Hindawi
Journal of Computer Networks and Communications
Volume 2024, Article ID 9571987, 13 pages
https://doi.org/10.1155/2024/9571987

Research Article

Development of an AI-Enabled Q-Agent for Making Data Offloading Decisions in a Multi-RAT Wireless Network

Murk Marvi,[1] **Adnan Aijaz**,[2] **Anam Qureshi**,[3] **and Muhammad Khurram**[4]

[1]*Department of Computer Science and Information Technology, NED University of Engineering and Technology, Karachi, Pakistan*
[2]*Bristol Research and Innovation Laboratory, Toshiba Europe Ltd., Bristol, UK*
[3]*Department of Computer Science, National University of Computer and Emerging Sciences, Karachi, Pakistan*
[4]*Department of Computer and Information Systems Engineering, NED University of Engineering and Technology, Karachi, Pakistan*

Correspondence should be addressed to Murk Marvi; marvi@cloud.neduet.edu.pk

Received 25 September 2023; Revised 23 December 2023; Accepted 8 January 2024; Published 24 January 2024

Academic Editor: Wanli Wen

Data offloading is considered as a potential candidate for alleviating congestion on wireless networks and for improving user experience. However, due to the stochastic nature of the wireless networks, it is important to take optimal actions under different conditions such that the user experience is enhanced and congestion on heavy-loaded radio access technologies (RATs) is reduced by offloading data through lower loaded RATs. Since artificial intelligence (AI)-based techniques can learn optimal actions and adapt to different conditions, in this work, we develop an AI-enabled Q-agent for making data offloading decisions in a multi-RAT wireless network. We employ a model-free Q-learning algorithm for training of the Q-agent. We use stochastic geometry as a tool for estimating the average data rate offered by the network in a given region by considering the effect of interference. We use the Markov process for modeling users' mobility, that is, estimating the probability that a user is currently located in a region given its previous location. The user equipment (UE) plays the role of a Q-agent responsible for taking sequence of actions such that the long-term discounted cost for using network service is minimized. Q-agent performance has been evaluated and compared with the existing data offloading policies. The results suggest that the existing policies offer the best performance under specific situations. However, the Q-agent has learned to take near-optimal actions under different conditions. Thus, the Q-agent offers performance which is close to the best under different conditions.

1. Introduction

The remarkable surge in mobile data usage resulting from technological advancements in various domains like Internet-of-Things, online gaming, social networks, augmented/virtual reality, and more has significantly amplified the burden on wireless networks and introduced a wide variety of applications with varying characteristics. In order to meet such intensive and highly heterogeneous demands, the 5th generation (5G) wireless networks are expected to be ultradense, autonomous, or intelligent, operate on multiple higher frequency bands, and with multiple radio access technologies (multi-RATs). The deployment of ultradense networks increases the spatial reuse factor of the resources. The use of multiple RATs and multiple bands enables effective use of spectral bands and resources of the wireless network. Moreover, it also helps in alleviating congestion on cellular RAT by offloading data to underloaded RATs like wireless fidelity (Wi-Fi) [1].The concept of offloading across multiple RATs extends beyond data transfer; it has also been harnessed for the sharing of computational [2] and power resources [3].

The 5G networks are expected to be intelligent to meet the quality-of-service (QoS) requirements imposed by the users with heterogeneous demands. That is why, artificial intelligence (AI) has an important role to play in the

development of intelligent and autonomous algorithms for 5G networks. The efficient allocation and utilization of resources among users with different demands in a 5G multi-RAT wireless network is a challenging problem. The network densification further complicates this problem by introducing interference and coverage issues, often manifesting in terms of degraded network performance and user experience. Therefore, it becomes challenging to satisfy the QoS requirements imposed by the user. Various algorithms have been presented in the literature, as discussed in Section 2, for alleviating congestion on wireless networks and for efficient utilization of the resources by employing the concept of data offloading and AI techniques. However, some of these have been developed without taking into account the effect of interference caused due to the deployment of ultradense wireless networks [1], and others have been developed without using AI techniques.

Therefore, the main objective of this work is to develop an AI-enabled Q-agent for making data offloading decisions in a multi-RAT wireless network by taking into account the effect of interference caused by an ultradense network and by making use of a model-free reinforcement learning (RL) technique. Here, the term "model-free" means that an accurate or approximate mathematical representation of the system under consideration is not required; thus, the AI agent can learn through trial-and-error approach. A high-level diagram of the proposed work is shown in Figure 1. A multi-RAT wireless network scenario has been assumed which includes a cellular and a Wi-Fi RAT. The coverage provided by both the RATs has been divided into different zones based on the signal-to-interference (SIR) experienced by the user in a given region. Initially, the user is under coverage of cellular RAT only, and it has generated a request for downloading of a file. We use the Q-learning algorithm which is a model-free RL algorithm for training of the agent; that is why, we have named it as a Q-agent. It is responsible for taking a sequence of actions by observing states such that the long-term discounted cost of using a network service is minimized.

We formulate the problem of data offloading by assuming that the user equipment (UE) plays the role of the Q-agent as depicted in Figure 1. The users' request for downloading a file of certain size in a given duration and at a given location is used for defining states. The Q-agent can take three actions, that is, download data through cellular RAT, offload data through Wi-Fi RAT, and remain idle. We define a cost function for using network services which is a function of the actions taken by the Q-agent. We also define a penalty function for missing the defined delay limit. The task of the Q-learning algorithm is to minimize this cost function by learning an optimal data offloading policy that takes best action in a given state. We have evaluated the performance of the developed Q-agent for making data offloading decisions and also compared the results with existing model-based analytical offloading approach [4] and some other standard data offloading policies. In the end, we have also discussed about the issues and challenges imposed by the Q-learning algorithm and how we can tackle these for developing efficient agents that offer near to optimal performance. Advanced AI techniques based on deep learning such as the deep Q-network (DQN) or double DQN can also be adopted for tackling the issues imposed by the Q-learning algorithm. However, the data offloading problem considered in this work is user-centric. Therefore, its state and action space is small, and a simple algorithm such as Q-learning, with slight modifications, offers satisfactory performance. Moreover, according to the authors in [5], advanced AI techniques are not yet been considered for deployment in wireless networks because of the resource-constrained nature of these networks.

The rest of the paper has been organized as follows. The details of the related are covered in Section 2. The problem formulation and the assumptions are discussed in Section 3. The development of AI-enabled Q-agent is discussed in Section 4. Performance evaluation of the developed Q-agent and comparison with existing data offloading policies is in Section 5. Finally, Section 6 concludes the paper.

2. Related Work

In this section, we provide details of the studies presented in the existing literature and are related to this work in one or another way. A concise summary of these studies and our work is presented in Table 1.

A dynamic offloading algorithm has been presented in [11] for UEs by assuming a multi-RAT wireless network. An autonomous resource allocation policy has also been developed for multiaccess edge servers. The authors used a penalty-based genetic algorithm for learning offloading decisions and deep Q-neural network (DQN) for efficient allocation of resources by the edge servers. The authors assumed multiple RATs which include cellular and Wi-Fi RATs by defining different frequency bands. However, they did not take into account the effect of the channel-accessing scheme employed by Wi-Fi RAT which is different from cellular RAT. Moreover, the authors also did not consider the effect of interference due to ultradense deployment of base stations (BSs) while modeling the data rate experienced by a user. In another paper [8], a near to optimal policy for users' association in a heterogeneous wireless network has been obtained by employing DQN. Instead of assuming a multi-RAT wireless network scenario, they assumed a dual connectivity scenario wherein a user can associate with the BSs of different tiers under the same network, that is, macro- and micro-BSs. The authors selected DQN over SARSA or Q-learning algorithms because they were optimizing a network-centric user association policy which had large state and action spaces. Since in this work we propose a user-centric data offloading approach wherein each UE is responsible to minimize the cost of using a network service by making automated optimal data offloading decisions, the state and action spaces are small. That is why, we selected the Q-learning algorithm instead of DQN.

A multiagent RL-based algorithm for RAT access in a multi-RAT wireless network has been proposed in [9]. The authors assumed one cellular and one Wi-Fi RAT operating in different bands and with different channel accessing techniques. According to the authors, their proposed

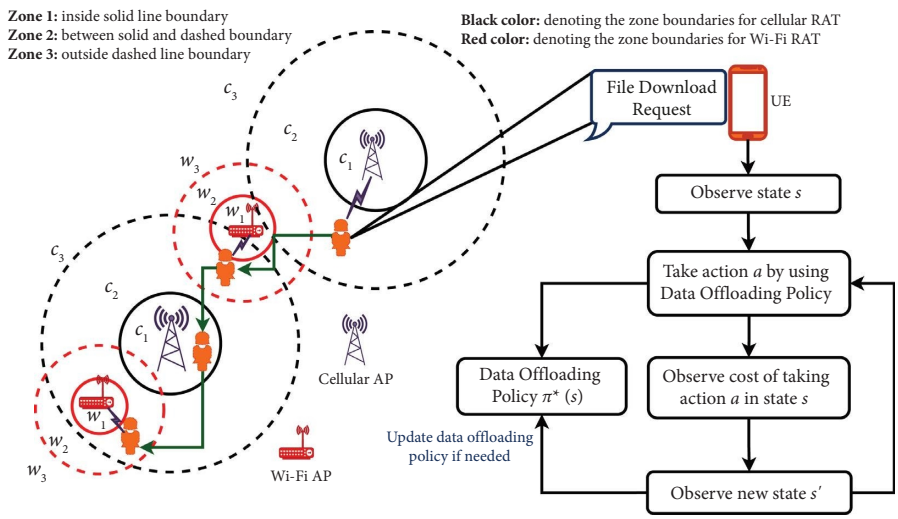


Figure 1: An illustration of the multi-RAT wireless network which includes cellular and Wi-Fi RATs and UE playing the role of Q-agent for taking sequence of optimal actions under different situations.

approach for RAT access offers better performance compared to the traditional data offloading schemes. However, the system model assumed by the authors did not incorporate the effect of dense deployment of the access points (APs) and the resulting interference, which highly impacts the data rate experienced by the users. The authors in [12] presented an incentive-based contract-theoretic approach to motivate the third-party operators, like Wi-Fi operators, to share their resources during peak time to overloaded cellular RAT. However, the main focus of this work was to propose optimal contracts to third-party operators such that they agree to accept the offloaded requests while the profit of the mobile network operators is maximized.

A delay aware offloading and network selection optimization algorithm has been proposed in [6] by assuming that unlimited cellular RAT coverage and limited Wi-Fi RAT coverage are available for the users. For solving the optimization problem, the authors used the backward induction algorithm which is computationally expensive. In another study [7], the authors proposed a data offloading approach by dividing the network coverage into various zones where each zone offers a different data rate to a user. They considered signal-to-noise ratio (SNR) for estimating the data rate offered to a user in a zone. However, for highly dense and heterogeneous wireless networks, SIR is considered as a better metric for estimating the coverage and data rate experienced by a user [13]. To better capture the coverage and data rate offered by Wi-Fi RAT, while estimating the data offloading gains, the authors in [10] employed stochastic geometry (SG) modeling techniques. However, this work is limited to estimation of data offloading gains that can be provided by a Wi-Fi RAT.

The authors in [4] proposed an automated data offloading framework by assuming a multi-RAT wireless network scenario and developed a *model-based* data offloading policy. Unlike [7], they divided the coverage provided by each RAT into different zones by using SIR and SG modeling techniques. They adopted a *model-based* RL approach for obtaining an optimal data offloading policy. A model-based RL algorithm requires a transition matrix which depicts the complete stochastic nature of the assumed environment. The authors in [4] utilized SG and Markov decision process (MDP) for modeling stochastic nature of the assumed wireless network and for obtaining the corresponding transition matrix so that it can be used by the model-based RL algorithm. However, due to various random factors in spatial and temporal domains of a wireless network, the traffic characteristics, load, and various other parameters are prone to change. Thus, at any point in time, the practical scenario may deviate widely from the transition matrix derived for a specific scenario. This problem initiates the need for the design of *model-free* data offloading policies which can learn the network or user behavior in real time and accordingly take the optimal actions. Nevertheless, such approaches pose various challenges and issues when it comes to their convergence and implementation in practical scenarios due to their trial-and-error-based learning approach.

3. Network Model and Problem Formulation

In this section, we provide details about the wireless network scenario assumed in this work. Moreover, we also formulate the data offloading problem by defining the Markov decision process (MDP) which includes details regarding Q-agent and its environment, that is, the set of states, the set of actions, cost, and penalty functions.

3.1. Multi-RAT Wireless Network Model. Similar to [4, 14], we make use of SG modeling techniques for simulating a multi-RAT wireless network which includes a cellular and a Wi-Fi RAT. We assume that each RAT is under the control of the same operator [15]. We adopt homogeneous Poisson point processes (HPPPs) Φ_c and Φ_w, with intensity λ_c and λ_w, for drawing the locations of APs under cellular and Wi-Fi RATs, respectively. The users are assumed to be

Table 1: A summarized comparative analysis of the existing approaches with the approach presented in this work.

Approaches	Multi-RAT	Dense network	Interference	Model-free RL
A delay aware offloading and network selection using the backward induction algorithm [6]	✓	✗	✗	✗
A data offloading approach for a multi-RAT network using the Q-learning algorithm [7]	✓	✗	✗	✓
A user association approach in a heterogeneous wireless network using DQN [8]	✗	✗	✓	✓
RAT access in a multi-RAT network using a multiagent RL algorithm [9]	✓	✗	✗	✓
Estimation of offloading gains provided by Wi-Fi RAT by using stochastic geometry [10]	✗	✓	✓	✗
Dynamic offloading and resource allocation in multiaccess edge servers using the genetic algorithm and DQN [11]	✓	✓	✗	✓
An incentive based contract-theoretic approach for third-party operators [12]	✓	✓	✓	✗
A model-based data offloading approach for a multi-RAT wireless network [4]	✓	✓	✓	✗
A model-free AI-enabled approach, presented in this work, for making the data offloading decision in a multi-RAT wireless network	✓	✓	✓	✓

distributed according to another HPPP Φ_u, with intensity λ_u. We assume that all APs belonging to $r \in \{c, w\}$ RAT operate at the same power level $\mathscr{P}_r$ over the entire bandwidth $\mathscr{B}_r$. Furthermore, we assume a saturated downlink channel wherein the same resources are shared by all the APs of cellular RAT, and a single channel is shared by all the APs of Wi-Fi RAT. As a result, the signal-to-interference ratio experienced by a typical user under RAT r can be approximated by using the following equation:

$$\mathrm{SIR}_r = \frac{\varsigma_{y_o}/l\left(\left\|d_{y_o}\right\|\right)}{\sum_{y\in\Phi_r/y_o}\hat{e}_y\varsigma_y/l\left(\left\|d_y\right\|\right)}, \tag{1}$$

where $l(\|d\|)$ denotes a free space path lass model, ς_{y_o} and ς_y denote small-scale fading from the tagged and other BSs, respectively, and $\hat{e}_y$ is a medium access indicator function which represents if an AP of RAT r, located at y, is active or not. For an AP under cellular RAT $(r = c)$, the indicator function is unity because all the APs are assumed to transmit simultaneously. For an AP under Wi-Fi RAT $(r = w)$, it can be either zero or unity because not all the APs are allowed to transmit simultaneously due to the contention-based nature of carrier sense multiple access with collision avoidance (CSMA/CA) channel accessing scheme [16]. The probability that the network offers a data rate to the user which is greater than a threshold ρ_r can be defined as

$$\varpi_r = \mathbb{P}\left(\mathscr{C}_r > \rho_r\right), \tag{2}$$

where

$$\mathscr{C}_r = \frac{\mathscr{B}_r\widehat{\mathscr{P}}_r}{\overline{\mathscr{N}}_r}\log\left(1 + \mathrm{SIR}_r\right), \tag{3}$$

$\overline{\mathscr{N}}_r = \lambda_u/\lambda_r$ is the average load per AP and $\widehat{\mathscr{P}}_r$ is the medium access probability (MAP) for an AP. Based on the data rate offered to a user under a RAT, we divide the given region under each RAT into three zones as depicted in Figure 1. The first zone offers the maximum data rate, the second zone offers the minimum data rate, and the third zone is like an outage for a user. The probability that a user is located in zone z of RAT r has been defined in [4], and it is given as

$$\mathbb{P}\left(r_z\right) = \begin{cases} \varpi_{r_z}, & z = 1, \\ \varpi_{r_z} - \varpi_{r_{z-1}}, & z = 2, \\ 1 - \varpi_{r_{z-1}}, & z = 3, \end{cases} \tag{4}$$

where

$$\varpi_{r_z} = \mathbb{P}\left(\mathrm{SIR}_r > \tau_{r_z}\right), \tag{5}$$

$\tau_{r_z} = 2^{(\rho_{r_z}\overline{\mathscr{N}}_r/\widehat{\mathscr{P}}_r\mathscr{B}_r)-1}$, ρ_{r_z} is the data rate threshold. Here, (5) is obtained after substituting (3) in (2) and rearranging.

The users can move in a given region with possible locations denoted by the set $\mathscr{K} = \cup_{i\in\{1,2,3\}}\cup_{j\in\{1,2,3\}}(c_i, w_j)$, by following a widely used Markovian model [6]. The probability that a user moves to location $k' = (c_{i'}, w_{j'})$ in the next time slot given the current location as $k = (c_i, w_j)$ can be defined as follows:

$$\mathbb{P}\left(k' = c_{i'}, w_{j'}\middle|k = c_i, w_j\right) = \mathbb{P}\left(c_{i'}\middle|c_i\right)\mathbb{P}\left(w_{j'}\middle|w_j\right), \tag{6}$$

where

$$\mathbb{P}\left(r_{z'}\middle|r_z\right) = \begin{cases} \beta_r\mathbb{P}\left(r_z\right), & z' = z, \\ 1 - \beta_r\mathbb{P}\left(r_z\right), & z' \neq z, z' = 2, \\ \dfrac{1 - \beta_r\mathbb{P}\left(r_z\right)}{2}, & z' \neq z, z = 2, \end{cases} \tag{7}$$

$r \in \{c, w\}, z \in \{1, 2, 3\}, \beta_r$ is the scaling factor capturing the speed of mobility, and $\mathbb{P}(r_z)$ is defined in (4). For readability and clarity, we have included details of the assumed network scenario in this section. For details of the derivation of these equations which are obtained by using SG modeling techniques, please see [4, 6, 14].

3.2. Markov Decision Process Formulation. An MDP is a discrete stochastic process which is used for sequential decision-making. It is defined by a tuple $(\mathscr{S}, \mathscr{A}, \mathbb{P}(s'|s, a), \Omega, \alpha)$, where $\mathscr{S}$ is the state space, $\mathscr{A}$ is the action space, $\mathbb{P}(s'|s, a)$ is the state transition probability, Ω is the cost function, and α is the discount factor. Since we employ a model-free RL algorithm, the transition probability matrix is not required for the problem formulation. The rest of the components have been defined in the following subsections.

3.2.1. Q-Learning Agent: States and Actions. The UE has been defined as a Q-agent in this work which is responsible for taking sequence of actions after observing states. Assume that a user generates a request to download ψ bits of data within $\mathscr{D}$ units of time. Here, the users' request is expressed in terms of a tuple $\mu_o(\psi, \mathscr{D})$. We suppose that the time axis is divided into slots $t \in \mathscr{T} = \{1, 2, \ldots, \mathscr{D}\}$ of fixed length, and the Q-agent is required to take action at each time epoch. It is assumed that the duration of a time slot is so small such that the state of the system does not change. The state of the user, $s \in \mathscr{S}$, at a time slot t, has been defined as $s_t = (k, h, d)$, where $h \in \psi$ represents the remaining file size in bits, $d \in \mathscr{D}$ denotes the remaining time, and $k = (c_i, w_j) \in \mathscr{K}$ denotes the location of the user specified by available zones of cellular and Wi-Fi RATs, respectively. As we assume stationarity, for simplicity, the notation t is omitted from this point onward.

Three possible actions $a \in \mathscr{A}$ are available for the Q-agent to make: remain idle $(a = 0)$, download data through cellular RAT $(a = 1)$, and offload data through Wi-Fi RAT $(a = 2)$. However, in any state s, with a given location k and $\forall h, d$, the number of permitted actions $a \in \widehat{\mathscr{A}}(s)$ is at most two as defined in the following:

$$\widehat{\mathscr{A}}(s) = \begin{cases} \{2\}, & \rho_{c_i} < \rho_{w_j}, \\ \{1,\ 2\}, & \rho_{c_i} > \rho_{w_j} > 0, \\ \{0,\ 1\}, & \rho_{c_i} > 0, \rho_{w_j} = 0, \\ \{0\}, & \rho_{c_i} = 0, \rho_{w_j} = 0. \end{cases} \tag{8}$$

Equation (8) refers to the decision of offloading through Wi-Fi RAT if the data rate supported by cellular RAT is smaller than the Wi-Fi RAT. Equation (8) refers to the permitted actions when the data rate supported by cellular RAT is greater than the Wi-Fi RAT. Equation (8) refers to the permitted actions when Wi-Fi RAT is not available. Equation (8) refers to the action when none of the RATs are available.

3.2.2. Feedback from Environment: Cost and Penalty. Similar to [6], we assume that the cost for using cellular RAT for data download is higher than the Wi-Fi RAT. It means, by making an offloading decision to Wi-Fi RAT, the Q-agent can minimize the cost of data usage for the user. Moreover, while waiting for the availability of Wi-Fi RAT, the deadline limit associated with the generated request cannot be ignored which results, in the end, a huge penalty if exceeded. Thus, through the offloading process, the Q-agent is required to minimize the overall cost of downloading the file while maintaining the given QoS requirement.

We adopt a usage-based cost scheme, where a user is charged proportional to its data usage. Let $\varphi(a)$ represents the cost for downloading per unit of data by choosing action a. Let us assume that $\rho(k,a)$ denotes the average supported data rate in bits per second at location k when action a is chosen. Thus, the total cost during a time slot, when the Q-agent chooses action a in state s such that the delay timer is not expired, is given by

$$\Omega(s,a) = \min\{h, 60 * \rho(k,a)\}\varphi(a), \quad d > 0. \tag{9}$$

The penalty for the Q-agent when it is not able to complete the download within $\mathscr{D}$ units of time has been defined as follows:

$$\Omega(s, \forall a) = \Upsilon(h), \quad d = 0, \tag{10}$$

where $\Upsilon(h)$ is a nondecreasing function of h [6]. Thus, the objective function can be defined as

$$\min_a \left\{ \sum_{0 \le d \le \mathscr{D}} \Omega(s,a) \right\}, \tag{11}$$

which implies that the Q-agent is responsible to choose sequence of actions such that the accumulated cost of using a network service is minimized.

4. Development of a Q-Learning Agent for Data Offloading

Q-learning is a model-free RL algorithm in which an agent interacts with its environment and tries to learn optimal actions for given states through the trial-and-error approach. The quality of an action taken in a given state by the agent is recorded by defining a quality function, which is denoted by $Q_\pi(s,a)$. It denotes the expected long-term discounted reward of taking action a in state s by using policy π. In this work, the UE plays the role of an agent, named as Q-agent, and the Q value is defined as the expected long-term discounted cost for taking action a in state s by using policy π. Thus, here the aim of the agent is to find the best policy $\pi^*(s)$, that minimizes this quality function for each (s,a) pair, by choosing the optimal action in a given state, i.e., $\pi^*(s) = \text{argmin}_a Q(s,a)$.

Assume that at the current time epoch, the agent observes state s and takes action a. As a result, it receives the cost $\Omega(s,a)$ from the environment for taking action a in state s, and it ends in state s' at the next time epoch. Thus, the Q value for (s,a) pair can be defined as follows:

$$Q(s,a) \longleftarrow Q(s,a) + \gamma\left(\Omega(s,a) + \alpha \min_{a'}\left\{Q\left(s', a'\right)\right\} - Q(s,a)\right), \tag{12}$$

where α is the discount factor and γ is the learning rate, and it is defined in [7] as

$$\gamma(s,a) = \left(\sqrt{\beta(s,a) + 3}\right)^{-1}, \tag{13}$$

where $\beta(s,a)$ denotes the number of times action a is taken in state s. It has been proved that while $\beta(s,a)$ is sufficiently large and γ is reduced to zero over time, $Q(s,a)$ is guaranteed to converge to $Q_{\pi^*}(s,a)$ [17]. The algorithm for training of the Q-agent has been defined in Algorithm 1, and its details are discussed in the following:

4.1. Initialization of Q Values. Poor initialization of Q values, like $Q(s,a) \longleftarrow 0, Q(s,a) \longleftarrow 1$ or $Q(s,a) \longleftarrow$ uniformly distributed $\forall(s,a)$ pairs, can badly affect the overall learning curve of the Q-agent and convergence speed of the Q-learning algorithm. However, initialization based on context of the problem can greatly help in speeding up the learning process. Therefore, in this work, as clear from Algorithm 1, we initialize the Q values by exploiting a single back-up sweep and using uniform random policy. Since at the end of each episode, the Q-agent is supposed to observe a penalty if the deadline is missed, and the single back-up sweep spreads the influence of penalty throughout the Q values for all (s,a) pairs, which overall improves the learning process.

4.2. Q-Learning Algorithm without Employing ϵ−Greedy Approach. In the Q-learning algorithm, at decision epochs, the agent decides randomly or based on previously learned Q values, which action should be taken in a given state. For minimizing cost, the agent may take low-cost actions it has tried in the past. This is known as the exploitation mode. The agent also needs to try actions it has not taken before, which may play a role in further minimization of the accumulated cost. Therefore, the agent may take one of the actions randomly from the set of available actions, to enhance its future decisions. This is known as the exploration mode.

Initialization
$\pi(a|s)$ as a random uniform policy
$Q(s,a) \longleftarrow \Omega(s,a) + \sum_{\forall a_i} \sum_{\forall s'} \pi(a_i|s) Q(s', a_i)$
$\beta(s,a) \longleftarrow 0 \quad \forall s \in \mathcal{S}, a \in \mathcal{A}$
for each download request $\mu_o(\psi, \mathcal{D})$ - episode **do**
  define state $s(k,h,d)$ - $d = \mathcal{D}, h = \psi, k$ randomly generated using (6)
  while download is not complete $(h > 0)$ **do**
    if $Q(s, \forall a)$ is same **then**
      choose action a at random
    else
      choose $a = \text{argmin}_a Q(s,a)$
    end if
    take action a
    update $\gamma(s,a)$ by using (13)
    if $d > 0$ **then**
      obtain $\Omega(s,a)$ using (9)
      obtain $s'(k', h', d')$ - $d' = d - 1, h' = h - \min\{h, 60 * \rho(k,a)\}, k'$ randomly generated using (6)
      update $Q(s,a)$ by using (12)
      $s \longleftarrow s'$
    else
      obtain $\Omega(s,a)$ using (10)
      update $Q(s,a) = \Omega(s,a)$
      break
    end if
  end While
end for

Algorithm 1: Training of Q-agent.

Since Q-learning is a model-free iterative learning algorithm, it is important that exploration and exploitation should be simultaneously performed. The agent must observe the effect of taking different actions in a given state and progressively favor ones with the minimal cost [17].

In most of the existing literature [7], the $\epsilon-$greedy method is utilized, in which an agent explores with probability ϵ and exploits with probability $1 - \epsilon$. However, in this work, we did not employ any method for coping up with this trade-off as the Q-learning algorithm by default has a feature which causes it to switch between exploration and exploitation modes, during the training of the Q-agent. For example, $\forall(s,a)$ pairs if $Q(s,a)$ values are initialized to the same value, then random policy can be executed for breaking the ties; here, the use of random policy is equivalent to the exploration mode. Moreover, if we carefully evaluate (12), when an (s,a) pair is visited for a number of times, its Q value increases. Since, in this work, the agent is required to find the action in a state with the smallest Q value, the less visited (s,a) pairs by default get a chance to be explored. Thus, this insight shows that the Q-learning when defined in terms of a minimization optimization problem, by default, has the capability of switching between exploration and exploitation modes.

5. Results and Discussion

We used Python for creating simulation setup and implementation of Algorithm 1. Unless otherwise specified, the parameters used for generating the results, presented in this section, are mentioned in Table 2.

Table 2: The default parameters used for simulating multi-RAT wireless network scenario and training of Q-agent.

Parameter(s)	Value(s)
P_c, P_w	46 dBm, 23 dBm
$\lambda_c, \lambda_w, \lambda_u$	10 AP/km^2, 100 AP/km^2, 300 users/km^2
$\mathcal{B}_c, \mathcal{B}_w$	50 MHz, 10 MHz
$\rho(c_1), \rho(c_2)$	3 Mbps, 1 Mbps
$\rho(w_1), \rho(w_2)$	3 Mbps, 1 Mbps
$\varphi(0), \varphi(1), \varphi(2)$	0\$/GB, 6\$/GB, 2\$/GB
$\Upsilon(h)$	bh^2, h is in Mbits and $b = 0.001$

For training of the Q-agent, we executed 120×10^4 episodes of Algorithm 1. The numbers of times the Q-agent observed certain states, irrespective of the actions taken, are reported in Figure 2. According to [17], the Q-learning algorithm is guaranteed to find an optimal solution if the number of visits to each (s,a) pair is sufficiently large. However, in practical scenarios, it is highly likely that some states are observed more often as compared to others. We have reported the results in Figures 2(a)–2(c) for the states when the remaining file size (h) to download is 200 Mbits, 500 Mbits, and 800 Mbits, respectively, as a function of remaining delay (d) and users' location (k). It must be evident from the numbers reported in Figure 2 that some states are visited more often as compared to the others. One of the main reasons behind such results is users' mobility; that is, the locations with higher probabilities are visited more often as compared to the others. Furthermore, the states with higher d and larger h are visited less often as evident from Figure 2. Because, if $h = 800$ Mbits and $d = 10$ mint at the

current decision epoch, then at the next decision epoch, $h \leq 800$ Mbits and $d < 10$ mint. This implies that the states with $h \ll 800$ Mbits and $d \ll 10$ mint are visited more often.

We have reported the data offloading policy learned by the Q-agent in Figure 3, that is, the optimal actions taken by the Q-agent in the same states as mentioned in Figure 2. We have included data for only three most important locations wherein the decision-making is challenging just to give better insights. For example, the decision is simple at the locations where both the RATs offer the same data rate, that is, to offload data through Wi-Fi RAT. However, it is challenging for Q-agent to choose the correct action at the locations where both the RATs offer different data rates. For example, at location (c_2, w_1), the Wi-Fi RAT offers data rate which is greater than the cellular RAT; therefore, the Q-agent has learned to offload data through Wi-Fi RAT only, as evident from Figure 3(a). At location (c_2, w_3), the cellular RAT is available, but Wi-Fi RAT is not available. The Q-agent has learned to download data through cellular RAT in this case, as evident from Figure 3(a), although it can wait for the availability of Wi-Fi RAT for higher d. However, due to the stochastic nature of the wireless network, it is possible that the Q-agent observes locations where both the RATs do not offer any data rate. That is why, the Q-agent has learned to download data through cellular RAT even for higher d which is evident from Figure 3(a). Moreover, since Q-learning is a model-free learning algorithm, minor fluctuations in the Q-agents' decisions are possible because of the stochastic nature of the environment. At location (c_1, w_2), the data rate supported by cellular RAT is higher than the Wi-Fi RAT. Since $h = 200$ Mbits, it can be easily downloaded in zone w_2 of Wi-Fi RAT for $d > 4$ mint. Therefore, except for $d = 6$ mint, the Q-agent has learned the optimal actions at this location; that is, it has decided to download data through cellular RAT for lower d and offload data through Wi-Fi RAT for higher d, as evident from Figure 3(a).

All the actions learned by the Q-agent in Figure 3(b) are optimal. For larger h and $d \leq 7$ mint, it is important to download data through cellular RAT when Wi-Fi RAT offers a lower data rate or not available. In Figure 3(c), we have reported results for $h = 800$ Mbits as a function of d and k. The Q-agent has not learned optimal actions for most of the states in Figure 3(c), because of larger h, most of these states have not been visited as evident from Figure 2(c) and already discussed in previous paragraphs. Thus, if observed, the Q-agent employs random policy for taking actions in such never visited states. At location (c_1, w_2), the Q-agent has learned to download data through cellular RAT only, as evident from Figure 3(c), because the data rate supported by c_1 is greater than w_2. Moreover, since $h = 800\,M$bits, the RAT which offers a higher data rate must be selected to successfully complete the download before $d \longrightarrow 0$. Thus, the Q-agent has learned the optimal actions for these states. Similarly, at location (c_2, w_1), the Q-agent has learned to offload data through Wi-Fi RAT because the data rate supported by zone w_1 is greater than c_2. Thus, we can conclude that the Q-agent has learned the optimal actions for almost all the states. Although it looks like it has learned a few incorrect decisions as well, such decisions have been learned due to the stochastic nature of the environment and can change over time after sufficient experience.

After each learning episode, the remaining file size (h) after hitting the deadline (d) is reported in Figure 4. We have reported the results in Figures 4(a)–4(c) for the episodes in which the users have generated requests for $\psi = 200$ Mbits, 500 Mbits, and 800 Mbits, respectively. The episodes are denoted in sorted order from left to right as a function of delay limit $\mathscr{D}$, and the color bar is used to represent it. The file with $\psi \geq 200$ Mbits cannot be successfully downloaded within $\mathscr{D} = 1$ mint no matter which RAT Q-agent choose in the assumed scenario. Because the maximum data rate supported by both the RATs is 3Mbps and in 1 mint at maximum 180 Mbits can be downloaded given that the user is located in the zone which supports the maximum data rate. That is why, $h \approx 200$ Mbits for almost all the episodes with $\mathscr{D} = 1$ mint. However, for higher $\mathscr{D}$, the Q-agent is trying to minimize the remaining file size, that is, $h \longrightarrow 0$ as $\mathscr{D}$ increases which is evident from Figure 4(a). Moreover, with each learning episode, the Q-agent has improved its decision-making capability. As it is evident from Figure 4 that during initial episodes for most of the cases, the download is incomplete, that is, $h > 0$. However, with each passing episode, h has been reduced and it ultimately approaches to zero. It is important to note here that for larger ψ like in Figures 4(b) and 4(c), the successful download is possible only for higher $\mathscr{D}$. That is why, even after learning for a quite large number of episodes, the agent is unable to successfully complete the download for certain cases. Nevertheless, given the network availability and higher $\mathscr{D}$, the agent has learned to successfully download larger files by taking a correct sequence of actions.

The accumulated payment for downloading ψ bits of data in $\mathscr{D}$ mint is shown in Figure 5 for a few randomly selected episodes at the end of the training period of the Q-agent. The minimum payment for downloading ψ bits of data, by using Wi-Fi RAT only, is shown by a double-dashed line in Figure 5. The maximum payment for downloading ψ bits of data, by using cellular RAT only, is shown by a dashed-dotted line in Figure 5. This minimum and maximum payment limits serve as a reference for evaluating the performance of the data offloading policy learned by the Q-agent. The average payment for downloading of a file, as a result of actions taken by the Q-agent, has been represented by a solid line. For $\psi = 200$ Mbits, the file can be successfully downloaded for $\mathscr{D} \geq 4$ mints. That is why, for $\psi = 200$ Mbits in Figure 5, the Q-agent has taken the sequence of steps, for most of the episodes, which has resulted in the minimal payment. However, the average payment for downloading of the file, as a result of the actions taken by the Q-agent, is above the minimum threshold. This is due to the stochastic nature of the wireless network because Wi-Fi RAT may not be available at certain locations and the Q-agent must download data through cellular RAT to complete the download in such situations. As a result, the average payment for download of the file is slightly higher. However, it must be interesting to note that the average payment is much

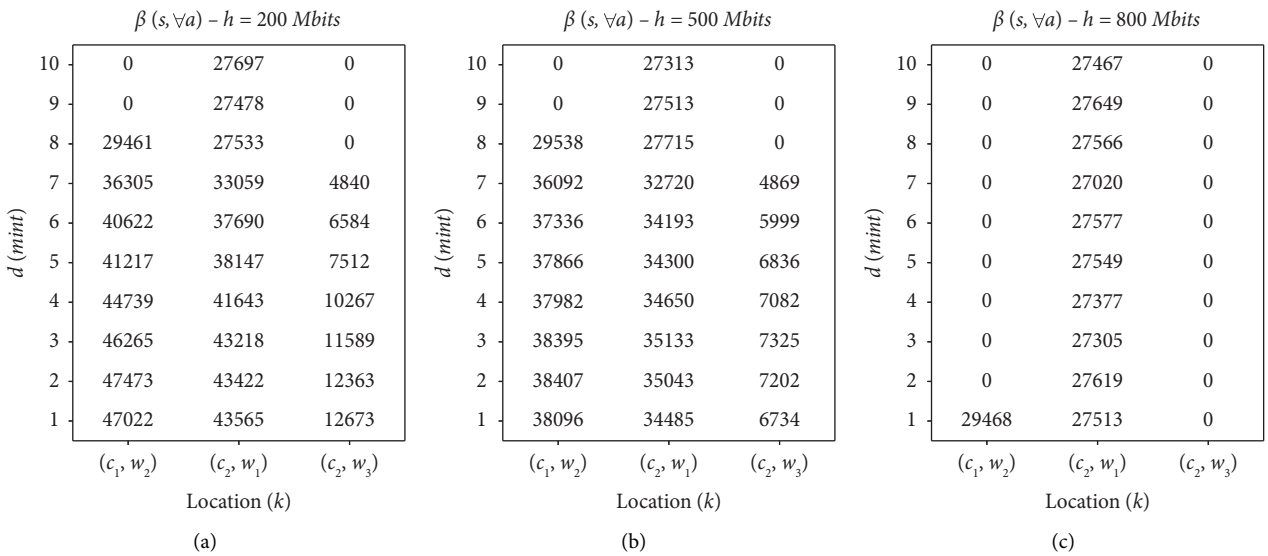


FIGURE 2: The number of visits by the Q-agent to selected states $s(k = (c_z, w_z), h, d)$ accumulated over all actions.

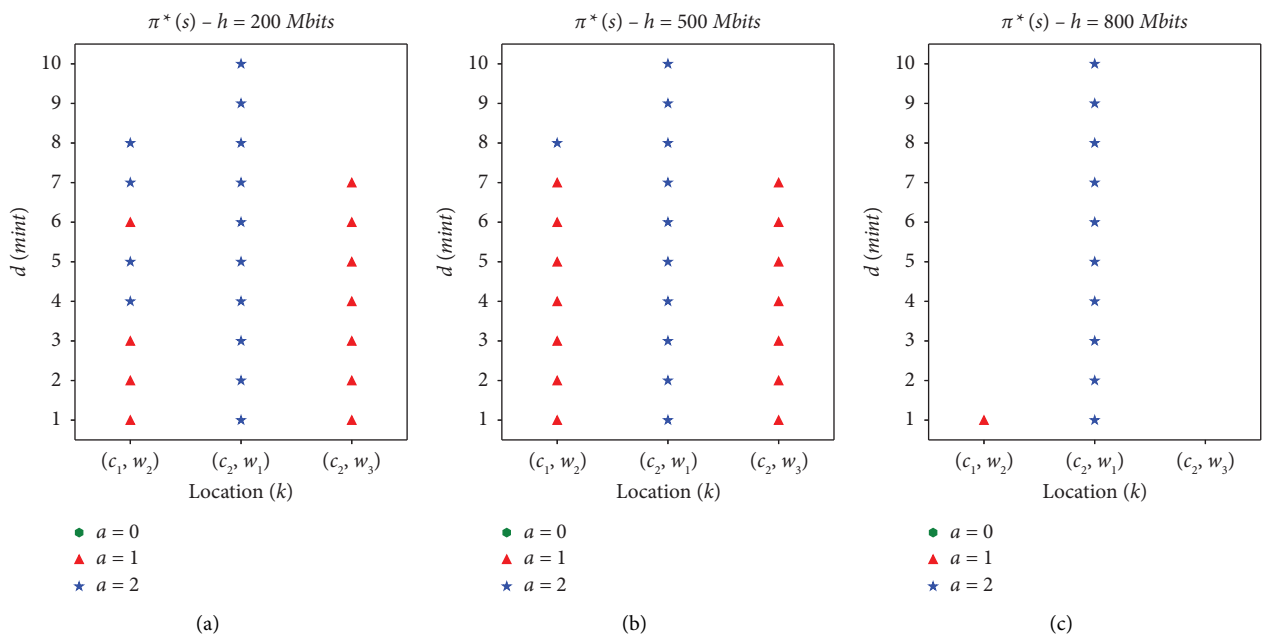


FIGURE 3: The actions learned by the Q-agent, $a = \pi^*(s)$, as a function of states $s(k = (c_z, w_z), h, d)$. Here, ⬡ denotes remain idle, △ denotes download data through cellular RAT, and ∗ denotes download data through Wi-Fi RAT.

smaller than the maximum threshold and closer to the minimum threshold which implies that the Q-agent has mostly offloaded data through Wi-Fi RAT.

It must be evident from Figure 5(a) that for $\psi \geq 500$ Mbits, the average payment for downloading the file is approximately equal or smaller than the minimum threshold. This is possible only in those situations in which the file download has not been completed successfully within the defined $\mathscr{D}$. Since $\mathscr{D}$ in Figure 5(a) is only 4 mints, larger files cannot be successfully downloaded even if the Q-agent chooses the RAT which supports maximum data rate always. On the other hand, in Figure 5(b), $\mathscr{D} = 8$ mints. As a result, the average payment for the file download is above the minimum threshold because for most of the episodes, the Q-agent has successfully downloaded the files with larger size as well.

We have evaluated the performance of the developed data offloading policy learned by the Q-agent and compared it with the standard policies and analytical (Ana.) approach presented in [4]. The evaluation results are reported in Figure 6. In always offload (AO) policy, the data are downloaded by using Wi-Fi RAT only. In no offload (NO) policy, the data are downloaded using cellular RAT only. In on-the-spot offload (OTSO) policy, the data are offloaded through Wi-Fi RAT whenever it is available. Otherwise, it is

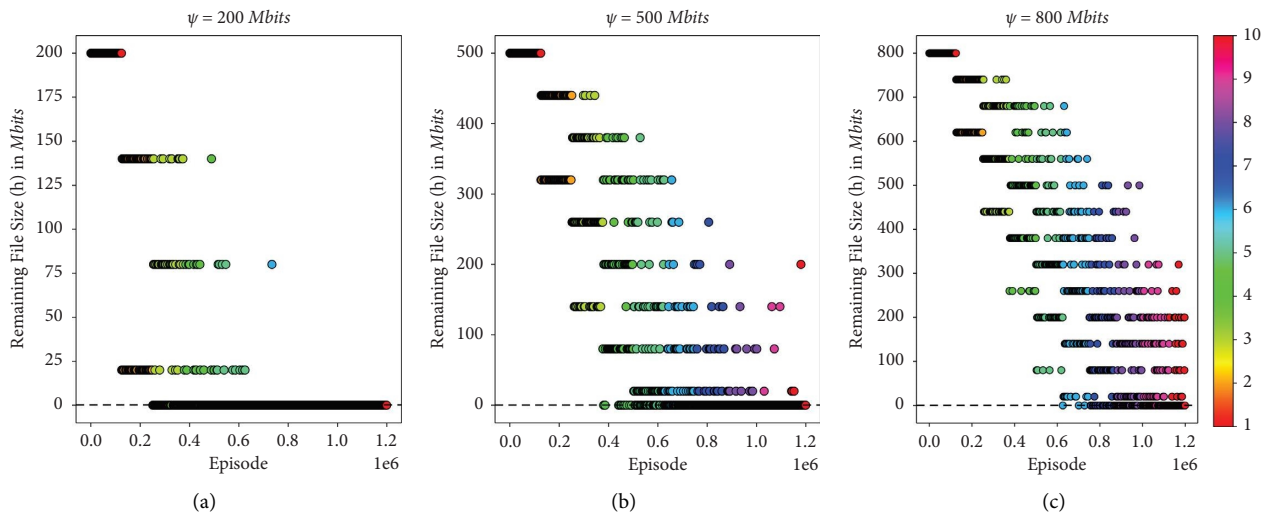
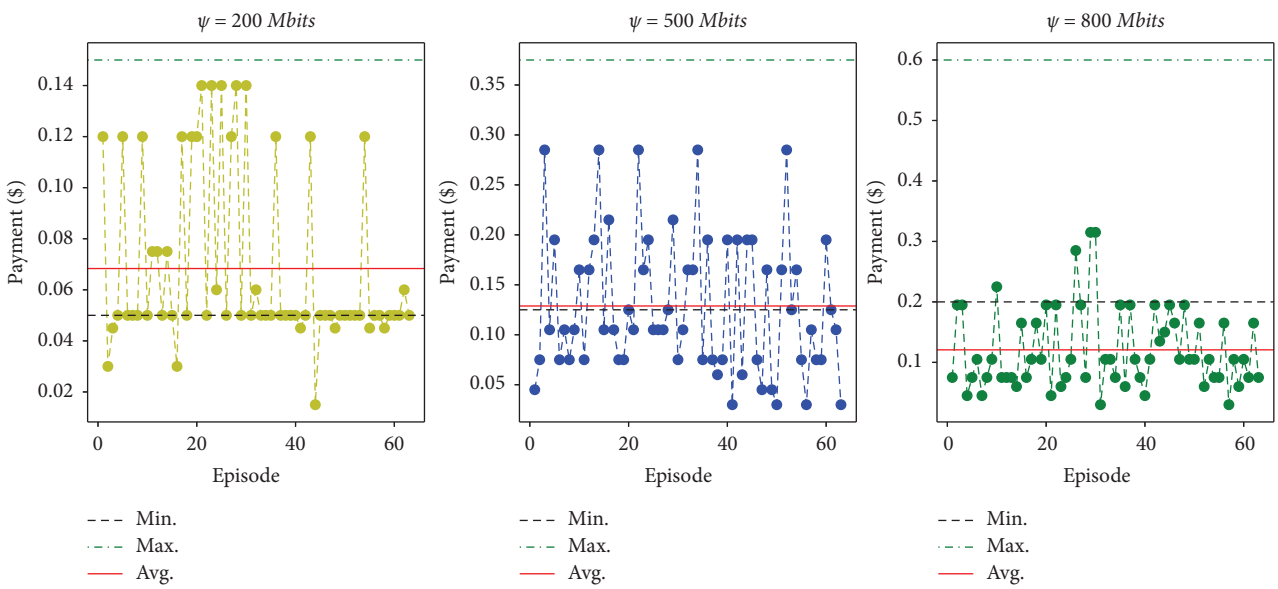


Figure 4: Remaining file size (h) at the end of each episode as a function $\mathscr{D}$.



(a)

Figure 5: Continued.

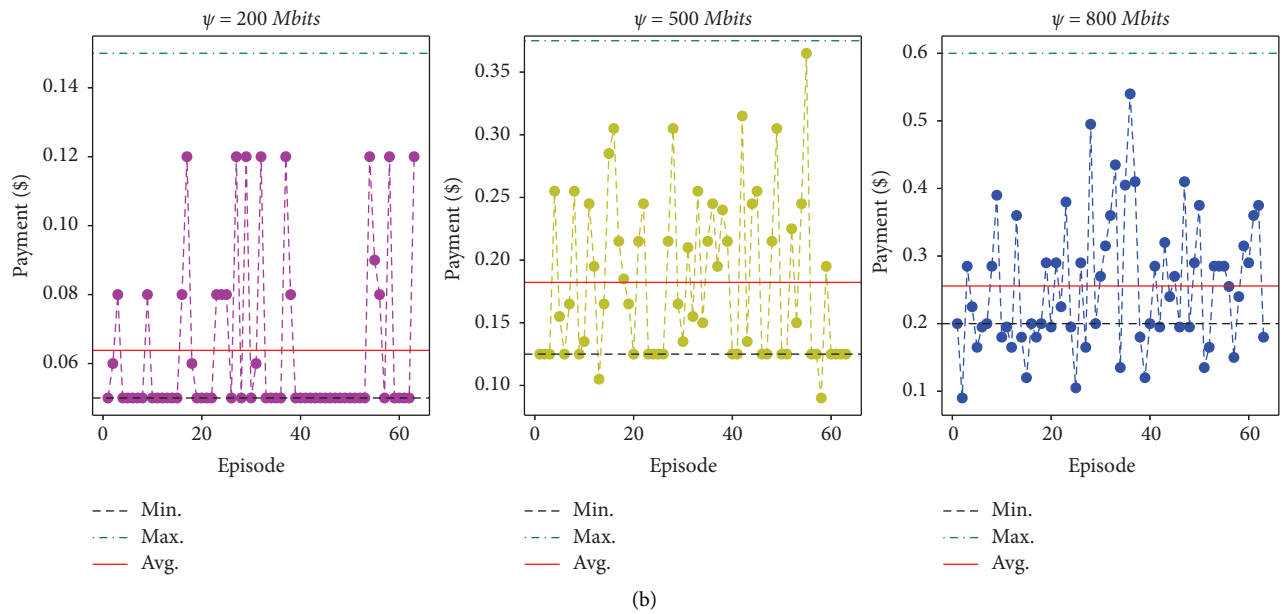


Figure 5: Accumulated payment at the end of a few randomly selected episode as a function $\mathscr{D}$. (a) $\mathscr{D}$ = 4 mint. (b) $\mathscr{D}$ = 8 mint.

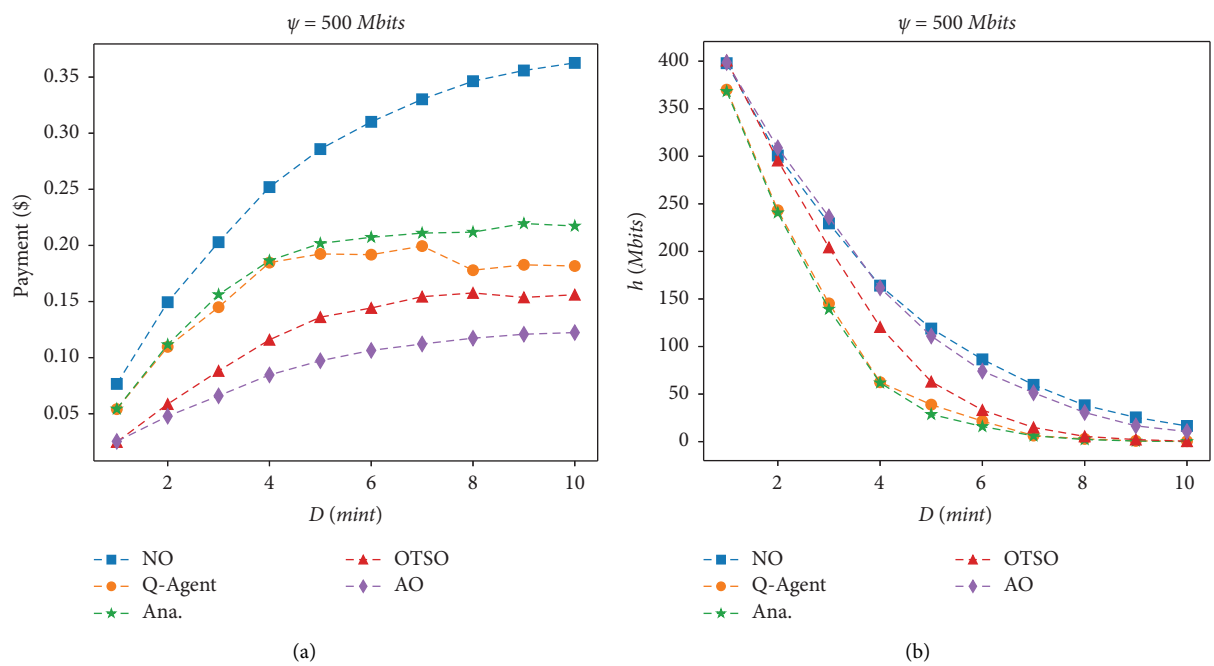


Figure 6: Performance evaluation of the data offloading policy learned by the Q-agent developed in this work and existing offloading policies, as a function of users' request $\mu(\psi = 500\,\text{Mbits}, \mathscr{D})$: (a) the payment in ($) for entertaining a given user request and (b) remaining file size (h) after the defined delay timer $(\mathscr{D})$ expires.

downloaded using cellular RAT. The data offloading policy presented in [4] is obtained by using a policy iteration algorithm which is the model-based RL algorithm. In Figure 6(a), we report the average payment a user has to pay for downloading ψ = 500 Mbits as a function of $\mathscr{D}$. Similarly, in Figure 6(b), we report the remaining file size (h) after the given delay timer $(\mathscr{D})$ expires. These average results have been obtained after executing each existing policy and the one learned by the Q-agent for 1000 iterations.

The NO approach has resulted maximum payment, which is evident from Figure 6(a), because the cost of using cellular RAT is larger as compared to the Wi-Fi RAT. Moreover, due to users' mobility and unavailability of cellular RAT at certain locations, the NO approach could not successfully download the file even for larger $\mathscr{D} > 7$ mint, as evident from Figure 6(b). Similarly, the AO approach has resulted in minimum payment, as evident from Figure 6(a), because the cost of using Wi-Fi RAT is smaller as compared to the cellular RAT. However, due

to users' mobility and unavailability of Wi-Fi RAT at certain locations, it suffers from the same issue of incomplete data download even for larger $\mathscr{D} > 7$ mint, as evident from Figure 6(b). Since OTSO exploits both the RATs given their availability and prefers Wi-Fi RAT over cellular RAT, the cost for downloading data is smaller than NO and is slightly larger than AO which is evident from Figure 6(a). Moreover, since the OTSO approach is using both the RATs, it has successfully completed the download request for larger $\mathscr{D} > 7$ mint, as evident from Figure 6(b). The Ana. approach presented in [4] uses both the RATs for data download; however, for lower delay limits, it prefers the RAT which offers a higher data rate so that the data download can be completed. That is why, in Figure 6(a), the payment for Ana. approach is slightly larger and the remaining file size in Figure 6(b) is slightly smaller than the OTSO approach.

As evident from the results reported in Figure 6, for delay limits $\mathscr{D} < 7$ mint, the performance of data offloading policy learned by the Q-agent is comparable to the model-based analytical approach. Although the payment of the Ana. approach and Q-agent is slightly higher compared to the OTSO and AO policies, h is much smaller as evident from Figure 6(b). This implies that these approaches have tried to complete the download without waiting for the availability of Wi-Fi RAT because $\mathscr{D}$ is short. On the other hand, for $\mathscr{D} > 7$ mint, the OTSO, Ana., and Q-agent have successfully completed the download of the file. However, for larger $\mathscr{D}$, the payment of the Ana. approach is slightly higher because it has downloaded data through cellular RAT without waiting much for the availability of Wi-Fi RAT. Since $\mathscr{D}$ is large, the Q-agent has waited for the the availability of Wi-Fi RAT in this case and tried to minimize the payment as well. The AO policy has obtained minimal payment because it always downloads data through Wi-Fi RAT. Thus, it is clear that close to the best and stable performance has been offered by the Q-agent for different $\mathscr{D}$. For lower $\mathscr{D}$, it tried to complete the download at the cost of slightly larger payment. On the other hand, for higher $\mathscr{D}$, it tried to minimize the payment while successfully completing the download.

6. Conclusion and Future Work

In this work, we developed an AI-enabled Q-agent for making data offloading decisions in a multi-RAT wireless network by using a model-free Q-learning algorithm. Although model-free learning algorithms offer quite a good set of features, their successful implementation poses various challenges. Therefore, we also discussed a few of the challenges along with their possible solutions. For speeding up the learning process, we initialized the $Q(s, a)$ values of the Q-learning algorithm by employing a single back-up sweep. Moreover, we exploited an inherit feature offered by the Q-learning algorithm, by redefining it in terms of expectation minimization problem, to balance the trade-off between exploration and exploitation modes. We evaluated the performance of the trained Q-agent and also compared against an existing analytical data offloading approach [4] and other offloading policies like always offload, no offload, and on-the-spot offload. The results showed that the performance of the Q-agent developed in this work is near optimal for different data download requests. For lower delay limits, the performance of the Q-agent for making data offloading decisions is close to the model-based approach presented in [4] which tries to complete the download at the cost of a higher payment. For higher delay limits, its performance is close to the on-the-spot offloading policy which tries to minimize the payment. Thus, the Q-agent has learned to make intelligent and near optimal decisions under different situations. The future work includes the development of such adaptive and optimal agents for 6G wireless networks by using advanced AI techniques such as DQN or double DQN.

Abbreviations

RL: Reinforcement learning
QoS: Quality-of-service
RAT: Radio access technology
AI: Artificial intelligence
UE: User equipment
SINR: Signal-to-interference noise ratio
Wi-Fi: Wireless fidelity
HPPP: Homogeneous Poisson point process
SG: Stochastic geometry
AP: Access points.

Data Availability

No data were used to support the findings of this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Murk Marvi and Adnan Aijaz have made substantial contributions to conception, design, analysis, and interpretation of the results; Murk Marvi and Anam Qureshi have been involved in the drafting of the manuscript and coding; Murk Marvi and Muhammad Khurram have revised it critically for important intellectual content; all the authors have read the final version of the manuscript and have given the final approval for this version to be published. Each author has agreed to be accountable for all aspects of the work.

References

[1] L. Wang, C. Yang, and R. Q. Hu, "Autonomous traffic offloading in heterogeneous ultra-dense networks using machine learning," *IEEE Wireless Communications*, vol. 26, no. 4, pp. 102–109, 2019.

[2] X. Wang, Z. Ning, L. Guo, S. Guo, X. Gao, and G. Wang, "Online learning for distributed computation offloading in wireless powered mobile edge computing networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 8, pp. 1841–1855, 2022.

[3] A. Gao, S. Zhang, Y. Hu, W. Liang, and S. X. Ng, "Game-combined multi-agent DRL for tasks offloading in wireless

powered MEC networks," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 7, pp. 9131–9144, 2023.

[4] M. Marvi, A. Aijaz, and M. Khurram, "Toward an automated data offloading framework for multi-rat 5g wireless networks," *IEEE Transactions on Network and Service Management*, vol. 17, no. 4, pp. 2584–2597, 2020.

[5] A. Paleyes, R.-G. Urma, and N. D. Lawrence, "Challenges in deploying machine learning: a survey of case studies," *ACM Computing Surveys*, vol. 55, no. 6, pp. 1–29, 2022.

[6] M. H. Cheung and J. Huang, "Dawn: delay-aware Wi-Fi offloading and network selection," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 6, pp. 1214–1223, 2015.

[7] M. El Helou, M. Ibrahim, S. Lahoud, K. Khawam, D. Mezher, and B. Cousin, "A network-assisted approach for rat selection in heterogeneous cellular networks," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 6, pp. 1055–1067, 2015.

[8] M. Yi, Y. Zhang, X. Wang, C. Xu, and X. Ma, "Deep reinforcement learning for user association in heterogeneous networks with dual connectivity," in *Proceeedings of the 2021 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–5, IEEE, Nanjing, China, March 2021.

[9] M. Yan, G. Feng, J. Zhou, and S. Qin, "Smart multi-RAT access based on multiagent reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 5, pp. 4539–4551, 2018.

[10] Z. Hu, Z. Lu, X. Wen, and Q. Li, "Stochastic-geometry-based performance analysis of delayed mobile data offloading with mobility prediction in dense IEEE 802.11 networks," *IEEE Access*, vol. 5, pp. 23060–23068, 2017.

[11] J. Yun, Y. Goh, W. Yoo, and J.-M. Chung, "5G multi-RAT URLLC and eMBB dynamic task offloading with MEC resource allocation using distributed deep reinforcement learning," *IEEE Internet of Things Journal*, vol. 9, no. 20, pp. 20733–20749, 2022.

[12] X. Song, R. Zhang, Y. Wang, Y. Yu, and S. Xu, "Incentive mechanism design for two-layer mobile data offloading networks: a contract theory approach," *Ad Hoc Networks*, vol. 144, Article ID 103154, 2023.

[13] H. ElSawy, A. Sultan-Salem, M. S. Alouini, and M. Z. Win, "Modeling and analysis of cellular networks using stochastic geometry: a tutorial," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 167–203, 2017.

[14] J. G. Andrews, F. Baccelli, and R. K. Ganti, "A tractable approach to coverage and rate in cellular networks," *IEEE Transactions on Communications*, vol. 59, no. 11, pp. 3122–3134, 2011.

[15] H. Yu, M. H. Cheung, L. Huang, and J. Huang, "Power-delay tradeoff with predictive scheduling in integrated cellular and Wi-Fi networks," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 4, pp. 735–742, 2016.

[16] M. Marvi, A. Aijaz, and M. Khurram, "Toward a unified framework for analysis of multi-RAT heterogeneous wireless networks," *Wireless Communications and Mobile Computing*, vol. 2019, Article ID 6918637, 19 pages, 2019.

[17] C. J. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3/4, pp. 279–292, 1992.