Efficient Handover Mechanism for Radio Access Network Slicing by Exploiting Distributed Learning

Yao Sun, Wei Jiang, Gang Feng, *Senior Member, IEEE*, Paulo Valente Klaine, Lei Zhang, *Senior Member, IEEE*, Muhammad Ali Imran, *Senior Member, IEEE* and Ying-Chang Liang, *Fellow, IEEE*

***Abstract*—Network slicing is identified as a fundamental architectural technology for future mobile networks since it can logically separate networks into multiple slices and provide tailored quality of service (QoS). However, the introduction of network slicing into radio access networks (RAN) can greatly increase user handover complexity in cellular networks. Specifically, both physical resource constraints on base stations (BSs) and logical connection constraints on network slices (NSs) should be considered when making a handover decision. Moreover, various service types call for an intelligent handover scheme to guarantee the diversified QoS requirements. As such, in this paper, a multi-agent reinforcement LEarning based Smart handover Scheme, named LESS, is proposed, with the purpose of minimizing handover cost while maintaining user QoS. Due to the large action space introduced by multiple users and the data sparsity caused by user mobility, conventional reinforcement learning algorithms cannot be applied directly. To solve these difficulties, LESS exploits the unique characteristics of slicing in designing two algorithms: 1) LESS-DL, a distributed Q-learning algorithm to make handover decisions with reduced action space but without compromising handover performance; 2) LESS-QVU, a modified Q-value update algorithm which exploits slice traffic similarity to improve the accuracy of Q-value evaluation with limited data. Thus, LESS uses LESS-DL to choose the target BS and NS when a handover occurs, while Q-values are updated by using LESS-QVU. The convergence of LESS is theoretically proved in this paper. Simulation results show that LESS can significantly improve network performance. In more detail, the number of handovers, handover cost and outage probability are reduced by around 50%, 65%, and 45%, respectively, when compared with traditional methods.**

***Index Terms*—Handover, RAN Slicing, Multi-agent Reinforcement Learning, Quality of Service**

I. INTRODUCTION

Network slicing has been widely accepted as a novel technology that will be of extreme importance in future mobile networks to support highly diverse quality of service (QoS) requirements from end-users [1]–[3]. Network slicing is defined as a technology that logically separates network functions and resources into multiple network slices (NSs) within a common physical infrastructure. Each NS represents an independent virtualized end-to-end network, providing tailored service for a specific communication scenario (e.g., enhanced Mobile Broadband, massive Machine Type Communication, or Ultra Reliable Low Latency Communications [4]). As such, due to its high flexibility and flexibility in terms of resource configuration, network slicing can provide great improvements in terms of network capacity, latency, transmission rate and reliability [5].

However, despite these benefits, network slicing also introduces many design challenges to the sliced radio access networks (referred as RAN slicing throughout this paper), such as network function virtualization, physical layer mixed-numerology coexistence, network resource allocation, and mobility management, to name a few [6]–[10]. Regarding mobility management, one critical component is the handover process, as it is essential for keeping users connected while they traverse the mobile network [11]. However, despite being essential, handovers bring other technical challenges to the network domain, as they can directly affect not only the performance of end users, in terms of QoS, but also of the overall network performance in terms of the amount of resources utilized, how many handovers occur, and (with the introduction of NS) NS re-configuration frequency, among others [12]. Thus, the introduction of NS is expected to bring other challenges in terms of handover design, as the conventional reference signal received power (RSRP) based handover schemes [13] will not be suitable for RAN slicing. This occurs because, if only RSRP information is used for making handover decisions, the target base station (BS) might not be able to provide the required service type for different users, which by its turn, will cause a severe increase in outage probability. On the other hand, although in the case that the target BS can provide the required service type, the achievable QoS performance of users may be poor due to the limited resource. In general, RSRP-based handover scheme cannot guarantee QoS provisioning for mobile users, which is exactly the main idea of network slicing. Therefore, it is of paramount importance to consider new handover mechanisms dedicated to the RAN slicing domain.

As it can be seen, designing handover procedures for RAN

The work was supported by the National Science Foundation of China under Grant number 61971099, China Postdoctoral Science Foundation under Grant 2019M663476, and U.K. Engineering and Physical Sciences Research Council under Grant EP/S02476X/1. (*Corresponding author: Wei Jiang.*)

Y. Sun, G. Feng and Y.-C. Liang are with the National Key Lab on Communications, and the Center for Intelligent Networking and Communications, University of Electronic Science and Technology of China, Chengdu 610051, China (e-mail: sunyao@.uestc.edu.cn; fenggang@uestc.edu.cn; liangyc@ieee.org.).

Wei Jiang is with the Shenzhen Key Laboratory of Digital Creative Technology, the Guangdong Province Engineering Laboratory for Digital Creative Technology, College of Electronics and Information Engineering, Shenzhen University, Shenzhen, China. (e-mail: weijiang@szu.edu.cn)

P. V. Paulo, L. Zhang and M. A. Imran are with the School of Engineering, University of Glasgow, Glasgow, G12 8QQ, UK (e-mail: {Paulo.ValenteKlaine, Lei.Zhang, Muhammad.Imran}@glasgow.ac.uk.).

slicing can be much more complicated than in traditional cellular networks. In the case of RAN slicing, user equipments (UEs) are now associated with not only a specific BS, but also a NS, forming a three-layer association relationship, UE-BS-NS. Therefore, in addition to the RSRP, now the service type of the NS should also be incorporated in order to guarantee the QoS of UEs when handovers occur. However, differently than traditional networks, in RAN slicing several handover scenarios are possible, e.g., switching NS only when UE changes service type; switching BS only when UE moves; switching both when UE moves with changed service type; or even deploy a new NS when the existing available NSs cannot provide the required service. Moreover, different types of handovers require different levels of signaling, thus resulting in different handover cost. For example, handover performed by switching only the NS of a UE should cost lower than that of a handover in which both NS and BS are changed. Thus, in order to address the aforementioned challenges, including the three-layer association, the diverse set of QoS requirements from UEs as well as different handover costs, it is imperative to resort to new effective techniques.

One promising method is to exploit artificial intelligence tools for designing a smart handover algorithm for RAN slicing. Specifically, reinforcement learning can be expected to solve such sequential decision problem under complex RAN slicing environment by continuously using trial and error learning process with environment interactions thus to optimize long-term handover performance. Moreover, some information such as user movement trajectory, channel quality, available communication resources, etc. is usually time-varying and cannot be obtained or described accurately and timely. Hence, learning tools come to rescue in designing handover schemes with certain unknown information in complex environment, such as our RAN slicing.

In this paper, a multi-agent LEarning based Smart handover Scheme (LESS) for RAN slicing is proposed, with the aim of minimizing long-term handover cost while also guaranteeing the QoS of UEs. In order to solve the scalability issues of Q-learning and the low accuracy of the value function due to limited data collected by each user in the network, the proposed LESS framework is divided into two parts. The first part, LESS-DL, chooses both the target BS and NS when a handover occurs, while the second, LESS-QVU, updates the Q-values of the Q-learning algorithm. More specifically, LESS-DL is a distributed Q-learning with a reduced action space. This allows each UE to separately update its own Q-value and make its own handover decisions without loss of the global optimality by maintaining a respective optimal action policy in parallel. LESS-QVU, on the other hand, is a modified Q-value update algorithm which uses data sharing in order to tackle the lack of collected data, such as around BSs where few UEs have been associated before. LESS-QVU updates the Q-value for UEs based on the traffic similarity of an NS, *i.e.*, UEs who access the same NSs share the reward when handover decisions are made due to similar service provisioning of this NS. Thus, the proposed solution requires less data to obtain accurate Q-value estimates. The convergence of LESS is theoretically proved in this paper. Comparing the performance of the proposed solution with other state-of-the-art approaches show that LESS can significantly improve not only network handover cost, but also the number of handovers and outage probability.

In the following, we overview the related work in Section II, and describe our system model in Section III. Then, we propose the learning based handover scheme LESS in Section IV, and elaborate LESS-DL for handover decisions and LESS-QVU for Q-value update with data sharing in Section V. In Section VI, we discuss the implementation of LESS, and present numerical results in Section VII. Lastly, conclusions are drawn in Section VIII.

II. Related Work

We overview the related work on handover schemes for traditional cellular network and RAN slicing respectively.

A. Handover Schemes for Traditional Cellular Networks

In recent years, research on handover is mainly focused on heterogeneous cellular networks (HetNets) consisting of different types of BSs. A number of handover schemes have been proposed to optimize the instantaneous or long-term network performance in terms of the number of handovers, system throughput, outage probability, load balance, etc.

Starting from handover on instantaneous network performance. The authors of [14] develop a handover framework for HetNets based on game theory to improve energy efficiency. In [15], the authors first determine the candidate BSs by considering the constrains in terms of signal strength, BS load and UE dwelling time, and then use bargaining game model for resource allocation thus to reduce handover occurrence ratio as well as call drop probability. The authors of [16] investigate handover management in dense networks by considering network topology. They propose several handover skipping schemes to avoid unnecessary handovers in dense networks. Works [17] and [18] are mainly focused on the improvement of handover trigger conditions to optimize handover performance in terms of the number of unnecessary handovers [17], [18] and handover failure rate [17].

In recent years, some researchers began to investigate intelligent handover schemes aiming to optimize long-term network performance by using machine learning [19] and/or Markov decision process (MDP). In [20], we use reinforcement learning to solve the huge redundant handover issue in millimeter wave heterogeneous networks. Both handover trigger conditions and target BS selections are investigated in our work aiming at reducing the number of redundant handovers while guaranteeing the QoS of users. The authors of [21] propose a handover scheme based on an MDP model. The proposed BS selection scheme considers context parameters, such as user speed, channel gains and cell load information to maximize UE average capacity. The authors of [22] propose a proactive handover decision scheme for cognitive radio networks to reduce redundant handovers based on MDP model. Unfortunately, due to the lack of considering NS, the aforementioned handover schemes cannot be directly applied to RAN slicing.

B. Handover Schemes for RAN Slicing

Thus far, only a few researchers focus on the handover issue in RAN slicing. The authors of [2] and [11] point out that the handover should be one of the key issues in RAN slicing, while no handover mechanisms or algorithms are studied in their work. The authors of [23] propose a new network architecture based on NS to support mobility management between different radio access technologies, including 4G, Wi-Fi and 5G. They investigate the problem from network architecture perspective, and no specific handover mechanism is proposed in their work either. In [24] a novel handover scheme for integrated train-ground systems based on virtualized wireless networks is proposed. As the resource virtualization is only considered in core networks, the handover scheme in [24] is not appropriate for RAN slicing. To the best of the authors' knowledge, so far there is no specific handover mechanism proposed for RAN slicing to optimize the handover performance.

III. System Model

In this paper, a mobile network architecture with multiple end-to-end NSs, BSs and UEs is considered, as shown in Fig. 1. The coverage of each NS is indicated by different colors and filling shapes, and the areas with multiple colors/shapes are the overlapping area of the corresponding NSs. The forwarding routers in overlapping area are shared by the multiple NSs. It is assumed that NSs share physical resources in both RAN and core network domains. Further to that, it is considered that each NS has different network function modules to support different service types, i.e., connection and mobility management, security, etc.. A more detailed discussion on sliced network architectures can be found in [25]. Since in this paper the focus is on the handover procedure, we focus on mobility management in RAN slicing.

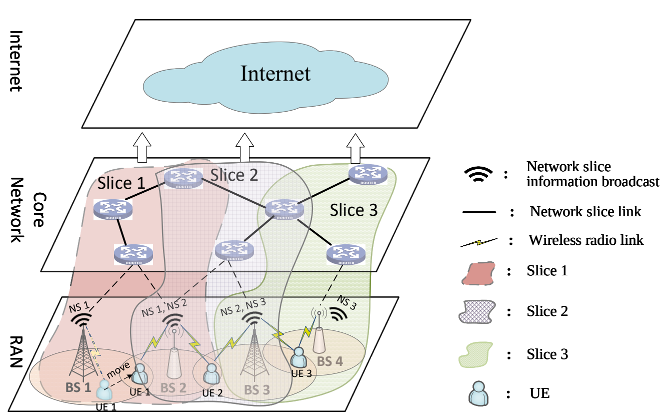


Fig. 1. NS-based mobile network architecture.

A. RAN Slicing Model

As shown in Fig. 1, a multi-BS and multi-NS RAN model is considered. Assume that $\mathcal{B}$, $\mathcal{N}$ and $\mathcal{U}$ denote the set of BSs, NSs and UEs, respectively. Moreover, it is considered that each UE moves at random speeds and random directions (a random mobility model). Regarding user requirements, a model similar to that in [20] is considered, in which two parameters are assumed in order to describe a UE QoS requirements. These parameters are:

- γ_i^{min}, which represents a minimum threshold of transmission rate;
- τ_i, which is an endurable time, or in other words, the maximum time a UE is allowed to have its 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 that UE i requires. We say $\psi_i = T_n$ when both γ_i^{min} and τ_i can meet the requirement of the service type T_n.

Further to that, a specific NS is also defined by two components:

- $\mathcal{T}_j$, which represents the service type of NS j provisions;
- $\mathbf{B}_j$, a bandwidth allocation vector of NS j from all BSs. The value of $\mathbf{B}_j$ is fixed, which means that bandwidth allocation for NS is static.

Given that $\bar{b}_j^{(k)}$ is the k-th element of vector $\mathbf{B}_j$, denoting the bandwidth of NS j allocated by BS k, when $\bar{b}_j^{(k)} = 0$ BS k is not in the coverage of NS j. For example, in the case of Fig 1, UEs can only access slice 1 via BSs 1 or 2, since BSs 3 and 4 do not cover that slice. For the way of bandwidth allocation to users, we assume that NS allocates the minimal required bandwidth to fulfil UE QoS requirement [26].

B. Handover Model

Before presenting the handover model considered in this paper, let us discuss two components of the handover procedure: its trigger condition and cost. In RAN slicing architecture, once the specific QoS of a UE is not satisfied, a handover should occur [23]. Thus, based on the QoS definition, the handover trigger condition for UE i can be expressed as Thus, based on the QoS definition, the handover trigger condition for UE i can be expressed as

$$\forall t_0 \in [t - \tau_i, t], r_i(t_0) < \gamma_i^{min}, \tag{1}$$

where $r_i(t_0)$ is the achievable transmission rate of UE i at time t_0. This condition states that UE i cannot achieve the minimum rate requirement γ_i^{min} in the past τ_i time. Here we should point out that the following design on target NS and BS selection will not be affected although the handover trigger condition changes. This is because that the handover trigger condition and target NS, BS selection are decoupled, and Section IV and Section V paper focus on how to optimize NS, BS selection when handovers occur.

Thus, whenever this condition is met, a US should select an appropriate target NS and BS to handover to. However, each type of handover incurs in a different cost. Thus, based on the handover procedure in RAN slicing [2], we define the handover cost by the four handover types as follows:

1) C_{NS}, the cost of switching serving NS only. This type of handover occurs when the UE changes the service type while staying in the coverage of the same BS. Therefore, the signaling exchanges only occur between the two NSs within the same BS. Some handover procedures such as synchronization, handover confirmation between different BSs and even data forwarding can be avoided. Thus, only a small amount of signaling overhead is caused for this type of handover.
2) C_{BS}, the cost of switching serving BS only. A handover of this type happens when a UE moves out of the coverage of the source BS with unchanged service type. The signaling of this type of handovers should be exchanged between two different BSs. Since the serving NS remains the same, the handover procedure is similar to that in conventional cellular networks. Thus, the handover cost C_{BS} should be greater than C_{NS}.
3) C_{NS-BS}, the cost of switching both the serving NS and BS. This type of handover happens because of UE movement and the change of service type. Thus, some additional handover processes should be performed, such as admission control for the new NS, handover confirmation between the two involved NS, resource allocation from the new NS, etc.. Therefore, the handover cost C_{NS-BS} is greater than C_{BS}.
4) C_{New}, the cost of deploying a new NS, which is exclusive to RAN slicing. If we cannot find an appropriate target NS to guarantee the QoS of the handover user, a new NS should be created for the user to provide the required service (This is consistent with the main idea of network slicing, providing tailored service for individual users). Thus, both the extra resources (in terms of power, bandwidth and computing) as well as the new network function chains should be introduced, leading to an extremely high handover cost for this type of handover.

Thus, we have $C_{NS} < C_{BS} < C_{NS-BS} < C_{New}$. Based on this, a novel handover procedure is designed with the goal of minimizing the overall handover cost, through the selection of both NS and BS, while guaranteeing user's QoS. Note that minimizing handover cost also implies in the improvement in terms of network signaling exchanges/overhead, the number of handovers and outage probability. Moreover, the cost value of these four types of handover may affect the absolute value of handover cost, but do not invalidate the relative performance enhancement of our proposed LESS scheme. This is because that LESS is a general solution based on a reinforcement learning (RL) model, which employs an obtained reward to guide the system to perform actions with the aim of optimizing long-term performance through interacting with the environment.

IV. Multi-Agent RL based Handover Framework

In this section, we first state the handover decision problem, and then formulate it as a multi-agent RL model. Finally, we propose an intelligent handover scheme, LESS, based on a modified distributed learning model.

A. Handover Decision Problem Description

Once the handover trigger condition (*i.e.*, equation (1)) for a UE is met, a handover decision should be performed in order to maintain the requested UE QoS by assigning the appropriate NS and BS. This is accomplished by making handover decisions, with the objective of minimizing the long-term handover cost while guaranteeing the desired QoS, subject to constrains on UE mobility, bandwidth resources, channel conditions and QoS provisioning of NSs. After careful investigation, the problem of selecting a target BS and NS can be formulated as a multi-agent RL model due to the complicated wireless environments, the long-term design objective as well as the multiple UEs in the network. Specifically, as mentioned before, the wireless environments are rather complicated in RAN slicing because of the three layer UE-BS-NS association association relationship, diversified QoS guaranteeing as well as multiple types of handover, requiring interactions with the environment for making handover decisions. Moreover, the long-term design objective can be achieved through an RL framework rather than traditional static optimization methods. Lastly, due to the resource competition among the UEs, intuitively a multi-agent model of RL framework could be exploited to achieve a near-optimal solution for the handover decision problem.

B. Multi-Agent RL Model for Handover

The proposed multi-agent RL model has four main components: agents, states, actions and a reward. More specifically, in our problem each UE acts as an agent that performs handover decisions. Similar to that in [20], the available bandwidth is discretized, and the available bandwidth of NS j and BS k at a specific time t is denoted by $s_j^k(t)$, after the discretization. Regarding the environment, a state is defined as the available bandwidth level of NSs, and thus the environment state at time t is represented by $S(t) = \left(s_j^k(t)\right)_{(|\mathcal{B}||\mathcal{N}|)\times 1}$.

On the other hand, regarding the UE, its actions consist of selecting a target NS and BS whenever a handover occurs. Moreover, the action space of this problem should contain the type of handover (i.e., switch NS/BS, switch the both or even create a new NS). This is because even the target BS and NS are same, the type of handover as well handover cost could be different as the different current serving BS and NS. Therefore, more specifically, an action taken by UE i at time t can be expressed by $\boldsymbol{a}_i(t) = (x_i(t), y_i(t), z_i(t))$, where $x_i(t)$, $y_i(t)$ and $z_i(t)$ represent the target BS, NS and handover type, respectively. It is also important to mention that if $y_i(t) \notin \mathcal{N}$, the action performed by the UE will result in a new NS being deployed. With $\mathcal{A}$ being the action space of a given UE, the action space of all UEs is given by $\mathcal{A}^{|\mathcal{U}|}$. Lastly, the system reward, given as $r_i(S(t), \boldsymbol{a}_i(t))$, corresponds to the handover cost of UE i at state $S(t) \in \mathcal{S}$ when performing action $\boldsymbol{a}_i(t) \in \mathcal{A}$ at time t. Mathematically, the reward is expressed as

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

Based on the aforementioned multi-agent RL framework, the optimization objective consists of minimizing the long-term handover cost, or in other words, the long-term reward, given by $\sum_{t=1}^{\infty} \sum_{i=1}^{|\mathcal{U}|} r_i(S(t), \boldsymbol{a}_i(t))$, which can be achieved by an intelligent handover mechanism. One of the most popular RL algorithms is Q-learning, introduced in [27]. However, despite Q-learning being extensively utilized in RL to find optimal or near optimal solutions, in the mobile networks domain, the performance of Q-learning can be degraded, mainly because of two issues. First, since UEs can have several different options of NS and BS to choose from, the system action space, $\left|\mathcal{A}^{|\mathcal{U}|}\right|$, can be extremely large, which can lead to a longer convergence time. Secondly, the original Q-learning algorithm is guaranteed to converge to an optimal solution only when all possible state-action pairs have been visited an infinite (or near infinite) number of times. However, if this condition is not met, the algorithm does not have enough data in order to accurately estimate the Q-values, which will eventually lead to a non-optimal solution. As such, using the standard Q-learning in the proposed model can be troublesome, as if UEs do not choose certain NS or BSs very frequently, the value of the Q will be inaccurate, leading to non-optimal solutions. Thus, in order to overcome these issues a novel handover scheme, named LESS, is proposed next.

C. Framework of LESS Handover Scheme

The proposed LEarning based Smart handover Scheme (LESS) consists of two algorithms: LESS-DL and LESS-QVU, shown in Fig. 2. LESS-DL consists of a distributed implementation of Q-learning, that chooses a target BS and NS for each UE in the network whenever handovers occur. By cooperating with UEs, LESS-DL is able to reduce the action space from $\left|\mathcal{A}^{|\mathcal{U}|}\right|$ to $|\mathcal{A}|$ without compromising handover performance when compared to the traditional Q-learning. More specifically, in LESS-DL each UE updates its Q-values and make handover decisions independently from one another (based only on its own Q-table). However, the calculation rule of the Q-value is modified, such that the minimum value of the original Q-table can be recovered from the distributed Q-tables of individual UEs. Besides these distributed Q-tables, LESS-DL also requires UEs to maintain a currently optimal policy in parallel, which guarantees the global optimality of the selected actions. We will elaborate LESS-DL later.

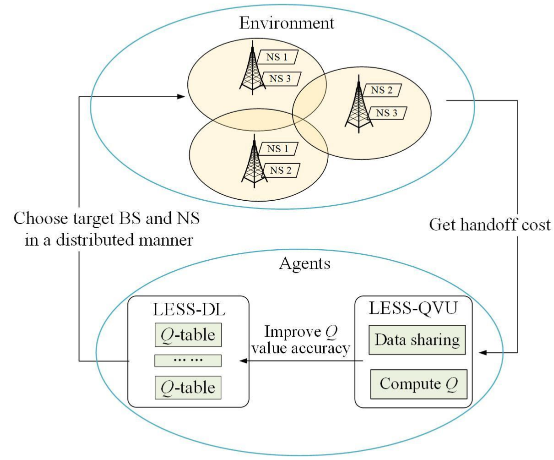


Fig. 2. The framework of LESS handover mechanism.

On the other hand, LESS-QVU is a modified Q-value update algorithm, which shares the reward of UEs with similar QoS requirements. By considering a collaborative approach, the problem of accurately estimating the Q-values by infinitely visiting all state-action pairs is mitigated and the aforementioned data sparsity problem can be overcome. Moreover, LESS-QVU is designed with the following principle in mind: given that UEs served by the same NS should have similar QOS requirements, whenever handover decisions are made, UEs with the same service type should have their Q-values updated by LESS-QVU. In this way, accurate Q-values can be obtained by using less data.

V. LESS Algorithms

In this section, LESS-DL and LESS-QVU algorithms are discussed in detail. Moreover, we theoretically prove the convergence of LESS at the end of this section. Let us start with LESS-DL.

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

The conventional Q-learning algorithm is a simple yet effective solution for RL problems. Its implementation can be described as follows. Denote 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 $r(S, \mathbf{A})$ represent the Q-value and reward for state-action pair $(S, \mathbf{A}) \in \mathcal{S} \times \mathcal{A}^{|\mathcal{U}|}$, respectively, where $r(S, \mathbf{A}) = \sum_{i=1}^{|\mathcal{U}|} r_i(S(t), \boldsymbol{a}_i(t))$. The calculation 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 $\boldsymbol{A}(t)$ correspond to the state and action vectors at time t, respectively. M is a sufficiently large constant for initialization purposes and $\beta (0<\beta<1)$ is the discount factor. Lastly, an ϵ-greedy policy is considered, in which the target BS and NS are selected based on the smallest Q-value [27].

However, when the traditional Q-learning model is applied to our problem, a large action space (*i.e.*,$\left|\mathcal{A}^{|\mathcal{U}|}\right|$) hinders its performance. On top of that, when the original Q-learning algorithm is considered, handover decisions are performed for all UEs simultaneously, which is unrealistic. As such, in order to tackle these problems, LESS-DL, a distributed learning algorithm is proposed in order to select target BSs and NSs for individual UEs, according to Fig. 3. The main idea behind LESS-DL is to maintain a set of reduced Q-tables, where the action space consists of each UE's (*i.e.,* agent) own actions. The Q-values are calculated by considering the cooperation of other UEs. Moreover, different from that in traditional Q-learning algorithm, besides the reduced Q-tables, LESS-DL also maintains a currently optimal policy in parallel based on current Q-values for multiple UEs. Storing this currently optimal policy guarantees the global optimality of LESS-DL from these reduced Q-tables in a distributed way. Once a handover decision is made, the UE will get a reward which is used for the update of Q-values as well as the currently optimal policy, and then the new policy is used for the next handover decision. In the following, we will elaborate the Q-value calculations and currently optimal policy maintaining in LESS-DL respectively.

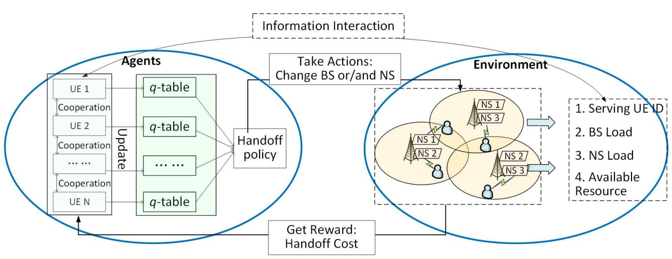


Fig. 3. The framework of LESS-DL.

First, let us study the Q-value calculation in LESS-DL. Considering that each individual UE maintains a reduced Q-table, denoted by q-table, the Q-value of UE i for a state-action pair $(S,\boldsymbol{a}_i)$, at time t is represented by $q_t^{(i)}(S,\boldsymbol{a}_i)$. For the readers convenience, we adopt q-value and Q-value to represent the values in the reduced and original tables respectively. Based on this concept and using a similar idea of [28], $q_t^{(i)}(S,\boldsymbol{a}_i)$ can be updated as:

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

As such, if the update is followed according to (4), the reduced q-tables for all UEs can be obtained. Despite the fact that the reduced q-tables of all UEs cannot reconstruct the original Q-table, it makes possible for all UEs to make decisions distributively. The proposition presented next highlights some properties of the reduced q-table.

Proposition 1. *When the action of a given UE i is $\boldsymbol{a}_i$, the q-value in the reduced q-table, given by $q_t^{(i)}(S,\boldsymbol{a}_i)$, is the minimum value in the original Q-table defined in* (3)*, 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 of the action vector $\mathbf{A}$.

Proof: Using the similar idea of [28], we prove Proposition 1 via mathematical induction.

First, when $t=0$, this equation naturally holds as all the values in both the reduced q-table and the original Q-table are equal to the initial value M.

Second, assume that (5) holds at the t-th iteration, and we should prove this equation also holds for $t+1$ in the following. If $(S(t),\boldsymbol{a}_i(t))\neq(S,\boldsymbol{a}_i)$, equation (5) holds for $t+1$, since no update is performed in both (3) and (4).
When $(S(t),\boldsymbol{a}_i(t))=(S,\boldsymbol{a}_i)$, for a specific UE i, we have

$$
\begin{aligned}
&q_{t+1}^{(i)}(S(t),\boldsymbol{a}_i(t))\\
&=\min\Big\{q_t^{(i)}(S(t),\boldsymbol{a}_i(t)),\\
&\qquad r_i(S(t),\boldsymbol{a}_i(t))+\beta\min_{\boldsymbol{a}'\in\mathcal{A}} q_t^{(i)}(S(t+1),\boldsymbol{a}')\Big\}\\
&\overset{(a)}{=}\min\Big\{\min_{\mathbf{A}\in\mathcal{A}^{|\mathcal{U}|},\boldsymbol{a}^{(i)}=\boldsymbol{a}_i} Q_t(S(t),\mathbf{A}(t)),\\
&\qquad r_i(S(t),\boldsymbol{a}_i(t))+\beta\min_{\mathbf{A}'\in\mathcal{A}^{|\mathcal{U}|}} Q_t(S(t+1),\mathbf{A}')\Big\}\\
&\overset{(b)}{=}\min\Big\{\min_{\mathbf{A}\in\mathcal{A}^{|\mathcal{U}|},\boldsymbol{a}^{(i)}=\boldsymbol{a}_i} Q_t(S(t),\mathbf{A}), Q_{t+1}(S(t),\mathbf{A}(t))\Big\},
\end{aligned}
\tag{6}
$$

where (a) is obtained from the assumption that equation (5) holds at the t-th iteration, and (b) is derived from (3).
Re-write $\min\limits_{\mathbf{A}\in\mathcal{A}^{|\mathcal{U}|},\boldsymbol{a}^{(i)}=\boldsymbol{a}_i} Q_t(S(t),\mathbf{A})$ as

$$
\begin{aligned}
&\min_{\mathbf{A}\in\mathcal{A}^{|\mathcal{U}|},\boldsymbol{a}^{(i)}=\boldsymbol{a}_i} Q_t(S(t),\mathbf{A})\\
&=\min\Big\{\min_{\mathbf{A}\in\mathcal{A}^{|\mathcal{U}|},\boldsymbol{a}^{(i)}=\boldsymbol{a}_i,\mathbf{A}\neq\mathbf{A}(t)} Q_t(S(t),\mathbf{A}), Q_t(S(t),\mathbf{A}(t))\Big\}.
\end{aligned}
\tag{7}
$$

For $\mathbf{A}\neq\mathbf{A}(t)$, Q-values do not update, *i.e*,

$$
Q_t(S(t),\mathbf{A})=Q_{t+1}(S(t),\mathbf{A}).
\tag{8}
$$

Moreover, due to the monotonicity of Q table, we have

$$
Q_{t+1}(S(t),\mathbf{A}(t))\leq Q_t(S(t),\mathbf{A}(t)).
\tag{9}
$$

Therefore, based on (8) and (9), we combine (6) and (7) as

$$\begin{aligned} & q_{t+1}^{(i)}\left(S(t), \boldsymbol{a}_i(t)\right) \\ & = \min \Big\{ \min_{\mathbf{A} \in \mathcal{A}^{|\mathcal{U}|}, \boldsymbol{a}^{(i)} = \boldsymbol{a}_i, \mathbf{A} \neq \mathbf{A}(t)} Q_{t+1}\left(S(t), \mathbf{A}\right), \\ & \quad Q_{t+1}\left(S(t), \mathbf{A}(t)\right) \Big\} \\ & = \min_{\mathbf{A} \in \mathcal{A}^{|\mathcal{U}|}, \boldsymbol{a}^{(i)} = \boldsymbol{a}_i} Q_{t+1}\left(S(t), \mathbf{A}\right). \end{aligned} \tag{10}$$

Hence, (5) holds for $t+1$.

Therefore, Proposition 1 is proved by using mathematical induction on t. ■

According to proposition 1, when (4) is used, each user will store in its reduced q-table, the minimum value of the original Q-table, $Q_t\left(S, \mathbf{A}\right)$. This, by its turn, enables us to design an optimal NS and BS distributed selection policy. Next, we illustrate how a global optimal policy for NS and BS selection can be achieved solely based on the q-values of each UE.

In conventional Q-leaning, after the algorithm has converged (when the values of the Q-table do not change), the followed policy is guaranteed to choose actions that yield the smallest Q-value, guaranteeing its optimality [27]. However, when the distributed reduced $q$tables is considered, if the smallest value for each individual UE is chosen, $q_t^{(i)}\left(S, \boldsymbol{a}_i\right)$, there is no guarantee that a global optimal policy is reached. In other words, choosing the best action values of each UE does not guarantee that the optimal action-vector $\mathbf{A}^*$ is chosen, *i.e.*, we cannot guarantee that

$$\left[\boldsymbol{a}_1{}^*, \boldsymbol{a}_2{}^*, \ldots, \boldsymbol{a}_{|\mathcal{U}|}{}^*\right] = \mathbf{A}^*. \tag{11}$$

In order to address this problem, a new policy for choosing actions is designed. The idea behind this new design is to store an action policy for UEs in parallel along with $q_t^{(i)}\left(S, \boldsymbol{a}_i\right)$ update. Once the value of $\min q_t^{(i)}\left(S, \boldsymbol{a}_i\right)$ decreases, the action policy is updated. Thus, since the policy has improved, a better action is available and, hence, it is stored as the currently optimal action. As such, when the algorithm converges, or in other words, when the value of $\min q_t^{(i)}\left(S, \boldsymbol{a}_i\right)$ does not change, the proposed action policy is stable, and the stored policy is the global optimal solution. The update rule of the stored action policy $\pi_t^{(i)}\left(S\right)$ of the i-th UE is:

$$\begin{aligned} & \pi_0^{(i)}\left(S\right) \in \mathcal{A}, \text{ arbitrarily}, \\ & \pi_{t+1}^{(i)}\left(S\right) = \\ & \begin{cases} \pi_t^{(i)}\left(S\right), \text{if } S \neq S_t \text{ or } \min\limits_{\boldsymbol{a}_i \in \mathcal{A}} q_t^{(i)}\left(S, \boldsymbol{a}_i\right) = \min\limits_{\boldsymbol{a}_i \in \mathcal{A}} q_{t+1}^{(i)}\left(S, \boldsymbol{a}_i\right), \\ \boldsymbol{a}_i\left(t\right), \text{ otherwise.} \end{cases} \end{aligned} \tag{12}$$

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

From [28] its corollary also states that for a given state S, we have

$$\left[\pi_t^{(1)}\left(S\right), \pi_t^{(2)}\left(S\right), \ldots, \pi_t^{(|\mathcal{U}|)}\left(S\right)\right] = \arg \min_{\mathbf{A} \in \mathcal{A}^{|\mathcal{U}|}} Q_t\left(S, \mathbf{A}\right). \tag{13}$$

As it can be seen, when the reduced q-tables converge the current stored action $\pi_t^{(i)}\left(S\right)$ for each individual UE guarantees the minimum handover cost.

In the case of LESS-DL, an ϵ-greedy action policy is chosen. This means that, before the q-values converge, each UE chooses as its target NS and BS pair the currently stored policy, $\pi_t^{(i)}\left(S\right)$, with a probability of $p = (1-\epsilon)$, or it can also choose randomly other actions with a probability of $p = \epsilon$. This allows the UEs to explore in earlier stages, and later on to exploit the information collected to their own benefit. Once the q-values have converged, the policy becomes a completely greedy one and the currently stored action is always chosen as the target NS and BS pair for each UE.

B. LESS-QVU Algorithm for Q-Value Update

Since the proposed LESS-DL algorithm relies on a variant of the Q-learning algorithm, it also requires a sufficient amount of data in order to accurately estimate its q-values and achieve a global optimal solution in terms of handover cost. However, since data among the mobile network is normally not evenly distributed, some BSs might be less visited than others, thus not generating enough data for the algorithm, and hindering its handover performance. Such areas, where data is not sufficient, is referred here as low-frequency activity (LFA) areas. As such, in order to surpass this data sparsity problem in LFA areas, a modified Q-value update algorithm, namely LESS-QVU, working in conjunction with LESS-DL is proposed. The core idea behind LESS-QVU, as shown in Fig. 4, is to exploit traffic similarity in the same slice for data augmentation, so as to expedite the convergence speed of the learning algorithm. Specifically, the agent trains the model by using not only its own data but also the shared data generated from other users of this slice with the same service type. For example, if a UE has a low latency requirement and it is in an LFA, it can utilize data from other UEs that also share low latency requirements to update its q-values, leading to better and more informed decisions. This idea ingeniously exploits the unique characteristics of NS: providing tailored service for UEs [2], leading to a high traffic similarity among the same type of NSs, thus the data sharing can be effectively exploited.

Considering an agent that makes decisions at time t as $\phi\left(t\right)$ and a given q-table stored by a given UE i, the q-value update policy for LESS-QVU is described as follows. If $\phi\left(t\right) = i$, the update policy of $q_{t+1}^{(i)}\left(S, \boldsymbol{a}_i\right)$ is the same as (4), whereas if $\phi\left(t\right) = j,\ (j \neq i)$, the update policy of $q_{t+1}^{(i)}\left(S, \boldsymbol{a}_i\right)$ is:

$$\begin{aligned} & q_{t+1}^{(i)}\left(S, \boldsymbol{a}_i\right) = \\ & \begin{cases} q_t^{(i)}\left(S, \boldsymbol{a}_i\right), \text{ if } \boldsymbol{a}_i\left(t\right) \neq \boldsymbol{a}_i \text{ or } S\left(t\right) \neq S \text{ } \textit{or } \psi_i \neq \psi_j, \\ \min\left\{q_t^{(i)}\left(S, \boldsymbol{a}_i\right), p_t^{(i,j)}\left(S, \boldsymbol{a}_i\right)\right\}, \text{ otherwise,} \end{cases} \end{aligned} \tag{14}$$

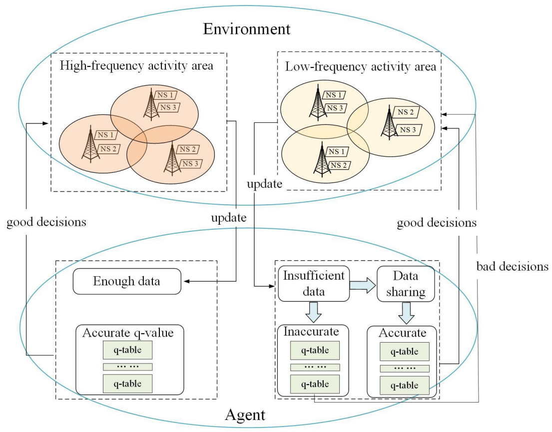


Fig. 4. LESS-QVU based LESS handover mechanism.

where the service type of the i-th UE is given by ψ_i and

$$\begin{aligned} & p_t^{(i,j)}\left(S, \boldsymbol{a}_i\right) \\ & = \Gamma_t^{(i)}\left(S, \boldsymbol{a}_i\right)^{\alpha}\left[\rho \cdot r_j\left(S, \boldsymbol{a}_i\right) + \beta \min_{\boldsymbol{a}' \in \mathcal{A}} q_t\left(\boldsymbol{a}', S\left(t+1\right)\right)\right] \end{aligned} \tag{15}$$

represents the calculated q-value of UE i considering the handover cost generated by UE j. Additionally, in $p_t^{(i,j)}\left(S, \boldsymbol{a}_i\right)$, $\Gamma_t^{(i)}\left(S, \boldsymbol{a}_i\right)$ represents how many times, at state S, the action $\boldsymbol{a}_i$ was chosen by UE i until time t, $\alpha > 0$ is a data sharing level parameter between UEs i and j, and $\rho > 1$ is a punishment factor to avoid excessive decrease of $q_{t+1}^{(i)}\left(S, \boldsymbol{a}_i\right)$.

Below a more in-depth explanation is given behind the q-value update policy of LESS-QVU. When a handover decision is made solely by UE i, the same update as in (4) is used to calculate its q-values. On the other hand, if the handover decision is performed by another UE j, the q-value is updated according to (14). On top of that, the update is designed in such a way that, if the chosen NS and BS pair area is regularly visited by UE i, implying that $\Gamma_t^{(i)}\left(S, \boldsymbol{a}_i\right)$ is a large number, the value of $p_t^{(i,j)}\left(S, \boldsymbol{a}_i\right)$ could be greater than $q_t^{(i)}\left(S, \boldsymbol{a}_i\right)$. Hence, $q_{t+1}^{(i)}\left(S, \boldsymbol{a}_i\right) = q_t^{(i)}\left(S, \boldsymbol{a}_i\right)$ is kept. In other words, this means that other UE's data does not interfere in the q-value update when a given UE is in a non-LFA area. On the other hand, in LFA areas, where $\Gamma_t^{(i)}\left(S, \boldsymbol{a}_i\right)$ is small, the handover cost of UE j, $r_j\left(S, \boldsymbol{a}_i\right)$, with the same service type of UE i, is utilized to calculate $p_t^{(i,j)}\left(S, \boldsymbol{a}_i\right)$ and to update the q-value $q_{t+1}^{(i)}\left(S, \boldsymbol{a}_i\right)$ of UE i. Lastly, in order to avoid excessively reducing the q-value of UE i, $q_{t+1}^{(i)}\left(S, \boldsymbol{a}_i\right)$, a punishment factor $\rho > 1$ is introduced. Based on this novel update, the impact that LESS-QVU has in the performance of the proposed framework is evaluated and verified by extensive simulations in Section VI.

Lastly, since LESS relies on the combination of both LESS-DL and LESS-QVU, this combination is briefly described here. First, whenever handover conditions are met and a handover event occurs, LESS-DL is activated and is utilized to choose the target NS and BS of multiple UEs in a distributed manner. After this process is performed, LESS-QVU comes into play and the q-values of all UEs are updated according to its novel update mechanism. Let us use the example in Fig. 1 to describe our proposed framework. In this scenario, there are 3 handover UEs, 4 BSs and 2 NSs. LESS-DL is first executed to make handover decisions for the 3 UEs based on their q-values respectively. According to the handover decisions, the corresponding reward values (handover cost) are generated, and LESS-QVU uses these reward values to update q-tables for the 3 UEs. In this way, whenever a new handover event occurs, LESS-DL is triggered again and chooses a new set of NS and BS based on the updated q-values.

C. Convergence Proof of LESS

We now start to theoretically prove the convergence of LESS algorithm. To obtain this, we first give the following Proposition 2 to show the convergence of LESS-DL.

Proposition 2. *Given by the q-value update rule in equation (4), LESS-DL converges to the minimum value in each action row of original Q-table with probability 1 (w.p.1) as long as each pair of state and action can be visited at infinite times.*

Proof: It is proved that the traditional single-agent Q-learning under the rule in equation (3) is converged w.p.1 if each pair of state and action can be visited infinitely [28]. In other word, denote by $Q^\star$ the converged single-agent based Q value, and thus

$$\lim_{t \to \infty} Q_t\left(S, \mathbf{A}\right) = Q^\star\left(S, \mathbf{A}\right) \tag{16}$$

Based on Proposition 1, we have that

$$q_t^{(i)}\left(S, \boldsymbol{a}_i\right) = \min_{\mathbf{A} \in \mathcal{A}^{|\mathcal{U}|}, \boldsymbol{a}^{(i)} = \boldsymbol{a}_i} Q_t\left(S, \mathbf{A}\right). \tag{17}$$

Therefore,

$$\begin{aligned} \lim_{t \to \infty} q_t^{(i)}\left(S, \boldsymbol{a}_i\right) &= \lim_{t \to \infty} \min_{\mathbf{A} \in \mathcal{A}^{|\mathcal{U}|}, \boldsymbol{a}^{(i)} = \boldsymbol{a}_i} Q_t\left(S, \mathbf{A}\right) \\ &= \min_{\mathbf{A} \in \mathcal{A}^{|\mathcal{U}|}, \boldsymbol{a}^{(i)} = \boldsymbol{a}_i} Q^\star\left(S, \mathbf{A}\right). \end{aligned} \tag{18}$$

Hence, we obtain that each q value is converged, thus the whole reduced q-table is converged. The convergence of LESS-DL under update rule of equation (4) has been proved. ∎

Then, moving to the convergence proof of LESS, we should in the following prove that the introduced data sharing scheme LESS-QVU does not affect the convergence property of LESS-DL.

Proposition 3. *Given by the q-value update rule in equation (4), and the data sharing rule in equation (14), LESS converges to the minimum value in each action row of original Q-table w.p.1 as long as each pair of state and action can be visited at infinite times.*

Proof: For an arbitrary q value, say $q^{(i)}\left(S, \boldsymbol{a}_i\right)$, denote by $\left\{q_n^{(i)}\left(S, \boldsymbol{a}_i\right)\right\}$ and $\left\{\hat{q}_n^{(i)}\left(S, \boldsymbol{a}_i\right)\right\}$ the update series under LESS-DL and LESS respectively. Note that the subscript n

denotes the n-th update not the time. Based on Proposition 2, we have that $\exists\ T_0 > 0$ and $\delta > 0$ satisfy

$$\left| q_{T_0}^{(i)}\left(S, \boldsymbol{a}_i\right) - \min_{\mathbf{A}\in\mathcal{A}^{|\mathcal{U}|}, \boldsymbol{a}^{(i)}=\boldsymbol{a}_i} Q^{\star}\left(S, \mathbf{A}\right) \right| < \delta. \quad (19)$$

Since $\alpha > 0$ in equation (15), it is reasonable to assume that $\forall\ n \geq T_0$, the following equation holds

$$p_n^{(i,j)}\left(S, \boldsymbol{a}_i\right) > q_n^{(i)}\left(S, \boldsymbol{a}_i\right), \forall\ j \neq i. \quad (20)$$

Thus, after T_0 updates, the data of other users does not affect the q-value update of UE i. Go back to the first T_0 updates, let us consider the worst case where all the first T_0 updates of UE i are performed by using others' data, i.e., for $\forall\ 0 < n \leq T_0$, $\hat{q}_n^{(i)}\left(S, \boldsymbol{a}_i\right)$ is updated based on $p_n^{(i,j)}\left(S, \boldsymbol{a}_i\right)$. Since $\rho \cdot r_j\left(S, \boldsymbol{a}_i\right) > r_i\left(S, \boldsymbol{a}_i\right)$ and $\alpha > 0$, we have

$$\begin{aligned} p_n^{(i,j)}\left(S, \boldsymbol{a}_i\right) &\geq r_i\left(S, \boldsymbol{a}_i\right) + \beta \min_{\boldsymbol{a}'\in\mathcal{A}} q_n\left(\boldsymbol{a}', S\left(n+1\right)\right) \\ &= q_n^{(i)}\left(S, \boldsymbol{a}_i\right) \end{aligned} \quad (21)$$

Thus, under this worst case, we have

$$\hat{q}_{T_0}^{(i)}\left(S, \boldsymbol{a}_i\right) = p_{T_0}^{(i,j)}\left(S, \boldsymbol{a}_i\right) \geq q_{T_0}^{(i)}\left(S, \boldsymbol{a}_i\right). \quad (22)$$

Since the value of $p_t^{(i,j)}\left(S, \boldsymbol{a}_i\right)$ does not affect the update of $\hat{q}_t^{(i)}\left(S, \boldsymbol{a}_i\right)$ after T_0 updates, the update rule of $\hat{q}_t^{(i)}\left(S, \boldsymbol{a}_i\right)$ now becomes the same as $q_t^{(i)}\left(S, \boldsymbol{a}_i\right)$. Combining equation (19) and (22) and the monotonous decrease of $\left\{\hat{q}_n^{(i)}\left(S, \boldsymbol{a}_i\right)\right\}$, we obtain that $\exists\ T_1 \geq T_0 > 0$ and $\delta > 0$ satisfy

$$\left| \hat{q}_{T_1}^{(i)}\left(S, \boldsymbol{a}_i\right) - \min_{\mathbf{A}\in\mathcal{A}^{|\mathcal{U}|}, \boldsymbol{a}^{(i)}=\boldsymbol{a}_i} Q^{\star}\left(S, \mathbf{A}\right) \right| < \delta. \quad (23)$$

Therefore, $\left\{\hat{q}_n^{(i)}\left(S, \boldsymbol{a}_i\right)\right\}$ is converged. ■

Here please note that the convergence speed of LESS should be faster than that of LESS-DL, although the required number of updates T_1 showing in the above proof of LESS could be larger than that of LESS-DL, i.e., T_0. This is because that there would be many updates under LESS are performed by using the data of others rather than the data generated by the UE itself after visiting the state-action pair. Hence, the convergence speed of LESS could be accelerated even more updates are introduced.

VI. Implementations of LESS

In this section, we first illustrate the implementations of LESS mechanism in a real communication system, and then analyze the signaling overhead.

A. Implementations

Due to the different architecture between RAN slicing and the traditional cellular network, the implementations of LESS mechanism in RAN slicing should be deliberately explained. In sliced mobile networks, a software defined network (SDN) controller needs to be deployed in each NS to handle handovers [2]. Note that the SDN controller will take responsibility for handovers besides routing and forwarding [2], [29]. With the cooperation of UEs, BSs and SDN controllers, the handover procedures based on LESS are described in Fig. 5.

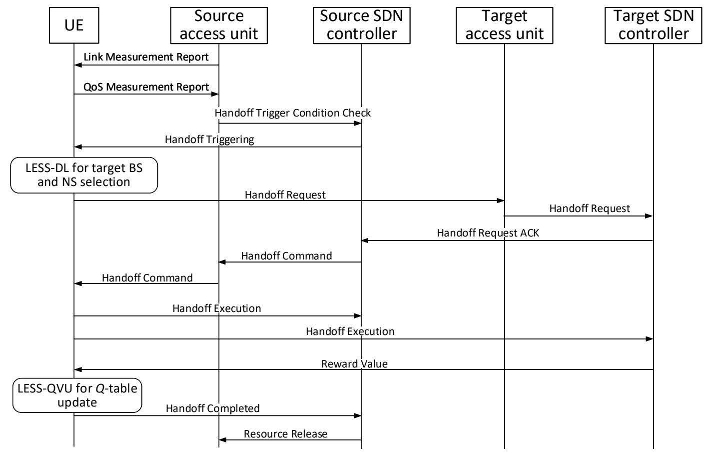


Fig. 5. Handover procedures for RAN slicing based on LESS.

In detail, the UE periodically measures and reports the obtained QoS to the source BS and NS, and the source SDN controller checks if the handover trigger condition (equation (1)) is satisfied. Then the UE uses LESS-DL to select the target BS and NS and sends the handover request to the corresponding SDN controllers. After the confirmation of handover command, this handover is executed by the target and the source SDN controllers. Before the handover is completed, the target SDN controller calculates the the reward value of this handover decision, and then broadcasts the reward to the UEs. The UEs served by the same type of this target NS use LESS-QVU to update Q-table. Finally, the resource of source BS and NS is released by the SDN controller.

B. Signaling Overhead

Next we analyze the signaling overhead of LESS mechanism. According to Fig. 5, we find that compared with conventional handover mechanism the extra procedures of LESS handover are executing LESS-DL and LESS-QVU. Thus, we focus on the signaling overhead caused by these two algorithms. First, in LESS-DL, the handover UE needs to send the handover request to the admissible NSs (the NSs that can provide the service type of the UE), and then the SDN controller deployed on these admissible NSs checks and notifies if the corresponding BSs have sufficient resources to serve the UE. Finally, the UE makes handover decisions from the admissible NSs and BSs. Therefore, the number of signaling exchanges can be approximately calculated as $(2|\mathcal{N}_a| + |\mathcal{B}_a|)$, where $|\mathcal{N}_a|$ and $|\mathcal{B}_a|$ are the number of admissible NSs and BSs respectively.

Second, let us examine the signaling overhead caused by LESS-QVU. In LESS-QVU, once a reward is generated by a handover UE, the corresponding target SDN controller should notify this reward value to all the serving UEs. Then, the

UEs need to update their Q-table according to (14). Thus, the number of signaling exchanges is $(|\mathcal{U}_a|)$, where $(|\mathcal{U}_a|)$ is the number of serving UEs of the target NS. Therefore, the total number of extra signaling exchanges of LESS is $(2|\mathcal{N}_a| + |\mathcal{B}_a| + |\mathcal{U}_a|)$ for one handover, and each signaling exchange uses only several bits.

VII. Simulation and Numerical Results

In order to evaluate the performance of the proposed LESS scheme, simulations are performed and the proposed solution is compared with three other baselines, namely: *Max-SINR, NS-Prior* and *LESS-DL*. The *Max-SINR* method consists of a modified version of the traditional RSRP-based handover scheme. It works by first selecting BSs with the highest signal-to-interference-plus-noise ratio (SINR) for every UE [13]. After that, it attempts to find a suitable NS in the BS that can satisfy the QoS required by the UEs. On the one hand, if such BS and NS pair exists, they are selected as targets for the UEs handover. On the other hand, if no such NS satisfies the UEs requirements, a new NS is deployed in the BS and the target is assigned for the respective BS and the newly created NS. The second baseline, *NS-prior*, is similar to the Max-SINR method, but it performs its operations in the opposite way. It first finds a suitable NS for UEs to handover to, and then later attempts to select a BS, with sufficient bandwidth, covered by this NS. Lastly, *LESS-DL* mechanism corresponds to the proposed handover mechanism, albeit without the data sharing component. This is done, so that we can compare the performance of LESS with and without data sharing and verify the effectiveness of the newly proposed data sharing policy.

In addition, the condition for a handover to be triggered is the same, as in (1), in order to provide a fair comparison between all methods. Lastly, all methods are evaluated in terms of three metrics, which are: handover cost, total number of handovers and UE outage probability, defined as the probability of the UE's QoS not being satisfied.

A. Simulation Settings

Regarding the simulation, a heterogeneous mobile network scenario is considered. In this network, a macro BS (MBS) located in the center of a circular area of 1000m radius is deployed. On top of that, a varying number of small BSs (SBS), such as femto BSs (FBS), and UEs are randomly distributed in the area. In terms of NSs, the total number of NSs in this network is 40. Each NS covers a random number of BSs, and also has different capabilities in terms of data rate and latency (given as γ_n^{min} and τ_n respectively in our model). Based on the levels of data rate (high, medium and low) and latency (high, medium and low) offered by the NSs, the total number of service types is set to 9, as shown in Table I. The transmit power of MBS and FBS is set to 46dBm and 20dBm respectively [30]. In terms of bandwidth, it is considered that all BSs share a 20MHz bandwidth [26], which is allocated to the deployed NSs based on the NS QoS provisioning. For UE movement, we consider a well-known user mobility patten straight-line motion with random bouncing (sLRB) defined by 3GPP [31], where users move at a constant speed along with a random direction in a straight line, and they will bounce in a random direction once reaching the edge of the considered area. Lastly, regarding user requirements, both their data rate and latency requirements are randomly distributed among the 3 levels defined in the NS provisioning (high, medium or low).

Since there is no reference to investigate the parameters of handover cost C_{NS}, C_{BS}, C_{NS-BS} and C_{New}, we set the values to them based on the relationship in Section III. Specifically, we normalize C_{NS} as 1, and set C_{BS}, C_{NS-BS} and C_{New} to 3, 5 and 20, respectively. Note that these four parameters may affect the absolute value of total handover cost, but do not invalidate the relative performance enhancement of our proposed mechanism. For convenience, simulation parameters are summarized in Table II.

TABLE I
Service Type

Service Type: Delay \ Rate	low	medium	high
low	1	2	3
medium	4	5	6
high	7	8	9

TABLE II
Simulation Parameters

Parameter	**Value**
MBS coverage radius	1000 m
The total number of deployed NSs	40
The number of service types	9
Handover cost C_{NS} of switching NS only	1
Handover cost C_{BS} of switching BS only	3
Handover cost C_{NS-BS} of switching the both	5
Handover cost C_{New} of deploying a new slice	20
The transmit power of MBS	46 dBm
The transmit power of PBS	30 dBm
The transmit power of FBS	20 dBm
Bandwidth of all BSs	20 MHz

B. Numerical Results and Discussions

In the first experiment, we verify the convergence of the proposed LESS algorithm with the comparison of LESS-DL. Fig. 6 shows the cumulative distribution function (CDF) of convergence step of LESS and LESS-DL under the number of BS equals to 15 and 25 respectively. The number of UE is set to 200. Note that both the two intelligent algorithms are distributed, and each agent (*i.e.,* UE) maintains a reduced Q-table separately. Hence, the convergence speed among these 200 UEs could be different. To make the results comparable here, we calculate the CDF of convergence step with respect to

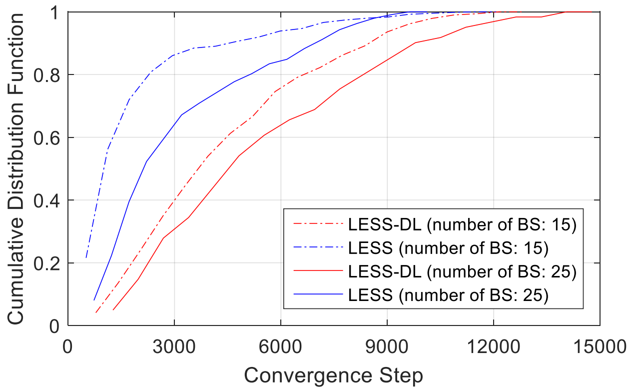


Fig. 6. Convergence of LESS and LESS-DL (number of UE: 200).

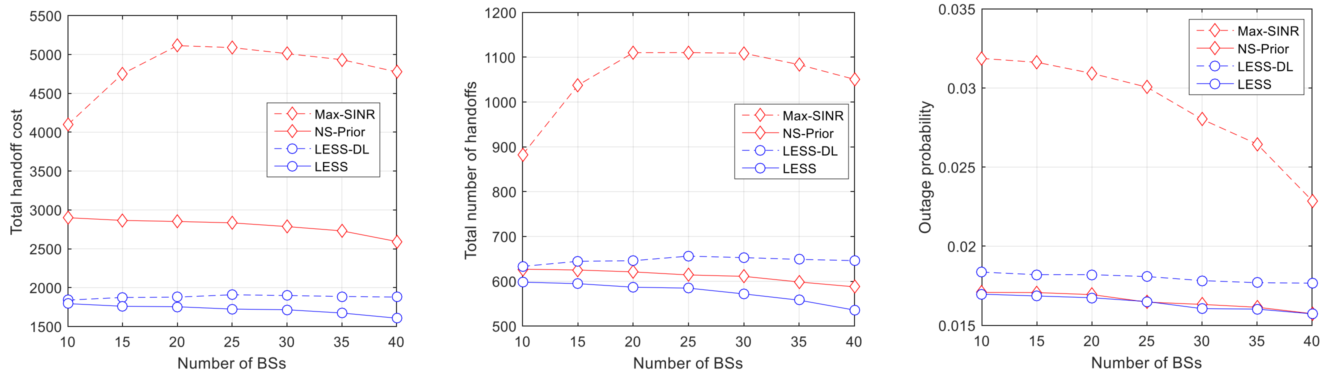
all 200 UEs for the two algorithms respectively. From Fig. 6, we can see that LESS has faster convergence speed compared with LESS-DL under the two different number of BSs. For example, more than 80% UEs can get converged before 3000 steps in LESS under the case of 15 BSs, while only about 63% UEs could converge in LESS-DL. These results clearly demonstrate the effectiveness of our data sharing scheme in LESS, which is that the data sharing scheme can increase the convergence speed of LESS-DL.

In the following several experiments we will evaluate the handover performance of the four methods. Fig. 7 shows the performance of all methods in terms of handover cost, number of handovers and UE outage probability, when the number of BSs varies from 10 to 40. From Fig. 7(a), it can be seen that the handover cost of the two intelligent methods (LESS and LESS-DL) is much lower than that of the Max-SINR and NS-Prior. For 25 BSs, for example, it can be seen that the handover cost gain of LESS is approximately of 51%, 40% and 10% when compared to Max-SINR, NS-Prior and LESS-DL, respectively. These results clearly validate the effectiveness of LESS. When comparing the total number of handovers, in Fig. 7(b), it can be seen that the proposed scheme, LESS, is able to achieve the lowest number of handovers among all methods. On the other hand, the performance of LESS-DL is worse than that of the NS-Prior method. This can be explained by the fact that LESS does not have enough data collected from the LESS-QVU new policy, thus it is not able to make as many good decisions as LESS-DL or even the NS-prior method. These results also demonstrate that the parameters of handover cost do not affect the performance gain of LESS mechanism since LESS achieves both the lowest handover cost and the smallest number of handovers. Finally, Fig. 7(c) shows that the UE outage probability of all the four mechanisms decreases with the number of BSs due to more available wireless resources. Moreover, NS-Prior achieves similar outage probability performance to LESS because of the prior consideration of NS service provisioning, while the Max-SINR method has the highest outage probability, even though it may achieve good SINR performance.

Fig. 8 compares the handover performance of the four handover methods for a fixed number of BSs, in this case 20, and a different number of UEs, varying from 50 to 400. Fig. 8(a) compares the handover cost for the four algorithms. It can be seen that the handover cost of all the mechanisms increases with the number of UEs, and that the learning methods LESS and LESS-DL significantly outperform the other two due to the exploitation of historical data. In terms of the total number of handovers, Fig. 8(b) shows that the performance of NS-Prior is similar to both LESS and LESS-DL. This can be explained by the fact that all these schemes consider finding a NS, whereas Max-SINR does not. Lastly, in term of UE outage probability, Fig 8(c) shows that it increases with the number of UEs for all the four handover methods due to limited network resources. The UE outage probability of Max-SINR is always much higher than that of the other three mechanisms.

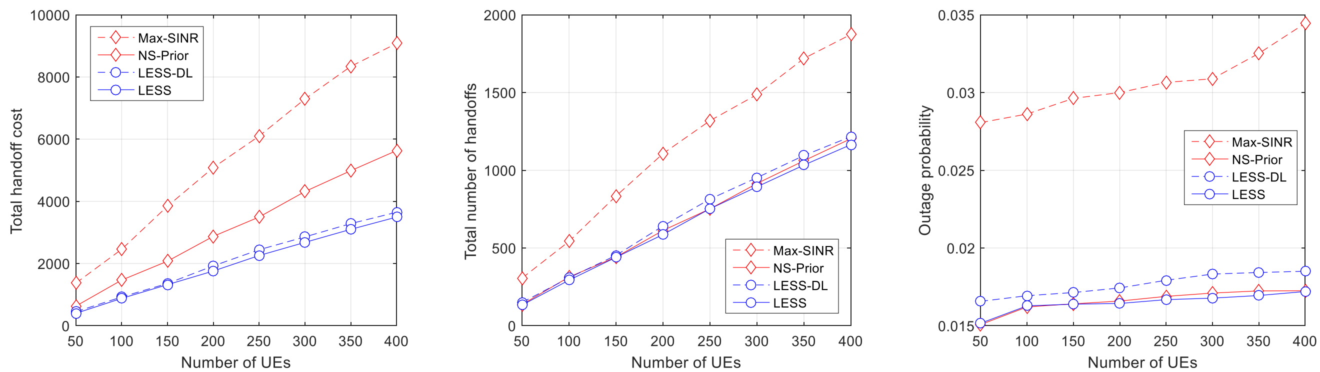
Fig. 9 compares the performance of all schemes for different UE movement speeds. Fig. 9(a) shows the handover cost for the four mechanisms. It can see that from a slow walking speed of 2 m/s (7.2 km/h) to faster driving speeds of up to 14 m/s (50km/h), the handover cost increases for all four methods. Moreover, the handover cost of LESS and LESS-DL increases slowly due to the interaction with environment by using historical data, while the handover cost of the traditional Max-SINR mechanism increases rapidly due to the lack of NS as well as UE service type. As expected, Fig. 9(b) shows that the number of handovers increases with UE movement speed for all the four mechanisms due to fast change of channel quality. Moreover, the differences of the number of handovers among LESS, LESS-DL and NS-Prior are relatively small. Fig. 9(c) compares the performance of UE outage probability. From this figure, we find that when UE movement speed is larger than 10 m/s, NS-Prior achieves the lowest UE outage probability. This is because in NS-Prior UEs always choose the most suitable serving NS to fulfill the QoS requirement in NS-Prior when a handover occurs. Moreover, the UE outage probability of LESS is always lower than 2%, implying that even if UE movement speed is not considered in our reinforcement learning model, the outage probability under fast movement scenario of LESS is only slightly higher than that of NS-Prior.

Lastly, the performance of the four methods is investigated in a scenario with different NS coverage. Note that the NS coverage is defined as the number of BSs covered by an NS. Fig. 10(a) shows the handover cost for the four mechanisms with different NS coverage. Intuitively, increasing NS coverage may reduce handover cost since the serving NS covers more BSs. However, from Fig. 10(a), we find that the handover cost of LESS, LESS-DL and NS-Prior remains stable when the NS coverage increases, and the handover cost of Max-SINR increases even more rapidly. This is because the total bandwidth of the network is fixed although we increase the NS coverage, implying that the average available bandwidth is decreased. Thus, the handover cost cannot be reduced when we increase the coverage of NS with a fixed amount of bandwidth. Similarly, Fig. 10(b) and Fig. 10(c) reveal the same behaviors of the number of handovers and UE outage



(a) Handover cost (b) Number of handovers (c) Outage probability of UEs

Fig. 7. Comparisons of handover performance for the four handover mechanisms with different number of BSs.



(a) Handover cost (b) Number of handovers (c) Outage probability of UEs

Fig. 8. Comparisons of handover performance for the four handover mechanisms with different number of UEs.

probability respectively. Therefore, we can draw a conclusion based on the results of these three figures: increasing NS coverage (i.e., the number of BSs covered by an NS) with fixed total wireless resources cannot improve handover performance.

VIII. Conclusions

In this work, we have identified the importance and challenges of designing new handover algorithms for RAN slicing, and exploited artificial intelligence to incorporate the information on the complicated RAN slicing environments for making precise handover decisions. The proposed efficient intelligent handover mechanism LESS is based on a multi-agent distributed learning with the objective of minimizing the long-term handover cost of the network, while guaranteeing users' QoS. By ingeniously exploiting the characteristics of network slice, we modified the learning rules including the way of decision making as well as reward function update, thus LESS archived significant performance gain in terms of handover cost, the number of handovers, outage probability. Even compared the performance improvement brought by other conventional learning-based algorithm, "LESS is more". Importantly, the idea of exploiting traffic similarity of a network slice and thus sharing the data for distributed learning can be generalized to address other problems of mobile networks in the case of limited available data, such as resource allocation, transmit power control and interference coordination.

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.

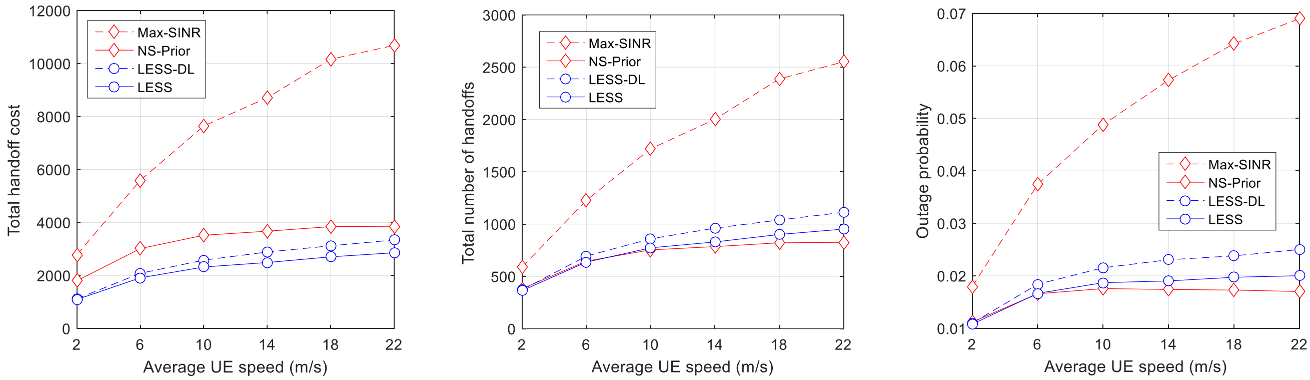
[3] Q. Ye, J. Li, K. Qu, W. Zhuang, X. S. Shen, and X. Li, "End-to-end quality of service in 5g networks: Examining the effectiveness of a network slicing framework," *IEEE Vehicular Technology Magazine*, vol. 13, no. 2, pp. 65–74, 2018.

[4] L. Zhang, A. Ijaz, P. Xiao, M. Molu, and R. Tafazolli, "Filtered OFDM systems, algorithms, and performance analysis for 5G and beyond," *IEEE Transactions on Communications*, vol. 66, no. 3, pp. 1205–1218, 2018.

[5] 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.

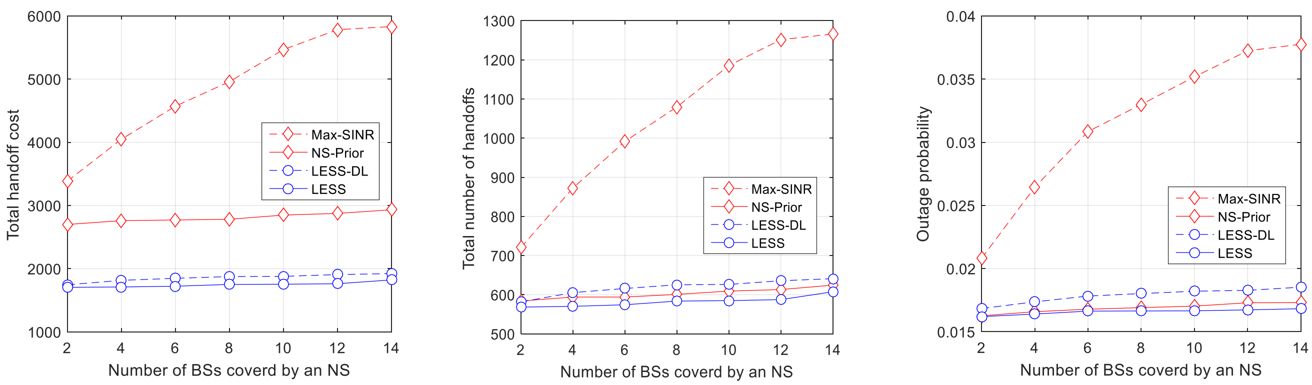
[6] Q. Ye, W. Zhuang, S. Zhang, A.-L. Jin, X. Shen, and X. Li, "Dynamic radio resource slicing for a two-tier heterogeneous wireless network," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 10, pp. 9896–9910, 2018.

[7] M. Yan, G. Feng, J. Zhou, Y. Sun, and Y.-C. Liang, "Intelligent resource scheduling for 5g radio access network slicing," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, pp. 7691–7703, 2019.



(a) Handover cost (b) Number of handovers (c) Outage probability of UEs

Fig. 9. Comparisons of handover performance for the four handover mechanisms with different UEs movement speed.



(a) Handover cost (b) Number of handovers (c) Outage probability of UEs

Fig. 10. Comparisons of handover performance for the four handover mechanisms with different NS coverage.

[8] B. Yang, L. Zhang, O. Onireti, P. Xiao, M. A. Imran, and R. Tafazolli, "Mixed-numerology signals transmission and interference cancellation for radio access network slicing." *IEEE Transactions on Wireless Communications*, pp. 1–16, 2020.

[9] F. Wei, G. Feng, Y. Sun, Y. Wang, S. Qin, and Y.-C. Liang, "Network slice reconfiguration by exploiting deep reinforcement learning with large action space," *IEEE Transactions on Network and Service Management*, 2020.

[10] 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.

[11] 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

[12] G. Wang, G. Feng, T. Q. S. Quek, S. Qin, R. Wen, and W. Tan, "Reconfiguration in Network Slicing - Optimizing the Profit and Performance," *IEEE Transactions on Network and Service Management*, 2019.

[13] 3GPP TS 36.331, "E-UTRA Radio Resource Control (RRC); Protocol specification (Release 9)," 2016.

[14] K. Vasudeva, S. Dikmese, I. Guvenc, A. Mehbodniya, W. Saad, and F. Adach, "Fuzzy Logic Game-Theoretic Approach for Energy Efficient Operation in HetNets," in *Communications Workshops (ICC Workshops), 2017 IEEE International Conference*, 2017, pp. 552–557.

[15] Y.-S. Liou, R.-H. Gau, and C.-J. Chang, "A Bargaining Game Based Access Network Selection Scheme for HetNet," in *2014 IEEE International Conference on Communications (ICC)*. IEEE, 2014, pp. 4888–4893.

[16] R. Arshad, H. Elsawy, S. Sorour, T. Y. Al-naffouri, and M.-s. Alouini, "Handover Management in 5G and Beyond : A Topology Aware Skipping Approach," *IEEE Access*, vol. 4, pp. 9073–9081, 2016.

[17] Z. Guohua, P. Legg, and G. Hui, "A network controlled handover mechanism and its optimization in LTE heterogeneous networks," in *IEEE Wireless Communications and Networking Conference, WCNC*, 2013, pp. 1915–1919.

[18] G. Araniti, J. Cosmas, A. Iera, A. Molinaro, A. Orsino, and P. Scopelliti, "Energy Efficient Handover Algorithm For Green Radio Networks," in *2014 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting*, 2014, pp. 1–6.

[19] N. C. Luong, D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y.-C. Liang, and D. I. Kim, "Applications of Deep Reinforcement Learning in Communications and Networking: A Survey," *arXiv preprint arXiv:1810.07862*, 2018.

[20] 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.

[21] F. Guidolin, I. Pappalardo, A. Zanella, and M. Zorzi, "Context-Aware Handover Policies in HetNets," *IEEE Transactions on Wireless Communications*, vol. 15, no. 3, pp. 1895–1906, 2016.

[22] N. Gupta, S. K. Dhurandher, I. Woungang, and M. S. Obaidat, "Proactive Decision Based Handoff Scheme for Cognitive Radio Networks," in *2018 IEEE International Conference on Communications (ICC)*, 2018.

[23] A. S. D. Alfoudi, M. Dighriri, A. Otebolaku, T. Baker, G. M. Lee, and R. Pereira, "Mobility Management Architecture in Different RATs Based Network Slicing," in *The 32-nd IEEE International Conference on Advanced Information Networking and Applications*, 2018, pp. 270–274.

[24] L. Zhu, F. R. Yu, T. Tang, and B. Ning, "Handoff performance improvements in an integrated train-ground communication system

based on wireless network virtualization," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1165–1178, 2017. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/20857278

[25] 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.

[26] Y. Sun, S. Qin, G. Feng, L. Zhang, and M. A. Imran, "Service provisioning framework for ran slicing: user admissibility, slice association and bandwidth allocation," *IEEE Transactions on Mobile Computing*, 2020. [Online]. Available: http://dx.doi.org/10.1109/TMC.2020.3000657

[27] 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

[28] 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.

[29] V. Yazici, U. C. Kozat, and M. O. Sunay, "A New Control Plane for 5G Network Architecture with a Case Study on Unified Handoff, Mobility, and Routing Management." *IEEE Communications Magazine*, vol. 52, no. 11, pp. 76 – 85, 2014.

[30] 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.

[31] E. U. T. R. Access, "Mobility enhancements in heterogeneous networks," *3GPP TR 36.839, Tech. Rep.*, 2012.

Yao Sun received the B.S. degree in Mathematical Sciences, and the Ph.D. degree in Communication and Information System from University of Electronic Science and Technology of China (UESTC), in 2014 and 2019, respectively. He has published widely in wireless networking research area, and received the IEEE ComSoc TAOS Best Paper Award in 2019. His research interests include intelligent access control, handoff and resource management in mobile networks.

Wei Jiang received her Ph.D. degree in School of Communication and Information Engineering at University of Electronic Science and Technology of China (UESTC) in 2019. She was a visiting student at the Pennsylvania State University (PSU) from 2017 to 2018. She is now a post-doctor with the Shenzhen Key Laboratory of Digital Creative Technology, the Guangdong Province Engineering Laboratory for Digital Creative Technology, College of Electronics and Information Engineering, Shenzhen University, Shenzhen, China. Her current research interests include next generation mobile communication systems, machine learning and content caching and distribution techniques in mobile networks.

Gang Feng (M'01, SM'06) received his BEng and MEng degrees in Electronic Engineering from the University of Electronic Science and Technology of China (UESTC), in 1986 and 1989, respectively, and the Ph.D. degrees in Information Engineering from The Chinese University of Hong Kong in 1998. He joined the School of Electric and Electronic Engineering, Nanyang Technological University in December 2000 as an assistant professor and became an associate professor in October 2005. At present he is a professor with the National Laboratory of Communications, UESTC. Dr. Feng has extensive research experience and has published widely in wireless networking research. A number of his papers have been highly cited. He has received the IEEE ComSoc TAOS Best Paper Award and ICC best paper award in 2019. His research interests include next generation mobile networks, mobile cloud computing, AI-enabled wireless networking, etc. Dr. Feng is a senor member of IEEE.

Paulo Valente Klaine (S'17, M'19) received his B. Eng. degree in electrical and electronic engineering from the Federal University of Technology - Paraná (UTFPR), Brazil in 2014, the MSc. degree from the University of Surrey, Guildford, U.K., in Mobile Communications Systems in 2015, and the PhD degree in Electrical and Electronics Engineering from the University of Glasgow, U.K., in 2019. He has 3 filed patents and authored/co-authored over 15 publications. He is currently a research associate at the University of Glasgow, U.K., and his research interests include self organizing networks, V2X communications, wireless blockchain and machine learning in wireless networks.

Lei Zhang (SM'18) received his Ph.D. from the University of Sheffield, U.K. He is now a Lecturer at the University of Glasgow, UK. His research interests broadly lie in the Communications and Array Signal Processing, including radio access network slicing (RAN slicing), wireless blockchain networks, new air interface design, Internet of Things (IoT), V2X, multi-antenna signal processing, massive MIMO systems, etc. He is holding 16 US/UK/EU/China granted patents on wireless communications. Dr Lei Zhang also holds a visiting position in 5GIC at the University of Surrey. He is an associate editor of IEEE ACCESS and a senior member of IEEE.

Muhammad Ali Imran (M'03, SM'12) Fellow IET, Senior Member IEEE, Senior Fellow HEA is a Professor of Wireless Communication Systems with research interests in self organised networks, wireless networked control systems and the wireless sensor systems. He heads the Communications, Sensing and Imaging CSI research group at University of Glasgow. He is an Affiliate Professor at the University of Oklahoma, USA and a visiting Professor at 5G Innovation Centre, University of Surrey, UK. He has over 20 years of combined academic and industry experience with several leading roles in multi-million pounds funded projects. He has filed 15 patents; has authored/co-authored over 400 journal and conference publications; was editor of 3 books and author of more than 20 book chapters; has successfully supervised over 40 postgraduate students at Doctoral level. He has been a consultant to international projects and local companies in the area of self-organised networks.

Ying-Chang Liang (F'11) is currently a Professor with the University of Electronic Science and Technology of China, China, where he leads the Center for Intelligent Networking and Communications and serves as the Deputy Director of the Artificial Intelligence Research Institute. He was a Professor with The University of Sydney, Australia, a Principal Scientist and Technical Advisor with the Institute for Infocomm Research, Singapore, and a Visiting Scholar with Stanford University, USA. His research interests include wireless networking and communications, cognitive radio, symbiotic radio, dynamic spectrum access, the Internet-of-Things, artificial intelligence, and machine learning techniques. Dr. Liang has been recognized by Thomson Reuters (now Clarivate Analytics) as a Highly Cited Researcher since 2014. He received the Prestigious Engineering Achievement Award from The Institution of Engineers, Singapore, in 2007, the Outstanding Contribution Appreciation Award from the IEEE Standards Association, in 2011, and the Recognition Award from the IEEE Communications Society Technical Committee on Cognitive Networks, in 2018. He is the recipient of numerous paper awards, including the IEEE Jack Neubauer Memorial Award, in 2014, and the IEEE Communications Society APB Outstanding Paper Award, in 2012. He was elected a Fellow of the IEEE for contributions to cognitive radio communications. He is the Founding Editor-in-Chief of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS: COGNITIVE RADIO SERIES, and the Key Founder and now the Editor-in-Chief of the IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING. He is also serving as an Associate Editor-in-Chief for China Communications. He served as a Guest/Associate Editor of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, the IEEE JOURNAL OF SELECTED AREAS IN COMMUNICATIONS, the IEEE Signal Processing Magazine, the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, and the IEEE TRANSACTIONS ON SIGNAL AND INFORMATION PROCESSING OVER NETWORK. He was also an Associate Editor-in-Chief of the World Scientific Journal on Random Matrices: Theory and Applications. He was a Distinguished Lecturer of the IEEE Communications Society and the IEEE Vehicular Technology Society. He was the Chair of the IEEE Communications Society Technical Committee on Cognitive Networks, and served as the TPC Chair and Executive Co-Chair of the IEEE Globecom'17.