Journal of Engineering Research

Volume 7 | Issue 3 Article 12

2023

Slice Admission control based on Reinforcement Learning for 5G Networks

Mai Z. Ibrahim, Mohamed TALAAT FAHIM, Nada Elshennawy

Journal of Engineering Research (ERJ)
Vol. 7 – No. 3, 2023
©Tanta University, Faculty of Engineering
ISSN: 2356-9441 https://erjeng.journals.ekb.eg/ e ISSN: 2735-4873

Slice Admission Control based on Reinforcement Learning for 5G Networks

Mai Z. Ibrahim[1], Mohamed TALAAT FAHIM[2], Nada Elshennawy[3]

[1] Computer and Automatic Control Department, Faculty of Engineering, Tanta University – email: mai_zaky@f-eng.tanta.edu.eg
[2] Computer and Automatic Control Department, Faculty of Engineering, Tanta University – email: mohamed_ahmed1@f-eng.tanta.edu.eg
[3] Computer and Automatic Control Department, Faculty of Engineering, Tanta University– email: nada_elshennawy@f-eng.tanta.edu.eg

***Abstract*— Network slicing empowers service providers to deploy diverse network slice architectures within a shared physical infrastructure. This technology enables the provision of differentiated services that cater for specific Quality of Service (QoS) requirements of different use cases which need to be adequately supported in 5G networks. By leveraging Network Slicing, operators can effectively meet these diverse requirements and provide customized services to different tenants in a flexible and efficient manner. However, infrastructure providers face a challenging dilemma of the slice admission control regarding whether to accept or reject slice requests. From one perspective, they strive to optimize the utilization of network resources through accepting a significant number of network slices. From another perspective, the availability of network resources is restricted, and it is crucial to fulfil the QoS requirements specified by the network slices. In this research, an Admission Control (AC) Algorithm founded upon Reinforcement Learning mechanisms, specifically Q-Learning (QL), Double Q-Learning (Double-QL), and a proposed mechanism based on Double QL is obtained to overcome this challenge. This algorithm is applied in order to make informed decisions regarding network slice requests. The simulation results demonstrate that the AC algorithm, leveraging the suggested mechanism, surpasses the Double-QL and QL mechanisms in relation to gained profit with average of 8% and 26%, respectively. In case of the acceptance ratio of slice requests, it achieves average of 13% and 28% higher than Double-QL and QL mechanisms, respectively. Finally, it obtains the maximum resource utilization, surpassing Double-QL and QL by 9% and 20%, respectively.**

***Keywords*- 5G mobile networks, Admission control mechanisms, Reinforcement learning.**

I. INTRODUCTION

The 5G networks provide extensive support for various services, categorized into Ultra-Reliable Low-Latency Communications (URLLC), enhanced Mobile Broadband (eMBB), and massive Machine-Type Communications (mMTC), considering their diverse QoS demands (e.g., latency, bandwidth and reliability) as defined by The International Telecommunications Union (ITU) [1]. To meet the different needs of customers, 5G cellular networks leverage a diversity of technologies, including Software Defined Network (SDN), Network Function Virtualization (NFV) and Network Slicing.

Network slicing is regarded as a fundamental technology that guarantees the isolation of E2E virtual networks of 5G, which are known as network slices (NSL). These NSLs are customized to meet distinct QoS needs of various services sharing the same physical infrastructure.

Network slices have the potential to encompass sequences of RAN elements and core VNFs. In 5G C-RAN, the arrangement of VNFs should integrate both the service category and its corresponding Service-Level Agreements (SLAs). Infrastructure provider (InP) receives network slices requests (SLRs) in order to implement NSLs of different types of services [2].

The rollout of 5G mobile networks is expected to lead to a substantial growth in the volume of customers seeking diverse network services. Increasing the network users make extra expected slice requests to be ignored. Accepting more slice-requests (SLR) will raise the revenue of the InP's due to maximizing the usage of network resources. Nevertheless, it is difficult to serve all incoming requests due to the limited network resources.

Hence, it is necessary to implement an effective admission control mechanism (AC) in order to decide on the acceptance or rejection of NSLRs depending on the importance of the services, putting both the support of QoS needs and the accessibility of physical resources into consideration [3]. The decision of AC mechanism must be performed for each individual SLR in such a way as to maximize the revenue of InPs over the extended term. Since slice-request traffic is typically unpredictable in practice, it is necessary to have algorithms that are capable of predicting future SLRs traffic from data of the past traffic and build slice admission choices due to these predictions [4] [21].

This paper employs Reinforcement Learning (RL) to introduce three admission control mechanisms: QL, Double QL, and a proposed mechanism based on Double Q-Learning. The aim is to derive an appropriate decision for incoming slice requests.

Reinforcement Learning (RL) falls within the branches of artificial intelligence and has the ability to learn how to optimize a specified objective, even in the absence of a comprehensive system model [6] [20]. RL involves an agent interacting with the environment to discover a strategy that maximizes the anticipated cumulative reward by suggesting actions. The popular offline RL algorithm is Q-learning, which can be used to maximize the predicted reward for any Markov Decision Processes (MDPs) [7]. This mechanism relies on the

Journal of Engineering Research (ERJ)
Vol. 7 – No. 3, 2023
©Tanta University, Faculty of Engineering
ISSN: 2356-9441 https://erjeng.journals.ekb.eg/ e ISSN: 2735-4873

principles of Q-learning, formulated to explore the optimal action for each state with the aim of maximizing the total reward. Nonetheless, because Reinforcement Learning (RL) involves a step of maximizing estimated action values that tends to favor overestimating rather than underestimating these values, it occasionally learns unrealistically high action values. Owing to such large overestimations of action values, Q-learning may have a poor performance [8].

The Double Q-learning algorithm employs a double estimator technique to overcome the previous issue as it occasionally underestimates of the maximum anticipated action value instead of overestimating it [9]. The third mechanism is a proposed mechanism utilizing the principles of the Double Q-learning algorithm, incorporating additional parameters to calculate the two estimators employed within the framework of the Double Q-learning algorithm. In addition, instead of updating these estimators randomly, we evaluate the values of geometric means for the other Q-values and choose which estimator to update based on these values.

In this research, we investigate the evaluation of the Admission Control mechanism using these three algorithms (Q-learning, Double Q-learning and our proposed algorithm based on Double Q-learning) in order to make decisions on accepting or rejecting network slice requests and to demonstrate the performance of each algorithm. Therefore, we debate the design of the network model and the implementation of the three mentioned algorithms.

Our study makes the following primary contributions:

- First, applying an Admission Control mechanism to the SLRs from various tenants using three machine learning algorithms: Q-learning, Double Q-learning and our proposed algorithm, in order to achieve the aim of 5G networks in serving diverse services and enhance the overall network performance.
- Second, proposing an algorithm that defines the double estimators of the Double Q-learning algorithm in a more precise manner to ensure accurate action selection.
- Third, enhancing the overall profit for the InP and acceptance ratio of the SLRs as a consequence of applying the proposed algorithm for the AC mechanism.
- Finally, comparing the proposed algorithm with Q-learning and Double Q-learning algorithms in terms of gained profit, acceptance ratio and the resource utilization. The simulation outcomes illustrate the effectiveness of our suggested algorithm.

The subsequent sections of this paper are organized as follows. Section 2 presents the literature review of the previous work. Sections 3 describes the system model and outlines the Admission Control mechanism based on three algorithms. Section 4 shows the performance evaluation and the simulation results. Finally, Section 5 provides the paper's conclusion.

II. Literature Review

Authors in [2] concentrate on enhancing the revenue of the INP through the utilization of reinforcement learning techniques. They present an approach that encompasses two mechanisms: one for Admission Control; and the other for Resource Allocation. For Admission Control, they use RL and Deep-RL to optimize the profit of INPs, considering the 5G use cases delineated by the International Telecommunications Union (ITU). The Resource Allocation mechanism focuses on balancing the workload across network nodes and enhancing network resource utilization. By employing these mechanisms, the approach aims to enhance overall network performance and efficiency.

The objective of the research in [3] is to create and assess an algorithm that addresses handover decisions in 5G sliced networks. It focuses on the design of a self-optimizing Fuzzy Q-learning algorithm to make a decision for slice handover. Through this research, an AC method is represented by utilizing a supervised learning algorithm. To tackle the challenge of elevated complexity, the combination of fuzzy logic and Fuzzy Q-learning is utilized for discretizing the state and associated action spaces. This approach aims to mitigate complexity and enable effective decision-making in the handover process.

In [4], authors devised an AC mechanism coupled with a RAN slicing solution, employing a straightforward multi-tier network configuration. They further employed a multi-agent Deep Reinforcement Learning (DRL) technique known as MADRL. They demonstrate that the proposed MADRL solution outperformed its single-agent counterpart in terms of performance. However, a limitation of their approach is the utilization of multi-layer perceptron (MLP) models for both slicing and Admission Control (AC). This choice may restrict scalability and hinder the generalizability of the solution when applied to diverse network architectures.

An approach is presented in [5], focusing on an online and simplified AC policy derived through reinforcement learning. It seeks to maximize revenue for Infrastructure Providers while mitigating penalties caused by SLA violations (specifically, rejecting slice requests) in diverse network conditions. A noteworthy characteristic of this solution is its suitability in scenarios involving slice requests for diverse types of services (e.g., URLLC, mMTC, and eMBB) are concurrently issued over the same infrastructure. It considers three potential algorithms (QL, DQL, and Regret Matching) to compute optimal admission policies.

Authors in [16] present an AC mechanism that leverages Big Data Analysis to forecast the traffic, with the objective of augmenting the profit of the infrastructure providers. This mechanism selectively approves slice requests solely under the condition that no degradation in service is anticipated. A utility model and multi-service solution to network slicing are proposed in [17], founded on the principles of Queuing Theory, with the objective of maximizing network utility. This approach involves separate queues for only two specific types of requests and takes into account the presence of impatient customers. A policy-based AC mechanism designed specifically for allocating intraservice slices is outlined in [18], which operates

Journal of Engineering Research (ERJ)
Vol. 7 – No. 3, 2023
©Tanta University, Faculty of Engineering
ISSN: 2356-9441 https://erjeng.journals.ekb.eg/ e ISSN: 2735-4873

at adaptable timescales. A Machine Learning algorithm, specifically a neural network (NN), is used and trained to acquire the most effective admission policies on only (uRLLC) services. And at runtime, the NN is employed to deliver nearly optimal admission decisions in network conditions where no precomputed optimal policy exists.

Authors in [19] proposes an approach for online slice AC in 5G/B5G networks which prioritizes fairness considerations. It addresses the challenge of efficiently admitting slices while ensuring fair resource allocation among multiple network slices. This approach dynamically adjusts admission decisions based on the current network conditions and the fairness requirements of different slices.

In [21], authors introduce a novel approach to admission control, focusing on concurrent slices, and utilizing an infrastructure resource reservation methodology. It considers the dynamic characteristics of network slice requests while proficiently managing uncertainties in the resource demands of the slices. From the standpoint of an Infrastructure Provider (InP), the paper suggests reservation schemes intended to optimize the allocation of infrastructure resources to maximize the number of granted slices, while concurrently minimizing costs incurred by the Mobile Network Operators (MNOs). This entails addressing a max-min optimization problem that encompasses non-linear constraints and a non-linear cost function. Moreover, the approach guarantees resilience against demand uncertainties and mitigates the influence of reservations on the concurrent background services.

Table 1. Literature Review

Paper	Strategy used for Slice AC	Performance Metrics	Domain	Objective
[2]	Q-Learning and Deep Q-Learning	Profit-resource utilization-acceptance ratio	Core Network	Optimizing the profit of Network Slice Providers.
[3]	Self-optimization Fuzzy Qlearning	New Call Blocking Probability (NCBP) - Handoff Call Blocking Probability (HCBP)	Core Network	provide effective decision-making in the handover process.
[4]	Multi-agent Deep Reinforcement Learning (MADRL)	Long-term InP revenue.	RAN Network	Increase the performance of MADRL solution over its single-agent counterpart.
[5]	Q-Learning, Deep Q-Learning, and Regret Matching	InfProv reward - Percentage of rejection the slice request	RAN Network	Maximizing the InfProv revenue and their ability to learn offline or online.
[16]	Big Data Analysis	Profit	RAN Network	Augmenting the profit of the infrastructure providers
[17]	Queuing Theory	Profit-Network utility-admission rate and the average request waiting time	RAN Network	Maximizing network utility
[18]	Neural Network (NN)	Resource utilization-fairness to the service providers-network owners' revenue and complexity.	Core Network	Raise the revenue to network owners and SPs with QoS guarantees for services with strict latency constraints (e.g., uRLLC services)
[19]	Heuristic Algorithm called Prioritized Slice Admission Control	Priority - Fairness degree and Resource utilization.	RAN Network	Dynamically adjusts admission decisions based on the current network conditions and the fairness requirements of different slices.

From Table 1, we found that most literatures that achieves satisfied results were using reinforcement learning to overcome the challenge of slice admission control. Hence, our work is based on the reinforcement learning, especially Q-Learning, Double Q-Learning and a proposed mechanism based on Double Q-Learning.

III. System Model

5G mobile networks consist of core nodes and edge nodes. For the Control Plane, core nodes are adequate since this plane encompasses VNFs that require substantial bandwidth and processing capacities. For the User Plane, edge nodes are preferable since they include VNFs that need to be positioned in proximity to end-users [10]. NSL can comprise VNFs positioned on core nodes or edge nodes.

Each slice request contains information such as the type of the slice (e.g., eMBB, URLLC, and MIOT), the requested bandwidth of the virtual link, the required processing capacity and specific QOS needs as shown in Figure 1. Then, the AC mechanism accomplishes the admission of each request in order to accept or reject it. In this paper, URLLC use case is given higher priority than the other types due to its stringent latency requirements, necessitating the deployment of virtual networks at the edge nodes to meet these requirements.

Our system focuses on two primary participants: (i) the Infrastructure Provider (InP), who owns the network infrastructure and is responsible for generating network slices via supplying the necessary network resources on the slices; and (ii) the tenants who ask the InP to create a network slice to provide services to their clients.

Journal of Engineering Research (ERJ)
Vol. 7 – No. 3, 2023
©Tanta University, Faculty of Engineering
ISSN: 2356-9441 https://erjeng.journals.ekb.eg/ e ISSN: 2735-4873

Each network has a finite state space that represents all possible states that can be experienced. A state indicates to the available network resources (e.g., the processing capacity available for edge and core nodes, as well as the available bandwidth across links) after executing an action by the agent in the 5G core network [2]. Core nodes have more abundant and less expensive resources compared to edge nodes. Our goal is to accept more slice requests with higher priority and optimize resource utilization to the greatest degree possible.

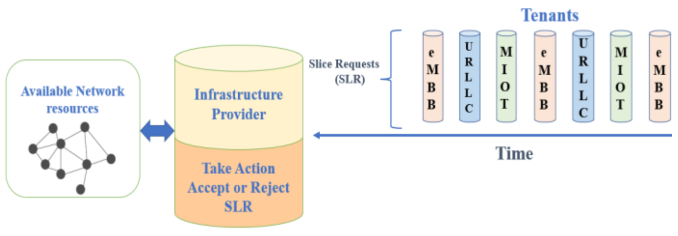


Figure 1. Admission Control System.

A. Admission control using QL

Q-learning belongs to the category of model-free reinforcement learning algorithms. A Q-learning agent chooses an action (a_t) for each state (s_t), receives a reward (r_t) for this action (e.g., the agent chooses the action that maximizes the profit), then turns to the next state (s_{t+1}). The aim of this algorithm is to recognize the most profitable SLRs. It assigns a quality value (Q-value) as in equation (1) for each action in a state and these Q-values will be stored in a lookup table called (Q-table). This process is referred to as exploring the undefined environment. In the initial state, Q-table is set to zero; then after each episode, it gets updated with the newly acquired Q-values [5].

$$Q_{t+1}(s_t\,,a_t) = Q_t(s_t\,,a_t) + \alpha\,.\,[r_t + \gamma\,.\,maxQ(s_{t+1},a_t) - Q_t(s_t\,,a_t)] \qquad (1)$$

Where:

$Q_{t+1}(s_t\,,a_t)$: The updated Q value of action a_t *in* state s_t.

$Q_t(s_t\,,a_t)$: The previous Q value of action a_t *in* state s_t.

α : The learning rate that ranges from 0 to 1. If $\alpha = 1$, the Q-agent ignores the old Q-values; and if $\alpha = 0$, the Q-agent learns anything new.

r_t : The reward or profit that be received from taking the action
a_t.

γ : The discount factor that is employed to find a trade-off between the immediate reward and the maximum anticipated future reward. The value of γ falls within the range of [0, 1].

$maxQ(s_{t+1}\,,a_t)$: The expectation of the optimal action in the future.

For each action taken, agent will implicitly accept SLRs with different types (e.g., URLLC, eMBB and MIoT) and consume network resources that have various costs. Therefore, the profit for each action taken will be calculated and is known as reward (r_t) and is given by:

$$r_t = \frac{\sum_{i=0}^{n} P(\,nsli\,)}{maxProfit\;(NSL\;,T)} \qquad (2)$$

where:
P(nsli) is the profit gained from accepting of the NSL.
maxProfit (NSL, T) is the utmost profit achievable by the network slice provider when utilizing all available network resources over a specific time period (T).
The QL-agent applies the epsilon-greedy approach [11] in order to explore an action for each step.
The QL-agent generates a random number *m* ϵ [0, 1]. If *m* > Ɛ, the QL-agent opts the action with the highest Q-value (exploiting a previously optimal action). Conversely, if *m* ≤ Ɛ, it picks a random action (exploring a new action) as follows:

$$a = \begin{cases} maxQ_t(s_t\,,a_t), & if\ m > \ Ɛ \\ random\ action, & otherwise \end{cases} \qquad (3)$$

After a sequence of episodes, the Q-table settles and serves as a guide for the agent to discern the optimal action based on the associated Q-values. If the requested resources are properly allocated for this action and also support QOS requirements, then the AC mechanism will accept the SLR. Otherwise, it will reject it as mentioned in Algorithm 1.

In some environments, the Q-learning algorithm has a poor performance caused by the massive overestimation of action values. It employs the highest action value as an estimator for the maximum expected action value. This denotes that the approximated Q values will nearly be strictly higher than the associated Q values from the actual action-value function.

Algorithm 1 Admission Control using Q-Learning

```
1: Initialize $Q$ , s
2: for episode → 1 to n do
3:     if all resources are available then the agent will observe
           the initial state ($s_t$)
4:     for next state ($s_{t+1}$) do
5:         The agent selects ($a_t$) from equation (3).
6:         for each SLR do
7:             check the available resources for the slice
8:             if SLR is mapped then
9:                 The agent accepts SLR.
10:            else
11:                The agent rejects SLR.
12:            end
13:        end
14:        The agent derives the reward ($r_t$) from equation (2).
15:        The agent also observes the subsequent state ($s_{t+1}$) //
               checks the current available resources.
16:        The Q-table will be updated using equation (1).
17:        The current state will be updated $s_{t+1} \rightarrow s_t$
18:    end
19: end
```

Doi: 10.21608/ERJENG.2023.228909.1209

Journal of Engineering Research (ERJ)
Vol. 7 – No. 3, 2023
©Tanta University, Faculty of Engineering
ISSN: 2356-9441 https://erjeng.journals.ekb.eg/ e ISSN: 2735-4873

B. Admission control using Double QL

Double Q-learning is an algorithm within the field of reinforcement learning [9] which employs double estimation in order to counteract the overestimation issues with the traditional Q-learning mechanism.

It retains two Q- tables (i.e., Q- estimators for the Q-functions): Q1 and Q2 from equations (4) and (5), respectively. One of the functions is utilized to estimate the maximum value of the Q-function and the second one is used for updating, then continuously exchanges the roles of these two estimators randomly. At each episode, for every action in state (s_t), one of the two Q function values is updated by employing the value of the other Q function estimated on the subsequent state [12].

$$Q1(s_t\, , a_t) = Q1(s_t\, , a_t) + \alpha * [r_t + \gamma * Q2\,(s_{t+1}, a^*) - Q1\,(s_t\, , a_t)] \tag{4}$$

where : $a^* = \arg max_a$ Q1($s_{t+1}\, , a_t$), which is referred to the maximum action value in the next state s_{t+1} according to the Q1 value.

$$Q2(s_t\, , a_t) = Q2(s_t\, , a_t) + \alpha * [r_t + \gamma * Q1(s_{t+1}, b^*) - Q2(s_t\, , a_t)] \tag{5}$$

where : $b^* = \arg max_a$ Q2($s_{t+1}\, , a_t$) , the maximum action value in the next state s_{t+1} according to the Q2 value.

Nevertheless, instead of using Q1 (s_{t+1},a^*) = $maxQ(s_{t+1}\, , a_t)$ in order to update Q1 in equation (4), as mentioned in Q-learning algorithm, the value Q2 (s_{t+1},a^*) will be used. As the value of Q2 was updated, albeits with different experience samples, it can be considered as unbiased estimate of this action's value. With reference to Q2, the same concept is used to update it, using b^* and Q1.

Although it is crucial for Q1and Q2 functions to learn from distinct sets of experiences, both value functions will be utilized when selecting the action to be performed. So, for each action, the average of both Q values is calculated, then the Ɛ -greedy strategy is performed for the resulting average of the Q values to select the action to execute [13].

The AC mechanism based on Double-QL algorithm will follow the same structure as described in Algorithm 1. However, the steps corresponding to Q-Learning will be replaced with the steps of Double Q-Learning steps as demonstrated in Algorithm 2.

C. Admission control using the proposed algorithm based on Double QL

To obtain the values of *Q1* and *Q2* as mentioned before in Double-QL algorithm, we also take into consideration the other Q-values in Q-Table1 and Q-Table2 by assigning the Root Mean Square error (RMS) [15] for (s_t) in each table. RMS is a standard statistical metric to determine the average for certain values. Accordingly, in our modified algorithm, the value of *Q1* will be obtained as follows:

Algorithm 2 Double Q-Learning

```
1: Initialize Q1, Q2, s
2: for episode → 1 to n do
3:     select a random number (m) ϵ [0,1]
4:     if m < Ɛ then
5:         choose a random action ($a_t$)
6:     else
7:         select the arg $max_a$ ($Q1(s_t) + Q2(s_t)$ )
8:     end
9:     Take action ($a_t$) then observe ($s_{t+1}$) and ($r_t$)
10:    Choose randomly either to UPDATE(Q1) or
       UPDATE(Q2) with probability of 0.5, respectively.
11:    if UPDATE(Q1) then
12:        Obtain $a^* = \arg max_a\ Q1(s_{t+1}\, , a_t)$
13:        $Q1(s_t\, , a_t)$=$Q1(s_t\, , a_t)+\alpha*\ [r_t+\gamma * Q2(s_{t+1},a^*) -$
               $Q1(s_t\, , a_t)]$
14:    else if UPDATE(Q2) then
15:        Obtain $b^* = \arg max_a\ Q2(s_{t+1}\, , a_t)$
16:        $Q2(s_t\, , a_t)$=$Q2(s_t\, , a_t)+\alpha*\ [r_t + \gamma * Q1(s_{t+1},b^*)-$
               $Q2(s_t\, , a_t)]$
17:      end
18:      The current state will be updated $s_{t+1} \rightarrow s_t$
19: end
```

$$Q1(s_t\, , a_t)=Q1(s_t\, , a_t) + \alpha * rms1 * [r_t + \gamma * Q2(s_{t+1},a^*) - Q1(s_t\, , a_t)] \tag{6}$$

Where: *rms1* is the root mean square error for all the Q-values of (n) actions for a certain state in Q-Table1 and can be calculated as:

$$rms1 = \sqrt{\frac{1}{n}\sum_{i=0}^{1=n} Q1\left(s_t\, , a_i\right)^2} \tag{7}$$

The value of *Q2* will be obtained as follows:

$$Q2(s_t\, , a_t)=Q2(s_t\, , a_t) + \alpha * rms2 * [r_t + \gamma * Q1(s_{t+1},a^*) - Q2\,(s_t\, , a_t)] \tag{8}$$

Where: *rms2* is the root mean square error for all the Q-values of (n) actions for a certain state in Q-Table2 and can be calculated as:

$$rms2 = \sqrt{\frac{1}{n}\sum_{i=0}^{1=n} Q2\,(s_t\, , a_i)^2} \tag{9}$$

Instead of randomly update the values of *Q1* and *Q2*, we consider the geometric mean [14] of the values in Q-Table1 and Q-Table2 for the state (s_t). Every state is associated with a specific set of actions, and each of these actions is characterized

Doi: 10.21608/ERJENG.2023.228909.1209

Journal of Engineering Research (ERJ)
Vol. 7 – No. 3, 2023
©Tanta University, Faculty of Engineering
ISSN: 2356-9441 https://erjeng.journals.ekb.eg/ e ISSN: 2735-4873

by two Q-values (e.g., $Q1$ stored in Q-Table1 and $Q2$ stored in Q-Table2). For a certain state (s_t), we calculate the geometric-mean for the Q- values of (n) actions in Q-Table1 (gmean1) using the below formula:

$$gmean1 = \sqrt[n]{Q1\,(s_t\,,a_0) * \,Q1\,(s_t\,,a_1) * \dots\dots\dots * \,Q1\,(s_t\,,a_n)} \tag{10}$$

Similarly, we calculate the geometric-mean for all the Q- values of n actions in Q-Table2 (gmean2).

$$gmean2 = \sqrt[n]{Q2\,(s_t\,,a_0) * \,Q2\,(s_t\,,a_1) * \dots\dots\dots * \,Q2\,(s_t\,,a_n)} \tag{11}$$

If gmean1 is higher than gmean2, then we will update $Q1(s_t\,,a_t)$ as in equation (6) and if it's less than or equal gmean2, then we will update $Q2(s_t\,,a_t)$ as in equation (8). In order to take an action, we calculate the RMS of the Q-values of Q1-table and Q2-table for actions in (s) as mentioned below:

$$rms_i = \sqrt{\frac{1}{2}\,(\,Q1\,(s\,,a_i)^2 + Q2\,(s\,,a_i)^2} \tag{12}$$

Afterwards, we will apply the Ɛ -greedy method for the resulting values of (rms_i)s to determine the action that will be performed. The AC mechanism based on this proposed algorithm will maintain the same structure outlined in Algorithm 1. However, the specific steps related to Q-Learning will be substituted with the steps outlined by the proposed algorithm introduced in Algorithm 3.

Algorithm 3 Proposed Modification on Double Q-Learning

```
1: Initialize $Q1$, $Q2$, $s$
2: for episode → 1 to n do
3:     select a random number ($m$) ϵ [0,1]
4:     if $m$ < Ɛ then
5:         choose a random action ($a_t$)
6:     else
7:         select the arg $\boldsymbol{max}_a\ (rms_i)$
8:     end
9:     Take action ($a_t$) then observe ($s_{t+1}$) and ($r_t$)
10:    Determine gmean1 and gmean2 from equations (10)
         and (11) respectively.
11:    if gmean1 > gmean2 then
12:        Obtain $a^*$ and rms1
13:        UPDATE($Q1$) using equation (6)
14:    else if UPDATE($Q2$) then
15:        Obtain $b^*$ and rms2
16:        UPDATE($Q1$) using equation (8)
17:    end
18:    The current state will be updated $s_{t+1} \rightarrow s_t$
19: end
```

IV. Performance Evaluation

This section provides an overview of the evaluation of our proposed algorithm, along with a comparison to the Q-learning and Double Q-learning algorithms to achieve the Admission Control mechanism. The metrics which are used for estimating the performance are presented. After that, the simulation variables that were used in the experiment are presented. Then, the simulation outcomes are acquired, and the best performance among the three mechanisms will be revealed.

A. Performance Metrics:

- **The profit**: is obtained from the acceptance of SLR, (P) is the income which an InP receives for constructing the network slice (i) minus the cost of bandwidth and processing for used resources

$$P = (revenue_i\ - cost_i) * T_i \tag{13}$$

where : T_i is the slice operational time.

- **The acceptance ratio**: is the ratio of accepted SLRs to the total number of SLRs.

$$Acceptance\ Ratio\ = \frac{accepted\ SLR}{Total\ no.of\ SLR} \tag{14}$$

- **The resource utilization**:

$$Resource\ utilization = \frac{\frac{\sum_i cpu(NSL_i)}{Total\ CPU} + \frac{\sum_i bw(NSL_i)}{Total\ BW}}{2} \tag{15}$$

Doi: 10.21608/ERJENG.2023.228909.1209

Journal of Engineering Research (ERJ)
Vol. 7 – No. 3, 2023
©Tanta University, Faculty of Engineering
ISSN: 2356-9441 https://erjeng.journals.ekb.eg/ e ISSN: 2735-4873

where: $\sum_i cpu(NSL_i)$ is the processing capacity used by all slice requests launched in the 5g core network and $\sum_i bw(NSL_i)$ is the bandwidth employed by all slice requests.

B. The Setup of the Experiment:

The simulation is developed using Python 3.8 executed on Intel Core i7-9750H CPU and RAM 16 GB. The network topology that are used in the experiments, 16-nodes topology which contains 12 edge nodes and 4 core nodes, with a 100, and 300 processing units, respectively. Moreover, each link has a bandwidth capacity of 100. The VNFs have computational demand of 5 processing units. The required virtual links bandwidth in the MIoT, URLLC, and eMBB graphs are 1, 2 and 3, respectively. The operational time used for the SLRs is 12 time units. The three types of SLRs have the same arrival rate. The arrival rate was varied from the range of 1 to 100 requests for each time unit to assess the performance of our proposed algorithm under different load scenarios. The simulation parameters and their respective values are detailed in Table 1.

Table 2. Simulation parameters

Parameter	Value
Topology	16 nodes
Bandwidth of Links	100
Processing (CPU) capacity of nodes	Edge nodes: 100, Core nodes: 300
Time window (time units)	2
Average operational time	12
Load (number of requests per time unit)	From 1 to 100
Learning rate (α)	0.9
Exploration factor ($\mathcal{E}$)	0.1
Discount factor (γ)	0.9
Number of Episodes	300
Number of state-action pairs	20 x 10^6 (10^6 states and 20 actions)

C. Results and Discussion:

Figure 1.a shows the acceptance ratio of the AC mechanism based on the three algorithms (Q-Learning, Double Q-Learning and our proposed algorithm)) in relation to the arrival rate of the slice requests. The proposed algorithm achieves acceptance ratios with average of 13% and 28% higher than produced by Double Q-Learning and Q-Learning, respectively. Due to the network's limited resources, all the algorithms exhibited low acceptance ratios in case of high arrival rate. In case of lower arrival rates (e.g., from 1 to 60 request per time unit), the proposed algorithm obtains values that somewhat exceed those provided by QL and Double-QL. This is because the proposed algorithm has a more accurate manner to obtain the Q-values that affect the actions to be handled. It is crucial to determine the type of the accepted slice requests; Figure 2.b presents the acceptance ratio of the proposed algorithm based on the types of SLRs. This figure ensures the fact that the proposed approach accepts a higher portion of URLLC slice requests compared to the other service types. URLLC use case has a higher priority than eMBB and MIoT use cases as it requires rigid latency requirements.

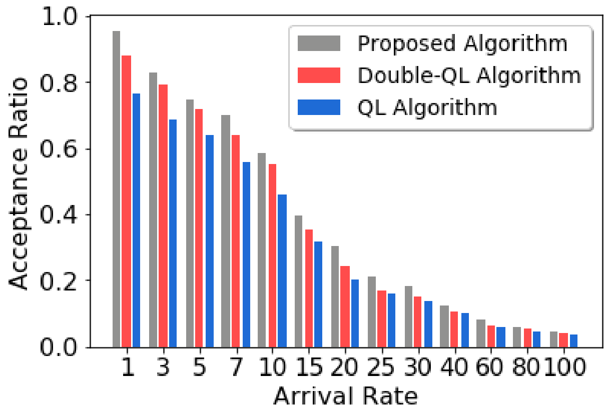


Figure 2. (a) Acceptance Ratio of the three algorithms vs Arrival Rate.

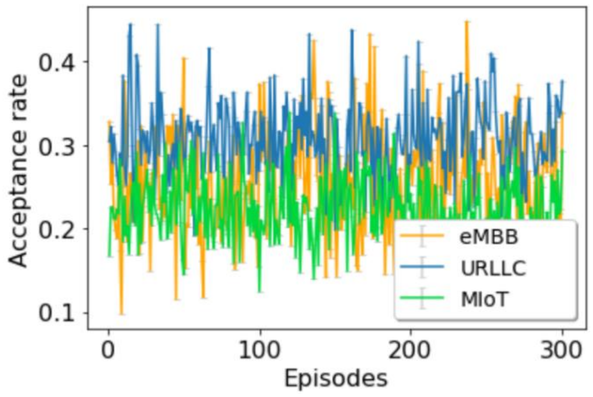
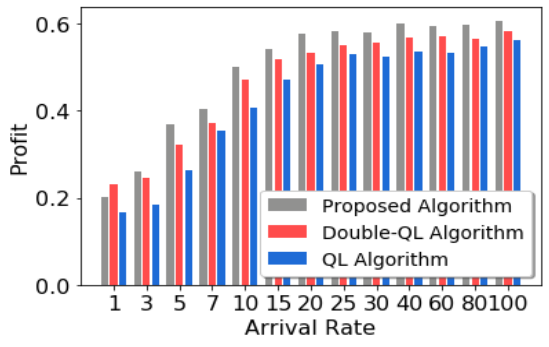


Figure 2. (b) Acceptance Rate of the Proposed algorithm per SLR Type.

Figure 3.a obtains the profit of the AC mechanism using the three algorithms as a function of the arrival rate. The proposed algorithm achieves superior performance compared to the other algorithms as it obtains the highest profit for almost all arrival rates even though with the highest ones. The profit of the proposed algorithm is greater than those achieved by Double Q-Learning and Q-Learning with average of 8% and 26%, respectively. This is because accepting more URLLC requests will increase the gained profit. The proposed approach learns to accept the appropriate portion of SLR type in order to improve profit. Figure 3.b shows how each type of accepted SLR contributed to the overall profit made by the proposed algorithm with an arrival rate 20 requests for each time unit. It is obvious from this figure that URLLC achieves higher profit than eMBB and MIoT.

Journal of Engineering Research (ERJ)
Vol. 7 – No. 3, 2023
©Tanta University, Faculty of Engineering
ISSN: 2356-9441 https://erjeng.journals.ekb.eg/ e ISSN: 2735-4873



.

Figure 3. (a) Profit of the three algorithms vs Arrival Rate.

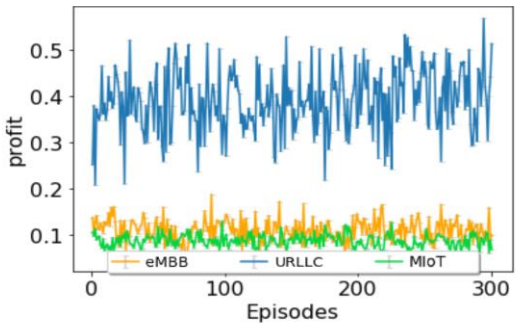


Figure 3. (b) Profit of the Proposed algorithm per NSLR Type.

Figure 4.a depicts the utilization of network resource with various arrival rates. When the load on the network increases, it leads to an increase in resource utilization. The proposed algorithm outperforms the other algorithms and this means that accepting more requests is essential for the purpose of promoting the profit. The proposed algorithm obtains the maximum resource utilization, that was 9% and 20% higher than those achieved by Double Q-Learning and Q-Learning, respectively.

The proposed algorithm uses the available resources in such a way to enhance the performance of the network. Figure 4.b presents the utilization of network resources resulted from using the proposed approach in edge nodes, core nodes, and links. The proposed algorithm learned to assign high priority to URLLC SLRs to increase the profit. URLLC use case utilizes more edge than core resources. Resources in core nodes are more numerous and less expensive than those for edge nodes. Consequently, the utilization in core nodes is higher than that in edge nodes in order to make more available edge nodes to serve URLLC services.

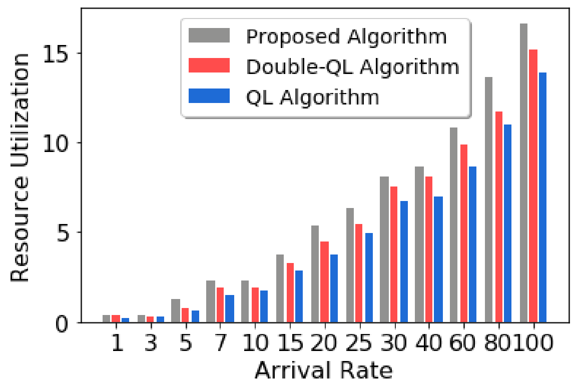


Figure 4. (a) Resource Utilization of the three Algorithms vs Arrival Rate.

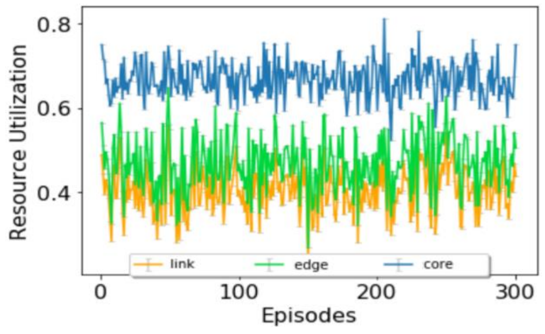


Figure 4. (b) Resource Utilization of the Proposed algorithm per Node Type.

Finally, a comparative analysis is conducted between our study and a prior piece of literature in [2] based on the information presented in Table 3. This comparison involves the proposed algorithm, Double QL, QL, Node Ranking algorithm (NR), the Always Admit Requests algorithm (AAR). The performance of all these algorithms is assessed under an arrival rate of 100.

Table 3. Comparative study with previous literature

	profit	Acceptance Ratio
Proposed Algorithm	0.5075	0.046
Double QL	0.4822	0.041
QL	0.4616	0.034
NR	0.4780	0.002
AAR	0.4415	0.002

V. Conclusion

In this context, the complexity of Admission Control for 5G slices requests encompassing URLLC, MIoT, and eMBB in 5G core network has been investigated. Numerous pieces of literature have successfully employed reinforcement learning to address the challenge of slice admission control, yielding satisfactory results. Hence, our research is grounded in reinforcement learning, specifically focusing on Q-Learning, Double Q-Learning, and a novel mechanism derived from Double Q-Learning. Unlike optimization-based approaches,

Doi: 10.21608/ERJENG.2023.228909.1209

Journal of Engineering Research (ERJ)
Vol. 7 – No. 3, 2023
©Tanta University, Faculty of Engineering
ISSN: 2356-9441 https://erjeng.journals.ekb.eg/ e ISSN: 2735-4873

these techniques are categorized as model-free, indicating that they operate without making assumptions about the underlying network structure.

The proposed algorithm obtains the Q-values in an appropriate manner which will enhance the decision of taking the most efficient action from the available state space. By applying this algorithm, the AC mechanism will achieve its aim of taking the most profitable slice requests within the specified time windows, as it prioritizes the URLLC use case with the strict latency demands and also increases the acceptance ratio of all types of the use cases concerning the available network resources. The simulation results validate the fact that the proposed algorithm outperforms QL and Double-QL algorithms in managing the network slices, in case of the gained profit, the acceptance ratio of slice requests, and the resource utilization.

Funding: The authors should mention if this research has received any type of funding.

Conflicts of Interest: The authors should explicitly declare if there is a conflict of interest.

REFERENCES

[1] M. Series, “IMT Vision–Framework and overall objectives of the future development of IMT for 2020 and beyond,” Recommendation ITU 2083.0, 2015.
[2] W. F. Villota-Jacome, O. M. C. Rendon, and N. L. da Fonseca, “Admission control for 5G core network slicing based on deep reinforcement learning,” IEEE Systems Journal 16.3, 2022, pp. 4686-4697.
[3] R. H. Puspita, J. Ali, and B. H. Roh, “An Intelligent Admission Control Scheme for Dynamic Slice Handover Policy in 5G Network Slicing,” Computers, Materials and Continua, CMC, vol.75, no.2, 2023, DOI: 10.32604.
[4] M. Sulaiman, A. Moayyedi, et al., “Multi-agent deep reinforcement learning for slicing and admission control in 5G C-RAN,” NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium, 2022.
[5] S. Bakri, B. Brik, and A. Ksentini, “On using reinforcement learning for network slice admission control in 5G: Offline vs. online,” International Journal of Communication Systems 34, no. 7, 2021, bv: e4757.
[6] D. Pandey and P. Pandey, “Approximate q-learning: An introduction,” in 2010 second international conference on machine learning and computing, pp. 317-320. IEEE, 2010.
[7] R. S. Sutton and A. G. Barto, “Reinforcement learning: An introduction,” MIT press, 2018.
[8] H. V. Hasselt, A. Guez, and D. Silver, “Deep reinforcement learning with double q-learning,” Proceedings of the AAAI conference on artificial intelligence. Vol. 30. No. 1, 2016.
[9] H. Hasselt, “Double Q-learning,” Advances in neural information processing systems 23, 2010.
[10] F. Giust, G. Verin, et al., “Mec deployments in 4g and evolution towards 5g,” ETSI White Paper, vol. 24, pp. 1–24, 2018.
[11] W. Dabney, G. Ostrovski, and A. Barreto, “Temporally-extended {\epsilon}-greedy exploration,” arXiv preprint arXiv:2006.01782, 2020.
[12] H. Xiong, L. Zhao, et al., “Finite-time analysis for double Q-learning,” Advances in neural information processing systems 33, pp. 16628-16638, 2020.
[13] Z. Zhang, Z. Pan, and M. J. Kochenderfer, “Weighted double Q-learning,” In IJCAI, pp. 3455-3461, 2017.
[14] B. Jeuris, R. Vandebril, and B. Vandereycken, “A survey and comparison of contemporary algorithms for computing the matrix geometric mean,” Electronic Transactions on Numerical Analysis 39, no. ARTICLE, pp. 379-402, 2012.
[15] T. Chai, and R. R. Draxler, “Root mean square error (RMSE) or mean absolute error (MAE),” Geoscientific model development discussions 7, no. 1, pp. 1525-1534, 2014.
[16] M. R. Raza, A. Rostami, et al., “A slice admission policy based on big data analytics for multi-tenant 5g networks,” Journal of Lightwave Technology, vol. 37, no. 7, pp. 1690–1697, 2019.
[17] B. Han, V. Sciancalepore, et al., “Multiservice-based network slicing orchestration with impatient tenants,” IEEE Trans. on Wireless Communications, pp. 1–1, 2020.
[18] M. Vincenzi, E. Lopez-Aguilera, and E. Garcia-Villegas. “Timely admission control for network slicing in 5G with machine learning,” IEEE access 9, pp.: 127595-127610, 2021.
[19] M. Dai, L. Luo, et al., “PSACCF: Prioritized Online Slice Admission Control Considering Fairness in 5G/B5G Networks,” IEEE Transactions on Network Science and Engineering 9, no. 6, pp. 4101-4114, 2022.
[20] H. P. Phyu, D. Naboulsi and R. Stanica, “Machine Learning in Network Slicing—A Survey,” in IEEE Access, vol. 11, pp. 39123-39153, 2023, doi: 10.1109/ACCESS.2023.3267985.
[21] Q. -T. Luu, S. Kerboeuf and M. Kieffer, “Admission Control and Resource Reservation for Prioritized Slice Requests with Guaranteed SLA Under Uncertainties,” in IEEE Transactions on Network and Service Management, vol. 19, no. 3, pp. 3136-3153, Sept. 2022, doi: 10.1109/TNSM.2022.3160352.