Distributed Learning based Handoff Mechanism for Radio Access Network Slicing with Data Sharing

Yao Sun*†, Gang Feng* *Senior Member, IEEE*, Lei Zhang† *Senior Member, IEEE*, Paulo Valente Klaine†, Muhammad Ali Imran† *Senior Member, IEEE*, Ying-Chang Liang* *Fellow, IEEE*
* National Key Laboratory of Science and Technology on Communications, University of Electronic Science and Technology of China, Chengdu, China
† School of Engineering, University of Glasgow, Glasgow
Email: fenggang@uestc.edu.cn

***Abstract*—Network slicing (NS) has been identified as a fundamental technology for future mobile networks to meet extremely diverse communication requirements by providing tailored quality of service (QoS). However, due to the introduction of NS into radio access networks (RAN) forming a UE-BS-NS three-layer association, handoff becomes very complicated and cannot be resolved by conventional policies. In this paper, we propose a multi-agent reinforcement LEarning based Smart handoff policy with data Sharing, named LESS, to reduce handoff cost while maintaining user QoS requirements in RAN slicing. Considering the large action space introduced by multiple users and the data sparsity problem due to user mobility, LESS is designed to have two components: 1) LESS-DL, a modified distributed Q-learning algorithm with small action space to make handoff decisions; 2) LESS-DS, a data sharing mechanism using limited data to improve the accuracy of handoff decisions made by LESS-DL. The proposed LESS mechanism uses LESS-DL to choose both the target base station and NS when a handoff occurs, and then updates the Q-values of each user according to LESS-DS. Numerical results show that in typical scenarios, LESS can significantly reduce the handoff cost when compared with traditional handoff policies without learning.**

I. Introduction

It has been widely agreed that network slicing (NS) will play a paramount role in future mobile networks to support highly diverse quality of service (QoS) requirements from end-users [1]–[3]. NS aims to logically separate network functions and resources within a common physical infrastructure, guaranteeing the specific QoS provisioning in different communication scenarios. By exploiting NS 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 configurations [4].

Besides these significant benefits, the introduction of NS also brings many design challenges to the new radio access networks (denoted as RAN slicing throughout this paper), including network function virtualization, network resource allocation, radio frame mixed numerology [5] as well as mobility management [2], [4]. In particular, handoff is crucial for keeping users connected while communication environment changes (e.g., user movement) [6], as it affects not only QoS of users but also network performance in terms of handoff rate, resource utilization, NS re-configuration rate, etc. Considering the NS-based network architecture, conventional reference signal received power (RSRP) based handoff mechanisms [7] are not applicable to RAN slicing. This happens because the target base station (BS) could not provide the required service for users if only considering RSRP when handoffs occur, thus RSRP-based handoff policy is not able to achieve the aim to provide guaranteed QoS for mobile users. Therefore, it is obligatory to design new handoff mechanisms dedicated for RAN slicing.

Indeed, handoff in RAN slicing is much more complicated when compared with that in the traditional cellular networks. Specifically, user equipments (UEs) should be associated with an NS via a specific BS, forming a UE-BS-NS three-layer association structure. Therefore, both the service type of NSs and the RSRP of BSs need to be considered to guarantee the QoS of UEs when handoffs occur. In addition to QoS, we also need to take handoff cost into consideration in RAN slicing. Unlike that in traditional networks, there are several types of handoff, e.g., switch NS only, switch BS only, switch both, or even apply for deploying a new NS. Different types of handoff with specific level of signaling exchange may incur different handoff costs. For example, switching NS only needs to exchange signaling in the same BS, implying a low handoff cost, while switching both NS and BS requires a large handoff cost. Therefore, considering the aforementioned challenges including the three layer associations, QoS guaranteeing as well as different handoff costs, artificial intelligence tools that incorporate information on surrounding environment could be used to design a smart handoff mechanism dedicated for RAN slicing.

In this paper, we propose a multi-agent LEarning based Smart handoff policy with data Sharing, named LESS, for RAN slicing. Our design objective is to minimize the long-term handoff cost while guaranteeing the QoS of UEs. Considering the high action space introduced by multiple users in the learning framework and the limited collected data due to user mobility, LESS is designed to consist of two parts, namely LESS-DL and LESS-DS. In LESS, we use LESS-DL to choose

This work was supported by the National Science Foundation of China under Grant number 61631005 and 61871099, the U.K. Engineering and Physical Sciences Research Council under Grant EP/S02476X/1, and the Basic Research on Innovation Projects under Grant number IFN2018206.

both the target BS and NS when a handoff occurs, and then update Q-values according to LESS-DS. Specifically, LESS-DL is a modified distributed Q-learning algorithm to reduce the action space. It allows each UE to separately update its own Q-value and make handoff decisions according to its own Q-table. LESS-DS is a data sharing mechanism to tackle insufficient data issues, which could exist around the BSs that UEs hardly associate with before. LESS-DS updates the Q-value for UEs with the same QoS by sharing the reward when handoff decisions are made, thus it requires less data to obtain accurate Q-values. Combing the two parts, LESS is practicable to RAN slicing. Numerical results show that in typical scenarios, LESS can significantly reduce the handoff cost when compared with traditional handoff policies without learning.

In the following, we describe our system model in Section II. Then, we propose LESS-DL handoff decision algorithm and LESS-DS data sharing mechanism in Section III and IV respectively. In Section V, we present the numerical results. Finally, Section VI concludes this paper.

II. System Model

We consider an NS-based mobile network architecture shown in Fig. 1, which consists of multiple end-to-end NSs, BSs, as well as UEs. These NSs share the physical resources in both core networks and RAN. Each NS has different network function modules, such as connection management, mobility management, security, etc., thus to provide a specific service for UEs. The detailed descriptions of the network architecture can be found in [8]. Here we focus on RAN slicing from the mobility management perspective.

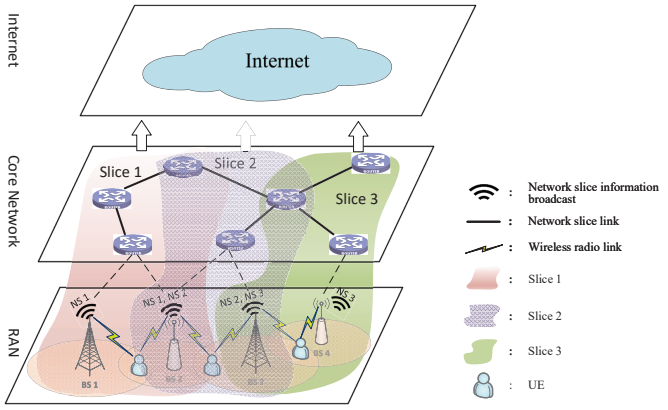


Fig. 1. NS-based mobile network architecture.

A. Radio Access Network Model

We consider a multi-BS and multi-NS RAN model shown in Fig. 1. Let $\mathcal{B}$, $\mathcal{N}$ and $\mathcal{U}$ be the set of BSs, NSs and UEs, respectively. We assume that UEs in the system move at a random speed and in a random direction. Similar to that in [9], we use two parameters to describe QoS requirements: minimum threshold of transmission rate γ_i^{min} and endurable time τ_i, which is the maximum time a UE is allowed to have the transmission rate lower than the minimum threshold. Let $\mathcal{T} = \{T_1, T_2, \ldots, T_L\}$ be the set of all service types, and $\psi_i \in \mathcal{T}$ be the service type of UE i. We say $\psi_i = T_n$ when both γ_i^{min} and τ_i can fulfill the requirement of the service type T_n.

We identify a specific NS, say NS j, by the two elements $(\mathcal{T}_j, \mathbf{B}_j)$, where $\mathcal{T}_j$ is the set of service types that NS j can provide, and $\mathbf{B}_j$ is a vector denoting the bandwidth allocation of NS j from all BSs. Let $\bar{b}_j^{(k)}$ be the k-th element of vector $\mathbf{B}_j$ denoting the bandwidth of NS j allocated by BS k. $\bar{b}_j^{(k)} = 0$ when BS k is not in the coverage of NS j. UEs can access to the NS via only the covered BSs. In the example of Fig. 1, UEs can access to NS 1 via only BSs 1 and 2.

B. Handoff Model

We describe the handoff model from two aspects: handoff trigger condition and handoff cost. Handoff should occur once the QoS of the UE cannot be satisfied [9]. Based on the definition of QoS, the handoff trigger condition for UE n can be written as

$$\forall t_0 \in [t - \tau_n, t], r_n(t_0) < \gamma_n^{min}, \tag{1}$$

where $r_n(t_0)$ is the achievable transmission rate of UE n at time t_0. This condition states that UE n cannot achieve the minimum rate requirement γ_n^{min} in the last τ_n time.

Once the handoff trigger condition is met, UEs need to select suitable target BSs and NSs. As aforementioned, each type of handoff corresponds to a specific handoff cost. Here we define 4 handoff costs generated by the 4 handoff types: 1) C_{NS}, switch NS only; 2) C_{BS}, switch BS only; 3) C_{NS-BS}, switch both NS and BS; 4) C_{New}, deploy a new NS (this type can be seen as a special handoff in RAN slicing); with the relationship $C_{NS} < C_{BS} < C_{NS-BS} < C_{New}$. Based on this, we design a handoff mechanism to minimize the overall handoff cost through target BS and NS selections while guaranteeing the QoS of UEs.

III. Multi-Agent Reinforcement Learning based Handoff Framework

In this section, we first formulate the handoff decision problem as a multi-agent reinforcement learning (RL) model, and then propose an intelligent handoff mechanism based on the learning model, LESS.

A. Multi-Agent RL Model for Handoff

Once the handoff trigger condition for a UE is met, it should choose an appropriate serving NS and BS in order to maintain the desired QoS. We model this target BS and NS selection problem as a multi-agent RL consisting of four main elements: agents, states, actions and reward. In detail, each UE is an agent to make handoff decisions. The states are defined as the available bandwidth levels of NSs. Let $s_j^k(t)$ denote the available bandwidth level of NS j via BS k at time t after discretization. The environment state can be written as $S(t) = \left(s_j^k(t)\right)_{(|\mathcal{B}||\mathcal{N}|)\times 1}$ at time t.

An actions means selecting both target BS and NS when a handoff occurs. In detail, we denote by $\boldsymbol{a}_i(t) = (x_i(t), y_i(t))$ the action taken by UE i at time t, where $x_i(t)$ and $y_i(t)$ is the target BS and NS respectively. If $y_i(t) \notin \mathcal{N}$, the action denotes deploying a new NS. Let $\mathcal{A}$ be the action space for a UE, and thus the system action space for all UEs is $\mathcal{A}^{|\mathcal{U}|}$. The reward denoted by $r_i(S(t), \boldsymbol{a}_i(t))$ is the handoff cost for UE i in state $S(t) \in \mathcal{S}$ with action $\boldsymbol{a}_i(t) \in \mathcal{A}$ at time t, which can be expressed as

$$r_i(S(t), \boldsymbol{a}_i(t)) = \begin{cases} C_{NS}, & \text{if } x_i(t) = x_i(t-1),\ y_i(t) \neq y_i(t-1), \\ C_{BS}, & \text{if } x_i(t) \neq x_i(t-1),\ y_i(t) = y_i(t-1), \\ C_{NS-BS}, & \text{if } x_i(t) \neq x_i(t-1),\ y_i(t) \neq y_i(t-1), \\ C_{New}, & \text{if } y_i(t) \notin \mathcal{N}. \end{cases} \tag{2}$$

Our objective is to minimize the total long-term handoff cost $\sum_{t=1}^{\infty}\sum_{i=1}^{|\mathcal{U}|} r_i(S(t), \boldsymbol{a}_i(t))$ by designing an intelligent handoff mechanism. Traditional Q-learning algorithm [10] is widely used to get the optimal solution to RL problems. However, considering the requirements of mobile networks, two issues prevent the use of traditional Q-learning algorithm to solve the proposed problem: 1) the system action space $\left|\mathcal{A}^{|\mathcal{U}|}\right|$ is very large, implying that Q-learning algorithm needs a long time to converge; 2) Q-learning requires enough data to obtain accurate Q-values, which can be problematic if UEs do not visit some BSs frequently leading to insufficient exploration of the environment. To address these two issues, we propose the LESS handoff mechanism in the following.

B. Framework of LESS Handoff Mechanism

LESS mainly consists of two parts: LESS-DL and LESS-DS as shown in Fig. 2. LESS-DL is a modified distributed Q-learning algorithm to choose the target BS and NS for each individual UE when handoffs occur. LESS-DL allows each UE to separately update its own Q-value and make handoff decisions according to its own Q-table, thus the action space is reduced to $|\mathcal{A}|$. By modifying the Q-value update method, LESS-DL can also converge to the optimal policy, which will be illustrated later.

LESS-DS is a data sharing mechanism to overcome the data sparsity problem mentioned before. FAs some BSs are not visited by a specific UE frequently, not enough data is gathered to update the UE's Q-values, thus the handoff decisions may be not optimal. Considering that the UEs served by the same NSs should have similar QoS requirements, we design LESS-DS to update Q-values of UEs with the same service type when handoff decisions are made, requiring less data to obtain accurate Q-values. In the following, we illustrate LESS-DL and LESS-DS in detail.

IV. LESS-DL Algorithm for Target BS and NS Selection

Q-learning is a simple yet effective algorithm for solving RL problems, and it can be briefly described as follows. Denote

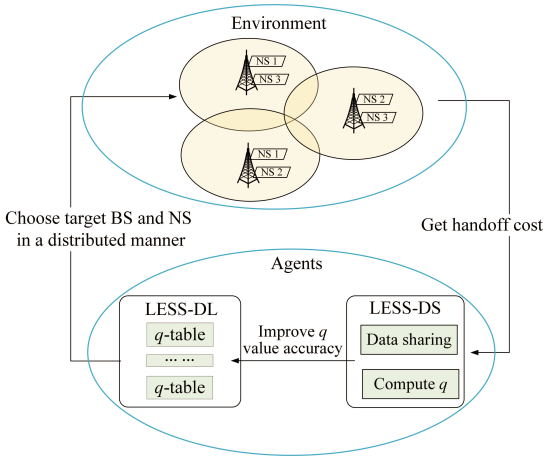


Fig. 2. The framework of LESS handoff mechanism.

by vector $\mathbf{A} = \left[\boldsymbol{a}_1(t), ..., \boldsymbol{a}_{|\mathcal{U}|}(t)\right] \in \mathcal{A}^{|\mathcal{U}|}$ the actions for all UEs. $Q_t(S, \mathbf{A})$ and $r(S, \mathbf{A})$ is respectively the Q-value and reward for state-action pair $(S, \mathbf{A}) \in \mathcal{S} \times \mathcal{A}^{|\mathcal{U}|}$, where $r(S, \mathbf{A}) = \sum_{i=1}^{|\mathcal{U}|} r_i(S(t), \boldsymbol{a}_i(t))$. The update rule of Q-value can be expressed as:

$$\begin{aligned} &Q_0(S, \mathbf{A}) = M, \text{ for all } \mathbf{A} \in \mathcal{A}^{|\mathcal{U}|} \text{ and } S \in \mathcal{S}, \\ &Q_{t+1}(S, \mathbf{A}) = \\ &\begin{cases} Q_t(S, \mathbf{A}), & \text{if } \mathbf{A}(t) \neq \mathbf{A} \text{ or } S(t) \neq S, \\ r(S, \mathbf{A}) + \beta \min\limits_{\mathbf{A}' \in \mathcal{A}^{|\mathcal{U}|}} Q_t(S(t+1), \mathbf{A}'), & \text{otherwise}, \end{cases} \end{aligned} \tag{3}$$

where $S(t)$ and $\boldsymbol{A}(t)$ is respectively the state and the action vector at time t, M is a large constant for initialization and $\beta (0 < \beta < 1)$ is the discount factor. The target BS and NS selection policy is that choosing the NS-BS pair with the smallest Q-value with respect to ϵ-greedy policy.

However, applying traditional Q-learning to solving our problem requires a large action space (i.e.,$\left|\mathcal{A}^{|\mathcal{U}|}\right|$), which takes a long time to converge. Moreover, it requires all UEs to make handoff decisions simultaneously, which is unrealistic. Thus, to overcome these issues, we develop a distributed learning algorithm, LESS-DL, shown in Fig. 3 to select target BS and NS for each individual UE. The main idea of LESS-DL is that each UE only needs to maintain a reduced Q-table where the action space is composed of his own actions without distinguishing the actions from other UEs.

A. q-Value Update Policy

Denote by q^i-table the reduced Q-table maintained by UE i, and $q_t^{(i)}(S, \boldsymbol{a}_i)$ the Q-value of UE i at time t with state-action pair $(S, \boldsymbol{a}_i)$. For convenience, we use q-value and Q-value to denote the value in the reduced and original Q-table respectively. Using a similar idea of [11], $q_t^{(i)}(S, \boldsymbol{a}_i)$ can be

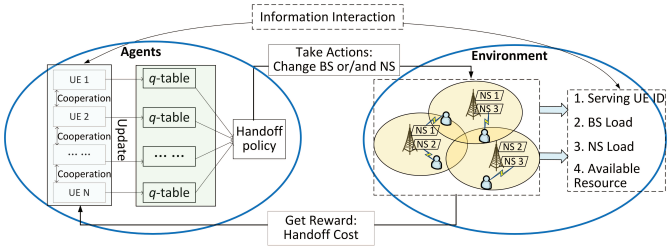


Fig. 3. The framework of LESS-DL.

updated as:

$$q_0^{(i)}(S,\boldsymbol{a}_i) = M, \text{ for all } \boldsymbol{a}_i \in \mathcal{A} \text{ and } S \in \mathcal{S},$$
$$q_{t+1}^{(i)}(S,\boldsymbol{a}_i) =$$
$$\begin{cases} q_t^{(i)}(S,\boldsymbol{a}_i), \text{ if } \boldsymbol{a}_i(t) \neq \boldsymbol{a}_i \text{ or } S(t) \neq S, \\ \min\left\{q_t^{(i)}(S,\boldsymbol{a}_i), r_i(S,\boldsymbol{a}_i) + \beta \min\limits_{\boldsymbol{a}' \in \mathcal{A}} q_t(\boldsymbol{a}', \delta(S,\mathbf{A}))\right\}, \\ \qquad\qquad \text{otherwise.} \end{cases} \tag{4}$$

By using this update method, we can get the reduced q-tables for all UEs. Although the reduced q-table of all UEs cannot construct the original big Q-table, the following proposition gives a very good property of the reduced q-table, which makes it possible to take actions in a distributed manner.

Proposition 1. *The value of $q_t^{(i)}(S,\boldsymbol{a}_i)$ in the reduced q^i-table is the minimum value in the original Q-table defined in (3) when the action of UE i is $\boldsymbol{a}_i$, i.e.,*

$$q_t^{(i)}(S,\boldsymbol{a}_i) = \min_{\mathbf{A} \in \mathcal{A}^{|\mathcal{U}|} \boldsymbol{a}^{(i)} = \boldsymbol{a}_i} Q_t(S,\mathbf{A}) \tag{5}$$

where $\boldsymbol{a}^{(i)}$ denotes the i-th element in action vector $\mathbf{A}$.

Proof: Similar to that in [11], we can easily obtain our Proposition 1 by replacing all the notation 'max' to 'min'. ■

Proposition 1 states that by using the q-value update method in (4), we can obtain the minimum value $Q_t(S,\mathbf{A})$, which makes it possible to design an optimal NS-BS selection policy for UEs in a distributed manner. In the following, we illustrate how to use the q-value to obtain the optimal target BS-NS selection policy.

B. Optimal Action Policy

For traditional Q-learning, we know that once we get a proper converged Q-table, the policy that we choose the action with the smallest Q-value can guarantee the optimality [10]. However, for the reduced q^i-table, if we choose the action with the smallest value $q_t^{(i)}(S,\boldsymbol{a}_i)$ for each UE, it cannot guarantee the optimal policy. In other words, choosing action $\boldsymbol{a}_i{}^*$ with the smallest value $q_t^{(i)}(\boldsymbol{a}_i{}^*, S)$ for each UE may not obtain the optimal action-vector $\mathbf{A}^*$ of all UEs with the smallest Q-value $Q_t(\mathbf{A}^*, S)$ [11], i.e., we cannot guarantee that

$$\left[\boldsymbol{a}_1{}^*, \boldsymbol{a}_2{}^*, \ldots, \boldsymbol{a}_{|\mathcal{U}|}{}^*\right] = \mathbf{A}^*. \tag{6}$$

To overcome this issue, we use the following policy to choose actions. The main idea is to update and store an action policy parallelly with $q_t^{(i)}(S,\boldsymbol{a}_i)$ update. Once the value of $\min q_t^{(i)}(S,\boldsymbol{a}_i)$ decreases, we update our action policy as we can find a better action, and then we store the better one as the current optimal action. When the q-table converges, implying that the value of $\min q_t^{(i)}(S,\boldsymbol{a}_i)$ stays unchanged, our action policy is stable, and the current stored policy is optimal. The update rule of stored action policy $\pi_t^{(i)}(S)$ of UE i is stated as:

$$\pi_0^{(i)}(S) \in \mathcal{A}, \text{ arbitrarily,}$$
$$\pi_{t+1}^{(i)}(S) =$$
$$\begin{cases} \pi_t^{(i)}(S), \text{if } S \neq S_t \text{ or } \min\limits_{\boldsymbol{a}_i \in \mathcal{A}} q_t^{(i)}(S,\boldsymbol{a}_i) = \min\limits_{\boldsymbol{a}_i \in \mathcal{A}} q_{t+1}^{(i)}(S,\boldsymbol{a}_i), \\ \boldsymbol{a}_i(t), \text{ otherwise.} \end{cases} \tag{7}$$

where $\boldsymbol{a}_i(t)$ is the action of UE i at time t.

From [11] we can get the corollary that for an arbitrary state S, we have

$$\left[\pi_t^{(1)}(S), \pi_t^{(1)}(S), \ldots, \pi_t^{(|\mathcal{U}|)}(S)\right] = \arg \min_{\mathbf{A} \in \mathcal{A}^{|\mathcal{U}|}} Q_t(S,\mathbf{A}). \tag{8}$$

Thus, when we get the converged q-table, choosing the current stored action $\pi_t^{(i)}(S)$ for each individual UE can guarantee the minimum handoff cost.

In general, based on ϵ-greedy policy LESS-DL can be described as: before q-value is converged, we choose the current stored policy $\pi_t^{(i)}(S)$ as the target NS-BS pair for each UE respectively with probability $p = (1-\epsilon)$, and choose other pairs randomly with probability $p = \epsilon$. Then, we update the q-values in a distributed manner according to (4) by using the obtained handoff cost. Finally, we update the current stored action policy based on (7) for the next handoff decision. Once we get the converged q-tables, we always choose the current stored action as the target NS-BS pair for each individual UE.

V. LESS-DS for Data Sharing

From the proposed LESS-DL algorithm, we know that it requires enough data to get the accurate q-value, and thus to achieve the minimum handoff cost. However, some unexplored BSs do not get enough data to update q-values, and the handoff performance maybe degraded. For convenience, we call this area as low-frequency activity (LFA) area. To overcome the insufficient data issue in LFA areas we propose a data sharing policy LESS-DS cooperating with LESS-DL algorithm, shown in Fig. 4. The main idea of LESS-DS is that the q-value of a UE should be updated by not only its own data but also the data of others who have the same service type.

Denote by $\phi(t)$ the agent who is making decisions at time t. For a specific q^i-table maintained by UE i, the q-value update policy based on LESS-DS can be described as follows. If $\phi(t) = i$, the update policy of $q_{t+1}^{(i)}(S,\boldsymbol{a}_i)$ is the same as

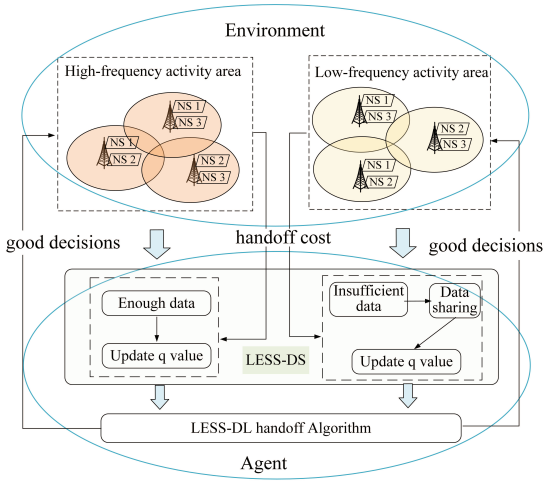


Fig. 4. LESS-DS based LESS handoff mechanism.

(4). If $\phi(t) = j,\ (j \neq i)$, the update policy of $q_{t+1}^{(i)}(S, \boldsymbol{a}_i)$ is:

$$q_{t+1}^{(i)}(S, \boldsymbol{a}_i) = \begin{cases} q_t^{(i)}(S, \boldsymbol{a}_i), \text{ if } \boldsymbol{a}_i(t) \neq \boldsymbol{a}_i \text{ or } S(t) \neq S \text{ or } \psi_i \neq \psi_j, \\ \min\left\{q_t^{(i)}(S, \boldsymbol{a}_i), p_t^{(i,j)}(S, \boldsymbol{a}_i)\right\}, \text{ otherwise,} \end{cases} \tag{9}$$

where ψ_i is the service type of UE i and $p_t^{(i,j)}(S, \boldsymbol{a}_i) = \Gamma_t^{(i)}(S, \boldsymbol{a}_i)^{\alpha}\left[\rho \cdot r_j(S, \boldsymbol{a}_i) + \beta \min_{\boldsymbol{a}' \in \mathcal{A}} q_t(\boldsymbol{a}', \delta(S, \mathbf{A}))\right]$ is defined as the calculated q-value for UE i by using the handoff cost generated by UE j. In $p_t^{(i,j)}(S, \boldsymbol{a}_i)$, $\Gamma_t^{(i)}(S, \boldsymbol{a}_i)$ is the number of times that UE i chooses action $\boldsymbol{a}_i$ with state S until time t, $\alpha > 0$ and $\rho > 1$ are parameters.

Here we give some explanations for this LESS-DS based q-value update policy. We use the same way as (4) to update the q-value when the handoff decision is made by UE i. If the decision is made by other UEs (e.g., UE j), we update the q-value according to (9). In (9), if $\Gamma_t^{(i)}(S, \boldsymbol{a}_i)$ is a large number, implying that the area covered by the corresponding BS-NS pair is frequently visited by UE i, the value of $p_t^{(i,j)}(S, \boldsymbol{a}_i)$ could be larger than $q_t^{(i)}(S, \boldsymbol{a}_i)$, and thus we keep $q_{t+1}^{(i)}(S, \boldsymbol{a}_i) = q_t^{(i)}(S, \boldsymbol{a}_i)$. This means that in the non-LFA areas, we do not use other UEs' data, while in LFA areas where $\Gamma_t^{(i)}(S, \boldsymbol{a}_i)$ is small, we use the handoff cost $r_j(S, \boldsymbol{a}_i)$ generated by UE j with the same service type to calculate $p_t^{(i,j)}(S, \boldsymbol{a}_i)$ and to update the q-value $q_{t+1}^{(i)}(S, \boldsymbol{a}_i)$ of UE i. To avoid decreasing $q_{t+1}^{(i)}(S, \boldsymbol{a}_i)$ value excessively, we add a punishment factor $\rho > 1$. The effectiveness of LESS-DS can also be verified by our simulations in Section VI.

Combing LESS-DL and LESS-DS, we propose the LESS handoff mechanism that runs as follows. When a handoff occurs, we use LESS-DL algorithm to choose the target BS and NS in a distributed manner. Then we update q-values according to LESS-DS mechanism. The updated q-value is used by making decisions when the next handoff occurs.

VI. Simulation and Numerical Results

In this section, we compare the performance of LESS with three other handoff mechanisms: Max-SINR, NS-Prior and LESS-DL. In detail, Max-SINR first selects the BS with the maximum signal-to-interference-plus-noise ratio (SINR) for UEs [7], and then finds the NS deployed in this BS with satisfied QoS provisioning. If such a BS-NS pair is found, select them as the target, otherwise deploy a new NS that satisfies the UE's QoS. NS-Prior mechanism first selects the NS that satisfies the QoS requirement of UEs, and then finds the BS covered by this NS with sufficient bandwidth. Lastly, by comparing with LESS-DL, we can verify the effectiveness of LESS-DS data sharing policy. The handoff trigger condition is the same for all the four mechanisms in (1).

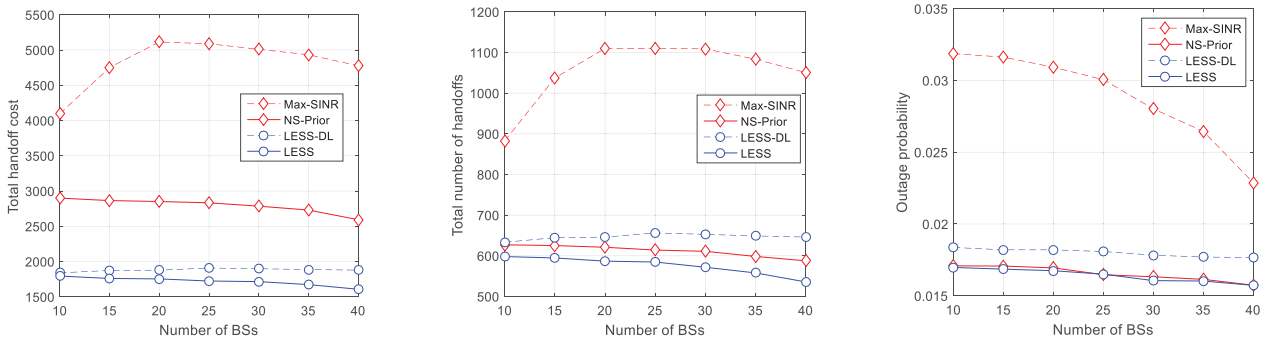
We consider a network which consists of a macro BS (MBS) located at the central of a circular area with a radius of 1000m and varying number of pico BSs (PBS), femto BSs (FBS) and UEs. The number of deployed NSs is 40. Each NS covers 8 BSs randomly, and provides different rate and delay (in term of τ_n in our model) performance. The transmit power of MBS, PBS and FBS is set to 46dBm, 30dBm and 20dBm, respectively [12]. All the BSs share a 20MHz bandwidth, and allocate them to the deployed NSs based on the NS QoS provisioning. UEs are randomly distributed in the area with different rate and delay requirements. In the following, we examine the performance of the proposed LESS handoff mechanism.

In the first experiment, we compare the handoff cost, the number of handoffs and the UE outage probability for the four handoff mechanisms when the number of BSs varies from 10 to 40 as shown in Fig. 5. As expected, we find that the handoff cost of the two learning based mechanisms LESS and LESS-DL is much lower than that of the other two traditional mechanisms shown in Fig. 5(a). In particular, when compared with Max-SINR, NS-Prior and LESS-DL, the handoff cost gain of LESS is about 51%, 40% and 10%, respectively when the number of BS is 25. From Fig. 5(b), we can see that LESS achieves the smallest number of handoffs, while the number of handoffs in LESS-DL is higher than that in NS-Prior due to the lack of data gathered from the LESS-DS policy. Finally, in terms of UE outage probability, Fig. 5(c) shows that NS-Prior achieves the performance as good as LESS due to the prior consideration of NS service provisioning.

Next, we compare the handoff performance for the four handoff mechanisms with different number of UEs, as shown in Fig. 6. Fig. 6(a) shows that the learning based mechanisms LESS and LESS-DL significantly outperform the other two in term of handoff cost. As the learning objective is handoff cost, the performance in terms of the number of handoffs and UE outage probability of LESS is very close to that of NS-Prior mechanism, which considers NS service type when making handoff decisions. This can be concluded from Figs. 6(b) and 6(c).

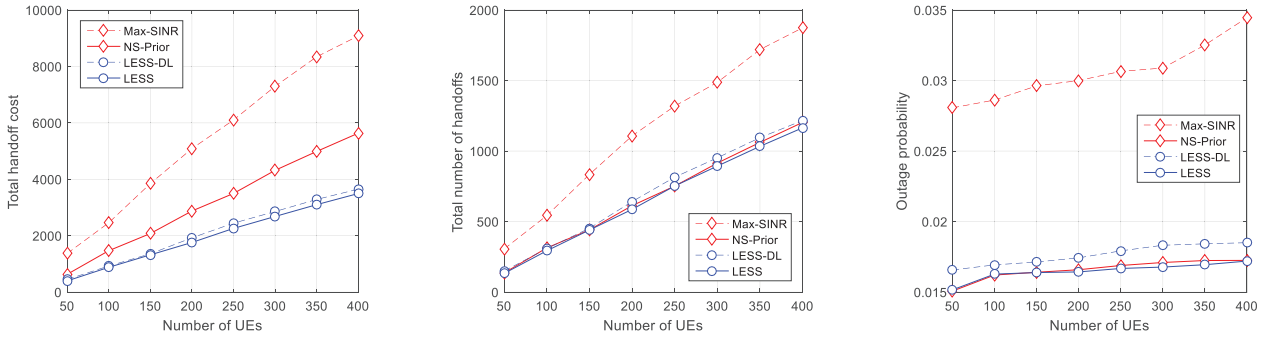
VII. Conclusions

In this paper, we proposed the LESS handoff mechanism for RAN slicing based on multi-agent RL with the aim of



(a) Handoff cost (b) Number of handoffs (c) Outage probability of UEs

Fig. 5. Comparisons of handoff performance for the four handoff mechanisms with different number of BSs.



(a) Handoff cost (b) Number of handoffs (c) Outage probability of UEs

Fig. 6. Comparisons of handoff performance for the four handoff mechanisms with different number of UEs.

minimizing the long-term handoff cost while guaranteeing the QoS of UEs. To make it practicable to mobile networks, LESS is designed to contain two parts, namely LESS-DL, a modified distributed Q-learning algorithm with small action space to select target BSs and NSs when handoffs occur, and LESS-DS, a data sharing policy using limited data to improve the accuracy of handoff decisions made by LESS-DL. Numerical results showed that LESS can significantly reduce the handoff cost by about 50% compared with traditional handoff policies.

REFERENCES

[1] 3GPP TR 23.799, "Study on Architecture for Next Generation System," 2016.

[2] H. Zhang, N. Liu, X. Chu, K. Long, A. H. Aghvami, and V. C. Leung, "Network Slicing Based 5G and Future Mobile Networks: Mobility, Resource Management, and Challenges," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 138–145, 2017.

[3] A. Ijaz, L. Zhang, M. Grau, A. Mohamed, S. Vural, A. U. Quddus, M. A. Imran, C. H. Foh, and R. Tafazolli, "Enabling massive iot in 5g and beyond systems: Phy radio frame design considerations," *IEEE Access*, vol. 4, pp. 3322–3339, 2016.

[4] X. Foukas, G. Patounas, A. Elmokashfi, and M. K. Marina, "Network Slicing in 5G: Survey and Challenges," *IEEE Communications Magazine*, vol. 55, no. 5, pp. 94–100, 2017.

[5] L. Zhang, A. Ijaz, P. Xiao, A. Quddus, and R. Tafazolli, "Subband filtered multi-carrier systems for multi-service wireless communications," *IEEE Transactions on Wireless Communications*, vol. 16, no. 3, pp. 1893–1907, 2017.

[6] H. Tabassum, M. Salehi, and E. Hossain, "Mobility-Aware Analysis of 5G and B5G Cellular Networks: A Tutorial," *arXiv preprint arXiv:1805.02719*, pp. 1–19, 2018. [Online]. Available: http://arxiv.org/abs/1805.02719

[7] 3GPP TS 36.331, "E-UTRA Radio Resource Control (RRC); Protocol specification (Release 9)," 2016.

[8] X. An, C. Zhou, R. Trivisonno, R. Guerzoni, A. Kaloxylos, D. Soldani, and A. Hecker, "On end to end network slicing for 5G communication systems," *Transactions on Emerging Telecommunications Technologies*, vol. 28, no. 4, 2016.

[9] Y. Sun, G. Feng, S. Qin, Y.-C. Liang, and T.-S. P. Yum, "The SMART Handoff Policy for Millimeter Wave Heterogeneous Cellular Networks." *IEEE Transactions on Mobile Computing*, vol. 17, no. 6, pp. 1456–1468, 2018.

[10] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3-4, pp. 279–292, 1992. [Online]. Available: http://link.springer.com/10.1007/BF00992698

[11] M. Lauer and M. Riedmiller, "An algorithm for distributed reinforcement learning in cooperative multi-agent systems," in *the Seventeenth International Conference on Machine Learning*, 2000, pp. 535–542.

[12] Q. Ye, B. Rong, Y. Chen, M. Al-shalash, C. Caramanis, and J. G. Andrews, "User Association for Load Balancing in Heterogeneous Cellular Networks," *IEEE Transactions on Wireless Communications*, vol. 12, no. 6, pp. 2706–2716, 2013.