Journal of ELECTRICAL ENGINEERING, VOL 74(2023), NO1, 48–56

sciendo

PAPERS

Deep Q-Learning based resource allocation and load balancing in a mobile edge system serving different types of user requests

Önem Yıldız*, Radosveta Ivanova Sokullu[1]

With the expansion of the communicative and perceptual capabilities of mobile devices in recent years, the number of complex and high computational applications has also increased rendering traditional methods of traffic management and resource allocation quite insufficient. Recently, mobile edge computing (MEC) has emerged as a new viable solution to these problems. It can provide additional computing features at the edge of the network and allow alleviation of the resource limit of mobile devices while increasing the performance for critical applications especially in terms of latency. In this work, we addressed the issue of reducing the service delay by choosing the optimal path in the MEC network, which consists of multiple MEC servers that has different capabilities, applying network load balancing where multiple requests need to be handled simultaneously and routing selection based on a deep-Q network (DQN) algorithm. A novel traffic control and resource allocation method is proposed based on deep Q-learning (DQL) which allows reducing the end-to-end delay in cellular networks and in the mobile edge network. Real life traffic scenarios with various types of user requests are considered and a novel DQL resource allocation scheme which adaptively assigns computing and network resources is proposed. The algorithm optimizes traffic distribution between servers reducing the total service time and balancing the use of available resources under varying environmental conditions.

K e y w o r d s: cellular network, deep Q-learning, mobile edge network, resource allocation

1 Introduction

With continuously evolving concept of smart city new advanced services come into the picture of communication networks especially mobile and cellular networks. Services like following the activity crowds in smart cities, enabling augmented reality (AR), virtual reality (VR) and real-time internet of things (IoT) applications are associated with increasingly more demanding and versatile traffic requirements for the network to deal with. Moreover, advanced smart city services (*eg*, augmented reality and interactive gaming) not only need intensive resources but are also extremely sensitive to latency. Therefore, in order to facilitate these services, especially in densely user populated environments, a wide network coverage and a large number of resources located near the end devices are needed. Mobile edge computing (MEC) has emerged as a promising solution in this line, which is able to provide services with significantly reduced latency for dense user populations located in the access area. The symbiosis of MEC and cellular network technologies will have a significant impact on service delivery and allow to meet the needs of large numbers of densely located end user devices.

With the proliferation of real-time and IoT-related applications connected to the edge network, the increased variability of service requests will lead to extremely dynamic changes in traffic loads. Furthermore, the service requests sent by various devices will have different service requirements, and the changing heavy traffic load will significantly affect the quality of experience (QoE) of users waiting to be served. Thus, network traffic control is becoming a decisive component in the edge network in order to ensure the required user quality of service (QoS). On the flip side however, it is getting increasingly difficult to design routing strategies to meet such stringent and ever-changing network requirements. Therefore, novel, and intelligent routing mechanisms are needed to cope with the large growth in traffic within an ever more complex network environment.

2 Related work

Several network studies are addressing delay reduction, routing issues and resource allocation in mobile edge networks. These problems have attracted the attention of researchers from many different areas. A group of works [1,2] focuses on limiting the end-to-end delay in the downlink. Another group of studies [3-5] examines the uplink case and the possibilities of different resource allocation mechanisms incorporating offloading to improve the performance of the network. In [5], DRL based resource allocation scheme is proposed for allocating computing and network resources adaptively with the goal to reduce the

[1] Department of Electrical and Electronics Engineering, Ege University, Izmir, 35100, Turkey * onem.yildiz@gmail.com.

https://doi.org/10.2478/jee-2023-0006, Print (till 2015) ISSN 1335-3632, On-line ISSN 1339-309X

average service time and balance the use of resources under a varying MEC environment.

Some other works address the issues of quality insurance and energy consumption in relation to delay. The authors in [6] proposed deep learning-based offloading technique to minimize the cost function which considers service delay, energy consumption, radio and computing resources of mobile devices and mobile edge servers. In another work [7], the authors construct a MEC-based computation offloading framework considering the delay cost, energy computation cost and bandwidth cost for vehicular networks. To minimize the defined cost, they propose a DRL-based computation migration and resource allocation scheme that requires no prior knowledge. In [8], an intelligent resource allocation framework is proposed to solve the complex resource allocation problem for the collaborative mobile edge computing network on a multi-task DRL algorithm with self-play training. In [9], an adaptive service offloading scheme for MEC is proposed to minimize the service latency, including service execution and offloading latencies, and offloading price. In study [10], the problem of placement of VRCs to minimize the average response time including overall network delay and processing delay in the MEC architecture is considered. In [11], the authors formulate a queue optimization problem to minimize the total service time in a multi-server system. Then, an intelligent task transition strategy is developed using DRL and Q-learning techniques called DQTM. The authors considered locally processed or migrated packets to calculate the average service time. In [12], a DRL-based scheme is proposed to solve the problem of vehicular service placement and migration in a MEC-based vehicular network. The optimization problem is formulated by using total delay including the computation and communication delays, migration cost in terms of bandwidth usage and energy consumption. The simulation results showed that the proposed DQL scheme achieve a near-optimal performance.

The above discussed body of research covers the issues of resource allocation while minimizing delay and energy, however the case of balancing the network load dynamically while reducing the task service time under varying environment and traffic conditions is not addressed. In this study, we consider a dense population of users generating requests to the edge network and focus on optimal load distribution in the MEC network under delay constraints. Tasks coming to the MEC network can be served by specific applications on specific servers. Not all servers have equal computational capabilities and/or host all possible applications. In light of this the main contributions of our work are:

- Definition of an optimization function which takes into consideration both uplink delay and traffic load in the MEC network.
- User packets coming from the cellular network and used as network inputs to our hybrid algorithm are interacted with signal to interference & noise ratio (SINR), which is also an important performance criterion in the cellular network, and an inter-layer interaction is revealed between cellular network and MEC network.
- The proposed algorithm considers real-time and non-real time types of services which have varying service requirements and constraints.
- The proposed algorithm ensures that the edge network load is optimally balanced over a set of servers with different application/computational capabilities and different connection capacities by ensuring that MEC servers are used as possible and will provide service with the least delay for balancing in case of multiple instances.
- The variation of packet sizes belonging to different applications and service time experienced by users at different SINR values were examined and simulation results were presented.

3 System model

The study considers a scenario where a large number of mobile device users generate real-time and non-real-time traffic demands on an ultra-dense cellular network equipped with access points (APs).

As shown in Fig. 1, there are a set of N access points (APs), and a set of K users (U). Since it is important that the service takes place with minimal delay, the scenario incorporates a set of MEC_m $m = 1, 2 \ldots, M$, servers with different computational and application capabilities. The set of the application request is X_i, $\{i = 1, 2 \ldots, I\}$. In the MEC network, high-speed (backhaul) links have different capacities, and the set of backhaul connections between MEC servers is expressed by B_e, with $\{e = 1, 2 \ldots, E\}$.

As the tasks are generated, the time they were issued is recorded too. Each task contains the components $\varphi_k = \{d_k, x_k\}$, where, d_k is the data size of the packet from the k-th user, and x_k is the type of application required by that user. The task generated at any time instant can be a real-time (RT) or non-real-time (NRT) application request type. According to the type of the incoming task φ_k, if the server has a serving application that can respond to this task, it processes it itself, and if not, it redirects it to the server that is closest and most suitable for responding to that task.

In the proposed intelligent routing architecture, a DRL module is built into the SDN controller, which has general knowledge of the network state and can control the routing of requests generated by the end devices. The proposed SDN-enabled MEC architecture is presented in Fig. 1. Packet forwarding takes place depending on several parameters. These are: the servers data processing capacity, the type of application the server can process and the connection load between the servers.

Since the speeds $\{v_e\}$: $e = 1, 2, \ldots, E$ of the connections between the servers are different and the data capacity that each server can handle is different, the routing

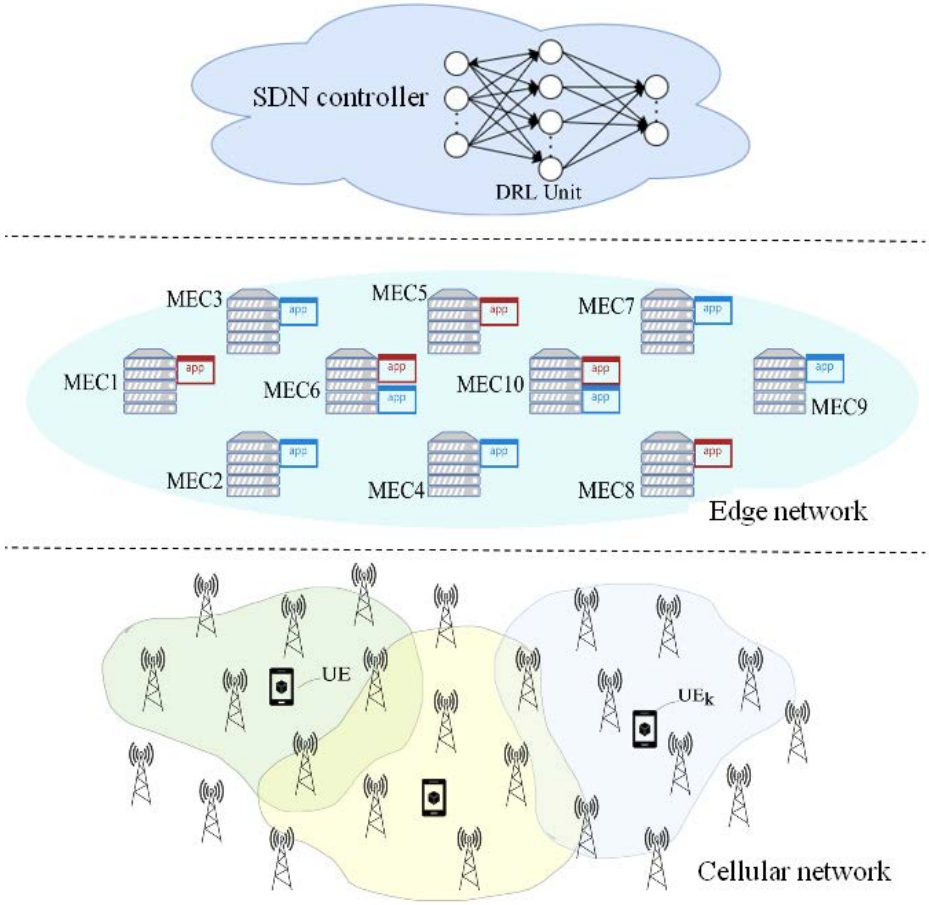

Fig. 1. SDN featured MEC architecture - general system architecture

of incoming requests should keep in balance both the load distribution between the servers for the current instance and the available capacity on the connections.

3.1 Problem Definition

The designed network is based on a hierarchical network architecture in which the upper layer control is provided by SDN administering distributed task processing for MEC servers and radio access is based on a cellular network. In addition, considering that each MEC server has limited resources, only a limited number of services can be processed simultaneously on a given server.

Although MEC servers have much better computing and storage capabilities than mobile end devices or IoT devices, to provide optimal overall service, their resources need to be carefully managed. Considering very high density of incoming requests, high traffic variability and resource demanding requests with different and strict latency constraints, the process of routing the requests to the relevant MEC server or servers has to be very efficiently optimized. This requires that the optimization is done both in terms of service time and load distribution between the MEC servers. So, the goal of this study is:

- To define the optimization function that will meet both traffic load and delay constraints
- To develop an ML based (DQN based) algorithm that will provide a solution.
- To evaluate the performance of the algorithm.
- To investigate the effects of the SINR parameter and optimize service time experienced by user with different type of requests.

4 Problem solution

4.1 Definition of the optimization function

To achieve the goals of this work, first the delay parameters that make up the service time are determined and the mathematical background for minimizing the service access delay for variable service requests generated by a large number of user devices while optimizing and balancing the load distribution on the MEC network which consists of multiple servers where multiple instances of requests occurs simultaneously. The total service delay is considered to have three major components: access delay, transfer delay, and processing delay.

Access delay refers to the time it takes for the request to be transmitted from the mobile device (end device) and reach the MEC server. Although this period refers to the time elapsed between the user (U) and an access point

(AP), the delay between AP and MEC servers is assumed to be negligible. For this reason, in the continuation of the study, access delay will be referred to as the delay between the U and the MEC server for ease of expression. Access delay

$$T_{k,m} = \frac{d_k}{R_k}, \tag{1}$$

where d_k is the packet size sent from U_k (bit) and R_k is the data rate of U_k (bit/s). The d_k packet sizes for each user are obtained based on the 3GPP R1-070674 document, [13]. According to the information contained in this document, package sizes differ depending on the type of traffic. In the study, ftp and gaming traffics are based on the types of traffic that 3GPP also specifies under the main heading of non-real-time and real-time traffic. The data rates R_k can be determined according to the respective cellular network standard as in [14]. After obtaining the access delay values experienced by each user, the highest value was chosen and used in the analysis as the access delay value.

We proposed an equation that allows to determine the user packet sizes transmitted depending on the SINR value of each user

$$\gamma_{kS} = \frac{P_{\mathrm{S}}}{P_{\mathrm{S}} + P_{\mathrm{N}} + P_{\mathrm{I}}}, \tag{2}$$

where $P_{\mathrm{S}}, P_{\mathrm{N}}$, and P_{I} are power of signal, noise and interference, respectively.

It is well known that the greater the strength (power) of the desired signal, the lower the bit error rate (BER) value will be, and the packets will reach the MEC network with less errors *ie*, the "goodput" is higher. Thus, the packet size to be processed in the MEC network is calculated as

$$d_{\mathrm{u}} = \gamma_{kS} d_{\mathrm{k}}, \tag{3}$$

where d_{u} -is called "usable packet size"

Requests are generated by mobile devices accessing the network at the nearest MEC server and they are routed to MEC servers hosting the applications which correspond to the specific request. The total usavle data size of the combined requests for an application on the MEC server is expressed as $D^{(i)}_{w,m},\ i \in I$, where MEC_w is a server where the requests are collected, the upper index represents the i-th application needed, and MEC_m is the server that will meet the request.

The transfer delay which is the time it takes for the request to be transferred from the initially accessed server MEC_w to the server that will meet the request MEC_m, is expressed as follows

$$\begin{aligned} &T^{(i)}_{w,m} = \sum_{e=1}^{I} \frac{D^{(e)}_{w,m}}{v_i} P^{(e)}_{w,m}, \\ &w, m \in M, \quad P^{(i)}_{w,m} \in \{0,1\}, \end{aligned} \tag{4}$$

here, the term $P^{(e)}_{w,m}$ indicates probability of whether the connection $e \in E$ is on the routing route of the requests collected on the MEC_w server and forwarded to the MEC_m server, and v_e refers to the edge network routing capacity (bit/s).

After a request arrives at the MEC_m server, the delay caused by the processing of the data on this server is $T^{(i)}_m$, while $F^{(i)}_{w,m}$ defines the size of the combined requests that can be processed on MEC_m server at that instance. The time required for the processing is

$$T^{(i)}_m = \frac{F^{(i)}_{w,m}}{\nu_m}, \tag{5}$$

ν_m refers to the processing speed of the MEC_m server.

Finally, the goal of minimizing the total service delay T can be formulated as

$$\begin{aligned} &\min_{e(v_e, c_m)} T = T^{(i)}_{w,m} + T_{k,m} + T^{(i)}_m \\ &\text{st} \quad P^{(e)}_{w.m} \in \{0,1\}, \quad e \in E \\ &T_{k,m} = \max_{(T)} \Big\{ T_{1,m}, T_{2,m}, \ldots, T_{K,m} \Big\}, \\ &k \in U,\ m \in M \\ &0 \le F^{(i)}_{w,m} \le \sum_{m=1}^{M} c_m, \quad w, m \in M,\ i \in I \end{aligned} \tag{6}$$

The parameter "e" in the optimization equation is related to the application type, the current MEC capacity, and the current inter-MEC connection speed. Since we are focusing on optimizing the edge network portion of the delay, this parameter will also be a key component in defining the rewards for the proposed DQN algorithm. The proposed optimization function allows us to optimize not only the load distribution on the MEC servers, but also the distribution of the loads on the connections between the MEC servers, making the network accessible for more new requests. Thus, we maximize the number of available MEC servers by minimizing the number of MEC servers used in the network. Furthermore, instead of distributing the packets of a single request to more than one MEC server, the workload on specific MEC servers is maximized, and the load on the network as a whole is reduced because there are less data transfers within the network.

4.2 The DQN based algorithm

Q-learning is a model-free learning method and one of the classical RL algorithms. At the beginning of each chapter in Q-learning, the environmental situation is discussed. For each step in the section, the action a_t should be chosen based on the current situation s_t and policy. Then the corresponding reward r_t and the next state s_{t+1} can be obtained. Next, the action value $Q(t) = Q(s_t, a_t)$ should be updated using the Bellman equation

$$Q(t) \leftarrow Q(t) + \beta\Big(r_t + \gamma_{\max,t+1} Q(t+1) - Q(t)\Big). \tag{7}$$

Due to the large complexity of the environment considered in our case such a Q-table will be of unacceptable size. So, a Q-learning algorithm based on neural networks (NN) is used instead. In DQL, the learning process uses two neural networks: the trained network and the target network. These networks have the same architecture but different weights. The parameters of the target DQN are updated according to the evaluation of trained DQN at a certain rate where the parameters of the trained DQN are backpropagation using Adam [15] optimizer algorithm. It is important to emphasize that the parameters of the target network are not trained but synchronized periodically with the parameters of the main Q-network. Using the Q-values of the target network to train the main Q-network increases the stability of the training.

For a state value given as input to the NN, Q-values corresponding to more than one action will be obtained as output. From these outputs, the action maximizing the Q-value is selected and the target action specified as

$$\gamma_i = r_i + \gamma_{\max,t+1} Q(t+1), \tag{8}$$

$$L = \frac{1}{N} \sum_{i=0}^{N-1} \Big(Q_i - y_i \Big)^2. \tag{9}$$

All other possible outcomes are then used to approximate this target action by checking the difference between the Q-values and the target values. By calculating the error between the Q-values and the target values the loss function calculation is also performed

The three main components of the proposed DQN are defined as follows:

- State: The state vector $s \in S$ contains information about the data packet size (Mbit) that the MEC server will process at the time when the requests generated for the $x \in X$ service are collected.
- Action: The action vector consists of the elements $|X|\square|S|$. Any element in this vector represents an action that can be taken by the MEC server. are the generated service requests for the service X and they can be forwarded to a MEC server that has an service. The action here is defined as the MEC server's respond to the user generated requests. The MEC server can either process the request if it has the processing and application capabilities to do so or it can initiate its routing to the nearest available MEC server that can respond to this request.
- Rewards: The rewards and the Q-values are determined by the DQN algorithm. Since the reward depends on the connection speed and MEC server data capacity, the most suitable servers and backhaul connections at that time are evaluated to ensure optimal data transfer. The path selection for the data transfer from server to server depends on the type of packet that the server can process (*ie* , the type of application that it can serve), the current load of connections between servers and the data processing speed (*ie* , amount of data the server can process). The reward equation is

$$R_Q = \begin{cases} v_e c_m \xi & \text{if } m_i \text{ is feasible for } x_i \\ -100 & \text{if } m_i \text{ is not feasible for } x_i \end{cases}, \tag{10}$$

were, the parameter ξ is a weighting coefficient and was set to 0.001.

A novel hybrid algorithm (Algorithm 1) is designed combining Q-learning and DQL. Path planning is realized using Q-learning in the initial part of the algorithm and the Q-values obtained are used as initial Q-values for the DQN. These values are reinforced with the DQL algorithm. Then the obtained reward values in the DQN are used for Q-learning stage in the next iteration.

A novel hybrid algorithm (Algorithm 1) is designed combining Q-learning and DQL. Path planning is realized using Q-learning in the initial part of the algorithm and the Q-values obtained are used as initial Q-values for the DQN. These values are reinforced with the DQL algorithm. Then the obtained reward values in the DQN are used for Q-learning stage in the next iteration.

5 Simulation settings

A simulator in Python is developed and proposed DQN algorithm is trained on a computer with AMD Ryzen 9 3950X 16-Core CPU, 3.49 GHz frequency, 64 GB RAM and GIGABYTE GeForce RTX 2080 Ti graphic card. For simulations, UEs are distributed randomly and uniformly throughout the cellular network. Thanks to the wrap-around technique, an infinite-sized network is approached with 1000 antennas/km^2 and 250 UEs/km^2 using the 4 km^2 square area. The edge network comprises 10 MEC servers, each one with predefined capabilities: some servers are designed to respond only to NRT applications (blue color code), some only to RT applications (red color code), and some can serve both types of applications as shown in edge network in Fig. 1. As it can be understood, there are many MEC servers with different data processing capacities in our system and the application that each of these servers can serve is different. The backhaul connection speeds between servers has been taken into account, and in addition the packets that come to any server has been processed instantly if the packets arrive at the appropriate server. To process the remaining data as soon as possible if the packet size is larger than the server capacity, the remaining packets will be redirected to the MEC server, which can process the data in the fastest way. If the packets did not arrive at a suitable server in the first place, they are redirected to another server that can handle the request, taking into account the server capacities and backhaul connection speeds. The complexity of this MEC network design used in our scenario increases its relevance to real life.

Algorithm 1: Proposed Hybrid DQN Algorithm

Initialize Preprocess (MEC data capacity, speed, pack size, UE data rate, application type)
Initialize Routing Algorithm
While there are packages to process:
  If MEC_i can run application type and packet are not processed:
    Realize Action (Process packet)
    Update reward R_Q (MEC data capacity, connection speed)
    Transfer remaining packets to next available MEC
    Update path
  Else:
    Transfer packets to next available MEC
  Return R_Q
Initialize the Main network, the Target network, the Experience Replay Mechanism D, the agent to interact with the environment. Initial $r = R_Q$
If convergence criteria is **not** met:
  Run Agent
  Update ε using ε – decay.
  For state s, choose an action a using ε – greedy.
  Agent a realizes the action, observes the reward r and new state s'.
  Stores the transition sequence $(s, a, r, s'$, done) into experience replay memory D.
    If D has enough experience:
    A minibatch sized of N is chosen randomly from D
      Foreach $(s, a, r, s'$, done) sequence in Minibatch:
        If $done_i$ is **True**;
          $y_i = r_i$
        Else
          $y_i = r_i + \gamma \max_{a' \in A} Q'(s'_i, a')$
        Loss function $L = \frac{1}{N} \sum_{i=1}^{N-1} \left(Q(s_i, a_i) - y_i\right)^2$
        Using Adam, minimize the loss L and update the Q-value
      Foreach C step:
        Duplicate the weights of Q network into Q' network.

6 Evaluation of the proposed algorithm

In this section the relation between the delays for RT and NRT services that the users experience and the SINR is examined. The values for the SINR are in the range from 9 dB to 60 dB. Taking into consideration the relationship between PER (packet error rate) and BER (bit error rate) as considered in [16] the packet sizes for the different SINR case are calculated. SINR-adaptive term is used for the cases where the channel conditions for each user are taken into consideration, while SINR-non-adaptive is used for the ideal case.

In the literature, acceptable BER rates are in the range of 0.1% to 10%. BER below 0.1% are considered highly successful, while BER above 10% are not acceptable. In the light of this for an SINR of 9 dB the observed 11.26% BER is not acceptable; while 6.19% BER for SINR 13.43 dB, and 0.28% BER 35 dB and 0.09% BER for 60 dB, data transfer is achieved with high success.

Another performance output examined at different SINR values is the access delay. Figure 2(a) shows the access delay values for the SINR adaptive and SINR non-adaptive scenarios. As the SINR value increases, there are significant improvements in access delay due to the fact that the access delay is related to the packet size and uplink data rate, and the uplink data rate is also related to the SINR. Access delay changes only slightly when the SINR value drops from 60 dB to 35 dB, because the channel conditions are good at both SINR values. When the SINR value decreases from 35 dB to 13.43 dB, it is seen that the change in the 35 dB to 13.43 dB range is slightly faster than the change in the 13 dB to 9 dB range.

Figure 2(b) shows the transfer delay values for the SINR adaptive and SINR non-adaptive scenarios. With the increase in SINR values, the size of the usable packet in the MEC network is increased, and therefore the size of the data to be transferred within the MEC network is increased. As a result, the transfer delay for the SINR adaptive scenario increases while the delay for the SINR non-adaptive case is a constant at 0.27 ms.

Table 1. Simulation parameters

Parameter	Value
Bandwidth	20 MHz
Number of Users	1000
Number of APs	1000
Number of Antennas per AP	4
Cellular Network Area	4 km^2
Number of MEC Servers	10
MEC Server Data Capacity	(3,6) Gbit
MEC Server Processing Capacity	50 GHz
Backhaul Connection Speed	(100,300) Mbit/s
Episode	1000
Learning Rate (β)	0.99
Discount Factor (γ)	0.99
Memory Size	1000
Minibatch Size	256
Epsilon Decay	0.99975

In the second part of the simulations, we discuss the scenario where the requests from the users belong to the real time application (RT service). The real-time packets are generated according to the gaming traffic packet size equation given in [13]. The total packet size that will be

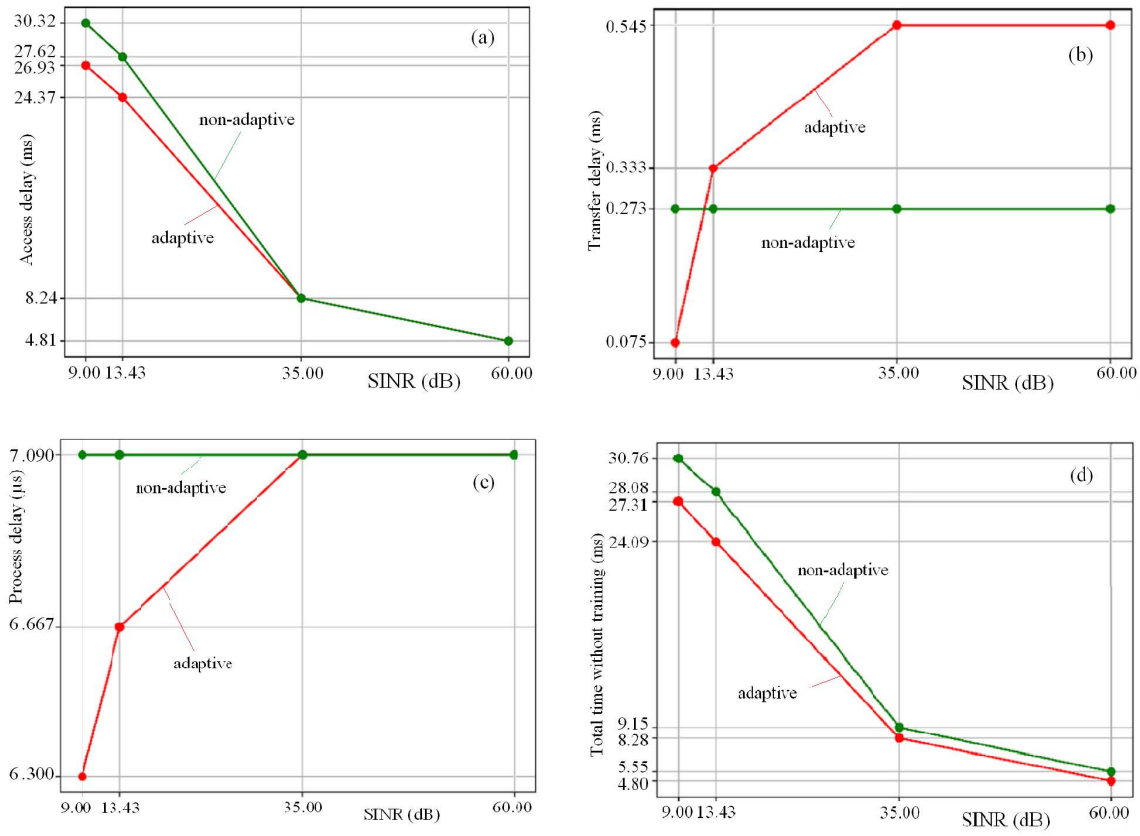

Fig. 2. (a) – Access delay, (b) – transfer delay, (c) – process delay, and (d) – total service time without training for non-real-time request

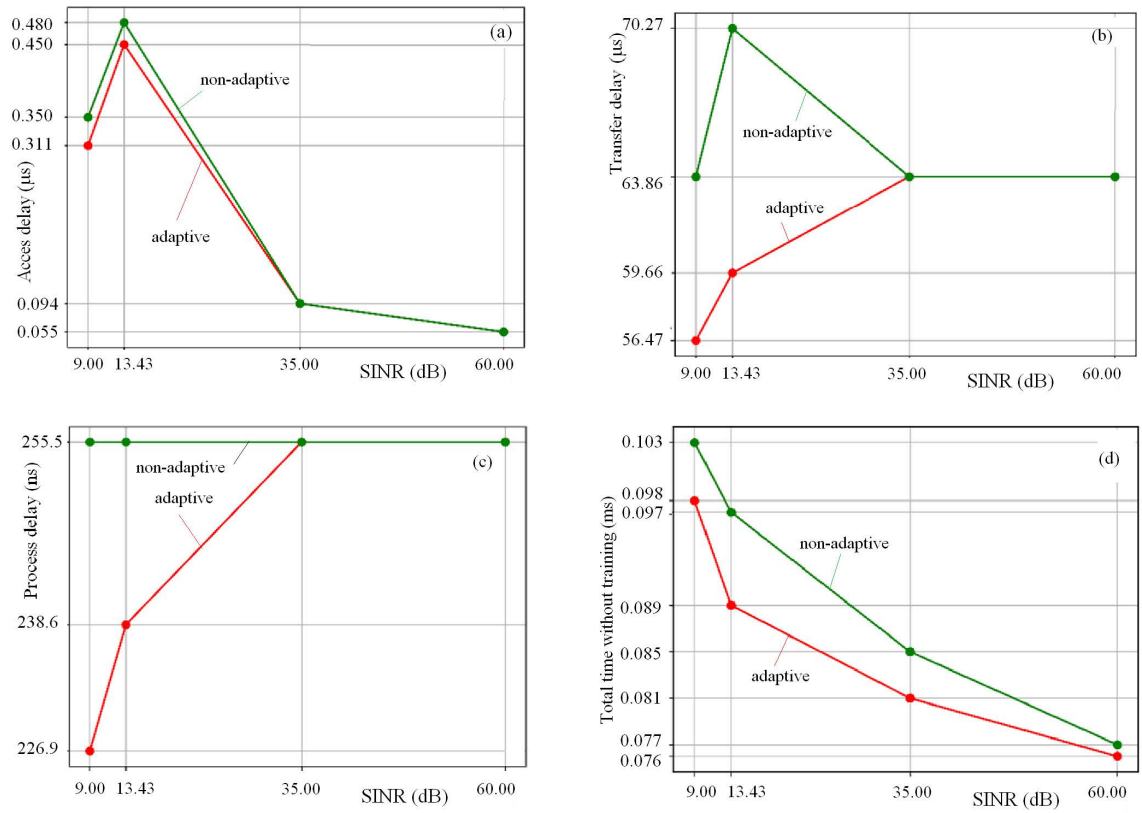

Fig. 3. (a) – Access delay, (b) – transfer delay, (c) – process delay, and (d) – total service time without training for real-time request

Table 2. Comparative average service time in ms; results for different processing capacity of MEC servers

GHz →	1	1.5	2	2.5	3	3.5	4
DRLRA (5)	330	220	160	140	130	115	90
Hybrid DQN	0.064	0.059	0.058	0.056	0.055	0.055	0.054

Table 3. Comparative average service time in ms; results for different data routing capacity

	Data routing capacity in Mbps			
	250	500	750	1000
DRLRA (5)	128	103	95	72
Hybrid DQN	0.056	0.030	0.022	0.018

Table 4. Comparative total service time in ms; results for different number of users

Number of users	200	160	120	80	40
ADORE (9)	30	17	10	7	3
Hybrid DQN	23.9	16.69	11.02	6.39	3.6

processed in the MEC network related to the RT traffic is adjusted for different channel conditions using (2) and (3). As theoretically expected, as the SINR value increases, the total packet size that can be processed increases.

In Fig. 3(a), the access delay for RT traffic from the cellular network to the MEC network is presented. Since the uplink data rate changes depending on the SINR value, there is a significant decrease in the access delay as the SINR value increases. When the results for the two scenarios are examined together, especially for low and medium SINR values, it is observed that the access delay is reduced by adapting the SINR related equations to the model. At high SINR values such as 35 dB and 60 dB, the access delay is equal due to the equal packet sizes.

Figure 3(b) shows the transfer delay values for the SINR adaptive and SINR non-adaptive scenarios. For low SINR values (9 dB and 13 dB) and SINR of 35 dB, the transfer delay is lower for the SINR adaptive scenario. For very high SINR values, the size of the usable packet in the MEC network increased, which result in the fact that the transfer delays almost converged for both scenarios.

In Fig. 3(c), the values for the process delay in the MEC network are given. Similar to the NRT traffic case, the processing delay for the SINR adaptive scenario increases with the SINR value. It is important to note that the transfer time does not depend only on the packet size but also on finding the most suitable server in the network and realizing the routing over the backhaul link with the highest capacity. This decision, which the proposed algorithm makes, is based on evaluating the suitability of the servers, their capacity and type of application, and the speeds of the backhaul links and is another important factor affecting the transfer delay. The results show that considering the more realistic case which is SINR adaptive has lower processing delay values.

The total service time without training is shown in Fig. 3(d). It can be seen that as the SINR value improves, there are improvements in the total service time for both scenarios, but SINR adaptive scenario performance output is better than SINR non-adaptive results.

Some analyzes were carried out to be able to examine them comparatively with some of the studies in the literature [5, 9]. In [5], an intelligent resource allocation algorithm (DRLRA) is introduced that can allocate computing and network resources that can adapt to changing MEC conditions using DRL. In the system model designed, it is stated that there are multiple MEC servers and there is a link between each server pair. On the other hand, in this study, servers that can run different types of applications are introduced and the average service time (including only transfer delay and processing delay) in the MEC network is examined taking into consideration several parameters: data routing capacity, processing capacity and *etc.* The performance results obtained when the experimental settings given in [5] were applied to the scenario in our study are given in Tab 2 and Tab. 3, comparatively. Table 2 shows the average service time values for different MEC processing capacities. Table 3 shows the average service time values experienced by users for different backhaul connection speeds. Both tables show that the Hybrid DQN algorithm presented in our study provides service to users with much less delay in both parameter value changes.

Compared to other studies in literature [9-12] our proposed model allows for a more detailed and comprehensive examination of the network performance especially in terms of service time and delays. While previous works have considered a single type of user requests, in this work we examine the delays for two different types of traffic: real-time traffic and non-real time traffic. Second, the various components contributing to the total service time delay that the user experiences, are defined and examined for different channel conditions: access delay, processing delay, transfer delay and the total service time without the training of the model. It is important to note that these different components, behave differently under different SINR values. It has been observed that for worse channel conditions the idealized delay values are higher than the more realistic channel aware case which our model allows to investigate. On the other hand, the

idealistic results *ie* SINR non-adaptive case are closer or lower for those with very good channel conditions.

Acknowledgements

This work has been developed under the FDK-2020-21953 project, funded by Ege University Research Fund.

7 Conclusions

The hybrid DQN based algorithm proposed in this work allows us to examine in detail the service delays and proposes a way to ensure that users are provided with the most efficient service both in respect to the service time and the optimal utilization of MEC network resources. The proposed algorithm establishes the optimal load balancing for the MEC application servers where multiple instances of requests occurs simultaneously which in turn allows us to serve more users in a shorter time. It is based on a correlation between the DQN reward values and the MEC server capabilities including the backhaul link capacities, forcing the agent to find the optimal path for routing the user traffic through the MEC network. Furthermore, the system considers the heterogeneity of user requests (real-time or non-real-time file type and size). The designed system allows for detailed examination of the effects of various delay components like access delay, process delay and transfer delay. Furthermore, it has been shown that when a more realistic, channel aware scenario is considered the proposed algorithm and optimal load balancing in the MEC network can lead to reducing the total service time experienced by the users.

References

[1] T. Yang, Y. Hu, M. C. Gursoy, A. Schmeink and R. Mathar, "Deep reinforcement learning based resource allocation in low latency edge computing networks", *15th International Symposium on Wireless Communication Systems* (ISWCS), pp. 1-5 , 2018.

[2] F. A. N. Qi, L. Zhuo and C. Xin, C, "Deep Reinforcement Learning Based Task Scheduling in Edge Computing Networks", *IEEE/CIC International Conference on Communications in China* (ICCC), pp. 835-840, 2020.

[3] G. S. Rahman, T. Dang and M. Ahmed, "Deep reinforcement learning based computation offloading and resource allocation for low-latency fog radio access networks", *Intelligent and Converged Networks*, vol. 1, no. 3, pp. 243-257 , 2020.

[4] Y. Yang, Y. Hu and M. C. Gursoy, "Deep Reinforcement Learning and Optimization Based Green Mobile Edge Computing", *IEEE 18th Annual Consumer Communications & Networking Conference*, (CCNC) pp. 1-2, IEEE , 2021.

[5] J. Wang, L. Zhao, J. Liu, and N. Kato, "Smart resource allocation for mobile edge computing: A deep reinforcement learning approach", *IEEE Transactions on emerging topics in computing*, 2019.

[6] Z. Ali, Z. H. Abbas, G. Abbas, A. Numani and M. Bilal, "Smart computational offloading for mobile edge computing in next-generation Internet of Things networks", *Computer Networks*, 198, 108356 , 2021.

[7] H. Wang, H. Ke, G. Liu, and W. Sun, "Computation migration and resource allocation in heterogeneous vehicular networks: a deep reinforcement learning approach", *IEEE Access*, vol. 8, pp. 171140-171153 , 2020.

[8] J. Chen, S. Chen, Q. Wang, B. Cao, G. Feng and J. Hu, "iRAF: A deep reinforcement learning approach for collaborative mobile edge computing IoT networks", *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 7011-7024 , 2019.

[9] A. Samanta and Z. Chang, "Adaptive service offloading for revenue maximization in mobile edge computing with delay-constraint", *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3864-387 , 2019.

[10] L. Zhao and J. Liu, "Optimal placement of virtual machines for supporting multiple applications in mobile edge networks", *IEEE Transactions on Vehicular Technology*, vol. 67, no. 7, pp. 6533-6545 , 2018.

[11] S. Z. Huang, K. Y. Lin and C. L. Hu, "Intelligent task migration with deep Qlearning in multiaccess edge computing", *IET Communications*, vol. 16, no. 11, pp. 1290-1302 , 2022.

[12] A. Abouaomar, Z. Mlika, A. Filali, S. Cherkaoui and A. Kobbane, "A deep reinforcement learning approach for service migration in mec-enabled vehicular networks", *IEEE 46th conference on local computer networks* (LCN), pp. 273-280 , 2021.

[13] C. M. Orange, "N. KPN, DoCoMo, S. T-mobile, vodafone and telecom italia", r1-070674: Lte physical layer framework for performance verification; 3gpp tsg ran wg1, *Tech. Rep. [Online]* http://www. 3gpp. org/ftp/tsg ran/WG1 RL1/TSGR1 48/Docs/R1-070674. zip.

[14] T. V. Chien and E. Björnson, *Massive MIMO communications in 5G mobile communications*, pp. 77-116, Springer.

[15] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization", *arXiv preprint*, arXiv:1412. 6980 , 2014.

[16] A. A. B. Salem, Y. W. Chong, S. M. Hanshi and T. C. Wan, "On the optimizing of LTE system performance for SISO and MIMO modes", *3rd International Conference on Artificial Intelligence, Modelling and Simulation* (AIMS),pp. 412-416 , 2015.

Received 7 February 2023

Önem Yıldız is currently pursuing the PhD degree at the Department of Electrical and Electronics Engineering, Ege University, Izmir, Turkey. She received her BSc and MSc degrees from the Department of Electrical and Electronics Engineering, Ege University in 2015 and 2017, respectively. Her current research interests include integration of deep learning methods in next generation networks, resource allocation, and mobile edge computing.

Radosveta Ivanova Sokullu (Senior Member, IEEE) received her MSc and PhD in Radio Communications from the Technical University Sofia in 1991. She is currently a Professor at the Department of EEE, Ege University, Chair of the Telecommunications Branch as well as Head of the Wireless Lab at the department. She is an IEEE Senior Member. Her main research interests are next generation networks, resource allocation and machine learning in wireless communications, PHY and MAC layer protocols for WSN and IoT, short range and body area networks, interference and interoperation issues in LTE networks.