Journal of Network and Computer Applications 157 (2020) 102573

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Resource slicing and customization in RAN with dueling deep Q-Network☆

Guolin Sun [a,*], Kun Xiong [a], Gordon Owusu Boateng [a], Guisong Liu [a,b], Wei Jiang [c]

[a] School of Computer Science and Engineering, University of Electronic Science and Technology of China (UESTC), Chengdu, Sichuan, 611731, PR China
[b] School of Computer Science, Zhongshan Institute, UESTC, Zhongshan, Guangdong, 528400, PR China
[c] German Research Center for Artificial Intelligence (DFKI GmbH), Kaiserslautern, 67663, Germany

ARTICLE INFO

Index Terms:
Resource slicing
Resource allocation
Network virtualization
Dueling deep Q-network

ABSTRACT

The emerging future generation 5G technology is expected to support service-oriented virtualized networks where different network applications provide unique services. 5G networks have the potential to allow completely different slices to co-exist in a substrate network and satisfy the differentiated requirements of various users. In networks with heterogeneous traffics, operators are required to provide services in isolation since each operator has its own defined performance requirements. However, achieving an efficient resource provisioning mechanism for such traffics is very challenging. This paper proposes a coarse resource provisioning scheme and a dynamic resource slicing refinement scheme based on dueling deep reinforcement learning for virtualized radio access network. Firstly, coarse resource provisioning scheme provisions and allocates radio resource to slices based on preferences and weights at different base stations. Secondly, reinforcement learning based slicing refinement adjusts the resource allocated to slices autonomously in order to balance satisfaction and resource utilization. The proposed dueling DQN algorithm unifies two objectives (QoS satisfaction and resource utilization) by weights to indicate the importance of each factor in the reward function. After the dueling DQN algorithm has output actions to provision resource at slice level, BS-level resource update is performed. Also, a common learning agent is used to control the activities of all the slices in the network. Then, a shape-based resource allocation algorithm is proposed to customize the diverse requirements of users to improve user satisfaction and resource utilization. Finally, a comprehensive performance evaluation is conducted against state-of-the-art solutions based on OFDMA air-interface design. The results reveal that the proposed algorithm balances satisfaction and resource utilization with 80% of the available resources. The algorithm also provides performance isolation such that, a sudden change in user population in one slice does not affect the others.

1. Introduction

The fifth generation (5G) wireless technology is envisioned to provide a wide variety of new services which will in turn increase the number of mobile cellular network subscribers at a rapid rate. These new services will come with new technological challenges and stringent service requirements to the various services. 5G upgrades the performance of the legacy 4G technology in various ways including increased system capacity, lower capital cost (CapEx) and operational cost (OpEx), massive increase in the number of connected devices, consistent quality of experience (QoE), higher throughput and reduced end-to-end (E2E) latency. The improvement on 4G services can be categorized into the following scenarios *i*) Enhanced mobile broadband (eMBB), which has a requirement of achieving the peak speed up to 20 Gbps, with the help of new radio access techniques (RATs), mmWave communication, massive multiple-input-multiple-output (MIMO), etc. *ii*) Ultra-reliable and low-latency communication (URLLC) which requires E2E delivery delay in few milliseconds for mission-critical services, such as smart-grid, self-driving cars, motion control and smart factory with sensors and remotes. *iii*) Massive internet of things (IoT) scenario where many devices are connected with hundreds of millions of IoT nodes like static sensors. *iv*) The scenario of high definition TV (Hdtv) and on-line video streaming results in low latencies and large flows such as virtual reality. All of these services will operate in a common mobile cellular network. The traffics of

☆ This work is supported by National Natural Science Foundation of China, China, Grant No. 61771098, by the Fundamental Research Funds for the Central Universities, China, under Grant No. ZYGX2018J068, by the fund from the Department of Science and Technology of Sichuan Province, China, Grant No. 2017GFW0128, 8ZDYF2265, 2018JYO578and 2017JY0007 and the ZTE Innovation Research Fund for Universities Program 2016.

* Corresponding author.
E-mail addresses: guolin.sun@uestc.edu.cn (G. Sun), lgs@uestc.edu.cn (G. Liu), wei.jiang@dfki.de (W. Jiang).

https://doi.org/10.1016/j.jnca.2020.102573
Received 7 July 2019; Received in revised form 1 December 2019; Accepted 8 February 2020
Available online 13 February 2020

different applications or services co-existing in the same network are referred to as heterogeneous traffics. To satisfy the versatile QoS requirements of the different users is one challenging aspect of heterogeneous traffics.

Network virtualization (NV) allows operators to share a common physical infrastructure and have networks co-existing in a flexible, dynamic manner utilizing the available resources more efficiently (Derakhshani et al., 2018). Lately, network virtualization has been explored heavily in many research fields (Kalil et al., 2015). The physical infrastructure and resource owned by the infrastructure provider (InP) are partitioned into virtualized infrastructure and resource known as slices, to be managed by different service operators. Each slice has a particular service or application it supports making the network a heterogeneous traffics network. Each application has its own service requirements such as maximum delay it can experience, minimum rate it can achieve, its maximum or aggregated capacity and its maximum QoS satisfaction. As the number of mobile subscribers increases daily at a rapid rate, it is laudable to extend virtualization in wired networks to wireless networks. Network function virtualization (NFV) and software-defined networking (SDN) are the main enabling technologies for network virtualization. NFV increases the utilization of the network hardware while decreasing the operational cost (OpEX) and capital cost (CapEx) of infrastructure deployment by decoupling the functionality of the network and the hardware. SDN enables programmability and innovation of the network by decoupling the control plane from the data plane. The combination of NFV and SDN technologies achieves the following in 5G, *i*) a programmable and extensible service-oriented architecture in terms of network infrastructure, network services and mobile applications; *ii*) network slicing that decouples operations of virtual network on top of physical infrastructures with slice isolation and customized resource allocation for each tenant.

Many works have investigated network and resource slicing especially in 5G. However, the following questions have still not been answered in heterogeneous traffic scenarios: (1) With scarce resource, how can resource efficiency be achieved for all slices while matching slice demand? (2) How can we guarantee the QoS satisfaction of heterogeneous slices by automatically partitioning resources for slices based on their service level agreement (SLA) requirement? (3) How can a resource allocation scheme interact with the varying nature of a highly dynamic wireless environment in real-time by adapting to rea-time changes in user population? In Kokku et al. (2012), the authors proposed a resource provisioning scheme for slices by configuring slice and flow schedulers. However, their only objective was to provide resource to slices in order to maximize the sum rate of users. Authors in Liang and Yu (2015) proposed a model-based solution with a common cost objective function for all slices. They jointly formulated virtualization and user resource allocation, which could not achieve the customization of resources. Practically, a heterogeneous network consists of different slices with different QoS requirements and different objectives. Authors in Mahindra et al. (2013) presented a network-wide radio access network (RAN) scheme for resource sharing which provides isolation for different slices existing on the BSs. However, the QoS constraints on the slices by SLA cannot account for the change in user population in request pattern in real-time. In Aijaz (2018), the author considered a resource slicing scheme for a specific application i.e. haptic communications without taking into consideration resource slicing for heterogeneous traffics. However, it is much reasonable to discuss a resource slicing scheme for heterogeneous traffic scenarios (Wang et al., 2016a). A typical 5G wireless network is expected to support differentiated services or applications ranging from delay-sensitive applications to delay-elastic ones. In this paper, we consider all of the services characterized by both delay and throughput requirements. The autonomous resource slicing and customization is discussed for heterogeneous traffics to balance resource utilization and QoS satisfaction of slices with minimum amount of resource.

To achieve dynamic resource slicing and customization in wireless networks, we first design an algorithm that reserves resources of slices among BSs based on weights and preferences of the slices on the designated BSs. Then, we present a dynamic resource slicing refinement scheme that adjusts the allocated resource of slices to balance QoS satisfaction and resource utilization of the slices based on deep reinforcement learning (DRL) technique. We introduce two forms of DRL; deep Q-network (DQN) and dueling DQN. Finally, we customize the physical resources for intra-slice resource allocation with logically shaped resource units (LSRUs). An LSRU describes a set of adjacent time-frequency resource blocks (RBs) which is determined by transmit rate and delay requirements of users. After users are admitted into a slice, they can be rescheduled or re-associated with designated BSs considering user priorities at the BSs, which can provide a better channel quality. The main contributions of this paper are summarized as:

- We develop a heuristic algorithm to perform slice resource provisioning considering different QoS characteristics of heterogeneous traffics to solve the problem of dynamically changing channel status considering the slices' preferences and weights at different BSs. In multiple BS scenario, radio resource provisioning with flexible rescheduling is considered.
- We propose an autonomous slice resource refinement scheme at the BSs using a DRL approach in a dynamic wireless network environment. Here, we unify the QoS satisfaction and resource utilization using weights to show the importance of each factor in the reward. After the learning agent receives feedback from the environment, it autonomously adjusts the resource of the slices at a specific decision epoch in order to balance QoS satisfaction and resource utilization with minimum amount of resources.
- We update the resource at BS-level and propose a shape-based intra-slice physical resource allocation algorithm for users to enhance the effective resource utilization and QoS satisfaction of users considering versatile users' QoS requirements. This low-complexity algorithm can be easily implemented in any orthogonal frequency division multiple access (OFDMA)-based RAN.
- Based on OFDMA air interface design, we conduct extensive simulations at system-level to evaluate the performance of the proposed scheme from the perspectives of mobile virtual network operators (MVNOs), infrastructure providers (InPs), service providers (SPs) and mobile users. The performance of our algorithm is compared against state-of-the-art baseline solutions.

The rest of paper is organized as follows: Section 2 presents the related works. Section 3 presents the system model segmented into virtualization model, network model and utility model. Section 4 presents the resource provisioning, Markov decision process (MDP) problem formulation and the DRL algorithm. Section 5 presents the resource update at BS-level and the proposed heuristic algorithm. Simulation results are discussed in Section 6. We conclude the paper in section 7.

2. Related works

Wireless network virtualization has received a lot of attention as a key enabler in future 5G technology. As such, several researches have been conducted on the resource slicing and allocation aspect of this field of study. The challenges involved in realizing everything-as-a-service (XaaS) in service-oriented 5G network was investigated in Chang et al. (2018). In da Silva et al. (2016), the authors investigated the effects of network slicing on some aspects of 5G RAN design. Some authors proposed a framework to separate the functions of network operator and SPs in a virtualized wireless network (Fu and Kozat, 2010). The interaction between the network operator and SPs was modeled as a stochastic game, which is played by the SPs and regulated by the network operator. The advantages of long term evolution (LTE) virtualization based on resource sharing between different MVNOs was discussed in Zaki et al. (2011). *Kokku etc.* proposed a network virtualization substrate (NVS) solution

that achieves effective virtualization of wireless resources in cellular networks in Kokku et al. (2012). In virtualizing a BS's resources into various slices, they considered three requirements: isolation, customization and efficient resource utilization. In Guo and Arnott (2013), the authors proposed a partial resource reservation scheme for LTE networks considering scheduling and admission control policy which was based on resource availability. *Panchal* et al. studied various resource sharing techniques for simple and complex methods and tested them via simulation in LTE network (Panchal et al., 2013). The authors in Garces et al. (2015) designed a RAN multi-tenant cell-slicing controller (RMSC) to allow the flexible sharing of RAN resources among multiple virtual operators. RMSC was benchmarked against distributed static slicing and centralized load balancing approaches to show its gains in terms of user satisfaction and resource utilization. A resource virtualization problem was discussed in Kalil et al. (2014). The problem was formulated as a binary integer programming problem which was solved by a low complexity iterative algorithm. In Kamel et al. (2014), authors also developed a resource allocation scheme to allocate resources of InP to SPs dynamically depending on the agreement between them. The aim of this scheme was to achieve the maximum system throughput and fairness among users. Wireless resource virtualization underlaying LTE network was proposed for D2D communications in Moubayed et al. (2015). The authors formulated the problem as an integer non-linear programming problem and suggested an optimality solution. Authors in Jiang et al. (2017a) investigated on-the-fly resource sharing between mobile users using an SDN approach to provide users with flexible access to available radio resources. In Caballero et al. (2019), authors analyzed a model for resource sharing well-known as 'share-constrained proportional allocation' for network slicing. Tseliou et al. (2016) proposed a resource slicing scheme to coordinate BSs in order to provide resources to multiple MVNOs in a heterogeneous network. The negotiation between MVNOs allows BSs with fewer resources to utilize the resource of other BSs. Authors in Khatibi and Carreia (2015) presented a virtual radio resource management model in a heterogeneous RAN to support shortage of radio resources. The proposed model allocates resources to MVNOs based on service-level agreements (SLAs) and priority. Jiang et al. (2017b) proposed an auction approach to slice the physical network in 5G wireless networks. The proposed approach tends to optimize network resources and increases network revenue. Authors in Jiang et al. (2016) proposed network slicing management and prioritization in 5G mobile system. *Mahindra* et al. designed and implemented *NetShare*, a framework for network-wide radio resource management for effective radio access network (RAN) sharing in Mahindra et al. (2013). *NetShare* provides isolation across entities and efficiently distributes resources for each entity across the network. An efficient resource allocation scheme was proposed in wireless networks that solves resource allocation problem to maximize the utility of MVNOs using alternating direction method of multipliers (ADMM) (Liang and Yu, 2015). The economic aspect of wireless network virtualization was studied by authors in Habiba and Hossain (2018). In their work, they emphasized on auction theory as a basic tool to design business model of virtualization in heterogeneous and multi-commodity scenarios. The authors in Zhang et al. (2017) proposed a software defined and virtualized network for multi-flow transmission considering multiple InPs and MVNOs. They then formulated an optimization problem to maximize the total utility of the system. In Chang et al. (2017), an energy efficient optimization for wireless network virtualization comprising a BS with multiple antennas was studied. A joint power, antenna allocation and subcarrier problems were formulated to achieve energy efficiency. Although these works investigated network virtualization and resource allocation, most of them solved the problem using traditional solutions. In a real-time wireless scenario where there is dynamic traffic, traditional solutions may fail to provide the best solution to the problem.

Some models only estimated the QoE requirements for applications which reflected user QoS satisfaction due to the different characteristics of applications. In the 5G mobile networks, more advanced scenarios require transfer of high-rate mission-critical traffic flows. With this in mind, *Vitaly Petrov* et al. proposed a softwarized architecture for E2E reliability in 5G wireless networks (Petrov et al., 2018). Network softwarization enables flexibility in system operation under challenging scenarios. *Aijaz* proposed Hap-SliceR, an application-specific radio resource slicing framework for 5G networks with haptic communications in Aijaz (2018). Hap-SliceR derives a network-wide radio resource slicing strategy for 5G networks and provides customization of radio resources for haptic communications over 5G networks. *Wang* et al. introduced user association and resource allocation scheme to guarantee the QoE of mixed traffics in heterogeneous networks (Wang et al., 2016a). The user association and resource allocation problem was formulated as a mixed integer non-linear programming problem and solved by a genetic algorithm. Authors in Han and Ansari (2015) presented a novel scheme of delay-aware user association, which considers the delays in both RAN and backhaul. Individual BSs decide on which users to allocate resources on both content and data rate requirements, given backhaul capacity and interference constraints. All these works have a limitation of considering the versatile QoS requirements and characteristics of heterogeneous traffics. In a network where there should be a balance between multiple objectives of multiple slices, the above literatures cannot suffice.

From the related work presented above, authors have successfully investigated network resource virtualization, joint association control and resource allocation in virtualized networks. However, few literatures considered heterogeneous traffics in their submissions. In this work, our novel ideas include the coarse slice resource provisioning, the slice-level autonomous resource refinement via DRL technique to balance resource utilization and QoS satisfaction and the user-level shape-based intra-slice resource allocation scheme.

3. System model

The system model constitutes five entities in the network i.e. *Users, Service, Controller, BSs and Resource. Users* send admission control signals to the BS. Each slice provides a specific *service* and is managed by an MVNO also known as SP. The *Controller* is in charge of system monitoring and making decisions on appropriate admission control, user association and effective slice resource allocation. The *BSs* schedule users and allocate resources to them at BS-level. *Resources* in the network are radio resources, meanwhile cache and computation resources can be considered in other scenarios which are out of scope in this work. Other than macro-BSs, sc-BSs can use digital subscriber lines, fiber lines or wireless as backhaul in practical systems (Liang and Yu, 2015). However, most backhauls do not have specific constraints. The InP owns and leases physical infrastructure and resources, e.g. radio resource and BSs to the MVNOs. MVNOs provide services to users with the slice resource they acquired from InPs. MVNOs care about the QoS satisfaction of users while InP targets on minimizing resource embedding cost and efficient resource utilization.

The proposed resource slicing and customization framework as illustrated in Fig. 1 functions as follows: Initially, resources are provisioned for slices based on preference and weight rankings of slices at each BS by coarse resource slicing scheme. A portion of a slice's resource is partitioned to its users and the remaining resource is reserved to cater for an increase in the number of users in a slice. Then, the resource allocated to the slice is adjusted dynamically by autonomous slicing refinement at slice level. At BS level, the reserved resource on each BS is updated to reflect the slice-level resource adjustment. If the user population of a slice increases, the reserved resource is adjusted to ensure that performance isolation is achieved. Next, users are re-associated with BSs based on the resource allocation weight of a slice to BS. Lastly, shape-based physical resource allocation scheme customizes the amount of resource for users to improve satisfaction and resource utilization. We classify the system model into the following: virtualization, network and utility models. For ease of comprehension, we provide a table of notations in Table 1.

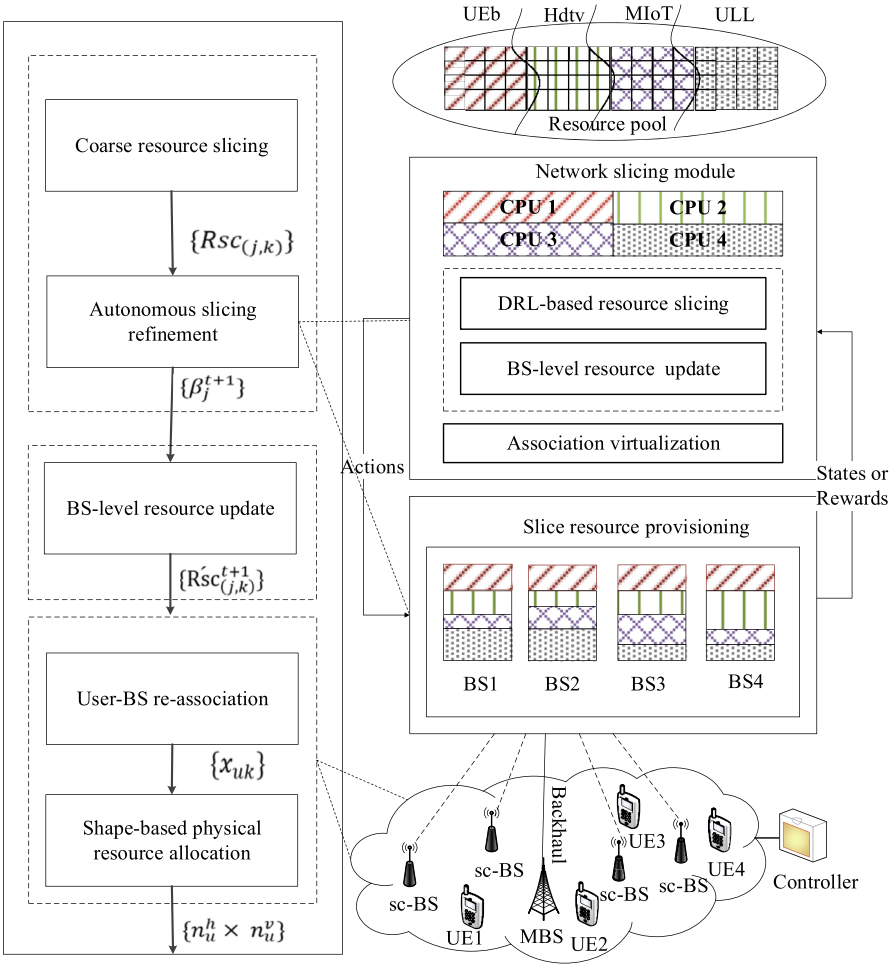

Fig. 1. The proposed resource slicing and customization framework.

3.1. Virtualization model

In virtualization, the physical resources are converted into virtual resource and partitioned into slices by network function virtualization (NFV). User association is also virtualized to be slice specific in this manner. It is assumed that the controller is made up of virtual machines (VMs) which are also known as CPUs. Each slice is connected to one VM or CPU in the network and each user is associated with only one VM since each user belongs to only one slice. A small-cell BS is expected to allocate physical RBs to a limited number of users in the network. The user association module (association manager) is responsible for assigning users in each slice to the cluster of BSs which is based on traffic distribution and QoS requirements of the slice. In case the logical resources allocated to a slice at the BS is not enough to guarantee the satisfaction of all the users associated with the BS, some users are re-associated with the next preferred BS that has enough resources to meet the QoS demands of the users. This reduces the rate of request rejection in the system. However, the users in the former slice whose demands are not met only utilize the remaining resources of the same slice at another BS which has enough resources.

For instance, if the RBs allocated to Slice 1 on BS 1 are not enough to serve some users belonging to the slice, these users of Slice 1 on BS 1 are re-associated with Slice 1 of BS 2 to be provided RBs allocated to Slice 1 on BS 2. The resource of the system is partitioned into four slices based on sum of user demand of the slice and the traffic distribution and the slices serve their users with the resource allocated to them.

The virtualized system bandwidth is shared amongst the slices each of which is managed by an MVNO. The BS that can satisfy the QoS demands of a user grants resources to the user based on the channel status and resource available to the BS. We assume an MVNO configures its *substrate network (SN)* with the resources it receives from the InP. A slice classifier groups the requested services and directs them to various slice queues to be allocated resources. The slicing controller then decides the resource allocation taking into consideration the associated preferences, the resource availability and the requested QoS demand. The scheme associates users of a slice to a specific BS that has sufficient resource to satisfy the demands of the users in a period assuming the slice availability is updated periodically. The scheme also allocates extra resources to the slices to solve the problem of isolation among the MVNOs with the least embedding cost, effective resource sharing among the slices and avoids fragmentation of resource at the BS.

Based on SLAs between the InP and the MVNOs, resources are allocated to slices by observing the states of the slices. A slice is given a weight based on the SLA between the InP and the MVNO. It is assumed that the controller in the RAN has an overview of the slices and other information such as the SLA and the flow belonging to each slice. From the QoS class identifier (QCI) index table (Mamman et al., 2019), the QoS class of each flow is known and the SLA enforced. For example, if the SLA of a slice is achieving low latency and high reliability such as virtual reality, the service data adaptation protocol (SDAP) sublayer maps any flow of this slice to a corresponding QoS class of index 69. The SDAP layer maps the QoS flow to its corresponding dedicated radio bearer (DRB) and the radio resource controller (RRC) signaling configures the SDAP layer. It should be noted that one QoS flow can be mapped to only one DRB but one or more QoS flows of the same slice can be mapped to the same DRB. However, the QoS flows of the same slice cannot be mapped to different

Table 1
Table of notations.

System parameters		Algorithm parameters	
Symbol	Meaning	Symbol	Meaning
K	A set of BSs	s	state
$\mathscr{J}$	A set of slices	a	Action
$\mathscr{U}$	A set of users	r	Reward
R_u	Requested transmit rate	s'	Next state
Td_u	Affordable time delay	η	Importance of utilization in reward
x_{uk}	Association indicator between user u and BS k	t	Time step
B	System bandwidth	ε	Epsilon
P_t	Transmit power	$V_\pi(s)$	State-value function for a policy π
d	Distance (in meters)	γ	Discount factor
f	Frequency (in MHz)	$A_\pi(s)$	Advantage function
$\mathfrak{g}$	Channel gain	$\varnothing$	Common network parameters
n_u^h	Width of LSRU	∂	Advantage stream parameters
n_u^v	Height of LSRU	ϑ	Stream parameters
Γ_{uk}	Data rate of user (u) on BS (k)	$\mathbb{H}$	Number of hidden layers
λ	Packet arriving rate	$\mathbb{N}$	Number of neurons
L	Length of packet	$W\_rat_{(j,k)}$	Weight-based ratio of slice (j) on BS (k)
α_j	Satisfaction of slice (j)	$rnk\_rat_{(j,k)}$	Rank-based ratio of slice (j) on BS (k)
θ_j	Resource utilization of slice (j)	$I_{(j,k)}^{BS}$	Weight of slice (j) on BS (k)
β_j	Fraction of resource allocated to slice (j)	$I_{(j,k)}^{slice}$	Weight of BS (k) on slice (j)
$Rsc_{(j,k)}$	Resource required by slice (j) at BS (k)	ϖ	Slice's weight
$Rsv_{(j,k)}$	Reserved resource of slice (j) at BS (k)	$\mathfrak{r}$	Slice level BS rank

DRBs. The enforcement of the SLAs solves any conflict between slices since the resource allocation is done in a dynamic manner (3GPP TS 38.300, 2018). We check the performance of the resource slicing scheme using QoS satisfaction and resource utilization with the number of user requests which vary with time. Resource allocation is done in such a way that the performance of one slice does not affect the performance of the other slices especially when there is change in user population in one slice. Hence, slice performance isolation is achieved in the network (Kasgari and Saad, 2018). The basic virtualized resource in the RAN can be a fraction of bandwidth and time slots in various systems (Richart et al., 2016). In this paper, we consider the granular virtualized resource to be physical RBs which are isolated and allocated to slices on the BSs. At the BSs, resources are also sliced and allocated to slices in order to provide isolation at BS level. The physical RBs should be mapped to the virtualized resources efficiently to clearly present the connection between the two. In this light, we map a fraction of the system bandwidth to the physical RBs in our work. The LSRU group of RBs are scheduled and allocated to a slice which is referred to as intra-slice shaped based physical resource allocation.

3.2. Network model

Given a set of BSs $k \in \mathscr{K} = \{1, 2...K\}$ and a set of slices $j \in \mathscr{J} = \{1, 2...J\}$, a slice (j) is made up of a set of users, $u \in \mathscr{U}_j = \{1, 2... U_j\}$. Due to the strategy of the resource provisioning scheme, each slice is assigned a priority. The request of user (u) with respect to QoS is denoted as $Rqst\ (R_u, Td_u)$, where R_u denotes the transmit rate requested by *user* (u) and Td_u represents the affordable time delay in terms of the inter-departure time of consecutive packets. The sum of transmit rate R_j requested by *user* (u) in slice (j) is $R_j = \sum_{u=1}^{U_j} R_u$.

Physical radio resource in this paper is of bandwidth B which consists of M RBs such that, an RB has a bandwidth B_m in the frequency domain. If we divide the time domain into scheduling frames where a frame consists of T sub-frames, the length of one sub-frame in the time domain is defined as t_l. In this case, the length of one scheduling frame is expressed as $T \times t_l$. Then, the fundamental unit of resource is one RB per sub-frame (RB_m^t). The total power transmitted by a BS (k) is $p_{uk}^{(t,m)}$ and the power allocated to *user* (u) from that BS is $p_{uk}^{(t,m)}$ considering RB $m \in \mathscr{M} = \{1, 2...M\}$ at time $t \in \mathscr{T} = \{1, 2...T\}$.

We consider a moderate load scenario where the difference between the number of users and the number of RBs is not very huge. In a specific scheduling frame, a flow may be allocated RBs based on its QoS requirements. Since RBs are defined in time and frequency domains, a very effective resource allocation scheme is able to schedule RBs to users dynamically in each scheduling frame. At a specific scheduling frame, a number of users occupy the RBs and transmit data. At another frame, the number of users occupying the RBs may change due to different scheduling at the current time slot. For ULL flows, only one RB may be enough to transmit data. However, Hdtv flows may need several RBs to transmit data. Delay-sensitive flows are allocated RBs to transmit data quickly whiles delay-elastic flows are delayed and later allocated RBs when available since they are considered as best-effort flows. This balances the satisfaction and resource utilization of the system.

We denote the channel gain for user (u) on RB m on BS k at time t as $\mathfrak{g}_{uk}^{(t,m)}$ assuming that the BS has an overview of the channel gain information of all users of SPs. Since there are multiple BSs, the co-tier and cross-tier interferences between the sc-BSs and users cannot be overlooked. To reduce strong interference of other users, interference avoidance, cancelation and randomization techniques are applied (Saquib et al., 2012; Song et al., 2014). With N_o as the noise spectral density, u' as a neighboring user and k' as a neighboring BS, the data rate associated with each RB_m^t for a given *user* (u) is defined as

$$RB_{uk}^{(t,m)} = \frac{B_m}{T} \log_2\left(1 + \frac{p_{uk}^{(t,m)} \mathfrak{g}_{uk}^{(t,m)}}{N_o B_m + p_{u'k'}^{(t,m)} \mathfrak{g}_{u'k'}^{(t,m)}}\right), \tag{1}$$

where u' and k' are the neighboring user and BS respectively.

To reduce transmission delay of users, appropriate resource allocation must be ensured by associating the BS with sufficient resource to the user and also at a channel, assigning appropriate sub-frames to the user. To ensure that the affordable delay of *user* (u) in slice (j) is not exceeded, we expect that the inter-departure time of consecutive packets for *user* (u) must be less than Td_u to guarantee smoothness of traffic flows with a common shape-based resource allocation algorithm for heterogeneous traffics. The minimum number of time slots required to satisfy its delay demand is denoted by n_u^h, and the number of RB_m^t needed by *user* (u) in a single time slot is denoted by n_u^v. We express n_u^h and n_u^v in equations (2) and (3) as follows:

$$n_u^h = \left\lceil \frac{T \times t_l}{Td_u} \right\rceil, \tag{2}$$

$$n_u^v = \left\lceil \frac{R_u}{n_u^h \times RB_{uk}^{(t,m)}} \right\rceil, \tag{3}$$

where R_u represents the data rate requested by the user within a single scheduling time frame T. A user is allocated a sub-channel of width n_u^h and height n_u^v which is defined as an LSRU. The achievable data rate Γ_{uk} that can be achieved by *user* (u) from BS (k) is computed as,

$$\Gamma_{uk} = n_u^h \times n_u^v \times RB_{uk}^{(t,m)}. \tag{4}$$

We make the following assumptions in an attempt to set up a delay model for heterogeneous traffics: (1) With λ_u as the packet arriving rate for user (u), the inter-arrival times between consecutive packets of a user (u) are independent of each other and exponentially distributed with a

mean of $1/\lambda_u$ seconds, (2) The packet lengths for slices are independent but that of a user (u), are with mean L_u bits. Under these assumptions, we formulate the delay model. The service time for traffic of user (u) is exponentially distributed with mean $t_{uk} = L_u/\ \Gamma_{uk}\ = 1/\gamma_{uk}$. With respect to the average packet length, γ_{uk} denotes the normalized rate achievable. It is obvious that an M/M/1 queue is preferred for the traffic of a slice to a user (u). Hence, we calculate the average delay of a packet in the queue from BS (k) to user (u) as

$$\tau_{uk} = \frac{1}{1/t_{uk} - \lambda_u} = \frac{1}{\gamma_{uk} - \lambda_u}. \tag{5}$$

3.3. Utility model

The satisfaction of a user is defined as the ratio of users allocated resources in a slice to the total number of users in the slice during scheduling in a specific period. The satisfaction on rate or delay of user (u) in slice (j) is denoted as Sat_u. The slice satisfaction α_j on all the BSs which measures the utility metric can be computed as

$$\alpha_j = \frac{\sum_{u=1}^{U_j} Sat_u}{U_j}, \tag{6}$$

where U_j represents the total number of users for *slice*(j) on all of BSs. We consider measuring the satisfaction of a user (u) with a sigmoid function (Sheng et al., 2014). The satisfaction metric with respect to rate and delay of user (u) is expressed in equations (7) and (8) respectively as:

$$Sat_u^{rate} = \frac{1}{1 + e^{-\phi(\Gamma_u - R_u)}}, \tag{7}$$

$$Sat_u^{delay} = \frac{1}{1 + e^{-\phi(\tau_u - Td_u)}}, \tag{8}$$

where R_u represents the rate requirement of user (u) and Γ_u is the data rate expected to be achieved by user (u). The achieved delay and the delay requirement of user (u) are represented by τ_u and Td_u respectively. ϕ is a constant deciding the steepness of the satisfactory curve. Γ_u is determined by the amount of allocated resources, transmission power, noise, interference and many other related factors. It is easy to verify that: 1) Sat_u^{rate} and Sat_u^{delay} are monotonic increasing functions with respect to Γ_u and τ_u respectively, because individual users will feel more satisfied if they receive higher throughput or lower delay above their minimum demand or below their maximum threshold and vice versa; 2) Sat_u^{rate} or Sat_u^{delay} of each user (u) is scaled between 0 and 1, i.e. Sat_u^{rate} or $Sat_u^{delay} \in (0,\ 1)$. The level of satisfaction can be used to determine whether a user is satisfied or not. For example, if the level of satisfaction is greater than or equal to 0.5, the user is said to be satisfied and vice versa. Similar to (Tsiropoulou et al., 2015; Tsiropoulou et al., 2013), the achievable data rate in Eqn. (4) and average delay in Eqn. (5) are functions of the SINR. Users' QoS characteristics are justified in a similar manner as shown in the above-mentioned literatures, which is commonly used in the literature of network utility maximization. The satisfaction at system level (α) is also expressed as $\alpha = \frac{1}{J} \sum_{j=1}^{J} \alpha_j$. The slice-level resource utilization (θ_j) is defined as follows:

$$\theta_j = \frac{\Omega_j}{\prod_j}. \tag{9}$$

where Ω_j is the number of RBs occupied by slice (j) and $\prod_j$ is the number of RBs allocated to slice (j). Finally, we express the system's resource utilization (θ) as

$$\theta = \frac{\Sigma_{j=1}^{J} \Omega_j}{\Sigma_{j=1}^{J} \prod_j}. \tag{10}$$

4. DRL-based resource slicing

4.1. Coarse slice resource provisioning

After the slice classifier groups user requests into slice queues, each user (u) finds a potential list of BSs that can satisfy its QoS demands to associate with. Each of the BSs is ranked based on the average data rate achievable per one RB, $RB_{uk}^{(t,m)}$. The rank of a specific BS (k) by user (u) is represented as δ_{uk}. In this paper, a logical resource is referred to as the amount of resource assigned by a BS to a slice. The total number of users in a slice and the slice preference determines the logical resource to be assigned to a slice. The sum of BS ranks at slice level (r_{jk}) is aggregated as the average of ranks of BS (k) by users of a slice (j) as

$$\bar{\mathrm{r}}_{jk} = \frac{\sum_{u=1}^{U_j} \delta_{uk}}{U_j},\ \forall\ u \in \mathcal{U}_j = \{1, 2 \ldots\ U_j\}. \tag{11}$$

The rank based ratio $rnk\_rat_{(j,k)}$ for a slice at BS (k) is expressed as

$$rnk\_rat_{(j,k)} = \frac{\mathrm{r}_{jk}}{\sum_{j=1}^{J} \mathrm{r}_{jk}}, \forall\ j \in \mathcal{J} = \{1,\ 2 \ldots J\}, \tag{12}$$

and the weight of a slice, ϖ_{jk} is defined as the sum of the *transmit rate requests* per user R_u at BS (k) which is expressed as

$$\varpi_{jk} = \sum_{u=1}^{\mathrm{U}_{jk}} R_u, \forall u \in \mathcal{U}_{jk} = \{1, 2 \ldots U_{jk}\}. \tag{13}$$

where U_{jk} denotes the total user population of slice (j) on BS (k). The slices have versatile weights because of the difference in traffic arriving rates, changes in channel quality and traffic distribution of users in the slices on the BSs. The weight of a slice is dependent on its ratio with the weights of the other slices on the same BS. The ratio of the weight of the slice to the weight of all the slices on a BS is $(W\_rat_{(j,k)})$.

We formulate $W\_rat_{(j,k)}$ as

$$W\_rat_{(j,k)} = \frac{\varpi_{jk}}{\sum_{j=1}^{J} \varpi_{jk}}, \forall j \in \mathcal{J} = \{1, 2 \ldots J\}, \tag{14}$$

From equations (12) and (14), we can derive the overall resource virtualization of our resource provisioning algorithm. The weight $I_{(j,k)}^{BS}$ of slice (j) to BS (k) is computed as follows;

$$I_{(j,k)}^{BS} = \frac{\left(rnk\_rat_{(j,k)} \times W\_rat_{(j,k)}\right)}{\sum_{j=1}^{J} \left(rnk\_rat_{(j,k)} \times W\_rat_{(j,k)}\right)} \tag{15}$$

A specific slice (j) at BS (k) requires an amount of resource $Rsc_{(j,k)}$ and is expressed as

$$Rsc_{(j,k)} = I_{(j,k)}^{BS} \times \text{Ł}_k,\ \forall\ j \in \mathcal{J} = \{1, 2 \ldots J\}, \tag{16}$$

where Ł_k denotes the overall resource of BS (k), and $Rsc_{(j,k)}$ is the fraction of resource belonging to slice (j) at BS (k).

4.2. MDP formulation of slicing refinement

The slice resource provisioning problem can be modeled as a Markov decision process (MDP) and a solution is formulated for the problem. The slicing problem for heterogeneous traffics can be solved using dynamic programming or model-free RL technique. Dynamic programming requires a perfect model and suffers from high computational complexity. In our scenario, we deal with a large and complicated state and action spaces which is costly to be solved by dynamic programming. Instead, we resort to two forms of RL; DQN and dueling DQN to formulate a solution for the slice resource provisioning problem. DQN is a combination of

artificial neural networks (ANN) and Q-learning. It is a good choice for the complicated machine learning problems where the size of state space is very large. The interaction between the learning agent and the RAN environment is represented as a tuple, $(s,a,r,s^{'})$, where s represents the set of possible states, a is the set of actions, r represents the reward achieved when an action a, in state s, is selected and $s^{'}$ is the next state. We begin to define the constituents of the MDP below.

State(s): Based on the present state of a slice, the agent selects an action to be enforced on the available resource of the slice. The states of a slice can be obtained from the following metrics: the QoS utility of slice (j), α_j, also known as the satisfaction ratio, the resource fraction allocated to slice β_j and the resource utilization of slice (j), θ_j. These metrics can be represented as a tuple $s = \{\alpha_j,\ \beta_j,\ \theta_j\}$. As defined in equations (6) and (9), α_j is the ratio of the total satisfaction of users in slice (j) to the total user population of slice (j) on all the BSs and θ_j is the ratio of used resources of slice (j) to the total amount of resource allocated to slice (j) respectively.

Action(a): According to the states of a slice, the learning agent selects the best action a^t of resource provisioning for each slice (j) based on the current state s^t. The possible actions to be selected for one slice comes from a set of percentages $a \leftarrow$ {-0.6, −0.4, −0.2, 0, 0.2, 0.4, 0.6}. The learning agent selects an action that is used to update the slice resource at BS level. We define the weight $I^{slice}_{(j,k)}$ of BS (k) to slice (j) as

$$I^{slice}_{(j,\,k)} = \frac{\left(rnk\_rat_{(j,k)} \times W\_rat_{(j,k)}\right)}{\sum_{k=1}^{K}\left(rnk\_rat_{(j,k)} \times W\_rat_{(j,k)}\right)} \tag{17}$$

The mapping from the overall resource allocation to the BS-level slice resource depends on the weights of the BSs for a specific slice.

Reward(r): The reward of the learning agent considers some metrics that affect the resource provisioning of the slice and a summation of these metrics can be taken as the reward function of the learning agent. DRL is concerned with how to interact with the environment by selecting enough possible actions and reinforcing the better ones. We define the reward function r^t of the agent as the summation of the satisfaction ratio α_j and the resource utilization θ_j of all the slices in the network which is expressed mathematically as

$$r(s,a) = \ min\left\{\alpha_j(s,a) + \eta\ .\theta_j(s,a)\right\}, \tag{18}$$

where η is a constant which indicates the importance of slice resource utilization.

Next state($s^{'}$): The next state tuple comprises the new satisfaction of the slice, the fraction of resource allocated to the slice and the resource utilization of the slice after an action is enforced by the learning agent. The next state parameters are affected by the action taken on the slice.

Given $\mathcal{G}_{\pi}([s^0, s^1, \ldots s^t],\ a^t)$ as a Markov chain utility, the future reward of state s^t at time step t expected on the long term is determined by the sum of future discounted rewards and can be expressed as

$$r(s^t) + \gamma.r\left(s^{t+1}\right) + \gamma^2.r\left(s^{t+2}\right) + \ldots, \tag{19}$$

where γ denotes the discount factor which is a value from 0 to 1. A discount factor of 0 means we only care about the immediate reward whiles a discount factor approaching the value of 1 means we care about future rewards.

The state-value function of a generic strategy at the time step t is denoted as

$$V_{\pi}(s) = \mathbb{E}\left\{\sum_{t=0}^{\infty}\gamma^t r(s^t, a^t)\middle| s^0 = s\right\} \tag{20}$$

The objective of MDP is to achieve the optimal strategy π^* in order to maximize the future discounted rewards. The strategy of a Markov chain utility can be expressed as

$$V_{\pi}(s) = \mathbb{E}\left\{r(s^t,\ a^t) + \sum_{s^{'}}P\left(s^{'}|s^t, a^t\right)V_{\pi}\left(s^{'}\right)\right\}, \tag{21}$$

where $V_{\pi}(s^{'})$ represents the expected utility given the best strategy. Based on the famous Bellman equation (Sutton and Barto, 1998), the state-value function for the best strategy can be expressed as

$$V_{\pi^*}(s) = \arg\max_{a^t}\left\{\begin{array}{c} r(s^t,\ a^t)+ \\ \gamma\sum_{s^{'}}P\left(s^{'}|s^t,\ a^t\right)V_{\pi}\left(s^{'}\right)\end{array}\right\}, \tag{22}$$

where $r(s^t,\ a^t)$ is the present reward, $V_{\pi}(s)$ is the present utility and $V_{\pi}(s^{'})$ is the future utility.

4.3. Dueling deep Q-Network

Dueling DQN is a form of model-free RL technique where the Q-function is made up of two estimator functions: state-value estimator function and state-dependent action advantage estimator function (Wang et al., 2016b). This form of RL technique is similar to the original DQN algorithm with the main difference being that, the original DQN algorithm has a single Q-network with one stream Q-function whiles the dueling DQN algorithm has a single Q-network with a two-stream Q-function, i.e. the state-value function and the advantage function. With the original DQN algorithm, we need to calculate the value of each action in a particular state. The question is, what if the choice of some actions have no effect on the learning environment? This will only make the algorithm slow at identifying the best action at a state. It is then unnecessary to calculate the value of each possible action at every state. The main advantage of dueling DQN is that it can generalize the learning process for all possible actions in the environment without causing a change to the underlying RL algorithm. The algorithm can also quickly identify the best action because it has the ability to learn which states are important to the learning agent without learning the effect of each action for each state.

In dueling DQN based algorithm, the goal of the learning agent is to find the optimal strategy π^* that maximizes the expected discounted reward. According to the strategy π, the state-action pair value $\boldsymbol{Q}_{\pi}(s,a)$ and the state value $V_{\pi}(s)$ at time step t can be expressed as

$$\boldsymbol{Q}_{\pi}(s,a) = \mathbb{E}\{r^t|s^t = s,\ a^t = a,\ \pi\}, \tag{23a}$$

$$V_{\pi}(s) = \mathbb{E}_{a\sim\pi(s)}[\boldsymbol{Q}_{\pi}(s,a)] \tag{23b}$$

From equation (23), we define the advantage function as

$$A_{\pi}(s) = \boldsymbol{Q}_{\pi}(s,a) - \mathrm{V}_{\pi}(s), \tag{24}$$

where $\mathrm{V}_{\pi}(s)$ measures how good it is to be in a specific state s and $\boldsymbol{Q}_{\pi}(s,a)$ measures the value of choosing a specific action a in state s. It should be noted that, $\mathbb{E}_{a\sim\pi(s)}[A_{\pi}(s,a)] = 0$. For deterministic policy $a^* = \arg\max_{a^{'}}\boldsymbol{Q}(s, a)$, it follows that $\boldsymbol{Q}(s,a^*) = V(s)$, hence $A(s,a^*) = 0$.

The output of the dueling network is given as;

$$\boldsymbol{Q}(s,a;\ \varnothing,\ \partial,\vartheta) = V(s;\ \varnothing,\vartheta) + A(s,a;\ \varnothing,\ \partial), \tag{25}$$

where $\varnothing$ represents common network parameters, ∂ represents advantage stream parameters and ϑ represents value stream parameters. However, $\boldsymbol{Q}(s,a;\varnothing,\partial,\vartheta)$ can be unidentifiable because it is only a parameterized estimate of the true Q-function and results in poor performance. To solve the issue of identifiability, we generate the Q-values for each action a at state s using the aggregation layer as follows:

$$\begin{aligned}\boldsymbol{Q}(s,a;\varnothing,\ \partial,\vartheta) = \ & V(s;\varnothing,\vartheta) + (A(s,a;\ \varnothing,\ \partial) \\ & -\frac{1}{|\mathrm{A}|}\sum_{a^{t+1}}A\left(s^t, a^{t+1};\varnothing,\ \partial\right)\Big)\end{aligned} \tag{26}$$

Algorithm 1 shows the operation mode of the dueling DQN algorithm. The process will be repeated until termination. The computational complexity of Algorithm 1 is of $O(\mathbb{H}^{\vartheta}\mathbb{N}_{\rho})$, where $\mathbb{H}^{\vartheta}$ denotes the number of hidden layers and $\mathbb{N}_{\rho}$ denotes the number of neurons in the ANN. The optimality of DRL has been discussed in Watkins (1989) which is applicable to our proposed algorithm. Both the algorithmic and statistical rates of convergence imply that the action-value function converges to the optimal counterpart up to an unimprovable statistical error in geometric rate (Yang et al., 2019).

Algorithm 1
Dueling Deep Q-Network Based Resource Slicing.

1. *Initialization*: A replay memory, action-value $\boldsymbol{Q}$ with random weights ∂ and ϑ, epsilon ε and Q-table
2. Observe state $s = \{\alpha_j, \beta_j, \theta_j\}$
3. **For** each time step t **do**
4. Select a random action a^t with probability ε, otherwise
Select $a = \arg\max_{a^t} \boldsymbol{Q}(s^t, a^t; \partial, \vartheta)$
5. Update the resource fraction β_j by (27) and observe reward r^t by (18) and new state s^{t+1}
6. Update the required allocation $Rsc^t_{(j,k)}$ by (29) or reserved Rsv$^{t+1}_{(j,k)}$ by (31) on all BSs
7. Store (s^t, a^t, r^t, s^{t+1}) in the replay memory
8. Sample mini-batch of transitions (s^j, a^j, r^j, s^{j+1}) from the replay memory
9. Combine $A_{\pi}(s,a)$ and $V_{\pi}(s)$ using Eqn. (26)
10. Set $y^j = r^j + \gamma . \max_{a'} \boldsymbol{Q}(s^{j+1}, a^{j+1}; \partial^-, \vartheta^-)$
11. Perform a gradient descent step on $(y^j - \boldsymbol{Q}(s^j, a^j; \partial, \vartheta))^2$
12. Every C steps reset $\widehat{\boldsymbol{Q}} = \boldsymbol{Q}$
13. **Until** terminated

5. BS-level resource update and customization

5.1. BS-level resource update

We update the resource of the slice based on the action selected by the agent. Let β_j^t represent the resource provisioned to a slice at slicing time t and β_j^{t+1} represent the overall resource update at slicing time $t+1$. We update the resource at slicing time $t+1$ as:

$$\beta_j^{t+1} = \begin{cases} \beta_j^t, & if\ a_j = 0 \\ (1 + a_j)\beta_j^t, & otherwise \end{cases}, \tag{27}$$

where a_j denotes the action selected by the agent to be enforced on the resource of slice (j).

After the resource is updated at the slice level, the update has to reflect at all the BSs. Hence, the resource update at slice level is converted back to BS update. We consider two types of resource; allocated resource and reserved resource. Allocated resource is the fraction of the overall resource provisioned to a slice that is partitioned to the users in the slice. Reserved resource is the remaining fraction of the provisioned resource of the slice that is only allocated back to the slices to provide resource isolation when there is increase in user population of the slice. If we define $Rsc^t_{(j,k)}$ as the resource allocated to slice (j) at BS (k) at slicing time t, the updated allocated resource of slice (j) at BS (k) at slicing time $t+1$ is given by $Rsc^{t+1}_{(j,k)}$. With the importance of the BSs to a specific slice (j) as $I^{slice}_{(j,k)}$, we calculate the fraction of resource of the slice (j) at BS (k) by multiplying the weight $I^{slice}_{(j,k)}$ by the resource update of the slice β_j^{t+1}, but this value is not normalized. We update the BS level resource at slicing time $t+1$ as:

$$\text{Rsc}^{t+1}_{(j,k)} = \beta_j^{t+1} . K . I^{slice}_{(j,k)}, \tag{28}$$

If the values of β_j^{t+1}, K and $I^{slice}_{(j,k)}$ are known, we can calculate the resource provisioning at BS level at slicing time $t+1$. Finally, resource of the slice to the BSs is reconfigured and the slice allocates the resource to its users.

The original DQN and dueling DQN algorithms operate as follows; The necessary resource is allocated to the slice and the unused resource is kept as reserved resource. The allocated resource fraction of slice (j) at BS (k) is expressed as:

$$\acute{\text{Rsc}}^{t+1}_{(j,k)} = \begin{cases} \text{Rsc}^{t+1}_{(j,k)}, & if \sum_{j=1}^{J} \text{Rsc}^{t+1}_{(j,k)} < 1 \\ \dfrac{I^{BS}_{(j,k)} \text{Rsc}^{t+1}_{(j,k)}}{\sum_{j=1}^{J} I^{BS}_{(j,k)} \text{Rsc}^{t+1}_{(j,k)}}, & otherwise \end{cases} \tag{29}$$

The reserved resource fraction of slice (j) is calculated as:

$$\overline{Rsv}_j^{t+1} = \begin{cases} 0, & if \sum_{j=1}^{J} \beta_j^{t+1} \geq 1 \\ \dfrac{\left(1 - \sum_{j=1}^{J} \beta_j^{t+1}\right) . \beta_j^{t+1}}{\sum_{j=1}^{J} \beta_j^{t+1}}, & otherwise \end{cases} \tag{30}$$

The reserved resource fraction of slice (j) at BS (k) is:

$$\text{Rsv}^{t+1}_{(j,k)} = \frac{I^{BS}_{(j,k)} \text{Rsc}^{t+1}_{(j,k)}}{\sum_{j=1}^{J} I^{BS}_{(j,k)} \text{Rsc}^{t+1}_{(j,k)}} \tag{31}$$

Here, the fraction of unused resource $\overline{Rsc}_j^{t+1} = 0$.

5.2. Shape based resource allocation and association control

User association and intra-slice resource allocation follow slice-level resource allocation. Here, the controller associates users with slices and the slices' resources at the BS are allocated to the users. Resources are referred to as LSRUs which are RBs shaped in time and frequency domains. To minimize resource fragmentation, the demand of users in a slice should be checked against the available resource of the slice at the specific BS (Han and Ansari, 2015). Intra-slice resource allocation ensures that each user's QoS satisfaction is achieved. For resource to be provisioned to a user on a BS, we consider the following conditions: (1) if and only if the data rate requirement of the user can be achieved by the available resource on the BS (2) if and only if the BS can provide the number of RBs needed by the user based on the available resource of the slice on the BS. The achievable data rate Γ_{uk} from BS (k) to user (u) at one frame period is calculated as:

$$\Gamma_{uk} = \sum_{(t,m)=(1,1)}^{(n_u^h, n_u^v)} \frac{B_m}{T} \log_2 \left(1 + \frac{p_{uk}^{(t,m)} \times g_{uk}^{(t,m)}}{N_o B_m + p_{u'k'}^{(t,m)} g_{u'k'}^{(t,m)}}\right),\ n_u^h \leq T. \tag{32}$$

Furthermore, the achievable data rate Γ_{jk} of slice (j) at a specific BS (k), is calculated as

$$\Gamma_{jk} = \sum_{u=1}^{U_j} \Gamma_{uk} . x_{uk}, \tag{33}$$

The nature of available RBs at BS (k) can be determined by denoting a BS's available radio resource by a Karnaugh-map-like table which constitutes rows of M sub-channels and columns of T sub-frames.

The total data rate Γ_k in air-interface for BS (k) is calculated as follows:

$$\Gamma_k = \sum_{j=1}^{J} \Gamma_{jk}. \tag{34}$$

On a given BS, τ_u represents the transmit rate of user (u) and Γ_u is the time delay of user (u). We formulate the shape-based resource allocation and association problem as an NP- complete 0–1 multiple knapsack problem as

$$min \quad \sum_{k=1}^{K} \sum_{j=1}^{J} \sum_{u=1}^{U_j} \left(n_u^h \times n_u^v\right).O_{uj}.x_{uk} \tag{35}$$

s.t.

$$x_{uk}(x_{uk}-1)=0 \; \forall k \in \mathscr{K}=\{1,\ 2\ldots K\},\ j \in \mathscr{J}=\{1,\ 2\ldots J\} \tag{35a}$$

$$\sum_{u=1}^{U_j} \left(n_u^h \times n_u^v\right).x_{uk} \le Rsc_{(j,k)}, \forall k \in \mathscr{K}=\{1,2\ldots K\}, \tag{35b}$$

$$\sum_{k=1}^{K} O_{uj}.x_{uk}=1, \forall u \in \mathscr{U}_j=\{1,2\ldots U_j\}, \tag{35c}$$

$$\tau_u \le Td_u, \forall u \in \mathscr{U}_j=\{1,2\ldots U_j\}, \tag{35d}$$

$$\Gamma_u \ge R_u, \forall u \in \mathscr{U}_j=\{1,2\ldots U_j\}, \tag{35e}$$

Constraint (35b) ensures that all users of slice (j) who are supposed to be allocated $Rsc_{(j,k)}$ or reserved $Rsv_{(j,k)}$ from BS (k) should be satisfied by the resource of slice (j) on BS (k). Constraint (35c) ensures that a user is allocated one channel from only one BS. We assume that a specific slice (j) can achieve its fraction of resource $Rsc_{(j,k)}$ on each BS (k). The association relationship between user (u) and slice (j) is represented by O_{uj}. Constraint (35d) and constraint (35e) ensure that the QoS constrained request $Rqst$ (R_u, Td_u) for user (u) is achieved. The computational complexity of the optimal solution is exponential (Johnson and Garey, 1990). To reduce the complexity, we propose a heuristic algorithm for the 0–1 multiple knapsack problem (35). The bottom-left bin-packing algorithm ensures that the LSRU of users are placed on RB image of each BS (Chazelle, 1983). Algorithm 2 is of computational complexity $O(J.K.U_{jk})$.

Algorithm 2
Shape-based User Resource Allocation and Association Control.

Initialization [K, J, $rsc_{(k)}$, ($Rsc_{(j,k)}$), Rqst$_u$ (R_u,Td_u)], $U_{(j)}$, δ_{uk}
For $j = 1$: J
  Waiting$_{(j,:)}$.Queue = $U_{(j,:)}$;
  Waiting$_{(j,:)}$.Queue = Sort (Waiting$_{(j,:)}$.Queue, Td$_u$ (asc));
  While Waiting$_{(j,:)}$.Queue. Empty () == false
    For $k = 1$: K
      If $u \cap U_{jk} == \phi$
        If $rsc_{(k)} < R_{(u)}$
        u.BS_Pref = next_pref (δ_{uk});
          u.Td$_u$.update;
          $U_{(j,\ u.BS\_Pref)}$.add (u);
        Else
          calculate n_u^h using equation (2);
          calculate n_u^v using equation (3);

(*continued on next column*)

Algorithm 2 (*continued*)

If ($n_u^h \times n_u^v$) < Rsc$_{(k,j)}$
  Assoc_mat (k).add (u);
  Place LSRU on BS via B-L B-P algorithm
  Update ($Rsc_{(k,j)}$) according to System status
    Update rsc$_{(k)}$ based on the remaining resource
  Else
    U$_{(j,\ u.BS_Pref)}$.add (u);
End
End
End
End
End

6. Performance evaluation

6.1. Scenario configuration and assumptions

In order to evaluate the performance of our proposed scheme, we consider a wireless mobile network scenario with four types of flows. We performed extensive simulations using Python software. We run the simulations on a 20-core i7 server with central processing unit (CPU) running on a processor speed of 2.4 GHz and 16 GHz RAM. We summarize most of our simulation parameters in Table 2. A given area of 700 m*700 m contains 4 BSs which are uniformly distributed with a coverage radius of 150 m. There is a fixed distance of 120 m between each two neighboring BSs. We define a path-loss (PL) model in terms of user-BS distance d (in meters) and channel frequency f (in MHz) as PL(dB) = 20 log10 (d) +20 log10 (f) – 27.55. We define 4 slices for this simulation based of 5G slicing categories as UEb, Hdtv, MIoT and ULL slices each of which has its own defined user population, packet arriving rates, packet sizes, maximum delay requirements and minimum rate requirements. We set the simulation time as 1000 s which is equal to 1 × 105 frames. We set the following values of parameters for two DQN algorithms, dueling DQN and (original) DQN: discount factor, learning rate and epsilon greedy value are 0.9, 0.01 and 0.07 respectively. We set a replay memory of size 8000 and a mini-batch of 32.

For the baseline approaches, we used Q-learning (Aijaz, 2018), NVS (Kokku et al., 2012) and NetShare (Mahindra et al., 2013) algorithms. The shortcomings of the baseline approaches are presented below:

6.1.1. Benchmark 1: Q-learning

In this paper, dynamic resource slicing and customization are done for a specific type of application; haptic communications, in 5G networks (Aijaz, 2018). The slicing policy is based on RL technique (Q-learning) which allocates resources to different slices with different requirements and seeks to achieve an optimal solution to the problem. The slicing policy initially provisions resource to the slices based on traffic demand estimation. However, resource slicing is done at RB-level which makes the state space very large and leads to curse of dimensionality. Since Q-learning cannot solve complicated machine learning problems, it is

Table 2
Simulation parameters.

Parameters	Settings	Parameters	Settings
The number of BS	4	Number of users per slice	[UEb:60, Hdtv:11, MIoT:240, ULL:124]
The number of slice	4	Packet arriving rate per user	[UEb:100,Hdtv:100,MIoT:100,ULL:100] (packet per second);
System Bandwidth	5 MHz	Minimum rate demand	[UEb:100,Hdtv:500, MIoT:12, ULL:10] (kbps)
Transmitting power	30 dBm	Maximum delay demand	[UEb:100,Hdtv:120,MIoT:105,ULL:10] (ms)
BS coverage	150 m	The packet size	[UEb:400,Hdtv:4000,MIoT:500,ULL:120](bits)
Inter-BS distance	120 m	Packet arriving time distribution	ULL: uniform, Others: exponential distribution
Channel gain	To be generated via path loss model	Learning rate	0.01
Sub-frame duration	1 ms	Discount factor	0.9
Frame length	10 ms	Epsilon/-greedy	0.07
Sub-channels per BS	25	Size of replay memory	8000
Noise power density	−174 dBm/Hz	Size of mini-batch	32
The number of users	Less than 435	Slicing period	200 ms
User distribution	Uniform	Episode duration	1400 ms

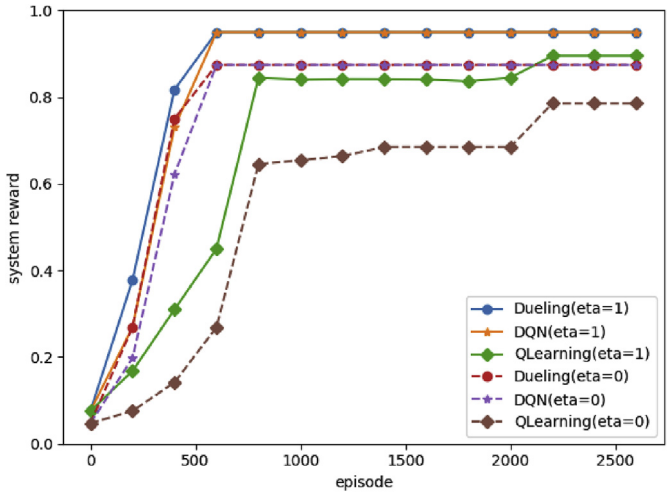

Fig. 2. Convergence analysis.

impracticable for the Q-table to converge and Hap-SliceR cannot find the optimal solution to the resource slicing problem for heterogeneous traffics.

6.1.2. Benchmark 2: network virtualization substrate (NVS)

This paper refers to the global view provisioning, as static slice resource provisioning, also known as NVS (Kokku et al., 2012). With this provisioning scheme, weight-based slicing is considered with the assumption that, the channel status of each user in the slice is known in prior. In this case, the resource is statistically provisioned i.e. no re-association is considered and for that matter, the resource provisioning is only based on the slices' weight in the network. After the *slice classifier* has classified users into slices, the slice's weight is computed as follows

$$\varpi_j = \sum_{u=1}^{U_j} R_u \ \ \forall u \ \in U_j, \tag{36}$$

where ϖ_j is the weight of a slice in the whole network, defined by the aggregated data rate demand of all its users and U_j is the number of users in the same slice (j). According to the static slice resource provisioning in Kokku et al. (2012), the fixed fraction W_rat_j that indicates the network share for this slice is defined as

$$W\_rat_j = \frac{\varpi_j}{\sum_{j=1}^{J} \varpi_j}, \quad \forall j \in \{1, 2, ..J\}. \tag{37}$$

The slice resource provisioning is calculated by W_rat_j to determine the resource allocation among BSs with equations (36) and (37).

6.1.3. Bechmark 3: NetShare

The resource fraction of the slices in NVS is calculated by the ratio of resource requirements of the slices in the initial slicing. However, there are at least two drawbacks for NVS. Firstly, such aggregate resource utilization of the slice across the network suffers from a static per-BS resource reservation. Secondly, NVS does not consider traffic classes of real-time and non-real-time. NetShare considers that resource fraction of slice (j) has a maximum and a minimum resource constraints on the system level. Furthermore, NetShare sets an upper bound and a lower bound of the BS-level resource allocation for each slice. They assume all resources of one BS are allocated to slices. With the principles of proportional fairness, dynamic resource allocation of slices is periodically determined in NetShare by maximizing a utility function of slice demand fraction scaled by resource allocation fraction at the BSs (Mahindra et al., 2013). The resource reserved in NetShare for a specific slice are dynamically distributed among all BSs.

6.2. Convergence analysis

In this simulation, the convergence of our proposed dueling DQN algorithm is evaluated and compared with original DQN (Sutton and Barto, 1998) and Q-learning (Aijaz, 2018) algorithms during 3000 episodes. Dueling DQN performs a faster action selection because it can learn which states are valuable without having to learn the effect of each iteration at each state. Due to this unique property, it is able to achieve faster convergence compared with traditional schemes and even original DQN/Q-learning. The result of this simulation is based on maximum value of every 50 episodes. The number of states in Q-Learning is 128. We consider two values of η, which is a constant that indicates the importance of the resource utilization on the reward function. Firstly, we set $\eta = 1$ and later change the value to 0. When the value of η is 1, the reward function is the sum of the slice satisfaction and resource utilization. A value of η as 0 defines the reward function in terms of slice satisfaction only. From Fig. 2, it can be observed that dueling DQN and original DQN algorithms with $\eta = 1$ converge after 500 episodes with the highest reward at 0.95. Q-learning algorithm under the same condition begins to converge around episode 2200 with a reward of 0.9. With $\eta = 0$, dueling DQN and original DQN achieve convergence at the same training stage as previously with a lower reward of 0.88. Q-learning algorithm achieves the lowest reward of 0.75 at episode 2200. It can be concluded that dueling DQN and original DQN perform better than Q- Learning in terms of rate of convergence and the level of system reward. This is because, the neural network fits the state definition better than the state discretization

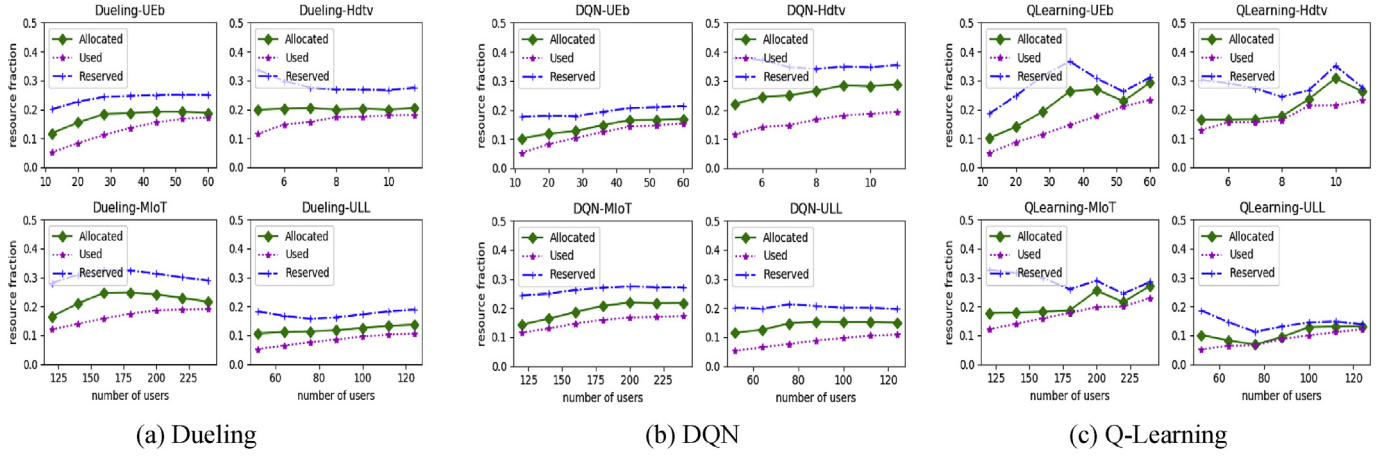

Fig. 3. Slice resource provisioning.

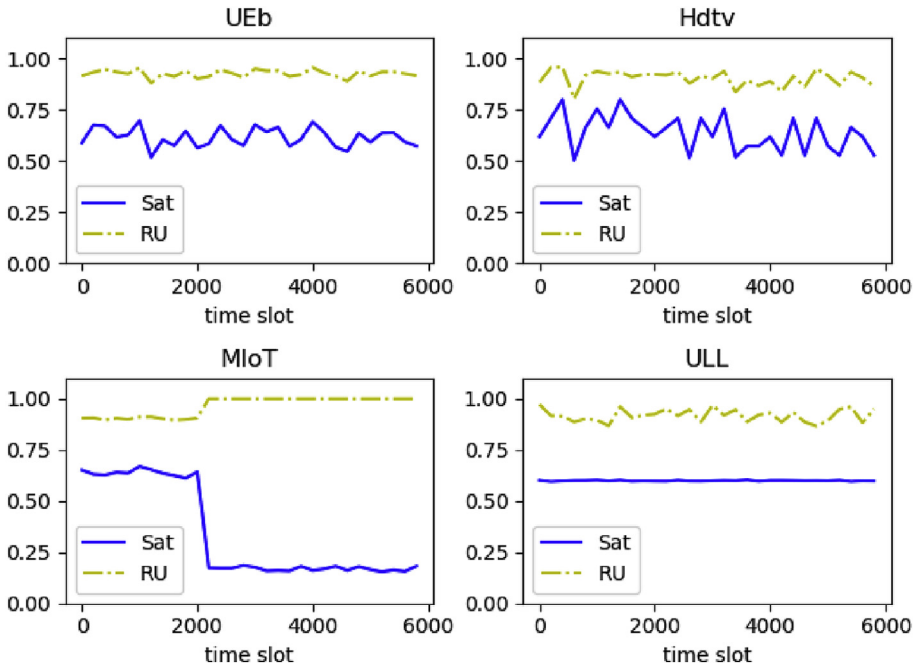

Fig. 4. Slice isolation.

in Q-learning. The convergence and complexity of the dueling DQN algorithm depends on the number of state and action sets involved in the learning process. As soon as convergence is achieved, any incoming states do not have to be learned anymore thereby making the learning curve a steady one. If the state space is too large, Q-learning is inapplicable to solve resource slicing problems in real-time. Leveraging the deep dueling networks, we develop optimal and fast algorithms to overcome this shortcoming.

6.3. Resource provisioning and allocation for slices

To evaluate the resource provisioning and allocation of slices, we compare the performance of the 4 defined slices under dueling DQN, DQN and Q-learning schemes based on reserved, allocated and used resources. The result of this simulation is the average of the last 100 episodes. Initially, we reserve the resources for each slice considering the user demand and traffic distribution which is referred to as the reserved resource. Then, we apportion part of the reserved resource to be allocated to users in the slice which is termed as the allocated resource. The unused resource is kept and can be allocated back to the slices when the user population increases in the slice. The reserved resource fractions of all the slices at a BS sum up to 1. The allocated resource fractions of all the slices at a BS is less than 1. The maximum user population in the UEb, Hdtv, ULL and MIoT slices are 60, 11, 240 and 124 respectively. A snapshot of the reserved, allocated and used resource results of the 4 slices are shown in Fig. 3(a), Fig. 3(b) and (c) with respect to the proposed dueling DQN, original DQN and Q-learning schemes respectively.

Fig. 3(a) shows that each slice does not fully utilize the allocated and reserved resources in dueling DQN. At heavy load, the used resource is very close to the allocated resource but far from the reserved resource. The sum of the fraction of resource allocated to all the slices under dueling DQN is 0.752. Similarly, in Fig. 3(b), none of the slices fully utilizes the allocated and reserved resources in DQN with the allocated resource and used resource curves of Hdtv slice wide apart. The sum of the allocated resource fraction of all the slices under dueling DQN is 0.824. However, in Fig. 3(c), the reserved, allocated and used resources follow an abnormal trend. At light load, Hdtv, MIoT and ULL slices utilize close to the allocated resource although none of the slices utilized more than the allocated resource. The Q-learning algorithm allocates the reserved resource of about 95% to the slices. Since all the slices do not utilize all their reserved or allocated resources, it can be concluded that our proposed algorithm and DQN algorithm can achieve a better resource provisioning and allocation than the Q-Learning algorithm. Our goal is to satisfy users with the minimum amount of resource. In dueling DQN and DQN, the allocated resource fraction is less than in Q-learning.

6.4. Performance on slice isolation

In this simulation, we observe the result of change in user population in a slice on the other slices based on the proposed dueling DQN algorithm. Fig. 4 shows the result of slice isolation on satisfaction and resource utilization in the 4 slices. In the UEb slice, we set the number of users at 60. In the Hdtv, MIoT and ULL slices, we set the number of users at 11, 240 and 124 respectively. These values are based on the outcome last episode captured in this simulation.

Initially, the fraction of resource allocated to the UEb, Hdtv, MIoT and ULL slices are 0.188, 0.211, 0.216 and 0.137 respectively which are the results of dueling DQN for the last episode. Assuming a time slot is1ms, we set one slicing period as 6000 time slots which is equal to 6000 ms.

From Fig. 4, it can be observed that the UEb and Hdtv slices do not keep steady levels of satisfaction and resource utilization as the number of time slots increases although the resources are not fully utilized in the slices. The satisfaction and resource utilization levels remain at constant levels throughout the time slots under a constant user population in the ULL slice. To observe the effect of sudden change in user population in one slice to other slices, we increase the number of users in the MIoT slice to 420 after the 2000th time slot. The slice achieves a satisfaction level of 0.25 and a resource utilization level of 1.00. The satisfaction and resource utilization levels of the other slices remain unchanged even though the increase in the user population was changed in the MIoT slice.

We conclude that since each slice's resource is isolated from the others, an increase in user population in a slice cannot affect resource utilization and satisfaction of other slices.

6.5. System stability analysis on queue length

In this simulation, we use the same configuration for performance on slice isolation to evaluate the stability of the system on queue length.

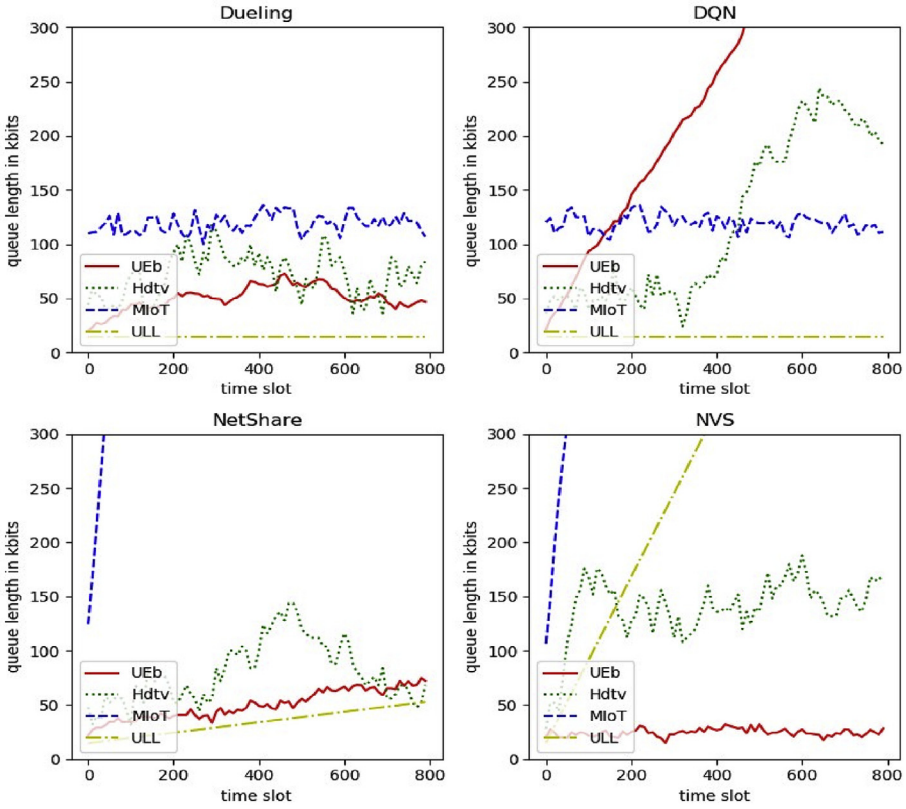

Fig. 5. Queue length per slice over dynamic scenarios.

Initially, the fraction of resource allocated to the UEb, Hdtv, MIoT and ULL slices are 0.169, 0.288, 0.217 and 0.15 respectively which are the results of DQN for the last episode. The different traffic characteristics and application types in the whole system results in different user requirements in terms of R_u and Td_u. From equation (35), n_u^h and n_u^v are two parameters defining the shape of LSRU for user (u) from QoS request $Rqst$ $(R_u,\ Td_u)$ based on equations (2) and (3) in the problem formulation. In our simulation, we assume that all users from a common slice have the same request on R_u and Td_u. Each slice has one specific queue for user requests. Since the queue is dedicated to each slice, all of the user flows share a common traffic arrival rate in one slice. The M/M/1 queuing model is used to calculate the expected waiting delay for each slice as described with equation (5).

To check the performance on system stability in view of queuing as a long-term dynamic process, we compare our dueling DQN algorithm with DQN, NVS and NetShare in terms of queue length. The results are shown in Fig. 5. At a maximum queue length of 300kbits, the buffer lengths of all slices always fluctuate around a certain value in dueling DQN due to reasonable resource allocation. The buffer length of UEb slice is always on the rise in DQN due to insufficient resources allocated to the UEb slice. In NetShare, the buffer length of MIoT slice is always on the rise due to insufficient resources allocated to the MIoT slice. In NVS, the buffer length of MIoT and ULL slices are high as a result of insufficient resources. Because of unfair resource allocation for MIoT in NetShare, its queue length increasing linearly with time. The fluctuation in queue length is due to the data packets arriving in exponential distribution. However, from the point of view on stability performance, dueling DQN algorithm outperforms the other three methods.

7. Conclusion

In this paper, we have proposed a hierarchical framework and dueling DQN based autonomous slicing refinement for heterogeneous traffics in virtualized RAN. Autonomous slicing refinement adjusts the resources provisioned to individual slices to balance QoS satisfaction and resource utilization using dueling DQN technique. Dueling DQN uses a two-stream Q-function that has the ability to learn which states are important to the agent without learning the effect of each action on each state. Slice satisfaction demand are met with the minimum amount of resource. Then, a shape-based heuristic algorithm for intra-slice resource allocation was proposed to map slices to the BSs which ensures efficient QoS satisfaction of slices. System-level performance evaluation was conducted with the following metrics; convergence analysis, performance analysis on slice and system levels, performance on slice isolation and stability analysis on queue length. The simulation results showed that the proposed dueling DQN algorithm converges to the optimal solution. The proposed algorithm balances QoS satisfaction and resource utilization with 80% of the available resource compared with existing state-of-the-art solutions, NVS and NetShare.

Author contributions section

Guolin Sun: Conceptualization, Methodology, Validation, Resources, Visualization, Project administration, Supervision. **Kun Xiong:** Investigation, Software. **Gordon Owusu Boateng:** Writing-Original draft preparation, Visualization. **Guisong Liu:** Writing-Review & Editing, Validation **Wei Jiang:** Writing-Review & Editing, Formal Analysis.

Declaration of competing interest

The authors declared that they have no conflicts of interest to this work.

References

3GPP TS 38.300, 2018. 5G-NR: Overall Description (version 15.3.1 Release 15).

Aijaz, A., Sept. 2018. Hap-SliceR: a radio resource slicing framework for 5G networks with haptic communications. IEEE Syst. J. 12 (3), 2285–2296.
Caballero, P., Banchs, A., de Veciana, G., Costa-Perez, X., Apr. 2019. Network slicing games: enabling customization in multi-tenant mobile networks. IEEE/ACM Trans. Netw. 27 (2), 662–675.
Chang, Z., Han, Z., Ristaniemi, T., April 2017. Energy efficient optimization for wireless virtualized small cell networks with large scale multiple antenna. IEEE Trans. Commun. 65 (4), 1696–1707.
Chang, Z., Zhou, Z., Zhou, S., Chen, T., Ristaniemi, T., 2018. Towards service oriented 5G: virtualizing the networks for everything-as-a-service. IEEE Access 6, 1480–1489.
Chazelle, Aug. 1983. The Bottom-left bin-packing heuristic: an efficient implementation. IEEE Trans. Comput. C-32 (8), 697–707.
da Silva, I., Mildh, G., Kaloxylos, A., Spapis, P., Buracchini, E., et al., Jun. 2016. Impact of network slicing on 5G radio access networks. In: European Conference on Networks & Communications(EUCNC) , Athens, Greece, pp. 153–157.
Derakhshani, Mahsa, Parsaeefard, Saeedeh, Le-Ngoc, Tho, Leon-Garcia, Alberto, Aug. 2018. Leveraging synergy of SDWN and multi-layer resource management for 5G networks. IET Netw. 7 (5), 336–345.
Fu, F., Kozat, U.C., Mar. 2010. Wireless network virtualization as a sequential auction game. In: 2010 IEEE Conference on Computer Communications (INFOCOM), San Diego, California, pp. 1–9.
Garces, P.C., Costa-Perez, X., Samdanis, K., Banchs, A., May 2015. RMSC: a cell slicing controller for virtualized multi-tenant mobile networks. In: 81st IEEE Vehicular Technology Conference. VTC Spring, Glasgow, UK, pp. 1–6.
Guo, T., Arnott, R., Sept. 2013. Active LTE RAN sharing with partial resource reservation. In: IEEE Vehicular Technology Conference Proceedings, Las Vegas, Nevada, pp. 1–5.
Habiba, U., Hossain, E., Mar. 2018. Auction mechanisms for virtualization in 5G cellular networks: basics, trends and open challenges. IEEE Commun. Surv. Tutor. 20 (3), 2264–2293.
Han, T., Ansari, N., Mar. 2015. User association in backhaul constrained small cell networks. In: 2015 IEEE Wireless Communication and Networking Conference (WCNC), pp. 1637–1642.
Jiang, M., Condoluci, M., Mahmoodi, T., May 2016. Network slicing management and prioritization in 5G mobile systems. In: 22nd European Wireless Conference, Olulu, Finland, pp. 1–6.
Jiang, M., Xenakis, D., Costanzo, S., Oassa, N., Mahmoodi, T., 2017a. Radio resource sharing as a service in 5G: a software-defined networking approach. Comput. Commun. 107, 13–29.
Jiang, M., Condoluci, M., Mahmoodi, T., 2017b. Network slicing in 5G: an auction-based model. In: Proc. IEEE International Conference on Communications (ICC), Paris, France, pp. 1–6.
Johnson, D.S., Garey, M., 1990. Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York, NY, USA.
Kalil, M., Shami, A., Ye, Y., May 2014. Wireless resources virtualization in LTE systems. In: IEEE Conference on Computer Communications Workshops, Toronto, Ontario, pp. 363–368.
Kalil, M., Shami, A., Al-Dweik, A., Jun. 2015. Wireless resources virtualization for cloud radio access networks (C-RAN). IEEE J. Comput. Sci. 120, 3366–3379.
Kamel, M.I., Le, L.B., Girard, A., Sept. 2014. LTE wireless network virtualization: dynamic slicing via flexible scheduling. In: 80th IEEE Vehicular Technology Conference (VTC Fall), Vancouver, British Columbia, pp. 1–5.
Kasgari, A.T.Z., Saad, W., March 2018. Stochastic optimization and control framework for 5G network slicing with effective isolation. In: 52nd Annual Conference on Information Sciences and Systems (CISS), Princeton, NJ, USA, pp. 1–6.
Khatibi, S., Carreia, L.M., Mar. 2015. A model for virtual radio resource management in virtual RANs. EURASIP J. Wirel. Commun. Netw. 2015 (68), 1–12.
Kokku, R., Mahindra, R., Zhang, H., Rangarajan, S., Oct. 2012. NVS: a substrate for virtualizing wireless resources in cellular networks. IEEE/ACM Trans. Netw. 20 (5), 1333–1346.
Liang, C., Yu, F.R., Apr. 2015. Distributed resource allocation in virtualized wireless cellular networks based on ADMM. In: IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Hong Kong, pp. 360–365.
Mahindra, R., Khojastepour, M., Zhang, H., Rangarajan, S., Oct. 2013. Radio access network sharing in cellular networks. In: 21st IEEE International Conference on Network Protocols (ICNP), pp. 1–10.
Mamman, M., Hanapi, Z.H., Abdullah, A., Muhammed, A., Jan. 2019. Quality of service class identifier (QCI) radio resource allocation algorithm for LTE downlink. PloS One 14 (1), 1–22.
Moubayed, A., Shami, A., Lutfiyya, H., Dec. 2015. Wireless resource virtualization with device-to-device communication underlaying LTE network. IEEE Trans. Broadcast. 61 (4), 734–740.
Panchal, J.S., Yates, R.D., Buddhikot, M.M., Sept. 2013. Mobile network resource sharing options: performance comparisons. IEEE Trans. Wireless Commun. 12 (9), 4470–4482.
Petrov, V., Lema, M.A., Gapeyenko, M., Antonakoglou, K., Moltchanov, D., Mar. 2018. Achieving end-to-end reliability of mission-critical traffic in softwarized 5G networks. IEEE J. Sel. Area. Commun. 36 (3), 485–501.
Richart, Matias, et al., Sept. 2016. Resource slicing in virtual wireless networks: a survey. IEEE Trans. Netw. Serv. Manag. 13 (3), 462–476.
Saquib, N., Hossain, E., Le, L.B., Kim, D.I., June 2012. Interference management in OFDMA femtocell networks: issues and approaches. IEEE Wirel. Commun. 19 (3), 86–95.
Sheng, M., Xu, C., Wang, X., Zhang, Y., Han, W., Li, J., Oct. 2014. Utility-based resource allocation for multi-channel decentralized networks. IEEE Trans. Commun. 62 (10), 3610–3620.
Song, L., Niyato, D., Han, Z., Hossain, E., June 2014. Game-theoretic resource allocation methods for device-to-device communication. IEEE Wirel. Commun. 21 (3), 136–144.
Sutton, R.S., Barto, A.G., Sept. 1998. Reinforcement learning: an introduction. IEEE Trans. Neural Netw. 9 (5), 1054.
Tseliou, G., Adelantado, F., Verikoukis, C., Aug. 2016. Scalable RAN virtualization in multi-tenant LTE-A heterogeneous networks. IEEE Trans. Veh. Technol. 65 (8), 6651–6654.
Tsiropoulou, E.E., Vamvakas, P., Papavassiliou, S., Dec. 2013. Joint utility-based uplink power and rate allocation in wireless networks: a non-cooperative game theoretic framework. Phys. Commun. 9, 299–307.
Tsiropoulou, E.E., Vamvakas, P., Katsinis, G.K., Papavassiliou, S., Dec. 2015. Combined power and rate allocation in self-optimized multi-service two-tier femtocell networks. Comput. Commun. 72, 38–48.
Wang, N., Fei, Z., Kuang, J., 2016a. QoE-aware resource allocation for mixed traffics in heterogeneous networks based on Kuhn-Munkres algorithm. In: 2016 IEEE International Conference on Communication Systems (ICCS), pp. 1–6.
Wang, Z., Schaul, T., Hessel, M., Van Hasselt, H., Lanctot, M., De Freitas, N., 2016b. Dueling network architectures for deep reinforcement learning. In: International Conference on Machine Learning. ICML, New York, NY, pp. 1–15.
Watkins, C.J.C.H., 1989. Learning from Delayed Rewards. Ph.D. Thesis. University of Cambridge, England.
Yang, Z., Xie, Y., Wang, Z., Jan. 2019. A Theoretical Analysis of Deep Q-Learning. https://arxiv.org/pdf/1901.00137.
Zaki, Y., Zhao, L., Goerg, C., Timm-Giel, A., Aug. 2011. LTE mobile network virtualization exploiting multiplexing and multi-user diversity gain. Mobile Network. Appl. 16 (4), 424–432.
Zhang, D., Chang, Z., Hamalainen, T., Yu, F.R., Dec. 2017. Double auction based multi-flow transmission in software-defined and virtualized wireless networks. IEEE Trans. Wireless Commun. 16 (10), 8390–8404.

Guolin Sun received his B.S., M.S. and Ph.D. degrees all in Comm. and Info. System from the University of Electronic Sci. and Tech. of China (UESTC), Chengdu, China, in 2000, 2003 and 2005 respectively. from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2000, 2003 and 2005 respectively. Since graduation with Ph.D. in 2005, he has got eight years industrial work experience on wireless research and development for 4G/5G, Wi-Fi, Internet of things, cognitive radio, localization and navigation. Before he joined the School of Computer Science and Engineering, University of Electronic Science and Technology of China as an Associate Professor in Aug. 2012, he worked at Huawei Technologies, Sweden. Dr. Sun has filed over 40 patents, and published over 40 scientific conference and journal papers, acts as TPC member of conferences. Currently, he serves as a vice-chair of the 5G oriented cognitive radio special interest group (SIG) of the IEEE Technical Committee on Cognitive Networks (TCCN). His general research interests include software defined networks, network function virtualization, artificial intelligence and radio resource management.

Kun Xiong received his BSc. degree in Computer Science from Civil Aviation Flight University of China, in 2016. He is currently studying MSc. Computer Science at the University of Electronic Science and Technology of China, due to finish in July 2019. He is also a member of the Mobile Cloud-Net Research Team – UESTC. His primary research interests include network virtualization, deep learning and internet-of-things.

Gordon Owusu Boateng received his bachelor degree in Telecommunications Engineering from the Kwame Nkrumah University of Science and Technology (KNUST), Kumasi-Ghana, West Africa, in 2014. He is currently studying his MSc. Computer Science and Engineering in University of Electronic Science and Technology of China (UESTC). From 2014 to 2016, he worked under sub-contracts for Ericsson (Ghana) and TIGO (Ghana). He is also a member of the Mobile Cloud-Net Research Team – UESTC. His interests include mobile/cloud computing, 5G wireless networks, data mining, D2D communications and SDN.

Guisong Liu received his B.S. degree in Mechanics from the Xi'an Jiao Tong University, Xi'an, China, in 1995, and his M.S. degree in Automatics, Ph.D degree in Computer Science from the University of Electronic Science and Technology of China, Chengdu, China, in 2000 and 2007, respectively. Prof. Liu was a visiting scholar at Humbolt University, Berlin from Sept. to Dec. 2015. Now, he is a full professor in the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China. He is also the dean of the School of Computer Science, Zhongshan Institute, UESTC, Zhongshan, China. His research interests include pattern recognition, neural networks, and machine learning

Wei Jiang received his Ph.D. degree from Beijing University of Posts and Telecommunications (BUPT) in 2008. Since Mar. 2008, he has been worked 4 years in Central Research Institute of Huawei Technologies, in the field of wireless communications and 3GPP standardization. In Sept. 2012, he joined the Institute of Digital Signal Processing, University of Duisburg-Essen, Germany, where he was a Postdoctoral researcher and worked for EUFP7 ABSOLUTE project and H2020 5G-PPP COHERENT project. Since Oct. 2015, he joined the Intelligent Networking Group, German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, Germany, as a senior researcher and works for H2020 5G-PPP SELFNET project.