Deep Q-learning for 5G network slicing with diverse resource stipulations and dynamic data traffic

Debaditya Shome
School of Electronics Engineering
KIIT University
Bhubaneswar, India.
ORCID ID: 0000-0001-9168-0379

Ankit Kudeshia
School of Electronics Engineering
KIIT University
Bhubaneswar, India.
ORCID ID: 0000-0002-3479-2099

***Abstract*—5G wireless networks use the network slicing technique that provides a suitable network to a service requirement raised by a network user. Further, the network performs effective slice management to improve the throughput and massive connectivity along with the required latency towards an appropriate resource allocation to these slices for service requirements. This paper presents an online Deep Q-learning based network slicing technique that considers a sigmoid transformed Quality of-Experience, price satisfaction, and spectral efficiency as the reward function for bandwidth allocation and slice selection to serve the network users. The Next Generation Mobile Network (NGMN) vertical use cases have been considered for the simulations which also deals with the problem of international roaming and diverse intra-use case requirement variations by using only three standard network service slices termed as enhanced Mobile Broadband (eMBB), Ultra Reliable Low Latency Communication (uRLLC), and massive Machine Type Communication (mMTC). Our Deep Q-Learning model also converges significantly faster than the conventional Deep Q-Learning based approaches used in this field. The environment has been prepared based on ITU specifications for eMBB, uRLLC, mMTC. Our proposed method demonstrates a superior Quality-of-experience for the different users and the higher network bandwidth efficiency compared to the conventional slicing technique.**

***Index Terms*—5G, wireless communication, Network slicing, Deep Q-learning, Quality-of-Experience, NGMN vertical use-cases.**

I. Introduction

After 2020, it is anticipated that the ongoing decade would experience a 1000 fold-increase in data traffic compared to the 2010 levels [1]. Unlike the earlier generations of communication networks, this huge amount of data traffic also includes a large variety of user requirements which vary from customer to customer. The 3G/4G network performance used to be evaluated using the hard metrics such as peak data-rates, coverage, and spectral efficiency wherein the 5G network performance is proposed to be evaluated in terms of user's Quality-of-Experience (QoE). Switching to QoE would assure positive experience of end users, which would mean the overall success of a network from the user perspective [2]. 5G networks will also offer a more user-centric and context-aware experience, delivering personalized content and assistance services [3]. therefore, a single network is not sufficient to satisfy every customer's specific requirements needing the concept of network slicing [4].

According to the Next Generation Mobile Network (NGMN) [5], a network slice is a set of virtual network functions and the resources to run these functions, forming a logical network that meets the requirements of a particular use case. As physical network slices would lead to higher costs and higher energy requirements hence creating virtualized network slices is more viable and sustainable solution. The implementation and the programmability of network functions is provided by network function virtualization (NFV) and software defined networking (SDN) [6]. Forums such as NGMN Alliance and ITU-R consider the following three main 5G service types: Enhanced Mobile Broadband (eMBB) which requires high bandwidth and ensures high network capacity, Massive machine type communication (mMTC) which provides a huge connection density, and Ultra Reliable Low Latency Communication (uRLLC) which provides very low latency and high reliability to satisfy mission critical use cases. Use cases have been grouped by many organisations with the aim of covering every possible requirements a user can have. The METIS-II project [7] defined 5 use cases which according to them covers most vertical use cases, but due to a large variety of intra use case requirement variations being provided, we use the 25 vertical use cases of the NGMN for this paper.

A large number of research articles are focused on the network slicing based on creating multiple slices without taking into account the complexities and challenges of roaming and interoperatibility. Therefore, the global system for mobile communications association (GSMA) [8] states that a standardization practice for defining Network Slices is necessary for Inter-operator roaming. In addition to the above, GSMA also defines that mobile network operators can deploy multiple network slices of different types that are together packaged as a single product targeted towards business customers (business bundle) having multiple and diverse requirements. For instance a vehicle may simultaneously need an eMBB slice for infotainment and an uRLLC slice for telemetry, assisted

driving etc. Thus the use cases also consider the multiple slice allocation to the network users. Next, we discuss the various works in the domain of 5G network resource allocation.

II. Related Works

The authors in [9] have worked on the problem of two level resource allocation in network slicing mainly from a financial point of view of resource bidding between the mobile virtual network operator (MVNO) and the infrastructure provider (InP) as higher level problem and assignment of resources to users from multiple MVNOs as lower level problem. Next, adding the machine learning framework to this, the authors of [10] have used supervised machine learning algorithms for the problem of slice selection as a classification task on a dataset which achieved a good accuracy while training and testing. The major issue with this work is that the deployment of the supervised machine learning model in a network would loose accuracy while encountering the real time network data. Next, in [11], a deep learning based slice selection approach is proposed based on key performance indicators (KPIs) for the varying traffic load prediction. The main drawback of this work is that predicting future traffic can be less accurate keeping in mind the wide variety and randomness in data traffic nowadays. Further deep reinforcement learning was also explored to be used for network the resource allocation. The work in [12] used the classical DQN for resource allocation among the network slices and in [13], the authors proposed an interesting modification to the classical DQN by using discrete normalized advantage functions to convert the discrete action space to a continuous which further enhanced the performance of the agent. However, in all the above works consider slice selection without taking into account whether a user with diverse requirements is satisfied or not and therefore the reward function has not been defined properly to guide the agent to maximize the user satisfaction and secondly the convergence time is significantly higher which would lead to a high delay in serving customers in real time scenarios.

Moreover, all the above discussed works in resource allocation among network slices have considered a user getting a single slice, not taking into account the fact that 3GPP rel 16 stated that a user equipment (UE) can be simultaneously connected with up to 8 network slices. Motivated by the above, our contributions in this work can be summarized as

- An online deep reinforcement learning algorithm has been proposed as a modified version of classical deep Q learning for bandwidth allocation and slice selection.
- A novel robust reward metric is proposed for our deep Q learning agent inspired from activation functions which guides the agent to converge in about 100 times lesser episodes than previous approaches [13].
- Our work analyses all the NGMN 25 vertical industry use cases and demonstrates that with only three standard slices all the diverse time-varying user requirements can be served using multi-slice connectivity to UEs.

III. System model and Problem formulation

A. Notations

The mathematical notations used in this paper are defined in Table I.

Symbol	Description
$\mathbf{B}_T$	Total available bandwidth
$\mathbb{A}$	Action space
s_t	State at time t
$\mathbf{B}_i$	Bandwidth allocated to i_{th} slice
$\mathbf{L}_i$	E2E latency of i_{th} slice
$\mathbf{R}_i$	Reliability of i_{th} slice
$\mathbf{C}_i$	Capacity of i_{th} slice
ϑ_i	Quality of Experience (QoE) of i_{th} slice
Θ_i	User Satisfaction ratio if i_{th} slice is assigned
Θ_u	User Satisfaction score of the user
π_i	Cost per unit of bandwidth of i_{th} slice
r	Reward
$\mathbf{S}_e$	Spectral efficiency

TABLE I: Symbols and notations

B. System Model

We consider a RAN slicing scenario of multiple MVNOs, each having multiple logical virtualized basestations (vBS) with limited system bandwidth of $\mathbf{B}_T$ each. There is a set of $\mathbf{n}$ users which belong to 25 different use cases as defined in [14]. A use case refers to a group of users with similar requirements. The user information of all the $\mathbf{n}$ users including the use case ID of each is passed on to a MVNO assigns a vBS to the user based on prioritised traffic scheme. For each vBS, a DQN agent is activated and according to the KPI requirements of the particular use case, the MVNO assigns the total bandwidth for a user which along with the user information is further passed on to a DQN agent. The agent further allocates the total bandwidth $\mathbf{B}_T$ into the single or multiple standard slices based on the diverse user requirements where

$$\mathbf{B}_T = \mathbf{B}_1 + \mathbf{B}_2 + \mathbf{B}_3. \tag{1}$$

Next the DQN assigns rewards to make the slices tailored to the specific use case requirements and passes the optimal bandwidth allocation information to the MVNO which finally serves the customer as either a single slice or a business bundle of two or three slices as described in [8]. Simultaneously other $\mathbf{n-1}$ DQNs allocate optimal bandwidth for the remaining $\mathbf{n-1}$ users. Once an already served user disconnects, the bandwidth gets freed from it and next user from the user queue is assigned to it's agent. We define generic slice templates for the three slices as per the idea from GSMA [8] for dealing with the problem of roaming. 1st slice template is based on uRLLC service type involving high reliability and very low latency, 2nd slice template is based on eMBB service type which involves higher bandwidth and extremely high data rates, 3rd slice template is based on mMTC service type which involves high connection density and scalability [7]. The QoE for each user of the i_{th} slice can be defined as a weighted sum of all the KPIs given as

$$\vartheta_i = \alpha_1 f_1\left(\frac{\mathbf{C}_i}{\mathbf{C}_t}\right) + \alpha_2 f_2\left(\frac{\mathbf{L}_i}{\mathbf{L}_t}\right) + \alpha_3 f_1\left(\frac{\mathbf{R}_i}{\mathbf{R}_t}\right). \tag{2}$$

The scaling functions $f_1(.)$ and $f_2(.)$ in (2) are mathematically defined as

$$f_1(x) = 1 - \frac{\log(10)}{\log(10+x)}, \tag{3}$$

$$f_2(x) = \frac{\log(10)}{\log(10+x)}, \tag{4}$$

where $f_1(.)$ scales the values of the capacity and the reliability satisfaction ratios in a monotonically increasing manner and $f_2(.)$ scales the value of the latency satisfaction in a monotonically decreasing manner. The target values $\mathbf{C}_t$, $\mathbf{L}_t$ of the QoE (ϑ) in (2) for all the 25 NGMN use cases from [14] are given in Table II. Further, α_1 , α_2 and α_3 are weights assigned to the KPIs signifying the importance of the satisfaction of the particular KPI to the user. The target values for the KPIs might be same within a use case but the weights can vary also among users of the same use case. For instance, if user of the use case group UC7 (High Speed train) is downloading a file with large size, he needs high data-rate, so the weight corresponding to the capacity term would be higher, whereas another user in the train might be playing a multiplayer game requiring low latency, then the weight corresponding to the latency term would be much higher. Similarly within every 25 use cases there would be users with different importance to each KPI. From a subscriber's point of view, getting good quality service at the lowest possible price is definitely the measure of satisfaction. Hence, the price satisfaction p_i of ith slice where $i = 1$, $i = 2$ and $i = 3$ denote the uRLLC, eMBB and mMTC slices respectively, can be defined as

$$p_i = \beta f_2\left(\frac{\mathbf{B}_i \pi_i}{\mathbf{B}_T}\right) \tag{5}$$

where β is the weight which refers to the importance of lower price for the user. The ratio π_1:π_2:π_3 is the price ratio where $\pi_1 > \pi_2 > \pi_3$ as uRLLC slice is given the highest priority, followed by eMBB and mMTC. Finally, the user satisfaction score Θ_i of the ith slice is defined as

$$\Theta_i = \frac{1}{1 + \exp\left(-\vartheta + \log(p_i \mathbf{S}_e)\right)} \tag{6}$$

A subscriber can have two types of operator plans: single slice or multi-slice. In single slice plan, the subscriber is allotted the slice with the highest reward Θ_i. In multi-slice plan, a subscriber is allotted multiple slices based on his/her service level agreement (SLA) operator plan. Let the SLA operator plan be defined as a binary mask vector $[w_1 \quad w_2 \quad w_3]$ where w_1, w_2, w_3 are value of w_i is equal to 1 if ith slice is required by the user, else is equal to 0. Let the individual user satisfaction score are arranged in the vector defined as $[\Theta_1 \quad \Theta_2 \quad \Theta_3]$. From this, the user satisfaction score Θ for the multi-slice user can be defined as

$$\Theta = \frac{[w1 \quad w2 \quad w3] \bullet [\Theta_1 \quad \Theta_2 \quad \Theta_3]^T}{w_1 + w_2 + w_3}. \tag{7}$$

The final user satisfaction score of any user can be summarised as :

$$\Theta_u = \begin{cases} \max(\Theta_i) & \text{if operator plan is single-slice} \\ \Theta \quad \text{from((7))} & \text{if operator plan is multi-slice} \end{cases} \tag{8}$$

C. Problem Formulation

The objective of the problem formulation is to make sure every diverse KPI requirement of each user is satisfied and the QoS parameters defined in Table II is maximised for every use case. For this objective we divide the problem into two sub-problems:

1) **Network slicing problem:** The task of slicing $\mathbf{B}_T$ and further slice selection based on user satisfaction is formulated as a Markov Decision process (MDP). A Markov Reward Process is a tuple $< S, A, P, R, \gamma >$. The goal of the MDP is to maximize the Quality of Experience (ϑ) of each user and also keeping into consideration the price satisfaction and fair bandwidth utilization.
2) **Heterogeneous traffic assignment problem:** We design a methodology to deal with the complex task of assigning users to a MVNO's vBS based on traffic and priority to different use cases.

IV. Proposed Models:

A. DQN based Network slicing

For solving the MDP problem to maximize rewards, Q learning can be used which works as an off-policy, model-free , online reinforcement learning algorithm. But due to the large action space of our problem leading to a huge Q-Table, Q-learning would take a long time to converge which would lead to delay faced by customers. To overcome this challenge we use Deep Q learning in which the Q function is approximated using a Deep neural network. As [15], we use a DQN agent with experience replay which consists of two Neural networks, a target neural network which calculates the target Q value and another is actor neural network which is used for updating the network parameters and generating the sampled values from experience replay pool and further choosing actions. The proposed DQN has the following parameters:

Action space: The actions are three discrete bandwidths allocated to the three slices which sum up to the total bandwidth as given in (1)

$$\mathbb{A} = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ \vdots & \vdots & \vdots \\ b_{n1} & b_{n2} & b_{n3} \end{bmatrix}$$

State Space: The state space contains environment states/observations which consists of Signal to noise ratio (SINR), arrival rate, packet size, latency and error rate values.

Reward: The reward r is defined according to (8)

$$r = \Theta_u \tag{9}$$

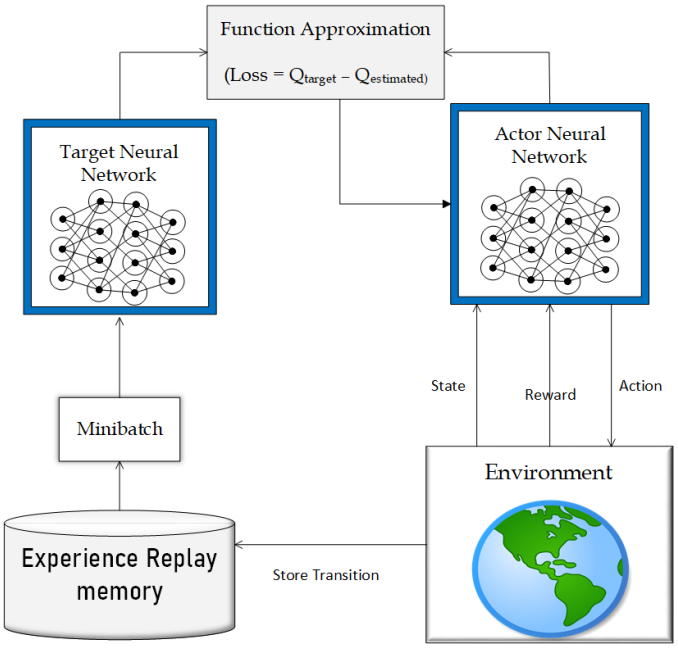

Fig. 1: Deep Q Learning illustration

use case	eMBB	uRLLC	mMTC	$\mathbf{C}_t$	$\mathbf{L}_t$	$\mathbf{R}_t$
UC1: Pervasive Video	✓	✓		300	10	0.999
UC2: Smart Office	✓	✓		1000	10	0.99999
UC3: Operator Cloud	✓	✓		300	10	0.99999
UC4: Open-Air Gathering	✓	✓	✓	25	10	0.99999
UC5: 50+ Mbps everywhere		✓	✓	50	10	0.999
UC6: Ultra-low Cost			✓	10	50	0.999
UC7: High Speed Train	✓	✓	✓	50	10	0.999
UC8: Remote Computing		✓		50	10	0.99999
UC9: Moving Hot Spots		✓		50	10	0.999
UC10: Aircraft		✓	✓	15	10	0.999
UC11: Smart Wearables			✓	0.1	100	0.999
UC12: Sensor Networks			✓	0.1	100	0.999
UC13: Video Surveillance		✓	✓	50	50	0.99999
UC14: Tactile Internet		✓		50	1	0.99999
UC15: Natural Disaster			✓	0.1	1000	0.999
UC16: Automated Driving	✓	✓		10	1	0.99999
UC17: Collaborative Robots		✓	✓	10	1	0.99999
UC18: eHealth		✓		10	10	0.99999
UC19: Remote Surgery		✓		10	1	0.99999
UC20: Drones		✓	✓	10	10	0.999
UC21: Public Safety		✓		10	10	0.99999
UC22: News Information	✓			200	100	0.999
UC23: Local Broadcast	✓			200	100	0.999
UC24: Regional Broadcast	✓			200	100	0.999
UC25: National Broadcast	✓			200	100	0.999

TABLE II: NGMN vertical use case requirements

Q-Values: We use the Bellman equation to update the Q values

$$\mathbf{Q}(s,a) = \mathbf{R} + \gamma \max_{a^{t+1}} \mathbf{Q}(\mathbf{s}_{t+1}, \mathbf{a}_{t+1})$$

Experience Replay: The agent stores the past experiences/states as Transition tuples ['state', 'action', 'reward', 'next state'] and uniformly selects some mini-batch of items from the stored values to update the Q-value. It improves the sample efficiency of the algorithm by enabling data re-usability and also improves the stability during training.

Model free: Model-free learning is when an agent can directly derive an optimal policy on it's own from it's interactions with the environment without the need to create a model beforehand.

Loss function: We use the loss function defined in [16]

$$L_i(\theta_i) = \mathbb{E}\left[L_\delta\left(r + \gamma \max_{a'} Q(s', a'; \theta_i^-) - Q(s, a; \theta_i)\right)\right] \tag{10}$$

Here, L_δ is the Huber loss function, defined as:

$$L_\delta(a) = \begin{cases} \frac{1}{2}a^2 & \text{for } |a| \le \delta \\ \delta\left(|a| - \frac{1}{2}\delta\right) & \text{otherwise} \end{cases} \tag{11}$$

Feed-forward Neural network architecture: The Neural Network architecture used in our DQN consists of two fully connected linear units with ReLU activation function. We keep the number of neurons less so as to deal with the problem of overestimation of Q values in DQN. Both [15] and [16] have considered finite number of episodes in their MDP problem, but we consider a different approach. In our network slicing scenario, we have to make sure that the Loss is minimized and function converges in every DQN for a user, but in an episodic algorithm we would not know that for which use case how many episodes it will take to converge. So, instead of running the algorithm over a defined set of episodes, we run it as a semi-continuous task which terminates only when the Loss is lesser than a threshold value, as experimentally we observed that after the loss gets near to 0.05 the agent stops it's exploration phase and allots the same action everytime, hence we took this threshold to be below 0.05 so that there is no extra computational resource utilization by running the DQN agent for yielding the same action. This methodology assures every user gets the most optimal slicing configuration and to make sure to deal with time varying user demand variations,

B. Priority based Traffic assignment

For the sub-problem of user scheduling and vBS assignment, we consider multiple MVNOs with subscribed users, each MVNO having a limited system bandwidth and can create multiple logical vBS as per growing user demand. A user is assigned a priority based on his/her requirements. A traffic demand analyzer keeps track of number of connection requests at the moment and creates multiple vBS each with a assigned DQN agent. Based on UE information, the DQN selects the slice / slices for the user and allocates bandwidth to each slice to maximize the user satisfaction score (Θ). A resource monitor keeps track of remaining spectrum at each vBS and assigns a user belonging to a low demand use-case to the vBS's DQN agent which further allocates the required bandwidth. A Utility monitor keeps in notice about the slope of User satisfaction score ($\frac{\partial \Theta_i}{\partial t}$). If the slope decreases for at least a threshold amount of time, this would mean that the user's bandwidth requirements have increased, as a DQN agent once converged does not let Θ decrease.

Algorithm 1 Modified DQN with Experience Replay

Result: Optimal slice configuration policy π
Initialize replay memory $\mathcal{D}$ to capacity N
Initialize Q neural network
Initialize Target Q neural network
while *loss* $>$ *0.05* **do**
  for t = *1,T* **do**
    Set/Update ϵ value with ϵ-decay
    Choose an action a from state s using ϵ-greedy policy(Q)
    Store transition (s_t, a_t, r_{t+1}, s_{t+1}) in replay memory $\mathcal{D}$
    if *enough experiences in* $\mathcal{D}$ **then**
      Sample random mini-batch of transitions $(\phi_j, a_j, r_j, \phi_{j+1})$ from $\mathcal{D}$
      Set $y_j = \begin{cases} r_j, & \text{for terminal } \phi_{j+1} \\ r_j + \gamma \max_{a'} Q(\phi_{j+1}, a', \theta) & \text{for non-terminal } \phi_{j+1} \end{cases}$
      Compute Target Q values w.r.t old paramters
      Calculate Huber Loss between Q-network and Q-learning targets
      Update Q using RMSprop optimizer to minimize loss function
      Every C steps copy weights from Q to $\hat{Q}$
    end
  end
end

Algorithm 2 Heterogenous Traffic scheduling in a MVNO

Result: Efficient assignment of Spectrum to each user of a MVNO
Initialize and fill the user waiting Queue
Sort the waiting queue based on Priority and create multiple logical vBS as per number of users Activate agents required to serve the users based on traffic.
while *user queue not empty* **do**
  for *each user* **do**
    Assign a DQN agent
    Agent calculates and allocates the number of RBs of the Slice with maximum Θ_i to be allocated to a user
    if $\frac{\partial \Theta_i}{\partial t} < 0$ **then**
      Send the user to another vBS and activate a new agent Reconfigure the number of RBs to again maximise the declining Θ_i
    end
  end
end

V. Experimental results and discussions

A. Computational requirements

We used Python 3.7 for creating the environment for the Agent based on ITU-R specifications. The DQN agent was created using PyTorch and we ran the simulations in PyCharm IDE. For the simulation of multiple vBSs, we have used Python's capability of multiprocessing and considered each vBS as a worker process. Furthermore, the hardware specifications of the system used for running our simulations are listed in Table III.

TABLE III: Hardware specifications

Component	Specification
Processor	Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz
GPU	NVIDIA GeForce GTX 1650 with Max-Q Design
GPU memory	8113 MB
CUDA cores	896

B. Results

For the simulation, as discussed in section III-B, the RAN slicing scenario consists of multiple MVNOs sharing resources along with each MVNO having many vBSs but for highlighting the achievements of our model we consider a MVNO with each vBS having a system bandwidth of 30 Mhz. This 30 MHz of bandwidth would be distributed among users by our DQN agent. We take initial ϵ value to be 0.9 so as to make the agent cover all possible observations to ensure global optimum of the Loss function is reached. The price ratio of the slices π_1:π_2:π_3 is taken as $3:2:1$. Keeping in mind the higher expectations of the NGMN use cases in [14] as compared to 3GPP, we consider allotting the total 30 MHz bandwidth to a single user if their operator plan is multi-slice or even as low as 1 MHz if the user is a single-slice user with low QoE expectations. We simulated for 100 users with each belonging to one of the 25 NGMN use cases.

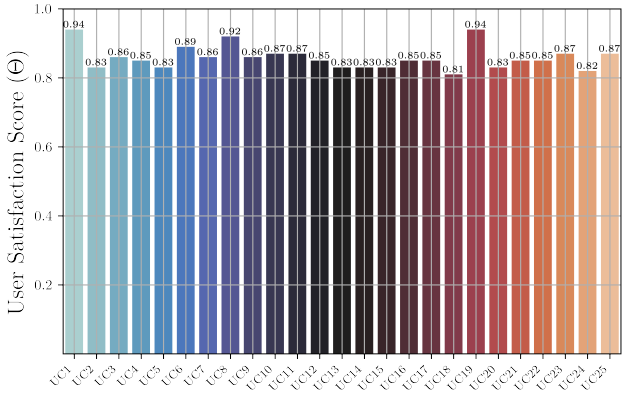

Fig. 2: Average User satisfaction score (Θ) achieved for the 25 NGMN use cases

Fig. 2 shows the average User Satisfaction score (Θ) achieved for users in a use case group. The best possible Θ achieved in our simulation is 0.94 and the least is 0.81.

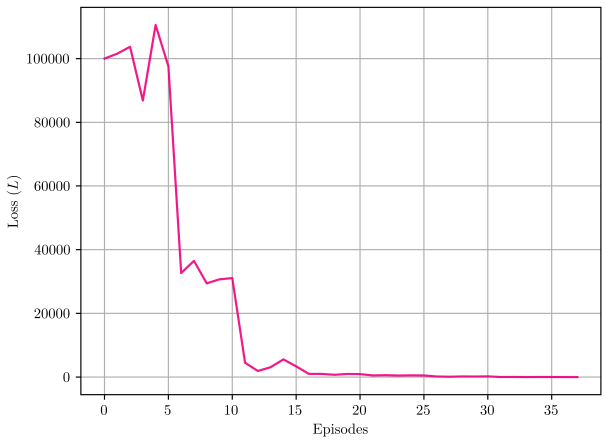

Fig. 3: Episodic Loss plot for the DQN agent

Due to the robust reward metric, for every use case our model converged in atleast 117 episodes in the worst case and 35 episodes in the best case, with each episode having 1000 training steps. This is about 100 or more times lesser than that of the the work in [13] and much more lesser than previously used DQN agents in this field.

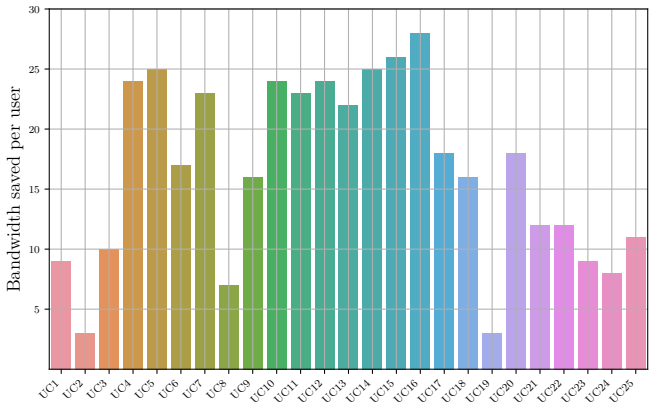

Fig. 4: Average bandwidth saved per use case

In Fig. 4, the average bandwidth saved per user in the use case by implementing our DQN agent is illustrated. It can be seen that even when only about 5 MHz or less bandwidth is used, our model provides a User satisfaction score more than 0.81 in an average for every use case, which is due to the fact that only use cases with higher throughput requirements require higher bandwidth and satisfying the use cases with other requirements can be done in minimal use of bandwidth. With much lesser use of bandwidth also our model is able to gain a high User Satisfaction score, in turn increasing the QoE, price satisfaction and also the spectral efficiency in every vertical NGMN use case.

VI. Conclusion

In this paper, we have presented a deep Q-learning based network slicing technique for an optimal resource allocation and slice selection in 5G wireless networks. The novel quality-of-experience based reward has been proposed for an efficient throughput, connectivity and latency requirement of the different services for a wide variety of NGMN use cases being allotted three standard network service slices termed as eMBB, uRLLC, and mMTC. The modified algorithm has also significantly reduced the convergence time for the DQN agent. The simulation results demonstrate the superior performance of the proposed slicing technique over the previously used methodologies in terms of user satisfaction score and the bandwidth efficiency of the network.

References

[1] P. K. Agyapong, M. Iwamura, D. Staehle, W. Kiess, and A. Benjebbour, "Design considerations for a 5G network architecture," *IEEE Communications Magazine*, vol. 52, no. 11, pp. 65–75, 2014.

[2] A. Mellouk, S. Hoceini, and H. A. Tran, "Quality of experience vs. quality of service : Application for a cdn architecture," in *2013 21st International Conference on Software, Telecommunications and Computer Networks - (SoftCOM 2013)*, pp. 1–8, 2013.

[3] B. Bangerter, S. Talwar, R. Arefi, and K. Stewart, "Networks and devices for the 5G era," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 90–96, 2014.

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

[5] N. Alliance, "Description of network slicing concept," *NGMN 5G P*, vol. 1, no. 1, 2016.

[6] C. Campolo, A. Molinaro, A. Iera, and F. Menichella, "5," pp. 38–45, 12 2017.

[7] S. E. Elayoubi, M. Fallgren, P. Spapis, G. Zimmermann, D. Martín-Sacristán, C. Yang, S. Jeux, P. Agyapong, L. Campoy, Y. Qi, *et al.*, "5G service requirements and operational use cases: Analysis and METIS II vision," in *2016 European Conference on Networks and Communications (EuCNC)*, pp. 158–162, IEEE, 2016.

[8] G. Association *et al.*, "Network slicing use case requirements," 2018.

[9] Y. K. Tun, N. H. Tran, D. T. Ngo, S. R. Pandey, Z. Han, and C. S. Hong, "Wireless network slicing: Generalized Kelly mechanism-based resource allocation," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 8, pp. 1794–1807, 2019.

[10] R. K. Gupta and R. Misra, "Machine learning-based slice allocation algorithms in 5G networks," in *2019 International Conference on Advances in Computing, Communication and Control (ICAC3)*, pp. 1–4, IEEE, 2019.

[11] A. Thantharate, R. Paropkari, V. Walunj, and C. Beard, "Deepslice: A deep learning approach towards an efficient and reliable network slicing in 5G networks," in *2019 IEEE 10th Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON)*, pp. 0762–0767, 2019.

[12] R. Li, Z. Zhao, Q. Sun, C. I, C. Yang, X. Chen, M. Zhao, and H. Zhang, "Deep reinforcement learning for resource management in network slicing," *IEEE Access*, vol. 6, pp. 74429–74441, 2018.

[13] C. Qi, Y. Hua, R. Li, Z. Zhao, and H. Zhang, "Deep reinforcement learning with discrete normalized advantage functions for resource management in network slicing," *IEEE Communications Letters*, vol. 23, no. 8, pp. 1337–1341, 2019.

[14] N. Alliance, "NGMN 5G white paper v1. 0," *approved and delivered by the NGMN board, 17th Feb*, 2015.

[15] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[16] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," 2013. cite arxiv:1312.5602Comment: NIPS Deep Learning Workshop 2013.