrm{C}(19\mathrm{c}), \mathrm{C}\,(19\mathrm{d}) \\ -1, \text{otherwise} \end{cases} \tag{27}$$

Next State: When enter into the next state, the load of RRU and the satisfaction of system will be changed. The next state parameters are affected by the action taken on the slice. After updating the parameters, they are stored in memory and used to predict possible actions during training.

Algorithm 2. DQN Based Cell-Level Load Balancing

Input: A replay memory D, action-valuefunction Q, The status information, $s_t = [\rho_1, \ldots, \rho_j, \ldots, \rho_J, E_1, \ldots, E_j, \ldots, E_J]$.
Output: Action $a = (j, n)$

while $s = s_t$ **do**
  Select action at satisfying $a = \arg\max_{a \in \mathcal{A}} Q_\pi^* (s, a)$
  Observe reward by (27) and new state
  Select both target RRU and slicing, excute $a = (j, n)$
  Store experience $\{s_t, a, r, s_{t+1}\}$ in the replay memory D
  Calculate output Q-values
  Update target Q-value of action by (22)
  Train Q Network using a loss function as the mean squared error of output and target
  Update $t = t + 1$
end while

5 Performance Evaluation

In this section, we conduct simulation on Python platform and TensorFlow tool to evaluate the performance of proposed mobile load balancing algorithm. We evaluate the performance in terms of system satisfaction and the total number of handoff.

Table 1. Simulation Parameters

Parameters	Value
Number of RRUs	7
Distance between RRUs	1000 m
Number of slices	4
Bandwidth	30 MHz
Transmit power	46 dBm
Noise power	−174 dBm/Hz
UE noise figure	9 dB
Pathloss	$37.6\log_{10}(d\,[km]) + 128.1$ dB
Overload threshold	0.95
Learning rate	0.2
Discount factor	0.1

In order to improve the accuracy and universality of the simulation, the number of RRU is fixed. And some users move randomly in the area, so that the overloaded cells will not be damaged due to the mobility of users. We also consider comparing proposed algorithm with only slice-level load control and only cell-level load control. The specific simulation parameter are shown in Table 1.

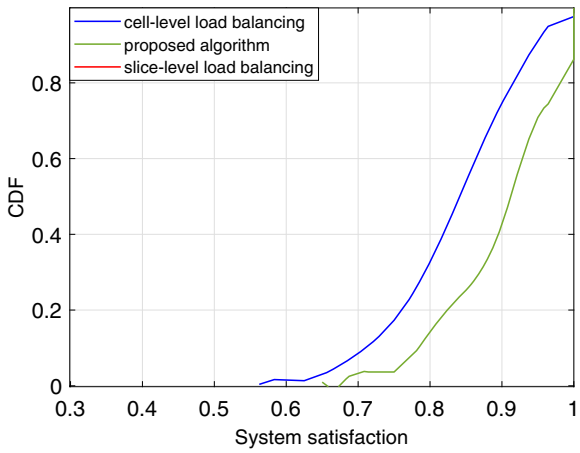

Fig. 3. CDF of system satisfaction.

It is necessary to consider both cell-level load balancing and slice-level load control for the introduction of RAN slicing. Figure 3 shows the distribution of system satisfaction in different situations. It can be seen that the system satisfaction with slice-level load control is distributed between [0.35,1], and the system satisfaction with cell-level load balancing only is distributed between [0.56,1]. The system satisfaction with the proposed algorithm can be effectively improved, which is distributed between [0.65,1]. It indicates that it is important to consider both slice-level load control and cell-level load balancing. In addition, it proves the effectiveness of the proposed algorithm in improving the system satisfaction.

Figure 4 illustrates the average system satisfaction with an increasing number of users. In general, as users increasing, the system satisfaction has decreased obviously. For slice-level load control, it only adjusts the radio resources allocated to slices by each RRU at each decision step. However, the radio resources of each RRU are limited. For overloaded cell, the system satisfaction cannot be effectively improved by adjusting radio resources of slices. For cell-level load balancing, if target RRU and slice are found, user will switch to the target RRU and slice; otherwise, a new slice needs to be deployed in the target RRU, which may resulting in large overhead. After adopting balancing strategy, if the gain of achievable data rate is less than overhead, handoff will not be carried out, so it can not effectively improve the system satisfaction. It can be seen that the performance of the proposed algorithm is always better than the other two cases, which also reflects the superiority of the proposed algorithm.

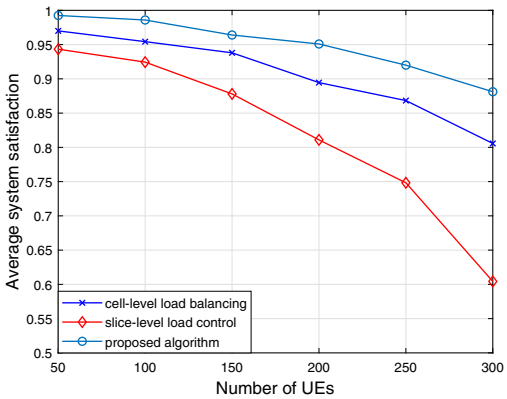

Fig. 4. Average system satisfaction with varying number of UEs.

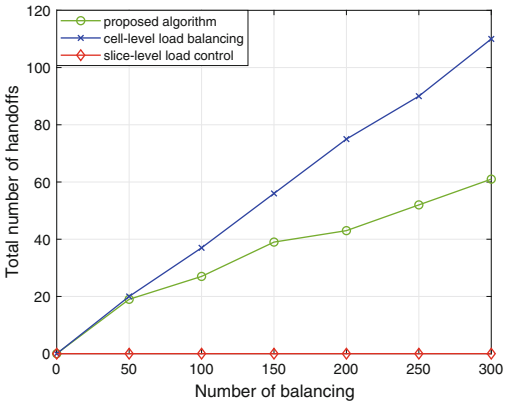

Fig. 5. Total number of handoffs with varying number of balancing.

Number of handoffs with an increasing number of balancing is shown in Fig. 5. As shown in the Fig. 5, the number of handoffs of proposed algorithm is less than cell-level load balancing, except that there is no handoff in slice-level load control. It can be obtained that if proposed algorithm can effectively improve the system satisfaction by slice-level load control, cell-level mobile load balancing will not be performed, the total number of handoffs and overhead will be effectively reduced.

6 Conclusion

To handle with mobile load balancing problem in 5G RAN slicing, we proposed the mobile load balancing strategy and a mobile load balancing algorithm based on DQN. The simulation results show that the proposed algorithm can effectively reduce the total number of handoff in the system and bring less balancing overhead. In addition, the proposed algorithm can effectively reduce the number of unsatisfied users and achieve higher system satisfaction.

References

1. Rost, P., Mannweiler, C., Michalopoulos, D.S., et al.: Network slicing to enable scalability and flexibility in 5G mobile networks. IEEE Commun. Mag. **55**(5), 72–79 (2017)
2. TR 22.864, Feasibility Study on New Services and Markets Technology Enablers - Network Operation, 3GPP, V15.0.0, 09 2016
3. Foukas, X., Patounas, G., Elmokashfi, A., Marina, M.K.: Network slicing in 5G: survey and challenges. IEEE Commun. Mag. **55**(5), 94–100 (2017)
4. Zhang, H., Liu, N., Chu, X., Long, K., Aghvami, A.-H., Leung, V.C.M.: Network slicing based 5G and future mobile networks: mobility, resource management, and challenges. IEEE Commun. Mag. **55**(8), 138–145 (2017)

5. Zhou, L., Zhang, T., Li, J., et al.: Radio resource allocation for RAN slicing in mobile networks. In: IEEE/CIC International Conference on Communications in China (ICCC) 2020, LNCS, pp. 1280–1285. IEEE, China (2020). https://doi.org/10.1109/ICCC49849.2020.9238905
6. TR 23.700, Study on enhancement of network slicing, 3GPP (2020)
7. RP-182105, Study on RAN-centric data collection and utilization for LTE and NR, 3GPP, 09 2018
8. Muñoz, P., Barco, R., Ruiz-Avilés, J.M., de la Bandera, I., Aguilar, A.: Fuzzy rule-based reinforcement learning for load balancing techniques in enterprise LTE Femtocells. IEEE Trans. Veh. Technol. **62**(5), 1962–1973 (2013)
9. Mwanje, S.S., Schmelz, L.C., Mitschele-Thiel, A.: Cognitive cellular networks: a Q-learning framework for self-organizing networks. IEEE Trans. Netw. Serv. Manage. **13**(1), 85–98 (2016)
10. Xu, Y., Xu, W., Wang, Z., Lin, J., Cui, S.: Load balancing for ultradense networks: a deep reinforcement learning-based approach. IEEE Internet Things J. **6**(6), 9399–9412 (2019)
11. Sun, Y., Jiang, W., Feng, G., et al.: Efficient handover mechanism for radio access network slicing by exploiting distributed learning. IEEE Trans. Netw. Serv. Manage. **17**(4), 2620–2633 (2020)
12. An, X., et al.: On end to end network slicing for 5G communication systems. Trans. Emerg. Telecommun. Technol. **28**(4), e3058 (2016)