IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 18, 2021, accepted March 14, 2021, date of publication March 31, 2021, date of current version April 16, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3070104

A User-Centric QoS-Aware Multi-Path Service Provisioning in Mobile Edge Computing

SAIF U. R. MALIK[1], TEHSIN KANWAL[2], SAMEE U. KHAN[3], HASSAN MALIK[4], (Member, IEEE), AND HARIS PERVAIZ[5], (Member, IEEE)

[1]Cybernetica AS, 12618 Tallinn, Estonia
[2]Department of Computer Science, COMSATS University Islamabad, Islamabad 45550, Pakistan
[3]Department of Electrical and Computer Engineering, Mississippi State University, Starkville, MS 39762, USA
[4]Department of Computer Science, Edge Hill University, Ormskirk L39 4QP, U.K.
[5]School of Computing and Communications (SCC), Lancaster University, Lancaster LA1 4YW, U.K.

Corresponding author: Saif U. R. Malik (saif.rehmanmalik@gmail.com)

This work was supported by the Estonian Personal Research under Grant 920.

ABSTRACT Recent development in modern wireless applications and services, such as augmented reality, image processing, and network gaming requires persistent computing on average commercial wireless devices to perform complex tasks with low latency. The traditional cloud systems are unable to meet those requirements solely. In the said perspective, Mobile Edge Computing (MEC) serves as a proxy between the things (devices) and the cloud, pushing the computations at the edge of the network. The MEC provides an effective solution to fulfill the demands of low-latency applications and services by executing most of the tasks within the proximity of users. The main challenge, however, is that too many simultaneous service requests created by wireless access produce severe interference, resulting in a decreased rate of data transmission. In this paper, we made an attempt to overcome the aforesaid limitation by proposing a user-centric QoS-aware multi-path service provisioning approach. A densely deployed base station MEC environment has overlapping coverage regions. We exploit such regions to distribute the service requests in a way that avoid hotspots and bottlenecks. Our approach is adaptive and can tune to different parameters based on service requirements. We performed several experiments to evaluate the effectiveness of our approach and compared it with the traditional Greedy approach. The results revealed that our approach improves the network state by 26.95% and average waiting time by 35.56% as compared to the Greedy approach. In addition, the QoS violations were also reduced by the fraction of 16.

INDEX TERMS Mobile edge computing, Internet of Things (IoT), quality of service (QoS), service provisioning, multi-path routing, high level petri nets.

I. INTRODUCTION

In the digital era, we are experiencing an explosive increase in the number of mobile devices accessing the wireless network. Advancements in cloud computing (CC) and wireless communication technology have been the motivating factor behind such explosive growth. The total number of mobile devices is expected to reach 75 billion by 2020, while the volume of data is expected to exceed 24.3 exabytes/mo [1]. Since the emergence of the IoT, Edge and Fog computing paradigm, smartphones and devices have undergone a huge transformation in the way they can be used. The more advanced technologies are evolving and attracting consumers, such as facial recognition, virtual reality, online immersive gaming, and natural language processing. These applications typically require high availability and are data intensive or computing intensive, requiring high resource and energy consumption. Mobile devices are known for the resource scarcity, having limited computational power and battery life. The conflict between the intensive application of compute/data and resource-constrained mobile devices prevents the efficient adaptation of new paradigms [2].

Mobile Edge Computing (MEC) offers a viable solution for tackling the tension between computing or data-oriented applications and constrained mobile devices [3]. The term mobile edge computing was regulated by the European Telecommunications Standards Institute (ETSI) and by the Industry Specification Group (ISG). According to

The associate editor coordinating the review of this manuscript and approving it for publication was Fan-Hsun Tseng.

ETSI, MEC generates an IT service based environment and cloud computing functionality on the edge of the mobile network, within the Radio Access Network (RAN) and mobile subscribers [4]. To reduce long gigabit transmission delays in the cloud, the MEC is designed to support pervasive high-performance computing, especially for delay-sensitive applications. In addition, MEC shifts publicly accessible computing resources, such as cellular base stations and WiFi access points, to the edge of the radio access network, so that mobile users (MUs) can easily outsource computational tasks [5]. However, utilizing the edge resources efficiently to maximize the service provisioning to a greater number of applications while maintaining the specified Quality of Services (QoS) is a complex task.

There are recent studies, such as [12], [14], [20], and [27] available in the literature that discusses the resource allocation and offloading in a MEC environment. However, to the best of our knowledge, an overlapping multi-cell environment, where a user is under the coverage region of multiple base stations is not studied in the literature. Multi-path routing is well studied in Data Center Network and Mobile Adhoc Networks (MANETs) and has proved to be effective in maintaining a unified state of the network. With the inception of 5G technologies, the density of base stations has been increasing to reach up to 50 base stations per KM [6]. In the said perspective, in this research, we have considered a densely deployed base stations (as shown in Fig. 1) to exploit the multiple available paths, so as to route the service request of the users. We propose an adaptive QoS-aware Multi-path Service Provisioning (QMSP) approach to maximize the number of requests served by the edge server. The QMSP initially identify the number of available paths based on the QoS attributes of the service request. Once the paths are identified, the requests are routed through the specified paths that ensures the QoS. As multiple paths are available for a single request to serve, the scheduler selects the shortest path initially to serve the request. As the number of requests routed to the same path increases, the congestion on that particular link increases that results in a longer waiting time or increased packet drop rate. In QMSP, when a certain congestion threshold is reached, then the requests are redirected to an alternative path to process the request. To demonstrate the effectiveness of proposed QMSP, we performed several experiments to evaluate: **(a)** the overall congestion state of the network, **(b)** the effect on waiting time of the requests, and **(c)** the number of QoS violations. We also compared our approach with the traditional Greedy approach. The results highlighted that the network state and average waiting time is improved by 26.95% and 35.56%, respectively.

Moreover, the QoS violations were also reduced by the fraction of 16% as compared to the Greedy approach. Furthermore, in this study, we tried to minimize the level of abstraction in a MEC environment, we have performed formal modeling and analysis using High-Level Petri Nets (HLPN). The HLPN(s) are used for system simulation, it also provide mathematical representation of proposed systems so as to analyze the behavior and structural properties [7]. The detailed discussion about the proposed approach and results are provided in the later sections. The contributions to this paper are summarized as follows:

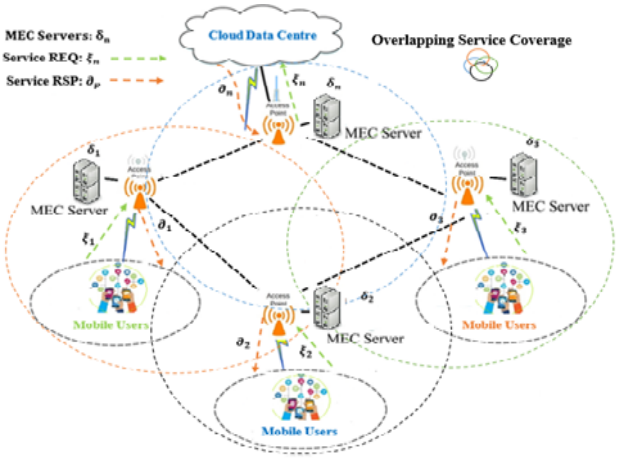

FIGURE 1. An overlapping base station system model of MEC environment.

- introducing a novel QoS-aware Multi-path Service Provisioning (QMSP) approach that exploits the multiple paths available in a multi-cell MEC environment to efficiently route the service requests;
- formalizing the QMSP approach using High Level Petri Nets;
- conducting the simulation experiments and comparison of the proposed strategy with the Greedy approach.

The remainder of the paper is organized as follows. Section 2 will review some of the related work done in the field of resource allocation and offloading in the MEC environment. The mathematical model and formulation of the problem will be discussed in Section 3. Section 4 will discuss the proposed QMSP approach along with the formal modeling of QMSP using HLPN. Section 5 will highlight the detailed discussion, evaluation results, and the comparison of our approach with the greedy approach. Finally, Section 6 concludes this paper, followed by the references.

II. RELATED WORK

This section includes the related research on task offloading and resource allocation in the MEC systems. In the task offloading, the task is assigned to the appropriate MEC server to be executed, which can results in minimum energy consumption of data transmission and thus increase the overall system utility [8]. A cooperative game-based job scheduling [9] is used to obtain the criteria of maximum job execution for meeting the target deadlines in MEC model. The proposed framework improves quality of service of the MEC system. In another work [10], by considering service latency, a distributed task scheduling algorithm increase throughput, reduce latency and improve device performance. In [11], a hybrid framework that incorporates wireless power transfer

technology uses an algorithm for resource allocation. It optimize the total energy consumption based on service latency constraints. A game-based computation offloading (GCO) algorithm has been proposed for the issue of task offloading. Experimental results proves that the proposed approach reduces energy consumption with increase in system throughput [12]. In [13], a new task offloading algorithm was proposed to reach the user tasks deadline while using low energy consumption. The problem formulation for task offloading has been performed in [14] as a non-linear program. The proposed offloading mechanism minimize the task delay and the battery consumption of mobile users, but did not consider the users mobility in the MEC environment. In [15], a task offloading algorithm based on the time and energy model has been proposed for the formalization of optimization based metrics. The proposed algorithm reduces the time for completion of task and energy utilization. In another work task optimization has been performed to reduce task latency and the mobile users energy consumption. The work given in 16] discussed the mobile users for the energy consumption in MEC system. The total latency cost and energy consumption assessment criteria for algorithm performance have not been precisely optimized. Power consumption of mobile user equipment UEs and task execution delay has been considered for optimization in the proposed online task unloading algorithm. It proves effective for multi-objective optimization and also provide improvements in the user's Quality of Service (QoS) in MEC systems [17]. A different approach to MEC systems designed to achieve quality-of-result and service response time criteria. The suggested scheme is used for edge applications with a certain degree of tolerance for quality loss [18]. The problem of computation offloading has been addressed in [19], it minimize the energy consumption of multiple mobile users in the multi-input multi-output (MIMO) network. In another work computation task offloading with multi-user resource allocation is considered which claims to reduce user energy consumption and delay [20]. A Lyapunov optimization approach based on the task offloading scheme has been proposed in [21], [22] to reduce the delay in the implementation of the proposed system for energy consumption. Multi-user multi-MEC server task offloading design [23] can reduce the pressure of single link communication in order to improve based network performance and reduce the amount of communication overhead. The proposed model was designed to maximize resource allocation energy. The problem of maximization was based on the Mixed-Integer Non-Linear Program (MINLP). The proposed optimal solutions for user selection and energy utilization have been achieved by following the transformation problem. Computational offloading decisions, physical resource block allocation and allocation of MEC resources were formulated as an optimization problem in [24]. Resource allocation and enormous computation resources in centralized cloud computing center has been considered for cloud MEC system in [25]. Cloud–MEC collaborative computation offloading has been investigated and proposed a scheme based on approximation and game theory. A cache-assisted scheme optimize task offloading and preserve a minimum delay in the mobile edge work. With this cache-assisted MEC system, the mobile user can upload and execute custom programs on the MEC server, while the server is configured for software reusability [26]. In another work [5], the computation delay and energy consumption of mobile MU users is minimized. Several parameters such as joint optimization of service caching placement, decision-making on task offloading and allocation of resources have been investigated. The problem of service optimization and request routing in MEC-enabled multi-cell networks is given in [27]. The proposed work includes multidimensional dimensions such as storage, energy computation and network communication. A randomized rounding technique-based algorithm has been suggested that is designed to maintain close-to-optimal performance and efficiently use the available resources. The approach given in [28] considers multi-user MEC scenario with server where the user equipment (UEs) may choose to unload their tasks through access point to the MEC server for QoS and reduced system. An ant colony optimization (ACO) algorithm has been designed to minimize system cost, provide better QoS and increase the MEC system performance. A mobility-aware task offloading problem has been studied in the MEC environment [29]. The mobility of edge users is used with a deep-learning approach to identify connectivity patterns and forecast potential user trajectories. An online approach is proposed to solve the quality-of-service (QoS) task offloading problem and predict user paths in real-time.

Machine learning methods like Deep Reinforcement Learning (DRL) based schemes with Q-learning models like DDQN and DRL have been efficiently utilized for task optimization and resource allocation in multi-user MEC systems [8], [30]–[35]. An adaptive task offloading and resource allocation approach based on DRL is given in [8], it decides about the task offloading and allocation of computational resources for these tasks. In addition, the mobile user equipment (UE) for mobility between base stations (BSs), is also considered in this scheme. It has been claimed that the proposed algorithm shows improved system performance with utility to satisfy the user service provider's requirements. In security and privacy context, there are various adversarial attacks in MEC environment like wireless jamming, denial of service, man-in-the-middle, spoofing attacks, privacy leakage, virtual machine manipulation, and unintended information injection [36], [38]. Recently, machine learning based approach for security and privacy in MEC have been studied from multidimensional perspectives. Deep Learning (DL) has been used for the detection of cyber-attack MEC networks [36]. An edge caching security mechanism as given in [37] provide anti-jamming approach for mobile offloading, physical authentication, and friendly jamming.

The literature in [38] studied Q learning based physical security in fog computing, it mitigate the smart QoS and security and privacy related parameters. A detailed comparative analysis of related work is given in Table 1. It is also clear

TABLE 1. A comparative analysis of related work in MEC systems.

Work	Techniques	Task offloading	Resource Allocation	Energy Optimization	Smart QoS	Security/ Privacy
[8]	Deep Reinforcement learning (DRL).	✓	✓	✓	✓	×
[10]	Distributed task scheduling with optimal decision engine.	✓	▪	✓	▪	×
[11]	MEC-Wireless Power Transfer (WPT) design with Lagrange duality method.	×	✓	✓	×	×
[12]	Game-based computation offloading (GCO) with Greedy-Pruning algorithm[12]	✓	×	✓	×	×
[13]	Partial and Binary computation offloading using convex optimization and Computation mode selection	✓	×	✓	×	×
[14]	Software defined network based optimal sub-problem solving approach.	✓	×	✓	×	×
[15]	TaSRD scheme for dependent and non-dependent subtask scheduling	✓	×	▪	×	×
[16]	Linear relaxation, semidefinite relaxation (SDR), an exhaustive search-based approach and SDR-based approach for different CPU frequency cases.	✓	×	✓	×	×
[17]	Lyapunov optimization based algorithm for local execution and computation offloading policy.	✓	×	▪	×	×
[18]	Optimization framework, MobiQoR, for optimizing the quality of results of all MEC nodes with offloading strategy.	✓	×	✓	▪	×
[19]	Time-Division Multiple Access (TDMA) and Orthogonal Frequency-Division Multiple Access (OFDMA) based approach.	✓	✓	▪	×	×
[26]	Approximation and game theory.	✓	×	×	×	×
[27]	Randomized rounding technique.	✓	✓	✓	×	×
[28]	Reinforcement learning (RL) using Dyna architecture and post decision learning.	✓	×	×	▪	✓
[29]	Deep learning in user mobility context.	✓	✓	✓	✓	×
[30]	Deep Reinforcement learning (DRL).	✓	×	×	▪	×
[31]	Deep-Q Network (DQN).	✓	✓	✓	▪	×
[32]	DRL based Q-learning.	✓	✓	✓	▪	×
[33]	Deep learning based task scheduling	✓	✓	✓	×	×
[34]	Reinforcement learning (RL), deep Q networks, recurrent neural network long short-term memory (RNN-LSTM), and deep reinforcement learning combined with LSTM (DRL-LSTM).	✓	✓	✓	▪	×
[35]	Asynchronous-Advantage-Actor-Critic (A3C) learning and Residual Recurrent Neural Network (R2N2).	✓	✓	✓	✓	×
[36]	Deep learning.	×	×	×	×	✓
[37]	Q-learning.	✓	×	×	×	✓
[38]	Reinforcement based Q-learning.	✓	×	×	▪	✓

from the related work comparison that most of the work has been performed for service offloading and resource allocation in MEC systems and performs improvement in mobile user equipment energy consumption. Security and privacy aspect have not been accommodated in a hybrid manner in proposed solutions. It is important to note that MEC users requires smart and application oriented QoS. In this regard there is need to propose an efficient MEC system routing algorithm that must be adaptively and automatically tune the network. Our proposed work provides QoS in terms of specific network parameters like response time and network link availability. Moreover, it has the capability to handle network congestion which will results in an improved response time for specific user application requests. We have described these QoS parameters in detail in Section III.

III. SYSTEM MODEL AND PROBLEM FORMULATION

A general MEC architecture usually comprised of mobile user, MEC server, services, and a traditional Cloud Data Center (CDC). In our model, we consider MEC architecture consisting of a set X having $|X|$ number of mobile users (represented as $N_{i\in X}$) and a set M having $|M|$ MEC servers, represented as $\delta_{n\in M}$. Each $\delta_{n\in M}$ has the following capabilities, as shown in (1).

$$\delta_{n\in M} = (S, C, B^u, B^d)_n, \tag{1}$$

where S_n represents the storage capacity of δ_n, C_n represents the computational power of δ_n, and B_n^u, B_n^d represents the upload and download bandwidth capacity of δ_n. The mobile users subscribe to different services provided by different service providers, such as online gaming, augmented reality, and video/audio streaming. Every service has a different requirement in order to perform successfully as per QoE of the users. A service, represented as ξ_i, hosted at δ_n can be represented as $\xi_{i\in S}^{\delta_n}$, where i belongs to the set of service S. The requirements of the services can be represented as in (2).

$$\xi_{i\in S}^{\delta_n} = \left(\xi^s, \xi^c, \xi^{B^u}, \xi^{B^d}\right)_i^{\delta_n}, \tag{2}$$

where ξ_i^{s,δ_n} represents the storage requirement of the ξ_i hosted at δ_n, ξ_i^{c,δ_n} represents the computational requirement of the ξ_i, and $\xi_i^{B^u,\delta_n}$, $\xi_i^{B^d,\delta_n}$ represents the uplink and downlink bandwidth requirement of the ξ_i hosted at δ_n, respectively. Every ξ_i has QoS attributes associated to it, such as in online banking availability and reliability is of top concern. Similarly, in online gaming service minimum delay or latency is of utmost importance. The online users can arbitrarily dispersed over the coverage region of the base stations. Moreover, in a densely deployed base station environment the coverage regions may overlap. In this research, we assume a densely deployed base station environment, where the coverage regions are overlapped. The user generates a service request that inherently has an associated QoS attribute determining the QoE of the users. The service request specification can be represented as $REQ_{\xi_i}^{N_i} = \{\partial, \alpha\}_{\xi_i}$, where ∂ and α state the response time and reliability requirements for ξ_i. The $REQ_{\xi_i}^{N_i}$ can be routed to a nearby δ_n given that the necessary specifications of the requests are met. If there is no such δ_n available, then the request is routed to the CDC, which may cause additional delay and thus must be avoided. The network operator is responsible for placing services and routing the user to them. To potentially increase the availability and performance of the services, the network operator may choose to replicate services to different δ_n. In such cases, routing a $REQ_{\xi_i}^{N_i}$ to the respective and appropriate δ_n to provide best services becomes a crucial task.

QoS-aware Multi-path Service Provision (QMSP) Problem: Given a set ψ of U mobile users, generating a $REQ_{\xi_i}^{N_i} = \{\partial, \alpha\}_{\xi_i}$ for the ξ_i hosted at multiple δ_n, the QMSP problem is to find the best δ_n that can fulfill the service request, such that: (a) the $\partial_p \leq \partial$, where ∂_p is the response time provided by the δ_n and (b) the $\alpha_p \geq \alpha$, where α_p represents the expected reliability of the service being requested.

We define $\pi_{\delta_n}^{N_i}$ as a set of all δ_n that can provide the services to the mobile node N_i. We have considered the MEC overlapping environment as an undirected graph, represented as $G = (V, E)$, where V denotes the nodes, represented as N (MEC server and mobile users) in the network and E represents the edges connecting the nodes (depicted in Fig. 2). A link between mobile node N_i and δ_n (if it exists) has a communication cost or weight associated to it, represented as

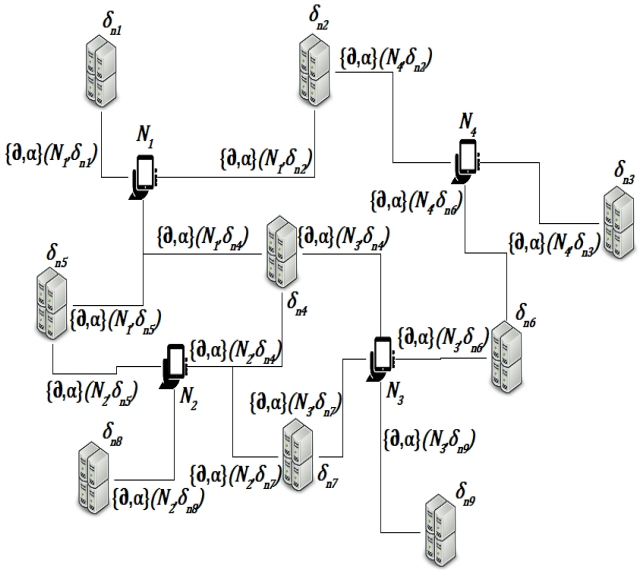

FIGURE 2. An undirected graph of MEC overlapping environment.

$w(N_i, \delta_n)$, where (N_i, δ_n) is a link between N_i and δ_n. Without loss of generality, we assume that $w(N_i, \delta_n) = w(\delta_n, N_i)$.

The weight of the communication link between $w(N_i, \delta_n)$, can generally be calculated as given in (3):

$$w(N_i, \delta_n) = \frac{D(N_i, \delta_n)}{pd} + \frac{s}{\beta_{N_i\delta_n}}, \tag{3}$$

where $D(N_i, \delta_n)$ is the distance between the nodes, pd represents the propagation delay of the medium, s is the message size in kilobytes, and $\beta_{N_i\delta_n}$ is the available network bandwidth between the nodes. It ensures that multiple users can harmoniously access and share a single communications medium, needs some kind of equitable link sharing. If a communication link provide R data rate shared by x active network users (with a minimum limit of one data packet in queue), every network user typically gains a throughput of approximately R/x, if fair queuing best-effort communication policy is assumed. Therefore, the $\beta_{N_i\delta_n}$ of the link will become (4):

$$\beta_{N_i\delta_n} = R_{N_i\delta_n}/x. \tag{4}$$

Considering the request specification that we have used previously, $REQ_{\xi_i}^{N_i} = \{\partial, \alpha\}_{\xi_i}$, the ∂ between N_i, δ_n, represented as $\partial_{\xi_i}^{N_i,\delta_n}$, can be calculated as(5):

$$\partial_{\xi_i}^{N_i,\delta_n} = \frac{e_{N_i}}{p_{\delta_n}} + \frac{d^{N_i,\delta_n}}{r^{N_i,\delta_n}} + w_{\delta_n}, \tag{5}$$

where e_{N_i} is the execution requirements (no. of processor cycles) of the request, p_{δ_n} represents the processing speed of the selected δ_n, d^{N_i,δ_n} represents the amount of data that is to be communicated over the link, r^{N_i,δ_n} represents the data transmission rate, and w_{δ_n} is the mean waiting time for δ_n to be free. The other specified QoS attribute is α that is mainly

computed based on the availability of the link. The network link availability $A(N_i, \delta_n)$, can be calculated as (6):

$$A(N_i, \delta_n) = \frac{u_{(N_i,\delta_n)}}{t_{(N_i,\delta_n)}}, \tag{6}$$

where $u_{(N_i,\delta_n)}$ represents the expected uptime of the medium or link and $t_{(N_i,\delta_n)}$ is the total time which is (uptime + downtime) of the communication link. The average link availability should be the real line interval. Suppose we have a constant $c > 0$, then average link availability, represented as $\alpha_{\xi_i}^{N_i,\delta_n}$, can be calculated as (7):

$$\alpha_{\xi_i}^{N_i,\delta_n} = A_{\sigma(N_i,\delta_n)} = \frac{1}{c}\int_{t=0}^{c} A(t)\, dt. \tag{7}$$

Once the request attributes are computed for all of the available servers, the job of the network operator is to select the best to process the REQ_{ξ_i}. The feasible servers that satisfy the requested QoS attributes of the $REQ_{\xi_i}^{N_i} = \{\partial, \alpha\}_{\xi_i}$ can be aggregated as given in (8):

$$\pi_{\delta_n}^{N_i} = \left\{ \begin{array}{l} 0 \le Max\left(\partial_{\xi_i}^{N_i,\delta_n}\right) \le \partial \\ 0 \le \alpha \le Min\left(\alpha_{\xi_i}^{N_i,\delta_n}\right) \end{array} \right\}_{\forall N_i \in X \wedge \forall \delta_n \in M}. \tag{8}$$

Once populated, the δ_n providing the $Min\left(Max\left(\partial_{\xi_i}^{N_i,\delta_n}\right)\right)$ and $Max\left(Min\left(\alpha_{\xi_i}^{N_i,\delta_n}\right)\right)$ from all of the available $\delta_n \in \pi_{\delta_n}^{N_i}$ is selected for the service request.

IV. QoS-AWARE MULTI-PATH SERVICE PROVISIONING

In this section, we present our main contribution of this work; a QoS-aware Multi-path Service Provisioning (QMSP) algorithm in MEC environment. As stated above, service provision is mainly based on two core aspects, which are mobile users (N_i) and MEC servers (δ_n). Our proposed algorithm (depicted in Fig.3) is also based on the core elements. The input to our algorithm is the request generated by the mobile user for a specific service. As soon as the request is generated, the first step is to identify the servers that can provide the requested service. Once the servers are identified, the next step is to compute the QoS attributes related to the services.

The QoS attributes is computed for each of the server that can provide the requested service. The QoS attributes are calculated according to the equations discussed in previous section.

Once the attributes are calculated for each server, all the servers that satisfies the minimum requested attribute requirements are selected. Finally, the best among the selected server that fulfills both of the service attribute is selected to process the request of the user. The key barriers to offloading computation in MEC are the network bandwidth and latency [39]. If too many simultaneous offloading requests are made through the wireless access to the cloud, then severe interference may be created, resulting in a reduced data rates for data transmission [40], [41]. In the said perspective, overlapping coverage of the services can be exploited to process the user requests efficiently. Our approach aims to fulfill the QoS requirements, rather than choosing the shortest path. The paths are computed according to the equations listed above. Suppose (in Fig. 2) N_1, N_2, N_3, and N_4 generates a *REQ* for ξ_i, which is hosted at several servers including $\delta_1, \delta_2, \delta_4, \delta_5$, and δ_8. The cloud operator analyzes the request and identify the server(s) ($\pi_{\delta_n}^{N_i}$) that can efficiently process the request. Let δ_4 being the closest to all of the requesting nodes is selected to process the requests. As δ_4 starts to process the requests, the congestion level also starts increasing. When the congestion level reaches a certain threshold value (Γ), all the requests generated from the nodes N_1, N_2, N_3, and N_4 are now processed by the other servers from the list $\pi_{\delta_n}^{N_i}$. It is noteworthy that our approach is mainly related to routing, we are not proposing a novel strategy on computational offloading. However, our approach can perform with any offloading strategy. The complexity of our algorithm is $O(n^2)$.

1. **Input**: $REQ_{\xi_i}^{N_i}$
2. **Output**: $\pi_{\delta_n}^{N_i}$
3. **for** $n \leftarrow 1$ to $|M|$ **do**
4. Populate $\pi_{\delta_n}^{N_i}$
5. **endfor**
6. **for all** $\delta_n \in \pi_{\delta_n}^{N_i}$ **do**
7. Compute $\partial_{\xi_i}^{N_i,\delta_n} \wedge \alpha_{\xi_i}^{N_i,\delta_n}$
8. **If** $\left(\partial_{\xi_i}^{N_i,\delta_n} \le \partial_{where\ \xi_i=\xi_i}\right) \wedge \left(\alpha \le \alpha_{\xi_i}^{N_i,\delta_n}{}_{where\ \xi_i=\xi_i}\right)$ **then**
   Update $\left(\pi_{\delta_n}^{N_i} \leftarrow \partial_{\xi_i}^{N_i,\delta_n}, \alpha_{\xi_i}^{N_i,\delta_n}\right)_{\delta_n=\delta_n}$
9. **else** Continue
10. **endif**
11. **end for**
12. Select $\left(\mathbf{Max}\left(\partial_{\xi_i}^{N_i,\delta_n}\right), \mathbf{Min}\left(\alpha_{\xi_i}^{N_i,\delta_n}\right)\right)_{\delta_n=\delta_n} \in \pi_{\delta_n}^{N_i}$

FIGURE 3. The steps involved in QMSP approach.

A. FORMAL MODELING AND ANALYSIS OF QMSP

We perform formal modeling of QMSP approach using HLPN in this section. As stated previously, the formal models help: *a*) analyze the interconnected system components and processes, *b*) provide in depth analysis of the flow of information, and *c*) depicts the information processing. The HLPN(s) are widely used in literature to formally model the systems. In formal modeling using HLPNs, we identify the required data types, Places (P), and mappings.

The description of the QMSP is described in the previous section, and now, we can define proposed approach-based formulas (preconditions and post conditions) to map on mathematical transitions. (Readers are encouraged to read Ref. [7], [40], and [44] for further details about the use of

TABLE 2. Places and mappings of data types involved in QMPS approach HLPN.

Places	Mapping	Place Description
φ (M-USERS)	$\mathbb{P}\,(Rq_{\xi_n}^{mob_n})$	*Place holds request set of n mobile user devices.*
φ (MEC-SRV)	$\mathbb{P}(\pi_{\delta_n}^{N_i} \times \partial_{\xi_i}^{N_i,\delta_m} \times \alpha_{\xi_i}^{N_i,\delta_m})$	*Place holding set of δ_m MEC servers, service response and service availability values of QoS.*
φ (TMP-SRV-VAL)	$\mathbb{P}(\pi_{\delta_n}^{N_i} tmp \times \partial tmp_{\xi_i}^{N_i,\delta_m} \times \alpha tmp_{\xi_i}^{N_i,\delta_m})$	*Place holding set of δ_m MEC servers, temporary service response and service availability values of QoS attributes.*
φ (SEL-SRV)	$\mathbb{P}\,(Sel\pi_{\delta_m}^{N_i})$	*Place holding set of δ_n selected or allocated servers to n mobile user requests.*

TABLE 3. Data types Used In QMPS.

Data Types	Data Type Description
$Rq_{\xi_n}^{mob_n}$	*An integer and string type value to repesent n mobile user request.*
$\pi_{\delta_m}^{N_i}$	*The integer type value for δ_m MEC servers.*
$\pi_{\delta_m}^{N_i} tmp$	*The temporary integer type value for δ_m MEC servers.*
$\partial_{\xi_i}^{N_i,\delta_m}$	*The integer type value for service response time in secs.*
$\alpha_{\xi_i}^{N_i,\delta_m}$	*The integer type value for average service availability time of the link in seconds.*
$\partial tmp_{\xi_i}^{N_i,\delta_m}$	*The integer type value for temporary updated service response availability time of the link in seconds.*
$\alpha tmp_{\xi_i}^{N_i,\delta_m}$	*The integer type value for temporary updated service availability time of the link in seconds.*
$Sel\pi_{\delta_m}^{N_i}$	*An integer type value for selected server id.*

HLPN). The HLPN model of the QMSP approach is given in Fig 4. The descriptions of the variable type, places, and description involved in HLPN for the proposed QoS-aware Multi-path Service Provisioning (QMSP) algorithm is given in Table 2 and Table 3. The HLPN model of the QMSP algorithm takes inputs request $REQ_{\xi_i}^{N_i} = \{\partial, \alpha\}_{\xi_i}$ from different mobile user devices and store it in place M-USERS in MEC system. Each mobile user service request consists of specific QoS attributes depending upon user requirement in distributed environment. The transition *Populate* places mobile users request by assigning request variable $R_{\xi_i}^{mob_i}$ to $\pi_{\delta_m}^{N_i}$ in MEC-SRV for identifying the servers that can potentially provide the requested services attributes. MEC-SRV is further updated with the newly variable values for MEC servers for further processing in (9).

$$
\begin{aligned}
R\,(Populate) \;&=\; \forall i2 \in x2 \wedge \forall i3 \in x3 \forall i4 \in x4| \\
i4\,[1]_{n_{\forall i4[1]\in x4}} \;&:=\; (i3\,[1]_{m_{\forall i3[1]\in x3}}) \\
&\quad \wedge x4' := x4 \cup \{i4\,[1]_{n_{\forall i4[1]\in x4}}, i4\,[2]_{r_{\forall i4[2]\in x4}}, \\
&\quad i4\,[3]_{l_{\forall i4[3]\in x4}}\} \qquad (9) \\
R\,(Compute) \;&=\; \forall i5 \in x5 \wedge \forall i6 \in x6 \wedge \forall i7 \in x7| \\
i7[1] \;&:=\; (i7\,[1]_{r_{\forall i7[1]\in x7}}) \\
&\quad \wedge (i7\,[2]_{s_{\forall i7[2]\in x7}}) := Compt \\
&\quad - val(i6\,[2]_{r_{\forall i6[2]\in x6}}) \\
&\quad \wedge (i7\,[3]_{t_{\forall i7[3]\in x7}}) := Compt \\
&\quad - val(i6\,[3]_{r_{\forall i6[3]\in x6}}) \\
&\quad \wedge x7' := x7 \cup \{i7\,[1]_{n_{\forall i7[1]\in x7}}, i7\,[2]_{s_{\forall i7[2]\in x7}}, \\
&\quad i7\,[3]_{t_{\forall i7[3]\in x7}}\} \qquad (10)
\end{aligned}
$$

The QoS attributes are taken from MEC-SRV and computed in transition *Compute* for each of the δ_m server that can provide the requested service. The transition *Compute* calculate the QoS parameters that are service response time $\partial_{\xi_i}^{N_i,\delta_m}$ with the routing strategy and average link availability time $\alpha_{\xi_i}^{N_i,\delta_m}$ values from expected link time and total link time with function $Compt - val()$. It update and assigns these calculated values in temporary variables $\partial tmp_{\xi_i}^{N_i,\delta_m}$ of service response time and link availability time $\alpha tmp_{\xi_i}^{N_i,\delta_m}$ in place TMP-SRV-VAL as given in (10).

$$
\begin{aligned}
&\mathrm{R}\,(Update) \\
&=\; \forall i9 \in x9 \wedge \forall i10 \in x10 \wedge \forall i11 \in x11| \\
&((i9\,[2]_{r_{\forall i9[2]\in x9}}) \\
&\leq\; i10\,[1]) \wedge ((i10\,[3] \leq i9\,[3]_{r_{\forall i9[3]\in x9}}) \\
&\rightarrow\; ((i11\,[2]_{n_{\forall i11[2]\in x11}} := Upd \\
&\quad - Rsp\left(i9\,[2]_{r_{\forall i9[2]\in x9}}\right) \\
&\quad \wedge i11\,[3]_{n_{\forall i11[3]\in x11}} := Upd \\
&\quad - avalablty(i9\,[3]_{r_{\forall i9[3]\in x9}})) \\
&\quad \wedge ((i11\,[1]_{n_{\forall i11[1]\in x11}} := ((i11\,[2]_{n_{\forall i11[2]\in x11}}) \\
&\quad \cup (i11\,[3]_{n_{\forall i11[3]\in x11}})) \\
&\quad \vee ((i11\,[2]_{q_{\forall i11[2]\in x11}}) \\
&:=\; (i10\,[2]_{p_{\forall i10[2]\in x10}})
\end{aligned}
$$

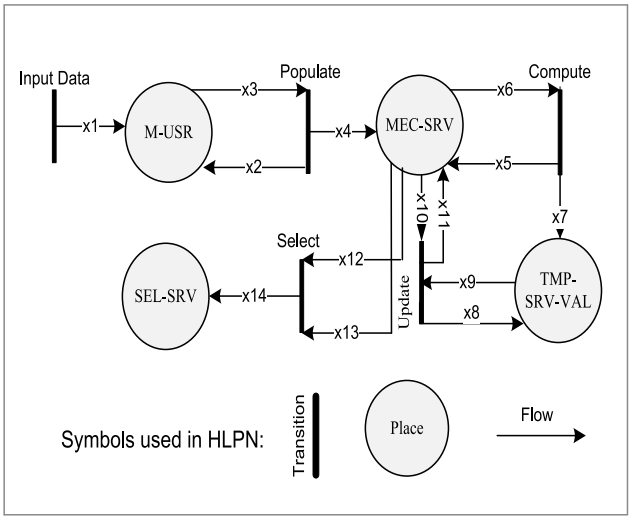

FIGURE 4. The HLPNs model of QMSP approach.

$$\wedge(i11\,[3]_{v_{\forall i11[3]\in x11}}) := (i10\,[3]_{u_{\forall i10[3]\in x10}}))$$
$$\wedge x11' := x11 \cup \{i11\,[1]_{n_{\forall i11[1]\in x11}}, i11\,[2]_{n_{\forall i11[2]\in x11}},$$
$$i11\,[3]_{n_{\forall i11[3]\in x11}}\} \tag{11}$$

In transition *Update* condition for QoS attributes are checked if it is TRUE then all the δ_m servers that satisfies the required value for QoS requirements are selected. Moreover In *Update* transition functions $Upd-Rsp()$ and $Upd-avalablty()$ assign the service response time and link availability time variable values to the corresponding final variables values in place. MEC-SRV. Moreover this transition update both the variable attribute values and stores these updated value of service response time $\partial_{\xi_i}^{N_i,\delta_m}$ and service link availability $\alpha_{\xi_i}^{N_i,\delta_m}$ in TMP-SRV-VAL, given in (11). In case if condition is FALSE then $\partial_{\xi_i}^{N_i,\delta_m}$ and $\alpha_{\xi_i}^{N_i,\delta_m}$ are again computed for next value of these variables from MEC-SRV.

$$R\,(Select) = \forall i12 \in x12 \wedge \forall i13 \in x13 \wedge \forall i14 \in x14|$$
$$i14[1] := Maxval(i12\,[2]_{r_{\forall i12[2]\in x12}})$$
$$\wedge i14[1] := Minval(i13\,[3]_{r_{\forall i13[3]\in x13}})$$
$$\wedge x14' := x14 \cup \{i14[1]\} \tag{12}$$

The QoS attributes for different mobile user's request $REQ_{\xi_i}^{N_i} = \{\partial, \alpha\}_{\xi_i}$ that are saved in MEC-SRV are taken as input to the Transition *Select* as given in equation (12). The network operator now select the most feasible server among the candidate servers. The functions *Minval* () and *Maxval* () calculate the minimum response time value and maximum link availability time variables from MEC-SRV and store the updated selected server variable value $Sel\pi_{\delta_m}^{N_i}$ place SEL-SRV. In the above-mentioned process, the selected server $Sel\pi_{\delta_m}^{N_i}$ will provide the most suitable minimum and maximum value of service response time $\partial_{\xi_i}^{N_i,\delta_m}$ and service average link availability time $\alpha_{\xi_i}^{N_i,\delta_m}$ to process the mobile users request in proposed scheme QMPR in MEC system.

V. RESULTS AND DISCUSSION

In this section, we discuss the results of our simulations, the simulation setup, the dataset used for the simulations, and analysis of our results. Our setup is mainly inspired from the previous work [7], [8]. We consider $N = 300$ mobile users distributed uniformly at random with $M = 12$ on a grid network of $300m \times 300m$ area. The coverage area of each δ is approximately $150m$ radius. There are $\xi = 100$ services available to the users that they can request. Moreover, the service requests generated by the users follows a usual Zipf distribution with a varying shape parameter from 0.5 to 0.9. The parameters for the servers, such as storage capacity, computation capacity, the upload and download capacities were set to $S_n = 500$ GB(s), $C_n = 3.2$ GHz, $B_n^u = 80$ Mbps, and $B_n^d = 250$ Mbps, respectively. The service requested attributes are varied over the course of simulations along with the aforesaid parameters.

We perform different experiments to demonstrate the effectiveness of our proposed approach. The objective of the experiments is to analyze the effect of multi-path routing on the QoS of the network. Moreover, we also demonstrate how a single path and multi-path routing effects the congestion state of the network. It is noteworthy, that multi-path routing is well studied in data center network and Mobile Adhoc Networks (MANETs). However, the concept of multi-path routing is less studied in the MEC environment. We implemented a Greedy algorithm for a single path routing. The path selection for a Greedy algorithm is based on the nearest server. Whenever a request is generated, the Greedy algorithm will select a server that is closest to the mobile user to process the request. The QMSP also selects the nearest server initially. However, when a certain congestion threshold ($\hat{C}$) is reached, the QMSP redirects the request to the other feasible available paths, as illustrated in Section IV.

QoS attributes as stated in the service level agreement. We evaluate the fractions of service requests that are generated and not processed during the time of simulations; as a QoS violation.

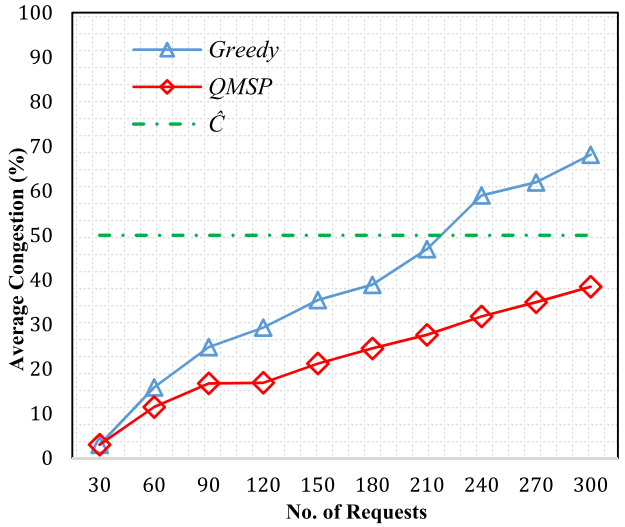

FIGURE 5. Average network congestion using QMSP and Greedy approach.

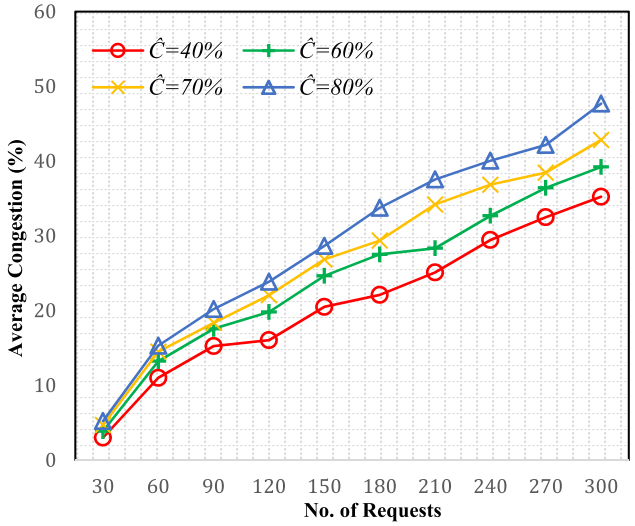

FIGURE 6. Effect of varying buffer threshold values over the average network congestion.

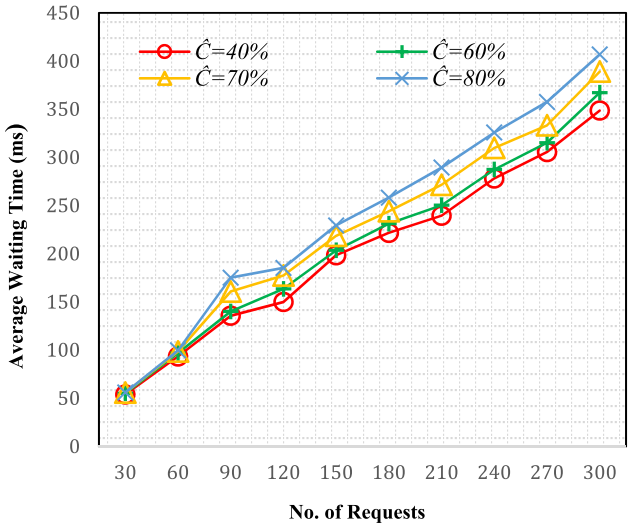

FIGURE 7. Effect of various buffer capacities on waiting time of the requests.

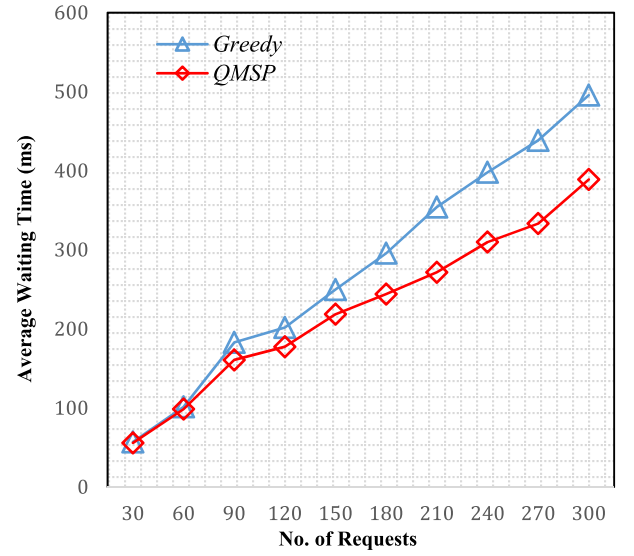

FIGURE 8. A comparison of average waiting time using QMSP and Greedy approach.

In case of Greedy algorithm, as it always selects the nearest server, the value of $\hat{C}$ has no impact. However, $\hat{C}$ can have a significant impact in case of QMSP. The overall load on the network is one of the most important parameters to evaluate the efficiency of the routing mechanism. Therefore, we evaluate the state of the network using our proposed approach and compared it with the Greedy approach. We plot the average congestion of the whole network against varying no. of service requests, as shown in Fig. 5. As seen in the Fig. 5, the congestion state of the network is significantly better in case of QMSP, as compared to the Greedy approach. As stated before, the Greedy algorithm always selects the closest server to process the request, causing some of the servers to reach higher congestion level, while the others are under- utilized. In case of QMSP, when the $\hat{C}$ value is reached, the alternative server (if exists) is selected to process the next average congestion level. The aforesaid uniformly distributes the service

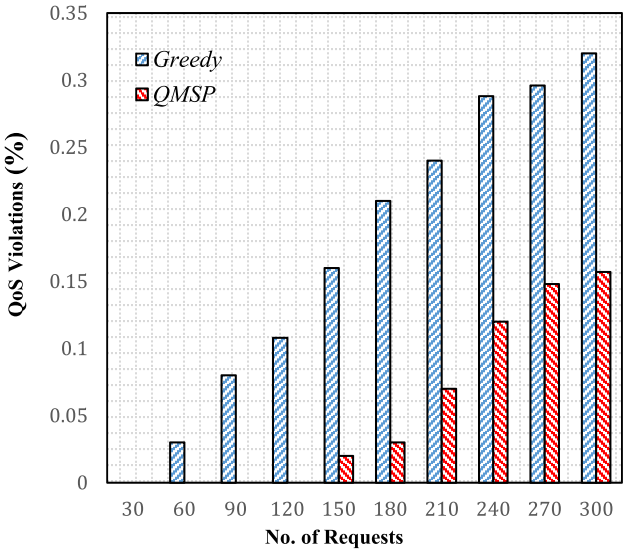

FIGURE 9. The fraction of QoS violations using QMSP and Greedy approach.

requests over the network and eventually maintain a lower average congestion level. Moreover, as the no. of requests are increased, the congestion state of the network is much more stable in case of QMSP as compared to Greedy approach. The decision of redirecting the request to another route is based on $\hat{C}$. Next, we analyze the effect of varying $\hat{C}$ on the overall state of the network. The $\hat{C}$ is computed based on consumed buffer capacity. The redirection of the service requests is performed based on the current consumption of the buffer. The results are depicted in Fig. 6. It shows that the lower the value of $\hat{C}$ the loess is the overall congestion in the Network quite straight forward. However, the important point related to $\hat{C}$ is that the lower the value, the higher will be the ratio of the packet drop. The reason for the aforesaid is that in case of QMSP, $\hat{C}$ value would redirect the request to the other available paths. If the value of $\hat{C}$ is low, then low number of service requests are served by a specific server In such cases, if there are no feasible paths (servers) available within the network topology, then the packets will be dropped. Therefore, selecting an optimal $\hat{C}$ value is very important. In our case, we are more interested in the overall state of the network. We choose few lower values of $\hat{C}$ so we can evaluate the process of redirections and how it effects the overall load of the network. We further investigate about the effect of $\hat{C}$ on the waiting time of the requests, shown in Fig. 7. The service waiting time is the time consumed by the request in the buffer, before it is processed. Waiting time represents the time a service has to wait until it is processed. The lower the time is the better will be the service response time, which improves the QoE of the users. In the said perspective, we analyzed the QoS violations in QMSP and Greedy approach, as shown in Fig. 9. The objective of any service provider is to deliver the Considering that we only had limited number of users and they can only generate certain number of requests, the fraction of QoS violations are not high. However, it is evident from the results that as the number of requests increases the QoS violations

also increase. The aforesaid is due to the fact that there are less number of servers as compared to the number of service requests. The QMSP seems to perform better than the Greedy approach and the rate of creep in the violations of QMSP is very low as compared to the Greedy approach. Our analysis concludes that multi-path routing can noticeably improve the state of the network, which ultimately have a positive impact the QoE of the users.

VI. CONCLUSION AND FUTURE DIRECTIONS

Efficiently exploiting the edge and cloud resources to improve the QoE of the users is a complex task. Moreover, improving the service request response time is vital, considering the mobile nature of the users in a MEC environment. In the said perspective, this research aims to provide an improved design for the overall state of the network and response time, by exploiting redundant paths available in the network for service provisioning. We propose a QMSP approach that can process a service request from multiple routes, while not violating the QoS attributes. We have performed several simulation experiments to illustrate the effectiveness and efficiency of our proposed approach. Moreover, we compared our approach with the traditional single path Greedy approach. The results revealed that QMSP improves the network state by 26.95% and average waiting time by 35.56% as compared to the Greedy approach. In addition, the QoS violations were reduced by the fraction of 16%.

In future perspective, we aim to implement the proposed approach in a real edge-cloud environment, which would require constant profiling of parameters related to new tasks, such as CPU, RAM, and Disk. Furthermore, an interesting aspect that we are planning to study is the mobility pattern of users and its effect on the service provisioning. In MEC environment, the users are mostly mobile and providing services to a mobile user is a complex task. The aforesaid involves questions related to what and where to offload the task and which server to select for the service request. Furthermore, as the environment at the edge of the network is highly dynamic in nature so it makes the network more vulnerable and difficult to protect. In the said perspective, a deep reinforcement schemes can be used to analyze the mobility patterns of the user, manage the task offloading, and resource allocation aspect in MEC systems in an intelligent and secure way [29], [42], and [43].

REFERENCES

[1] S. U. R. Malik, "Moving toward 5G: Significance, differences, and impact on quality of experience," *IEEE Consum. Electron. Mag.*, vol. 9, no. 6, pp. 9–14, Nov. 2020.

[2] K. Bilal, O. Khalid, A. Erbad, and S. U. Khan, "Potentials, trends, and prospects in edge technologies: Fog, cloudlet, mobile edge, and micro data centers," *Comput. Netw.*, vol. 130, pp. 94–120, Jan. 2018.

[3] S. U. Malik, H. Akram, S. S. Gill, H. Pervaiz, and H. Malik, "EFFORT: Energy efficient framework for offload communication in mobile cloud computing," *Softw., Pract. Exper.*, 2020, doi: 10.1002/spe.2850.

[4] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing—A key technology towards 5G," *ETSI White Paper*, vol. 11, no. 11, pp. 1–16, 2015.

[5] S. Bi, L. Huang, and Y.-J.-A. Zhang, "Joint optimization of service caching placement and computation offloading in mobile edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 19, no. 7, pp. 4947–4963, Jul. 2020.

[6] X. Ge, S. Tu, G. Mao, and C. X. Wang, "5G ultra-dense cellular networks," *IEEE Trans. Wireless Commun.*, vol. 23, no. 1, pp. 72–79, Feb. 2016.

[7] S. U. R. Malik, S. U. Khan, and S. K. Srinivasan, "Modeling and analysis of state-of-the-art VM-based cloud management platforms," *IEEE Trans. Cloud Comput.*, vol. 1, no. 1, p. 1, Jan. 2013.

[8] Z. Tong, X. Deng, F. Ye, S. Basodi, X. Xiao, and Y. Pan, "Adaptive computation offloading and resource allocation strategy in a mobile edge computing environment," *Inf. Sci.*, vol. 537, pp. 116–131, Oct. 2020.

[9] C. Liu, K. Li, J. Liang, and K. Li, "COOPER-SCHED: A cooperative scheduling framework for mobile edge computing with expected deadline guarantee," *IEEE Trans. Parallel Distrib. Syst.*, early access, Jun. 7, 2020, doi: 10.1109/TPDS.2019.2921761.

[10] A. Samanta, Z. Chang, and Z. Han, "Latency-oblivious distributed task scheduling for mobile edge computing," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–7.

[11] F. Wang, J. Xu, X. Wang, and S. Cui, "Joint offloading and computing optimization in wireless powered mobile-edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 1784–1797, Mar. 2018.

[12] J. Hu, C. Liu, K. Li, and K. Li, "Game-based multi-MD with QoS computation offloading for mobile edge computing of limited computation capacity," in *Proc. IFIP Int. Conf. Netw. Parallel Comput.* Cham, Switzerland: Springer, Aug. 2019, pp. 16–27.

[13] X. Cao, F. Wang, J. Xu, R. Zhang, and S. Cui, "Joint computation and communication cooperation for energy-efficient mobile edge computing," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4188–4200, Jun. 2019.

[14] M. Chen and Y. Hao, "Task offloading for mobile edge computing in software defined ultra-dense network," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 587–597, Mar. 2018.

[15] Y. Cao, H. Chen, J. Jiang, and F. Hu, "TaSRD: Task scheduling relying on resource and dependency in mobile edge computing," in *Proc. IEEE Int. Conf. Prog. Inform. Comput. (PIC)*, Dec. 2018, pp. 287–295.

[16] T. Q. Dinh, J. Tang, Q. D. La, and T. Q. S. Quek, "Offloading in mobile edge computing: Task allocation and computational frequency scaling," *IEEE Trans. Commun.*, vol. 65, no. 8, pp. 3571–3584, Aug. 2017.

[17] Y. Mao, J. Zhang, S. H. Song, and K. B. Letaief, "Power-delay tradeoff in multi-user mobile-edge computing systems," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2016, pp. 1–6.

[18] Y. Li, Y. Chen, T. Lan, and G. Venkataramani, "MobiQoR: Pushing the envelope of mobile edge computing via quality-of-result optimization," in *Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jun. 2017, pp. 1261–1270.

[19] Z. Cheng, Q. Wang, Z. Li, and G. Rudolph, "Computation offloading and resource allocation for mobile edge computing," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Dec. 2019, pp. 2735–2740.

[20] C. You, K. Huang, H. Chae, and B.-H. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1397–1411, Mar. 2017.

[21] O. Munoz, A. Pascual-Iserte, and J. Vidal, "Optimization of radio and computational resources for energy efficiency in latency-constrained application offloading," *IEEE Trans. Veh. Technol.*, vol. 64, no. 10, pp. 4738–4755, Oct. 2015.

[22] S. Sardellitti, G. Scutari, and S. Barbarossa, "Joint optimization of radio and computational resources for multicell mobile-edge computing," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 1, no. 2, pp. 89–103, Jun. 2015.

[23] H. Feng, S. Guo, A. Zhu, Q. Wang, and D. Liu, "Energy-efficient user selection and resource allocation in mobile edge computing," *Ad Hoc Netw.*, vol. 107, Oct. 2020, Art. no. 102202.

[24] C. Wang, F. R. Yu, C. Liang, Q. Chen, and L. Tang, "Joint computation offloading and interference management in wireless cellular networks with mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 66, no. 8, pp. 7432–7445, Aug. 2017.

[25] H. Guo and J. Liu, "Collaborative computation offloading for multiaccess edge computing over fiber–wireless networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4514–4526, May 2018.

[26] H. Xing, L. Liu, J. Xu, and A. Nallanathan, "Joint task assignment and wireless resource allocation for cooperative mobile-edge computing," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–6.

[27] K. Poularakis, J. Llorca, A. M. Tulino, I. Taylor, and L. Tassiulas, "Joint service placement and request routing in multi-cell mobile edge computing networks," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2019, pp. 10–18.

[28] M. Min, X. Wan, L. Xiao, Y. Chen, M. Xia, D. Wu, and H. Dai, "Learning-based privacy-aware offloading for healthcare IoT with energy harvesting," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4307–4316, Jun. 2019.
[29] C. Wu, Q. Peng, Y. Xia, and J. Lee, "Mobility-aware tasks offloading in mobile edge computing environment," in *Proc. 7th Int. Symp. Comput. Netw. (CANDAR)*, Nov. 2019, pp. 204–210.
[30] Z. Ning, P. Dong, X. Wang, J. J. Rodrigues, and F. Xia, "Deep reinforcement learning for vehicular edge computing: An intelligent offloading system," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 6, pp. 1–24, 2019.
[31] L. Huang, X. Feng, L. Qian, and Y. Wu, "Deep reinforcement learning-based task offloading and resource allocation for mobile edge computing," in *Proc. Int. Conf. Mach. Learn. Intell. Commun.* Cham, Switzerland: Springer, Jul. 2018, pp. 33–42.
[32] M. Cheng, J. Li, and S. Nazarian, "DRL-cloud: Deep reinforcement learning-based resource provisioning and task scheduling for cloud service providers," in *Proc. 23rd Asia South Pacific Design Automat. Conf. (ASP-DAC)*, Jan. 2018, pp. 129–134.
[33] F. Li and B. Hu, "DeepJS: Job scheduling based on deep reinforcement learning in cloud data center," in *Proc. 4th Int. Conf. Big Data Comput.*, vol. 5919, May 2019, pp. 48–53.
[34] G. Rjoub, J. Bentahar, O. A. Wahab, and A. S. Bataineh, "Deep and reinforcement learning for automated task scheduling in large-scale cloud computing systems," *Concurrency Comput., Pract. Exper.*, Jul. 2020.
[35] S. Tuli, S. Ilager, K. Ramamohanarao, and R. Buyya, "Dynamic scheduling for stochastic edge-cloud computing environments using A3C learning and residual recurrent neural networks," *IEEE Trans. Mobile Comput.*, early access, Aug. 17, 2020, doi: 10.1109/TMC.2020.3017079.
[36] R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges," *Future Gener. Comput. Syst.*, vol. 78, pp. 680–698, Jan. 2018.
[37] S. Tu, M. Waqas, S. U. Rehman, M. Aamir, O. U. Rehman, Z. Jianbiao, and C.-C. Chang, "Security in fog computing: A novel technique to tackle an impersonation attack," *IEEE Access*, vol. 6, pp. 74993–75001, 2018.
[38] L. Xiao, X. Wan, C. Dai, X. Du, X. Chen, and M. Guizani, "Security in mobile edge caching with reinforcement learning," *IEEE Wireless Commun.*, vol. 25, no. 3, pp. 116–122, Jun. 2018.
[39] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1628–1656, 3rd Quart., 2017.
[40] A. Ghosh, O. Khalid, R. N. B. Rais, A. Rehman, S. U. R. Malik, and I. A. Khan, "Data offloading in IoT environments: Modeling, analysis, and verification," *EURASIP J. Wireless Commun. Netw.*, vol. 2019, no. 1, pp. 1–23, Dec. 2019.
[41] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 4, pp. 974–983, Apr. 2015.
[42] K. Cao, Y. Liu, G. Meng, and Q. Sun, "An overview on edge computing research," *IEEE Access*, vol. 8, pp. 85714–85728, 2020.
[43] Q.-V. Pham, F. Fang, V. N. Ha, M. J. Piran, M. Le, L. B. Le, W.-J. Hwang, and Z. Ding, "A survey of multi-access edge computing in 5G and beyond: Fundamentals, technology integration, and state-of-the-art," *IEEE Access*, vol. 8, pp. 116974–117017, 2020.
[44] S. U. R. Malik, K. Bilal, S. U. Khan, B. Veeravalli, K. Li, and A. Y. Zomaya, "Modeling and analysis of the thermal properties exhibited by cyberphysical data centers," *IEEE Syst. J.*, vol. 11, no. 1, pp. 163–172, Mar. 2017.
[45] C. Singhal and S. De, Eds., *Resource Allocation in Next-Generation Broadband Wireless Access Networks*. Hershey, PA, USA: IGI Global, 2017.

SAIF U. R. MALIK received the Ph.D. degree from North Dakota State University, USA, in 2014. He is currently working as a Senior Researcher with Cybernetica AS, Estonia. Previously, he worked as an Assistant Professor with the Department of Computer Science, COMSATS University Islamabad, Pakistan. He is also an Active Researcher in the field of cloud computing, data centers, formal methods and its application in large scale computing systems, and data security and privacy. His work has appeared in highly reputable venues.

TEHSIN KANWAL is currently pursuing the Ph.D. degree in computer science with the Department of Computer Science, COMSATS University Islamabad. Her research interests include information security and privacy, cloud computing, the IoTs, and application of formal methods in security and privacy preserving mechanisms.

SAMEE U. KHAN received the Ph.D. degree from the University of Texas, in 2007. He is currently the Department Head and the James W. Bagley Chair of Electrical and Computer Engineering with the Mississippi State University (MSU). Before arriving at MSU, he was a Cluster Lead of Computer Systems Research with the National Science Foundation and the Walter B. Booth Professor with North Dakota State University from 2016 to 2020. His research interests include optimization, robustness, and security of computer systems. His work has appeared in over 400 publications. He is also an Associate Editor-in-Chief of the IEEE *IT Professional*, and an Associate Editor of IEEE TRANSACTIONS ON CLOUD COMPUTING, *Journal of Parallel and Distributed Computing*, and *ACM Computing Surveys*.

HASSAN MALIK (Member, IEEE) received the B.E. degree in information and communication systems from the National University of Science and Technology, Pakistan, the M.Sc. degree in wireless communication engineering from the University of Oulu, Finland, and the Ph.D. degree in electronic engineering from the University of Surrey, U.K. He is currently a Senior Lecturer with the Department of Computer Science, Edge Hill University, U.K. Previously, he was employed as a Researcher at the Thomas Johann Seebeck Department of Electronics, Tallinn University of Technology. Before that, he worked as a Postgraduate Researcher with the 5G Innovation Centre, University of Surrey, and as a Researcher Assistant with the Centre for Wireless Communication, University of Oulu. He also holds "Researcher to Innovator" Diploma from SETsquared Partnership, U.K. His research interests include wireless networking and communications, including energy-efficient green networking, cognitive radio networks, full-duplex communication, broadband wireless, context- & user-aware networks, the Internet of Things, interference and resource management, NB-IoT, URLLC, decentralized IoT networks, edge/fog computing, AI for wireless networks, vehicular networks, and mobile positioning.

HARIS PERVAIZ (Member, IEEE) received the M.Sc. degree in information security from the Royal Holloway University of London, Egham, U.K., in 2005, and the Ph.D. degree from the School of Computing and Communication, Lancaster University, Lancaster, U.K., in 2016. He was a Research Fellow with the 5G Innovation Centre, University of Surrey, Guildford, U.K., from 2017 to 2018, and an EPSRC Doctoral Prize Fellow with the School of Computing and Communication, Lancaster University, from 2016 to 2017. He is currently working as a Lecturer with the InfoLab21, Lancaster University. He has been actively involved in projects, such as CROWN, CogGREEN, TWEETHER, and Energy Proportional EnodeB for LTE-Advanced and Beyond and the DARE project, and an ESPRC funded project. His current research interests include green heterogeneous wireless communications and networking, 5G and beyond, millimeter wave communication, and energy and spectral efficiency. He is also an Associate Editor of IEEE ACCESS, *IET Quantum Communications*, *IET Network, Emerging Telecommunications Technologies* (Wiley), and *Internet Technology Letters* (Wiley).