Research Article

Service Provisioning in Sliced Cloud Radio Access Networks

Hong Zou ^{1,2,3}, Min Zhou ^{1,2,3}, Yaping Cui ^{1,2,3}, Peng He ^{1,2,3},
Hong Zhang ^{1,2,3} and Ruyan Wang ^{1,2,3}

¹School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing, China

²Advanced Network and Intelligent Connection Technology Key Laboratory of Chongqing Education Commission of China, Chongqing, China

³Chongqing Key Laboratory of Ubiquitous Sensing and Networking, Chongqing, China

Correspondence should be addressed to Hong Zhang; hongzhang@cqupt.edu.cn

Received 28 August 2021; Revised 8 November 2021; Accepted 2 December 2021; Published 3 February 2022

Academic Editor: Long Yang

Copyright © 2022 Hong Zou et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Network slicing- (NS-) based cloud radio access networks (C-RANs) have emerged as a key paradigm to support various novel applications in 5G and beyond networks. However, it is still a challenge to allocate resources efficiently due to heterogeneous quality of service (QoS) requirements of diverse services as well as competition among different network slices. In this paper, we consider a service provisioning allocation framework to guarantee resource utilization while ensuring the QoS of users. Specifically, an inter/intraslice bandwidth optimization strategy is developed to maximize the revenue of the system with multiple network slices. The proposed strategy is hierarchically structured, which decomposes into network-level slicing and packet scheduling level slicing. At the network level, resources are allocated to each slice. At the packet scheduling level, each slice allocates physical resource blocks (PRBs) among users associated with the slice. Numerical results show that the proposed strategy can effectively improve the revenue of the system while guaranteeing heterogeneous QoS requirements. For example, the revenue of the proposed strategy is 21% higher than that of the average allocation strategy.

1. Introduction

Many new applications have emerged due to the evolution of the Internet of Things (IoT) [1], such as Augmented Reality, Internet of Vehicles, and IoT devices. In order to serve the massively growing wireless data traffic, cloud radio access networks (C-RANs) have emerged as a key enabling technology for the next generation wireless communications [2, 3]. C-RAN consists of a baseband unit (BBU) and distributed remote radio heads (RRHs), in which BBU is a resource pool shared by RRHs. BBU and RRH are connected via optical fiber, and antennas are equipped with RRHs to transmit/receive radio frequency signals [3]. With the centralized cloud BBU pool, an agile and programmable software-defined environment in the RAN side can be achieved [4].

Network Functions Virtualization (NFV) and Software-Defined Networking (SDN) are the key technologies for implementing network slicing (NS) [5]. In NS, there is a service provider, which is the owner of each network slice, and it is also called a tenant [4]. NS is the operators' technique that separates multiple virtual end-to-end networks on a unified infrastructure and can provide different service segregation. In order to meet the diverse quality of service (QoS) requirements in different scenarios, that is, enhanced Mobile Broadband (eMBB), massive Machine Type Communication (mMTC), and Ultra-reliable Low Latency Communication (URLLC), the C-RAN should be sliced for provisioning tailored services [6, 7]. However, C-RAN slicing is more complex than core NS because of the uncertainty of the wireless channel. Thus, from both user QoS and network resource utilization perspectives, an

appropriate service provisioning scheme is needed to improve both network and user performance by inter/intraslice resources allocation.

Most works focus on the RAN slicing resource allocation framework [7–10]. In order to ensure QoS of users, the system would control the number of users in slice due to limited resources, and the authors depicted the NS problem as a two-stage resource allocation [7]. Similarly, references [8, 9] proposed the two-stage slicing framework, respectively. The scheduling idea of the earliest deadline priority was applied to RAN slice radio resource allocation, and the network-level resource demand was translated into cell-level resource demand by a translation framework [8]. In [9], dynamic BBU resource allocation and dynamic physical resource block (PRB) allocation were performed during user access control, where users with high priority can have priority access to the network. In [10], a joint resource allocation and admission control scheme is used in fog radio access network (FRAN) slicing. However, references [8–10] ignored the system revenue. Considering the uncertainty of wireless channel and the mobility of users, the dynamic allocation approach leads to frequent computations, which increases the burden on BBU.

There have been a few works concerning the intelligent resource allocation of RAN slicing because diverse QoS requirements make it complex for traditional resource allocation schemes. Reference [11] proposed demand-aware resource management based on deep distributional reinforcement learning (RL) [12], with the aim of dynamically and efficiently allocating resources for diversified services. In [13], a deep neural network (DNN) was used to estimate the resource provisioning according to the traffic of each slice, and a non-zero-sum game strategy was then used to solve the resulting offline optimization task. However, references [11, 13] did not consider the system revenue of the system. The system revenue was described as the stability of slicing [14]. The authors described the network slicing problem as a two-stage optimization model with time-averaged metric and solved the problem by learning the augmented optimization approach with deep learning and Lyapunov stability theory [14]. However, the authors ignored the total resource limit, which impacted the number of resources reserved for the long timescale. References [11–14] did not take into account the different performance requirements of different QoS. An online joint scheduling strategy was proposed in [15], and eMBB slices were considered as well as URLLC slices. When URLLC traffic is overloaded, it can opportunistically overlap the resources of eMBB at any minislot. However, this approach improved the QoS of the URLLC users, while it caused disruptions of the eMBB users.

Based on the mentioned previously, we propose an inter/intraslice bandwidth optimization strategy for C-RAN network slicing with the aim of maximizing the system revenue while guaranteeing the QoS of users. Since the instability of the channel state and the limitation of resources, the system revenue, which is obtained by resource allocation on one single timescale, is not optimal in the long term. Thus, the proposed strategy is a two-timescale method, including long timescale and short timescale. On a long

timescale, we consider resource allocation at the network slicing level, and we perform the interslice bandwidth allocation. On a short timescale, we consider resource allocation at the packet scheduling level, and we perform the intraslice bandwidth allocation to users. Specifically, we first allocate PRB resources to each slice, and then we perform bandwidth allocation to users associated with the slice. Specifically, for resources allocated among slices, noncooperative game theory is used for resource allocation among slices. For resources allocated among users associated with the slices, we use deep Q-learning (DQN) to find an optimal allocation solution for each user who is associated with slices [16].

We mainly consider two types of slices for resource allocation, that is, eMBB slices and URLLC slices. We can apply this situation to the Internet of Vehicles [17]. For example, there are two kinds of requirements in this network. One is automatic driving request, and the other is entertainment services. Then two types of slices will be established by the network operator to support these two kinds of requirements. That is, the eMBB slices deal with the entertainment requests, and the URLLC slices serve automatic driving requests. In order to answer the reviewers' doubt, we added some descriptions to explain why we did not take the mMTC slices into attention. Due to the large number of IoT devices, mMTC is mainly concerned with power optimization. In this paper, we mainly consider the bandwidth allocation problem, so we did not choose mMTC. We summarize our main contributions in the following.

We present a two-timescale resource allocation framework of C-RAN network slicing, which includes interslice and intraslice resource allocation. On a long timescale, we allocate resources at the network slicing level and allocate resources among slices. On a short timescale, we perform resource allocation at the packet scheduling level and allocate resources to users associated with slices.

We formulate a revenue modelling of C-RAN network slicing. The system revenue includes the cost of resource reservation on a long timescale and the profits obtained by serving users on a short timescale. The problem is solved to maximize the system revenue by finding the optimal bandwidth allocation strategy for users.

Simulation results demonstrate the advantage of the proposed strategy in terms of revenue in different situations. Compared with the baseline strategy, the system revenue of the proposed strategy is increased by 21%.

The rest of the paper is organized as follows: Section 2 presents the related work. Section 3 introduces the system architecture model and resource allocation model. Section 4 formulates the problem. Section 5 proposes an inter/intraslice bandwidth optimization strategy by game theory and the DQN method. Performance evaluation is conducted in Section 6, and we conclude this paper in Section 7.

2. Related Work

Network slicing provides the capability to allocate resources over a shared infrastructure [5, 18], and there are works on resource allocation for slicing using machine learning. The

current NS has faced several challenges, including slices isolation, mobility management, resource allocation for slices, and users associated with the slice [18]. In this paper, we mainly consider the resource allocation for slices and users.

2.1. Resource Allocation of Network Slicing. In order to make effective use of resources, it is necessary to allocate resources properly. Game theory is an effective tool to analyse the performance of resource allocation. References [19, 20] considered the resource allocation among slices. The authors studied the resource allocation and pricing problem for network slicing that captures interactions among access backhaul service providers and their UEs by using the Stackelberg game approach [19]. The authors considered that the system profit obtained by resource competition between individual slices is not the best [20]. Thus, the authors proposed an algorithm based on a bankrupt game to maximize the profit of the whole system inside of a single slice. Considering the waste of resources, a delay-limited over allocation prevention algorithm was proposed to minimize the oversupply ratio [21]. However, the flexibility of the algorithm still needs to be improved.

RL is a method for learning optimal resource allocation through trial and error. In [22], the authors proposed a model-free RL for resource optimization and security enhancement. Reference [23] proposed a resource block (RB) allocation method that is not affected by changes in the number of slices. The resource allocation is optimized using deep RL, which satisfies the slice requirements and allocates the minimum required number of RBs. In [24], a deep learning algorithm was proposed to assist service providers in allocating slices for tenants, and long short-term memory (LSTM) network is used to predict the channel conditions in the near future. A policy for resource allocation according to controlling slice admission was discussed in 5G RAN slicing environments. In [25], an ensemble learning method is used to reduce learning time and improve performance for the adaptive RAN. Due to the diverse QoS requirements in IoT network, different QoS requirements (i.e., latency and data rate) should be considered when performing resource allocation. However, the resource allocation policies in [22–25] ignored the heterogeneous QoS requirements in the IoT network.

2.2. Service Provisioning for RAN Slicing. Most works used SDN and NFV technologies to perform slice isolation and customization [5, 26–28]. Important efforts have been recently devoted to the development of network slicing in 5G, especially for RAN slicing. References [29, 30] considered a service framework of interslice and intraslice resource allocation. In [29], the authors proposed an autonomous resource slicing refinement scheme to adjust the allocated resources of slices. Moreover, the authors discussed a customized view of physical resources for intraslice resource allocation. In [30], the authors proposed a hierarchy system framework, which dynamically provisions and allocates appropriate quota of resources to a given slice at each base station based on its weight and preference ranking.

Meanwhile, some works investigate the wireless resource allocation among different RAN slices [7, 31]. In [31], the authors analysed the different options for configuring RAN slices by controlling certain parameters at the radio protocol layer. Furthermore, the authors considered resource allocation in a multiservice scenario with an eMBB slice and one mission critical slice. In order to meet the QoS requirements, considering three different scenarios, reference [32] proposed a resource allocation algorithm for unmanned aerial vehicle RAN slicing with limited bandwidth. However, from the operator's perspective, references [31, 32] ignored the long-term utility, which can prevent local optimal resource allocation strategy. In [33], the authors formulated the service multiplexing (i.e., URLLC and eMBB service) as an optimization problem to obtain the long-term slice utility. Specifically, a burst URLLC traffic model is used to guarantee the system performance of RAN slicing when URLLC slices are not well configured.

Thus far, there are a few studies on long-term revenue from the system perspective. Reference [4] maximizes the C-RAN operators' revenue by properly admitting the slice requests. To tackle this problem, a two-timescale framework can be adopted. Slice admission and bandwidth allocation were performed on a long timescale, and the operator generated beamformers on each short timescale. To achieve efficient resource utilization, some investigations have been conducted to design the resource reservation and intraslice resource allocation from the perspective of the system [34, 35]. In [34], a network slicing game was proposed to make a resource reservation with the aim of maximizing its user utility. However, this work considered the two steps on one single timescale. Considering the uncertainty of channel and the mobility of users, resource reservation and allocation on one timescale is a high computational complexity work. In [35], a two-timescale framework is used to perform resource reservation over a long timescale and resource allocation over a short timescale. However, the authors ignored the latency requirements of the URLLC service.

3. System Model

In this section, we present a C-RAN NS architecture, interslice resource allocation model at the network level, and intraslice resource allocation model at the packet scheduling level, respectively.

3.1. C-RAN Network Slicing Architecture Model. Figure 1 shows the C-RAN network slicing architecture. In 5G-beyond and 6G wireless networks, the BBU of each gNodeB is virtualized and centralized as a BBU pool, where each virtual BBU associates with gNodeB [17]. Each slice consists of a network slice management layer and a wireless interface protocol stack. It assumes that a tenant manages a slice, and the user has already been associated with the slice. There are multitenants in the system, and in C-RAN NS, BBU is responsible for resource management, including resource reservation and intraslice resource allocation, while the tenant performs resource reservation to request radio

FIGURE 1: C-RAN network slicing architecture.

resources of subchannels and power from physical infrastructure [35]. As shown in Figure 1, a two-timescale resource allocation framework is proposed; this framework consists of two levels: network level and packet scheduling level. At the network level, we allocate resources from the BBU pool to the slices. At the packet scheduling level, each slice allocates resources among users associated with the slice.

As the scenario is shown in Figure 1, the system consists of $\mathcal{D} = \{1, \dots, D\}$ RRHs, which connect the BBU pool via a fiber link. There are two kinds of slices, including eMBB slices $\mathcal{S}^e = \{1, \dots, S^e\}$ and URLLC slices $\mathcal{S}^u = \{1, \dots, S^u\}$. Suppose there are T long timescales and each long timescale contains Q equal short timescales. In different areas, the traffic is diverse because of the mobility of users. Therefore, we consider L areas, which consist of $\mathcal{T} = \{0, \dots, t, \dots, T-1\}$ long timescales. On a short timescale, there are U_q users, where $q \in Q = \{qT, \dots, (q+1)T-1\}$.

3.2. Interslice Resource Allocation Model. Network-level resource allocation can allocate resources to support the operation of slice according to the dynamic of service traffic. On a long timescale, in order to ensure the rationality of resource allocation, each slice needs to consider the other slices' allocation decision. Thus, to ensure fairness, this paper allocates resources in BBU to slices by game theory. From the perspective of slices, each slice is selfish to obtain more resources to maximize its own benefits. So, the slices are competitive, and the noncooperative game is used. We abstract the resource demands for different slices reasonably.

In this paper, the claimed PRBs by each slice have a relationship with the number of users and the typical traffic rates. We allocate PRBs to slices according to the bandwidth required by slices [36, 37]. A system with limited resources is considered, and let M represent the total number of PRBs in the system. There are $\mathcal{N}$ players in the game, where $\mathcal{N} = \{1, \dots, N\}$ represent the number of slices. The slice requests for the number of PRBs are formulated as follows [13]:

$$\hat{y}_{n,t} = U_{n,t} \times \frac{P_{n,t}}{D_{n,t} \cdot v}, \quad (1)$$

where $U_{n,t}$, $P_{n,t}$, and $D_{n,t}$ are the estimated value on a long timescale. $U_{n,t}$ is the average number of users of slices. $P_{n,t}$ is the average packet length and $D_{n,t}$ is the time required to transmit packets. Besides, v represents the processing rate of each PRB.

3.3. Intraslice Resource Allocation Model. At the packet scheduling level, each slice executes the PRB allocation to users based on the instantaneous service request. Based on the results of resource allocation in a slice of a long timescale, for intraslice, the beamforming decision is made for the users according to the QoS requirements, and each PRB is denoted by b .

To highlight the importance of the throughput to the eMBB slice and the delay sensitivity of the URLLC slice, it is simplified to the extent that the eMBB slice can satisfy the latency requirements and the URLLC slice can satisfy the throughput requirements of users [33]. Therefore, in the following, only the throughput constraint of eMBB and the latency constraint of URLLC are considered.

4. Problem Formulation

4.1. QoS Requirements of eMBB. In order to ensure the long-term benefits of the system, for eMBB slices, we maximize the throughput of the slices while considering the QoS. We suppose that the minimum throughput is R_q in a short timescale. Suppose the perfect channel state information can be known [38]. The channel vector can be given by $\mathbf{H} = [h_{1,d}^T, \dots, h_{u,d}^T]^T$. On account of the QoS requirement, the beamforming decision based on zero-forcing precoding [39] is denoted as $\mathbf{V} = \mathbf{H}^H(\mathbf{H}\mathbf{H}^H)^{-1}$, where $\mathbf{v} = \mathbf{V}/\|\mathbf{V}\| = [\mathbf{v}_{1,d}, \mathbf{v}_{2,d}, \dots, \mathbf{v}_{u,d}]$ is represented as the beamforming decision of user u to RRH d .

For users in the slice, the received signal of user u can be written as

$$\mathbf{h}_{u,d}^H(q)\mathbf{v}_{u,d}(q)i_u + \sum_{u' \neq U_q^e \setminus \{u\}} \mathbf{h}_{u,d}^H(q)\mathbf{v}_{u',d}(q)i_{u'} + \sigma^2, \quad (2)$$

where the first item on the right side of the equation denotes the desired signal of users and the second item is the interfering signal. The last item represents the noise signal.

We allocate PRBs to the user in order to obtain the target rate of u . Let $r_u^q(q)$ denote the rate that the user required, according to the Shannon formula, and the rate can be given by

$$r_u^q(q) = \lceil \delta_{q,n,u} \gamma_{n,t} b \rceil \log \left(1 + \frac{|\mathbf{h}_{u,d}^H(q)\mathbf{v}_{u,d}(q)|^2}{\sum_{u' \neq U_q^e \setminus \{u\}} |\mathbf{h}_{u,d}^H(q)\mathbf{v}_{u',d}(q)|^2 + \sigma^2} \right), \quad (3)$$

where $\delta_{q,n,u} \in [0, 1]$ represents the proportion of bandwidth allocated to users. Here, $\delta_{q,n,u} \gamma_{n,t} b$ is used to express the upward rounding of the bandwidth because the bandwidth is an integer and always needs to satisfy the QoS requirements.

Considering that an eMBB slice seeks high bandwidth [5], in order to ensure the throughput, the constraint is given as follows on a short timescale:

$$r_u^q(q) \geq R_q, \quad (4)$$

where R_q denotes the lower limit of the throughput of the q slot.

4.2. QoS Requirements of URLLC. Similarly, for URLLC slices $s \in \mathcal{S}^u$, let $I_u^u(q)$ be the received signal by the user in slice s during the q -th frame and $\mathbf{w}_{u,d}(q)$ be the transmit beamformer to the user from RRH d during the q -th frame. The received signal is

$$\mathbf{h}_{u,d}^H(q)\mathbf{w}_{u,d}(q)i_u + \sum_{u' \neq U_q^u \setminus \{u\}} \mathbf{h}_{u,d}^H(q)\mathbf{w}_{u',d}(q)i_{u'} + \sigma^2, \quad (5)$$

where the first item on the right side of the equation denotes the desired signal of users and the second item is the interfering signal. The last item represents the noise signal. The available rate for user u can be defined as

$$r_u^u(q) = \lceil \delta_{q,n,u} \gamma_{n,t} b \rceil \log \left(1 + \frac{|\mathbf{h}_{u,d}^H(q)\mathbf{w}_{u,d}(q)|^2}{\sum_{u' \neq U_q^u \setminus \{u\}} |\mathbf{h}_{u,d}^H(q)\mathbf{w}_{u',d}(q)|^2 + \sigma^2} \right). \quad (6)$$

In order to ensure the delay in URLLC slices, we model the latency tolerated by the user as

$$\frac{P_{q,u}}{r_u^u(q)} \leq D_n, \quad (7)$$

where $P_{q,u}$ is the packet length of user u in slice. D_n denotes the delay deadline of slice n .

4.3. The System Revenue. For resources allocation among slices, we perform resource reservation and game theory to allocate resources in the BBU pool. The profit is based on the number of users. In other words, the more the resources allocated to the slice, the more the users the slice can serve. Then, the tenants get more profit. To reduce resource waste, we introduce constant to express the cost of reserved resources [4, 35]. The cost function can be written as

$$C(\gamma_{n,t}) = c \cdot \gamma_{n,t}. \quad (8)$$

For users in the slice, we only consider the profit because the slice resources serve the users associated with the slice. Slices can gain more profit if slices can achieve higher throughput. Thus, the profit of serve users is modelled as

$$G_{\text{short}}^e(r_u^e(q), \mathbf{v}_{u,d}(q)) = A_1 r_u^e(q) - \sum_{u \in U_q^e} \mathbf{v}_{u,d}(q)^H \mathbf{v}_{u,d}(q), \quad (9)$$

where A_1 is a constant that expresses the unit profit of throughput. The second item on the right side of the equation denotes the cost of beamforming.

For the eMBB slice, the total profit can be given by the revenue on serve users and cost on resource allocation of slice:

$$Y^e(r_u^e(q), \mathbf{v}_{u,d}(q), \gamma_{n,t}) = \sum_{q \in Q} G_{\text{short}}^e(r_u^e(q), \mathbf{v}_{u,d}(q)) - C(\gamma_{n,t}). \quad (10)$$

For the URLLC slice, based on users packet length and bandwidth, the profit of serving users can be expressed as

$$G_{\text{short}}^u(P_{q,u}, r_u^u(q), \mathbf{w}_{u,d}(q)) = A_2 \frac{r_u^u(q)}{P_{q,u}} - \sum_{u \in U_q^u} \mathbf{w}_{u,d}(q)^H \mathbf{w}_{u,d}(q), \quad (11)$$

where A_2 is a constant. $r_u^u(q)/P_{q,u}$ is the reciprocal of the delay required by user because the tenant can obtain more profit when the lower latency is needed.

Thus, for each tenant, the total profit is as follows:

$$Y^u(P_{q,u}, r_u^u(q), \mathbf{w}_{u,d}(q), y_{n,t}) = \sum_{q \in Q} G_{\text{short}}^u(P_{q,u}, r_u^u(q), \mathbf{w}_{u,d}(q)) - C(y_{n,t}), \quad (12)$$

where the first item on the right side of the equation is the revenue of serving users, and the second item is the cost of resource reservation of slice. Above all, the profit of the whole system can be written as

$$Y^{\text{total}} = \sum_{s \in \mathcal{S}^e} Y^e(r_u^e(q), \mathbf{v}_{u,d}(q), y_{n,t}) + \sum_{s \in \mathcal{S}^u} Y^u(P_{q,u}, r_u^u(q), \mathbf{w}_{u,d}(q), y_{n,t}), \quad (13)$$

where the first item on the right side of the equation is the revenue from eMBB slices, and the second item is the revenue from URLLC slices.

We aim to allocate resources to maximize the system's profit, which consists of the whole tenants' profits in the system. Based on constraints (4) and (7), the optimization goal is given by

$$\begin{aligned} \max Y^{\text{total}} \\ \text{s.t. } c1: \sum_{q \in Q} [\delta_{q,n,u} y_{n,t}] &\leq y_{n,t} \\ c2: r_q^e(q) &\geq R_q \\ c3: \frac{P_{q,u}}{r_u^u(q)} &\leq D_n, \end{aligned} \quad (14)$$

where $c1$ denotes that the allocated resources cannot exceed the total resources in slice, $c2$ guarantees throughput requirements for eMBB slices, and $c3$ guarantees latency requirements for URLLC slices.

5. Service Provisioning Framework in RAN Slicing

We consider an inter/intraslice bandwidth optimization strategy to solve the problem (14). Firstly, resources are allocated to slices. Then, resources are allocated to users in slices based on the resources in the first step. Hence, the cost can be seen as a constant in the first step.

5.1. Resource Allocation among Slices. Since eMBB slices demand high bandwidth while URLLC slices require a low delay guarantee, a satisfaction function of slice is used to describe the fairness of resources allocation.

Generally, the more the resources the slice gets, the higher the satisfaction the slice has. Therefore, we use estimated and actual values to describe the satisfaction of each slice, which is expressed as

$$\phi_{n,t} = \frac{y_{n,t}}{\hat{y}_{n,t}}, \quad (15)$$

where $y_{n,t}$ and $\hat{y}_{n,t}$ denote the actual allocation of resources and estimated values, respectively. Specifically, $y_{n,t}$ represents the number of PRBs allocated to slices, while $\hat{y}_{n,t}$ is the number of PRBs that the slices requested.

From the system perspective, the optimal resource allocation strategy is to maximize the satisfaction of all slices. Because slices share their satisfaction with the BBU pool, BBU could in principle consider an allocation that optimizes the overall performance of the network. The optimal allocation is given by the following maximization:

$$\begin{aligned} \max \sum_{n \in \mathcal{N}} \sum_{t \in \mathcal{T}} \phi_{n,t} \\ \text{s.t. } \sum_{n \in \mathcal{N}} \sum_{t \in \mathcal{T}} y_{n,t} &\leq M, \end{aligned} \quad (16)$$

and the final allocation result of the slice is expressed as $Y_{n,t} = \{y_{n,1}, y_{n,2}, \dots, y_{n,T}\}$. The optimization objective (16) can affect objective (14) because the revenue is based on the allocation result $y_{n,t}$ in objective (14).

There are two optimizations in this paper, which are optimization formulations (14) and (16). However, this is not a multiobjective optimization problem. We aim to obtain the long-term revenue of the system, which cooperates with optimization formulation (14). We consider interslice and intraslice resource allocation to reach this goal. For interslice resource allocation, we use optimization formulation (16) to reach Nash equilibrium. For intraslice resource allocation, DQN is used to allocate resources for users in the slice. On the whole, the optimization formulation (16) is served for the cost, which expresses resource reservation on long timescales, and the optimization formulation (14) is served for the profit, which includes the cost and the revenue for serving users.

On the basis of slices' required PRB number and the entire number in the BBU pool, the noncooperation game is carried out among slices. When playing a game, we may not be able to meet all slices' requirements. Thus, this is a game between the resource pool BBU and the resources required by the slice. According to utility function (15), the network slicing has a Nash equilibrium [40] when the utility attains the maximum. In order to achieve the Nash equilibrium, we first gave all possible resource allocation decisions and then estimated the number of PRBs required for slices based on equation (1). Finally, we reached Nash equilibrium by traversing the strategy space based on equation (16).

In the following, we analyse the strategy to ensure that a Nash equilibrium exists after the allocation. We analysed two situations where resources are sufficient and insufficient to confirm the existence of Nash equilibrium. In case 1, there is no doubt that each slice can reach satisfaction of 1. In case 2, Nash equilibrium also can be reached according to equation (16):

Case 1: when resources are sufficient, BBU can always satisfy all requirements of slices. Each slice has the satisfaction with 1; thus, the existence of a Nash equilibrium can be guaranteed.

Case 2: when resources are insufficient, BBU cannot satisfy all requirements. In order to get the maximum satisfaction of the system, BBU would first satisfy the slices with fewer resource requirements. Since we assume that the requests of slices remain constant over a long timescale, there is always a resource allocation policy that achieves Nash equilibrium.

5.2. Packet Scheduling among Users. Since the channel state information is uncertain, the state space is large, while Q-learning is suitable for algorithms with small state space. Therefore, based on the first stage allocation results, we use DQN to allocate resources for users in slices. We describe the user arrival as a Markov model. Each slice acts as an agent and learns the optimal bandwidth allocation strategy to maximize its own revenue. Take URLLC slice as an example.

State space is defined as $\text{sta}_q = \{\mathbf{h}_{q,u,d}, P_{q,u}, y_{n,t}b\}$. Similar to the URLLC slice, the state space of the eMBB slice is represented as $\text{sta}_q = \{\mathbf{h}_{q,u,d}, r_{q,u}, y_{n,t}b\}$.

DQN performs the bandwidth allocation action based on the currently observed state. The action space is given as $\text{act}_q = \{\delta_1 y_{n,t}b, \delta_2 y_{n,t}b, \dots, \delta_k y_{n,t}b\}$. According to equation (11), the reward function is expressed as

$$\text{rewd}_q = A_2 \frac{r_u^u(q)}{P_{qu}}. \quad (17)$$

As mentioned above, the resources allocated among slices on a long timescale can be seen as a constant. When a new user arrives on a short timescale, the remaining bandwidth resources on the long timescale are updated according to the bandwidth allocated to the new user. The update is performed as follows:

$$\begin{cases} Q_{q+1}(\text{sta}_q, \text{act}_q, \theta_q) = Q_q(\text{sta}_q, \text{act}_q, \theta_q) + \alpha_q E_k, \\ E_k = \text{rewd}_q + \gamma \max Q(\text{sta}', \text{act}', \theta'_q) - Q_q(\text{sta}_q, \text{act}_q, \theta_q), \end{cases} \quad (18)$$

where α_q is the learning rate, γ represents discount, and sta' shows the next state. act' expresses the action that is the best reward that can be obtained at state sta' . E_k is the cumulative reward. In order to find the approximate optimal solution of the objective function as much as possible, the mean square error function is used as the loss function of DQN, which is defined as

$$L(\theta) = E[O_q - Q(\text{sta}_q, \text{act}_q; \theta)]^2, \quad (19)$$

where O_q is the optimization objective of the value function and θ is the parameter of DQN.

Above all, the resource allocation frame can be concluded as follows and as shown in Algorithm 1:

Step 1: for resources allocated among slices, a non-cooperation game is used to allocate resources among slices. When Nash equilibrium is reached, resources can be obtained for each slice.

Step 2: for resources allocated among users associated with the slice, based on the result of step 1, DQN is used

to learn a resource allocation strategy to achieve the best revenue of the system. We first train the DQN with state, action, and reward, then place these samples into the experience pool, and update the network by gradient descent until the loss function converges.

Next, we analyse the complexity of Algorithm 1. Let M represent the PRB number and N be the number of slices. In the first step, the computational complexity of a noncooperative game is $O(M^N)$. In the second step, the computational complexity of DQN is $O(n_t n_m)$. n_t is the number of time steps in each episode, and n_m refers to the number of episodes. Since MN is growing much faster than $n_t n_m$, the total computational complexity of the algorithmic 1 is $O(M^N)$.

6. Simulation and Results

In this section, simulation results are presented to evaluate the performance of the proposed inter/intraslice bandwidth optimization strategy. The proposed strategy is compared with the other four schemes, which are average resource allocation strategy, random resource allocation strategy, baseline strategy I, and baseline strategy II [35]. The average strategy is the equal distribution of resources to users associated with slices. Baseline I and baseline II are similar to our proposed scheme, baseline I addresses the issue of channel state information (CSI) uncertainty, and baseline II addresses the issue of user traffic variation. We consider the C-RAN architecture with 3 RRHs and 1 BBU pool. The parameter settings are mainly derived from references [38, 41]. The radius of each RRH is 100 m. Users are randomly generated in the coverage area. The entire number of PRBs of the BBU pool is 100, and the bandwidth of each PRB is 180 kHz. We consider two slices in the system: one is the eMBB slice, and the other is the URLLC slice. The parameter settings are shown in Table 1.

Figure 2 illustrates the resource allocation results of slices. We can see that the eMBB slice requires more PRBs than the URLLC slice. When the number of resources required is greater than system resources, there is a situation where the actual allocated resources are less than the required. For example, the black block in Figure 2 is the inadequate number of PRBs, which is expressed as eMBB lack in this paper. Since the number of PRBs requested by URLLC slices is small, the requests for URLLC slices can always be satisfied according to the utility function (16). Because the number of resources by URLLC slices is small, it is easy to achieve high satisfaction. Besides, in the case of insufficient resources, the proposed utility function can give priority to delay sensitive services.

Figure 3 shows the convergence of the reward. At the beginning of the iteration, there are fluctuations caused by the large initial exploration rate. As the number of iterations increases, the reward function gradually increases and then tends to converge at around 70,000 iterations. It indicates that the agent gradually finds an action, which maximizes the revenue of the system.

Input: PRBs in BBU, slice number N , estimated values of $U_{n,t}$, $P_{n,t}$, $D_{n,t}$ and processing rate v .
Output: allocation results $Y_{n,t} = \{y_{n,1}, y_{n,2}, \dots, y_{n,T}\}$ and PRBs are allocated to users.
Initialize a number of PRBs for slice requests.
Initialize the DQN parameter.
Step1: // **resource allocation** for slices
for $n = 1: N$ **do**:
 obtain the number of PRBs for slice requests according to equation (1).
end for
Calculate the utility value of different resource strategies through equation (15).
BBU adjusts the allocation policy according to equation (16) until Nash equilibrium is reached.
Step2: // resource allocation for users in slices
for each decision episode **do**:
 //selection
 with probability ε , randomly select an action, otherwise $\text{act}_q = \arg\max_{\text{act}} Q(\text{sta}_q, \text{act})$.
 //optimization
 Obtain the optimal allocation policy based on solving (14).
 //updating
 Observe the reward and new state sta_{q+1} .
 Store $(\text{sta}_q, \text{act}_q, \text{rewd}_q, \text{sta}_{q+1})$ into memory D .
 Updating DQN with gradient drop.
end for

ALGORITHM 1: Service provisioning of C-RAN slicing algorithm.

TABLE 1: Parameter settings.

Parameter	Value
BBU PRB numbers	100
Noise power σ^2	10^{-13} W
Transmit power	1 W
Cost c	0.5
A_1	0.2×10^{-6}
A_2	1
α	0.5×10^{-3}
Final exploration	0.2×10^{-2}

FIGURE 2: Resource allocation results among slices.

FIGURE 3: Reward.

In Figure 4, the performance of different package length and rate is examined in the URLLC slice. It can be seen that the revenue increases with the rate. The reason is as follows. For the same package size, when the user is allocated more PRBs, a lower latency can be achieved. What is more, according to (11), the revenue of slice increases when the user gets a higher rate. As is shown in the figure, there are two examples. These three values represent rate, packet length, and profits. When the rate is about the same, the larger packet can obtain the higher revenue because of the lower latency.

In Figure 5, we plot the revenue with different PRBs when the number of users is 5. Since the number of resources is constant on a long timescale, the revenue is influenced by the number PRBs. It can be observed that the revenue gradually increases with the number of PRBs. We note that,

FIGURE 4: The relationship between packet length, rate, and revenue.

FIGURE 5: Effect of PRB on revenue.

as the PRBs increase, the rate available to users increases, and slice can achieve higher revenue according to equation (11). The revenue improves slowly, that is, only 2.8%, when the number of resources goes from 15 to 17. We conclude that the user has almost reached the rate limit, so the system does not allocate more resources to users in order to avoid resource waste.

Figure 6 represents the two-slice revenues at different iterations. Specifically, the number of users in the slice is controlled by 5, and the number of PRBs is 15. We guarantee the same channel conditions for both service types in order to control variables. Similar to Figure 3, as the iteration increases, the revenue gradually increases because the agent gradually finds the optimal resource allocation strategy to maximize the revenue of slices. From Figure 6, we can see that the proposed strategy consistently outperforms the other two strategies, and the proposed strategy improves the revenue by 21% over the average resource allocation strategy. Moreover, for the same iteration, the revenue of the URLLC slice is lower than the eMBB slice. According to (9) and (11),

FIGURE 6: Revenue of URLLC and eMBB slice.

FIGURE 7: The impact of user rate on revenue.

the URLLC slice has strict requirements for latency, while the revenue of the eMBB slice is only concerned with data rate. Therefore, the eMBB slice obtains higher revenue than the URLLC slice under the same channel condition of users.

In Figure 7, we plot the effect of the user rate on revenue. As the number of users increases, the revenue gradually increases before levelling off. The revenue increases when the number of users is less than 4. We conclude that the slice is well resourced to serve those users. When the number of users is more than 4, the slice can only guarantee the QoS requirement and cannot allocate too many resources to them; as a result, the revenue increases slowly. The result shows that the revenue of the proposed strategy is improved by 87.5% compared with the baseline strategy I when the number of users is 2. Compared with the average strategy and random strategy, the proposed

FIGURE 8: Throughput of eMBB slice.

FIGURE 9: Package size of the URLLC slice.

algorithm improves the revenue by 19.1% and 68.3%, respectively. Since we assume that perfect channel state information is known, the profit is higher than baseline I. We can see that higher revenue can be obtained from serving more users and the impact of the traffic variation and CSI uncertainty.

Figure 8 shows the effect of different throughputs on the revenue of the eMBB slice. We conclude that the revenue gradually increases and then levels off as the throughput increases. If the throughput is greater than 11 Mb/s, the revenue increases slowly because the throughput of the slice almost reaches the upper limit. Compared with the average resource allocation strategy, the revenue of the proposed strategy is increased by 26.7%, and compared with the random allocation strategy, the revenue is increased by 58.3%.

Figure 9 depicts the impact of different package size on the revenue of URLLC slice. We have found that the revenue gradually decreases as the package size increases. The result indicates that, for the same strategy, the slice can obtain a smaller revenue when the package size is larger. The proposed strategy improves the revenue by 16.7% compared with the average strategy and 48.9% compared with the random strategy.

7. Conclusions

We proposed an inter/intraslice bandwidth optimization strategy to obtain the maximized system revenue from the long term. Considering the selfishness of slices and the limitation of resources, a noncooperation game is used for interslice resource allocation. We also illustrated that the game has a Nash equilibrium even when resources are insufficient. Then for intraslice resource allocation, DQN is used to find a resource allocation policy for the users associated with the slice to maximize the system revenue. Numerical results demonstrate the significant performance revenue of our proposed strategy when compared with the baseline mechanisms. For example, the proposed strategy improves the revenue by 21% over the baseline strategy. In this work, we consider the users who have already been associated with the slice, and we will subsequently study the problem of user-slice association in the future work.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was partially supported by the Natural Science Foundation of China (Grants 61901070, 61801065, 61771082, 61871062, and U20A20157), the Science and Technology Research Program of Chongqing Municipal Education Commission (Grants KJQN202000603, KJQN201900611, and KJQN202000626), the Natural Science Foundation of Chongqing (Grant cstc2020jcyj-zdxmX0024), and the University Innovation Research Group of Chongqing (Grant CXQT20017).

References

- [1] M. Youssef, M. Hassan, and G. Kortuem, "Next generation IoT: toward ubiquitous autonomous cost-efficient IoT devices," *IEEE Pervasive Computing*, vol. 18, no. 4, pp. 8–11, 2019.
- [2] Z. Xu, J. Tang, C. Yin et al., "ReCARL: resource allocation in cloud RANs with deep reinforcement learning," *IEEE Transactions on Mobile Computing*, vol. 1, 2020.

- [3] J. Wu, Z. Zhang, Y. Hong, and Y. Wen, "Cloud radio access network (C-RAN): a primer," *IEEE Network*, vol. 29, no. 1, pp. 35–41, 2015.
- [4] J. Tang, B. Shim, and T. Q. S. Quek, "Service multiplexing and revenue maximization in sliced C-RAN incorporated with URLLC and multicast eMBB," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 4, pp. 881–895, 2019.
- [5] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck, "Network slicing and softwarization: a survey on principles, enabling technologies, and solutions," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2429–2453, 2018.
- [6] M. Maule, J. Vardakas, and C. Verikoukis, "5G RAN slicing: dynamic single tenant radio resource orchestration for eMBB traffic within a multi-slice scenario," *IEEE Communications Magazine*, vol. 59, no. 3, pp. 110–116, 2021.
- [7] 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*, vol. 20, no. 12, pp. 3409–3422, 2021.
- [8] T. Guo and A. Suarez, "Enabling 5G RAN slicing with EDF slice scheduling," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2865–2877, 2019.
- [9] Y. L. Lee, J. Loo, T. C. Chuah, and L.-C. Wang, "Dynamic network slicing for multitenant heterogeneous cloud radio access networks," *IEEE Transactions on Wireless Communications*, vol. 17, no. 4, pp. 2146–2161, 2018.
- [10] G. Chen, Y. Zhang, Y. Shi, and Q. Zeng, "Joint resource allocation and admission control mechanism in software defined mobile networks," *China Communications*, vol. 16, no. 5, pp. 33–45, 2019.
- [11] Y. Hua, R. Li, Z. Zhao, X. Chen, and H. Zhang, "GAN-powered deep distributional reinforcement learning for resource management in network slicing," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 2, pp. 334–349, 2020.
- [12] M. Min, L. Xiao, Y. Chen, P. Cheng, D. Wu, and W. Zhuang, "Learning-based computation offloading for IoT devices with energy harvesting," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1930–1941, 2019.
- [13] C. Hatim and V. Christos, "OPEX-limited 5G RAN slicing: an over-dataset constrained deep learning approach," in *Proceedings of the ICC 2020-2020 IEEE International Conference on Communications (ICC)*, pp. 1–6, Dublin, Ireland, 2020.
- [14] X. Cheng, Y. Wu, G. Min, A. Y. Zomaya, and X. Fang, "Safeguard network slicing in 5G: a learning augmented optimization approach," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 7, pp. 1600–1613, 2020.
- [15] A. Arjun, V. Gustavo, and S. Sanjay, "Joint scheduling of URLLC and eMBB traffic in 5G wireless networks," *IEEE/ACM Transactions on Networking*, vol. 28, no. 2, pp. 477–490, 2020.
- [16] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: a brief survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, 2017.
- [17] J. Mei, X. Wang, and K. Zheng, "An intelligent self-sustained RAN slicing framework for diverse service provisioning in 5G-beyond and 6G networks," *Intelligent and Converged Networks*, vol. 1, no. 3, pp. 281–294, 2020.
- [18] X. Foulkas, 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.
- [19] T. D. Tran and L. B. Le, "Resource allocation for multi-tenant network slicing: a multi-leader multi-follower Stackelberg game approach," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 8, pp. 8886–8899, 2020.
- [20] Y. Jia, H. Tian, and S. Fan, "Bankruptcy game based resource allocation algorithm for 5G Cloud-RAN slicing," in *Proceedings of the 2018 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, Barcelona, Spain, April 2018.
- [21] H.-T. Chien, Y.-D. Lin, C.-L. Lai, and C.-T. Wang, "End-to-End slicing with optimized communication and computing resource allocation in multi-tenant 5G systems," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 2079–2091, 2020.
- [22] M. Z. Chowdhury and M. T. Hossain, "Applying model-free reinforcement learning algorithm in network slicing for 5G," in *Proceedings of the 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*, pp. 1–4, Dhaka, Bangladesh, May 2019.
- [23] Y. Abiko, T. Saito, D. Ikeda, K. Ohta, T. Mizuno, and H. Mineno, "Flexible resource block allocation to multiple slices for radio access network slicing using deep reinforcement learning," *IEEE Access*, vol. 8, pp. 68183–68198, 2020.
- [24] M. Toscano, F. Grunwald, and M. Richart, "Machine learning aided network slicing," in *Proceedings of the 2019 21st International Conference on Transparent Optical Networks (ICTON)*, pp. 1–4, Angers, France, July 2019.
- [25] S. II Moon, H. Hirayama, and Y. Tsukamoto, "Ensemble learning method-based slice admission control for adaptive RAN," in *Proceedings of the 2020 IEEE Globecom Workshops (GC Wkshps)*, pp. 1–6, Taipei, Taiwan, December 2020.
- [26] O. Sallent, J. Perez-Romero, R. Ferrus, and R. Agusti, "On radio access network slicing from a radio resource management perspective," *IEEE Wireless Communications*, vol. 24, no. 5, pp. 166–174, 2017.
- [27] D. A. Chekired, M. A. Togou, L. Khokhi, and A. Ksentini, "5G-Slicing-Enabled scalable SDN core network: toward an ultra-low latency of autonomous driving service," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 8, pp. 1769–1782, 2019.
- [28] S. Dawaliby, A. Bradai, and Y. Pousset, "Distributed network slicing in large scale IoT based on coalitional multi-game theory," *IEEE Transactions on Network and Service Management*, vol. 16, no. 4, pp. 1567–1580, 2019.
- [29] G. Sun, K. Xiong, G. O. Boateng, D. Ayepah-Mensah, G. Liu, and W. Jiang, "Autonomous resource provisioning and resource customization for mixed traffics in virtualized radio access network," *IEEE Systems Journal*, vol. 13, no. 3, pp. 2454–2465, 2019.
- [30] K. Xiong, S. Samuel Rene Adolphe, G. O. Boateng, G. Liu, and G. Sun, "Dynamic resource provisioning and resource customization for mixed traffics in virtualized radio access network," *IEEE Access*, vol. 7, pp. 115440–115453, 2019.
- [31] J. Prez-Romero, O. Sallent, and R. Ferrus, "On the configuration of radio resource management in a sliced RAN," in *Proceedings of the NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, pp. 1–6, Taipei, Taiwan, April 2018.
- [32] J.-W. Cho, P. Yang, and T. Q. S. Quek, "Service-aware resource allocation design of UAV RAN slicing," in *Proceedings of the 2020 International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 801–805, Jeju, South Korea, October 2020.
- [33] P. Yang, X. Xi, T. Q. S. Quek, J. Chen, X. Cao, and D. Wu, "How should I orchestrate resources of my slices for bursty

- URLLC service provision?" *IEEE Transactions on Communications*, vol. 69, no. 2, pp. 1134–1146, 2021.
- [34] P. Caballero, A. Banchs, G. de Veciana, and X. Costa-Perez, "Network slicing games: enabling customization in multi-tenant mobile networks," *IEEE/ACM Transactions on Networking*, vol. 27, no. 2, pp. 662–675, 2019.
 - [35] H. Zhang and V. W. S. Wong, "A two-timescale Approach for network slicing in C-RAN," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 6, pp. 6656–6669, 2020.
 - [36] Y. Yiwei Yu, E. Dutkiewicz, and M. Mueck, "Downlink resource allocation for next generation wireless networks with inter-cell interference," *IEEE Transactions on Wireless Communications*, vol. 12, no. 4, pp. 1783–1793, 2013.
 - [37] J. Yli-Kaakinen, T. Levanen, S. Valkonen et al., "Efficient fast-convolution-based waveform processing for 5G physical layer," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 6, pp. 1309–1326, 2017.
 - [38] W. Hao, Z. Chu, F. Zhou, S. Yang, G. Sun, and K.-K. Wong, "Green communication for NOMA-based CRAN," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 666–678, 2019.
 - [39] G. Interdonato, M. Karlsson, E. Bjornson, and E. G. Larsson, "Local partial zero-forcing precoding for cell-free massive MIMO," *IEEE Transactions on Wireless Communications*, vol. 19, no. 7, pp. 4758–4774, 2020.
 - [40] W. Saad, Z. Han, M. Debbah, A. Hjørungnes, and T. Basar, "Coalitional game theory for communication networks," *IEEE Signal Processing Magazine*, vol. 26, no. 5, pp. 77–97, 2009.
 - [41] E. Aqeeli, A. Moubayed, and A. Shami, "Power-aware optimized RRH to BBU allocation in C-RAN," *IEEE Transactions on Wireless Communications*, vol. 17, no. 2, pp. 1311–1322, 2018.