IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 13, 2021, accepted September 19, 2021, date of publication September 23, 2021,
date of current version October 5, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3115157

A Minimized Latency Collaborative Computation Offloading Game Under Mobile Edge Computing for Indoor Localization

MARWA ZAMZAM, (Graduate Student Member, IEEE), **TALLAL ELSHABRAWY**, AND **MOHAMED ASHOUR**
Faculty of Information Engineering and Technology, German University in Cairo, New Cairo 16482, Egypt
Corresponding author: Marwa Zamzam (marwah.zamzam@guc.edu.eg)

ABSTRACT Indoor localization has become one of the fundamental services that is required in a diverse set of applications these days, such as patient monitoring and smart parking. Highly accurate localization techniques impose high latency and high energy consumption on the underlying application system. Thus, for such indoor location-based application, offloading the computation of the localization process to a remote server with high resource capability has been recently introduced as an avenue to address such a challenge. In this paper, a computation offloading problem is formulated to find the optimal decision with regard to the operation of the localization process. This decision includes: a) Where to compute the localization task, either locally on the end device or on the edge server or on the cloud server, b) Which localization technique should be used, and finally, c) Which transmission technology is recommended to be chosen in combination with the localization technique. All these decisions are constrained by the device, and the servers resource capabilities load. They are also constrained by the fact that the localization algorithm has to satisfy a certain application QoS requirement. Within such context, three algorithms are proposed for task offload decision making. First, the Indoor Localization Latency Optimal Offloading algorithm, which finds the optimal offloading decision that minimizes the total latency of the system and is considered a benchmark for the other algorithms. Second, Indoor Localization Latency Centralized Offloading algorithm that finds a sub optimal solution with lower complexity. Third, Indoor Localization Latency Game-Theoretic Offloading decentralized algorithm that converges after finite improvement steps and achieves Nash equilibrium. Altogether, the paper finds the optimum localization strategy for all users with the minimum latency under mobile edge computing environment.

INDEX TERMS Localization, computation offloading, game theory, latency, mobile edge computing.

I. INTRODUCTION

With the progress of Internet of Things (IoT) use cases, the demand for indoor localization services is continuously growing. Indoor localization can be crucial in many aspects of smart automated systems such as smart parking [1], smart home [2], smart factories [3], smart buildings [4], smart e-health platform [5] and automated vehicles [6]. Although Global Positioning System (GPS) has been widely used for outdoor localization, it fails indoors, due to the effect of ubiquitous multi-path propagation and the existence of some obstacles that hinder the spread of electromagnetic signals [7], [8]. Different indoor positioning techniques can be used to find user's location, such as Fingerprinting (FP) [9], Received Signal Strength Indicator (RSSI-based) [10], Time of Arrival (ToA) [11], Time Difference of Arrival (TDoA) [12], Angle of Arrival (AoA) [13], and Channel State Information (CSI-based) [14]. In this paper, two indoor localization techniques are evaluated: FP and RSSI. FP is conducted into two phases: offline phase and online phase. In the offline phase, user devices of known locations collect features from anchors and then they send them to remote servers. These features are trained by the servers using machine learning techniques. In the online phase, the user devices repeat the preparation process of FPs, and then the servers compare the new collected data with the previously

The associate editor coordinating the review of this manuscript and approving it for publication was Chakchai So-In.

trained model. RSSI is one of the widely used indoor localization techniques due to its simplicity and the low hardware requirements. The actual received signal strength is used to estimate the distance between the transmitter and the receiver. This leads to poor localization accuracy due to additional signal attenuation resulting from transmission through walls and large obstacles [7]. These indoor localization techniques can be applied at different transmission technologies such as WiFi [15], Bluetooth Low Energy (BLE) [16], RFID [17], Ultra-wideband (UWB) [18], IEEE 802.15.4 [19], Infrared [20], Ultrasonic [21], and Zigbee [22]. In this paper, three types of transmission technologies are examined. The first one is WiFi, as most of the current IoT devices are enabled with it. Thus, it is considered as a part of the communication infrastructure. This reduces the deployment time and cost. Second, IEEE 802.15.4, which is a widespread standard for short-range and low-power communication. Finally, BLE is characterized by low energy consumption, but it suffers from low localization accuracy [16].

Each user has his own requirements for example maximum tolerable latency, battery lifetime and accuracy requirement. On the other hand, there are diverse metrics that are associated with the indoor localization problem, such as accuracy, precision, latency, energy consumption, cost, complexity and coverage [23]. In this work, we focus on the study of three performance metrics. First, the accuracy defines the Euclidean distance between the exact location and the one estimated by each indoor localization algorithm [24]. Second, the latency is the total time taken by the device to find its location [23]. Finally, power consumption is the rate of energy per unit time that is consumed by the device in order to find its estimated location [25]. Different indoor localization algorithms have been widely proposed in recent years [26]. The existing indoor localization techniques that provide high accuracy [27]–[29], require more data collection with more complex localization algorithms, which impose more processing time and power consumption. On the other hand, decreasing the latency requires eliminating some extensive tasks and time-consuming techniques in order to estimate the location of the user with the minimum delay [30]–[32]. Accordingly, this will affect the localization accuracy. Finally, in order to avoid draining the battery of the devices, the energy consumption of the localization system must be minimized by using a less complex localization algorithm that will also affect the accuracy of the system [33], [34]. This creates a fundamental trade off between latency, power consumption and localization accuracy.

Choosing the right combination of localization technique and transmission technology can exploit the variation in user requirements. Thus, according to the requirements of the users and the metrics of the localization model, the solution should choose the best localization combination. Meeting these extreme demands with all users requirements remains a challenging issue in indoor localization problems.

Mobile devices are characterized by low battery power and limited capability constraints [35]. Thus, the approach of offloading the localization technique to be processed on a remote server that is characterized by high processing capability and continuous power supply is a promising alternative to local processing [26], as it enhances the latency and the energy consumption of the system without affecting the accuracy. Computation offloading is the transfer of specific computing tasks to an external platform such as a cluster, a grid, or a cloud [36]. In this paper, offloading the indoor localization task to the cloud and Mobile Edge Computing (MEC) servers is examined. Traditional cloud computing provides unlimited capability services such as storage, memory, and computing capabilities to IoT devices. However, it suffers from high latency and weak system reliability caused by the long distance. Due to the centralized nature of the cloud servers, the system suffers from a single point of failure and network congestion [37]. To address these challenges, Cisco delivered the concept of MEC. It is a promising solution that extends the computation resources of the cloud at the edge of the network. Thus, by deploying MEC infrastructure, the system can support real-time and latency-sensitive applications. It proves to provide the mobile devices with swift and powerful computing, energy efficiency, storage capacity, mobility, location, and context awareness support [38].

Accordingly, the main contributions of this paper are summarized as follows:

1) A localization mathematical model is designed with different localization techniques and transmission technologies under MEC environment.

2) Indoor localization computation offloading problem is formulated to minimize the total latency of the system, taking into consideration the limited resources of MEC servers, the battery lifetime of the end devices and the accuracy requirement of each user.

3) The optimal solution is obtained by selecting the decision of the localization technique and transmission technology. It decides whether each user device should process the localization task locally on the mobile device, offload it to the edge server or run it on the cloud.

4) An approximated centralized offloading greedy algorithm is proposed to handle a large number of devices and to overcome the high complexity of the optimal offloading algorithm.

5) A decentralized offloading potential game is proposed in order to avoid the problem of a single point of failure and reduce the burden on the single center station.

The structure of the paper is organized as follows. In Section II, the related work is presented. In Section III, the system model is introduced. Section IV presents the formulation of the problem and the proposed algorithms. Simulation parameters and the analysis of the numerical results are conducted in Section V. Finally, conclusion and the future work are demonstrated in Section VI.

II. RELATED WORK

MEC servers suffer from a limited number of resources compared to cloud servers. These limited edge resources can only handle a limited number of service requests. Exceeding edge resources may lead to severe QoS degradation and may at some point cause service outage [39], [40]. Accordingly, despite the crucial role of deployment MEC platform to enhance QoS requirement of users. The problem of computation offloading introduces several critical challenges that still need to be addressed and studied [41]. The work in [42] presents a mathematical model to calculate the computational offloading latency and energy consumption for different mobile application models such as image processing and antivirus applications. The work in [43] highlights the significance of edge computing by providing real-life scenarios that have low application response time. Authors in [44] introduces the key issues through the offloading process, such as whether, where, and what to offload. Authors in [45] analyze and compare the existing computing offloading algorithms from the perspective of the minimum latency, energy consumption and trade off between both of them. The work in [41] gives an overview on offloading classification, the factors that influences it and offloading management. Authors in [46] present a framework for pre-compiled vector instruction offloading, their experimental results demonstrated that for smaller workloads, the edge server provided higher time and energy efficiency as compared to the cloud server. However, for larger workloads, the cloud server yields higher efficiency. The work in [47] studies the state-of-the-art of applying game theoretic framework, which is the same technique that is used in this paper for computation offloading to overcome the edge computing challenges.

Some recent studies have integrated MEC in indoor localization platforms. The work in [7] offloaded the localization process from the devices with low remaining battery to the edge nodes only. The results showed the improvement of the total energy consumption by 80% when they are compared with local computing. However, only FP localization technique is employed in this model which may not be acceptable for latency sensitive applications.

Authors in [8] proposed a cooperative localization model that improved both accuracy, reliability and location rate in the coverage area. The location information is processed on the edge nodes. In addition, this information is uploaded and stored on the cloud server. Moreover, the work was evaluated by constructing a real tunnel environment emulating a coal mining scenario. However, the authors did not tackle the trade-off between the MEC and the cloud servers. This may affect the latency due to offloading all the information to the cloud servers.

Authors in [48] proposed a new lookup algorithm based on chord protocol that finds the accurate location of an end device with the minimum latency and computational cost. The localization workload is distributed using a set of edge servers that are placed in different zones inside the building. However, the processing time to compute the localization process is neglected which may give misleading results for the total latency of the system.

The work in [49] proposed low cost indoor localization algorithm over a MEC system. Authors used both Bluetooth Low Energy (BLE) as it is easily accessible and IEEE 802.15.4 a compliant Ultra-wide band (UWB) for its high accuracy ranging. However, the algorithm did not take into account the energy and limited resources restrictions, where exceeding servers resources may cause service outage and affect the total performance of the system.

The work in [50] studies the problem formulation of minimizing the total energy consumption of the system. They focus only on choosing the best offloading decision for the indoor localization task, neglecting the type of the localization technique and the transmission technology.

To the best of our knowledge, this is the first paper that strives to find the optimal strategy of choosing indoor localization technique, transmission technology, where to compute the localization task with the minimum latency using computation offloading technique under MEC environment.

III. SYSTEM MODEL

This Section describes the network and the computation models. Three offloading schemes are considered: local computing, MEC offloading and cloud offloading, where l, M and C indicate that the task is computed on the local device, MEC and cloud servers respectively.

A. NETWORK MODEL

Figure.1 represents the architecture of the system which consists of three layers. The first layer consists of multiple user devices $\mathbb{N} = \{1, 2 \ldots N\}$. We denote by i the i^{th} user device. There are multiple anchors $\mathbb{G} = \{1, 2 \ldots G\}$ that can be any type of the three corresponding transmission technologies that are considered in this paper. We denote g the g^{th} anchor. The second layer consists of multiple edge servers $\mathbb{K} = \{1, 2 \ldots K\}$. We denote by j the j^{th} MEC server. Finally, the third layer which consists of a single cloud server. The aim of the problem is to find an accurate location for each user device i with the minimum latency. The system computes the localization process of the user device that receives more than three beacons from different anchors. There are multiple localization models $\mathbb{Q} = \{1, 2 \ldots Q\}$ for the localization process. We denote by q the q^{th} localization model.

B. COMPUTATION MODEL

The problem does not only choose the best localization technique and transmission technology, but also it selects the best offloading decision for processing this localization task. There are three cases, either to process the localization task locally on the user device or to offload it to remote servers (MEC or cloud servers).

1) LOCAL COMPUTING

In this case the user device calculates its estimated location without offloading the localization request to any remote server. For each device, the latency $L_l(i, q)$, energy consumption $E_l(i, q)$ and accuracy $\Lambda_l(i, q)$ is obtained for all possible

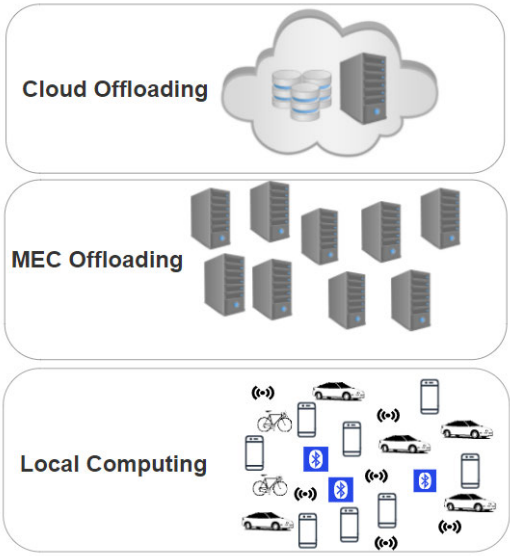

FIGURE 1. System architecture.

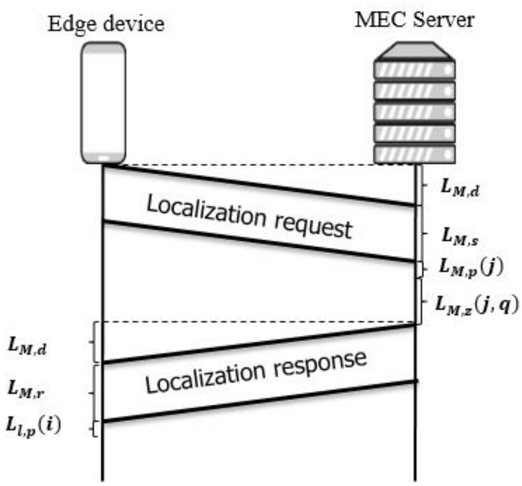

FIGURE 2. MEC offloading structure.

localization models, where these values will be defined in the simulation parameters.

2) MEC OFFLOADING

In this case the user device cannot calculate its location itself due to battery constraints or latency requirements. Thus, a localization request is sent to the MEC server as shown in Figure. 2. The localization process is conducted on the MEC server that has enough capacity and will obtain an accurate location with the minimum latency. The total latency to find the accurate location on MEC server $L_M(i,j,q)$ is presented by (1). It depends on: a) propagation time between the user device and the MEC server $L_{M,d}$ as defined in (2) where $\omega_{l,M}$ is the distance between the MEC server and the device, and c is the speed of light, we multiply the propagation time by 2 as there is a propagation time due to sending the localization request and the other due to receiving the localization result, as shown from Figure. 2, b) transmission time to send the localization request $L_{M,s}$ as described in (3) where ϱ is the size of the localization request and $\Gamma_{l,M}$ is the transmission rate between the user device and MEC server, c) transmission time to receive the localization response $L_{M,r}$ in (4) where υ is the size of the localization result, d) processing time of the localization request on the MEC server $L_{M,p}(j)$ as presented in (5) where ρ_M is the number of CPU cycles that are needed to process one bit on a given MEC server j and $f_M(j)$ is the computational capability of the MEC server j, e) processing time of the localization result on the user device $L_{l,p}(i)$ as shown in (6) where ρ_l is the number of CPU cycles that are needed to process one bit on the user device i and $f_l(i)$ is the computational capability of the user device i, f) time for localization process $L_{M,z}(j,q)$ as shown in (7) where $\kappa(q)$ is the total data required for computing a localization process q. The data required for each localization $\kappa(q)$ is defined in (8) where δ is a constant value that represents the average workload of the localization process and $L_l(q)$.

$$L_M(i,j,q) = 2L_{M,d} + L_{M,s} + L_{M,r} + L_{M,p}(j) + L_{l,p}(i) + L_{M,z}(j,q) \quad (1)$$

where:

$$L_{M,d} = \omega_{l,M}/c \quad (2)$$
$$L_{M,s} = \varrho/\Gamma_{l,M} \quad (3)$$
$$L_{M,r} = \upsilon/\Gamma_{l,M} \quad (4)$$
$$L_{M,p}(j) = \varrho\rho_M/f_M(j) \quad (5)$$
$$L_{l,p}(i) = \upsilon\rho_l/f_l(i) \quad (6)$$
$$L_{M,z}(j,q) = \kappa(q)\rho_M/f_M(j) \quad (7)$$
$$\kappa(q) = \delta L_l(q) \quad (8)$$

The total energy consumption $E_M(i,j,q)$ demonstrated by (9) depends on: a) energy consumed due to sending localization request $E_{M,s}(i)$ as presented in (10) where $\chi_s(i)$ is the power consumed by each user device i for sending, b) energy consumed due to receiving localization result $E_{M,r}(i)$ as shown in (11), where $\chi_r(i)$ is the power consumed by each user device i for receiving, c) energy consumed due to processing localization request on the MEC server $E_{M,p}(i,j)$ as defined in (12) where $\chi_o(i)$ is the power consumed by each user device i when being idle, d) energy consumed due to processing localization result on the user device $E_{l,p}(i)$ as described in (13) where $\chi_p(i)$ is the power consumed by each user device i for computing, e) energy consumed for processing localization process on the MEC server $E_{M,z}(i,j,q)$ as found in (14).

$$E_M(i,j,q) = E_{M,s}(i) + E_{M,r}(i) + E_{M,p}(i,j) + E_{l,p}(i) + E_{M,z}(i,j,q) \quad (9)$$

where:

$$E_{M,s}(i) = \chi_s(i)L_{M,s} \quad (10)$$
$$E_{M,r}(i) = \chi_r(i)L_{M,r} \quad (11)$$
$$E_{M,p}(i,j) = \chi_o(i)L_{M,p}(j) \quad (12)$$

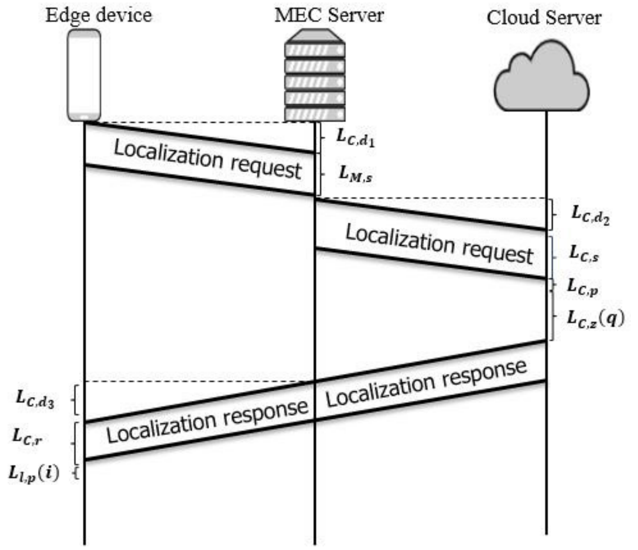

FIGURE 3. **Cloud offloading structure.**

$$E_{l,p}(i) = \chi_p(i)L_{l,p}(i) \tag{13}$$
$$E_{M,z}(i,j,q) = \chi_o(i)L_{M,z}(j,q) \tag{14}$$

3) CLOUD OFFLOADING

In this case, the user device cannot compute the localization task due to battery constraints or latency requirements, or the MEC servers reached their maximum capacity. Thus, the localization request is sent to the cloud server as shown in Figure. 3. The total latency $L_C(i,q)$ taken to find the accurate location using the cloud server is demonstrated by (15). It depends on: a) propagation time between the user device and the MEC server L_{C,d_1} as shown in (16), b) propagation time between the MEC server and the cloud server L_{C,d_2} as defined in (17) where $\omega_{M,C}$ is the distance between MEC and cloud servers, c) propagation time between the cloud server and the user device L_{C,d_3} as presented in (18) where $\omega_{l,C}$ is the distance between the cloud server and the user device, d) transmission time to send the localization request $L_{C,s}$ as found in (19) where $\Gamma_{M,C}$ is the transmission rate between the MEC server and cloud server, e) transmission time to receive the localization response $L_{C,r}$ as shown in (20) where $\Gamma_{l,C}$ is the transmission rate between the cloud server and the user device, f) processing time of the localization request on the MEC and the cloud server $L_{C,p}$ as defined in (21) where ρ_C is the number of CPU cycles that are needed to process one bit on the cloud server and f_C is the computational capability of the cloud server, g) processing time of the localization result on the user device $L_{l,p}(i)$, h) time for localization process on the cloud server $L_{C,z}(q)$ as presented in (22).

$$\begin{aligned} L_C(i,q) &= L_{C,d_1} + L_{C,d_2} + L_{C,d_3} \\ &\quad + L_{C,s} + L_{C,r} + L_{C,p} + L_{l,p}(i) + L_{C,z}(q) \end{aligned} \tag{15}$$

where:

$$L_{C,d_1} = \omega_{l,M}/c \tag{16}$$
$$L_{C,d_2} = \omega_{M,C}/c \tag{17}$$
$$L_{C,d_3} = \omega_{l,C}/c \tag{18}$$
$$L_{C,s} = L_{M,s} + \varrho/\Gamma_{M,C} \tag{19}$$
$$L_{C,r} = \upsilon/\Gamma_{l,C} \tag{20}$$
$$L_{C,p} = L_{M,p} + \varrho\rho_C/f_C \tag{21}$$
$$L_{C,z}(q) = \kappa(q)\rho_C/f_C \tag{22}$$

The total energy consumption $E_C(i,q)$ demonstrated by (23) depends on: a) energy consumed due to sending localization request $E_{C,s}(i)$ defined in (24), b) energy consumed due to receiving localization result $E_{C,r}(i)$ as shown in (25), c) energy consumed due to processing energy on the MEC and the cloud servers $E_{C,p}(i)$ as presented in (26), d) processing energy for the localization result on the user device $E_{l,p}(i)$, e) energy consumed for processing localization process on the cloud server $E_{C,z}(i,q)$ as described in (27).

$$\begin{aligned} E_C(i,q) &= E_{C,s}(i) + E_{C,r}(i) \\ &\quad + E_{C,p}(i) + E_{l,p}(i) + E_{C,z}(i,q) \end{aligned} \tag{23}$$

where:

$$E_{C,s}(i) = \chi_o(i)L_{C,s} \tag{24}$$
$$E_{C,r}(i) = \chi_o(i)L_{C,r} \tag{25}$$
$$E_{C,p}(i) = \chi_o(i)L_{C,p} \tag{26}$$
$$E_{C,z}(i,q) = \chi_o(i)L_{C,z}(q) \tag{27}$$

IV. OPTIMAL CHOICE OF LOCALIZATION TECHNIQUE AND TRANSMISSION TECHNOLOGY OFFLOADING PROBLEM FORMULATION AND PROPOSED SOLUTIONS

In this Section, first, the problem is formulated stating the decision variables, objective function and constraints. Second, the proposed solutions are presented, i.e., optimal solution, centralized algorithm and decentralized game theory schemes.

A. PROBLEM FORMULATION

In order to formulate the optimization problem, the decision variables must be defined first. These decision variables are the output of the problem. In our problem, they indicate whether the task will be computed locally $\alpha(i,q)$ or on the MEC server $\mu(i,j,q)$ or on the cloud server $\beta(i,q)$. They are defined as follows:

$$\alpha(i,q) = \begin{cases} 1, & \text{if user } i \text{ chooses localization model } q \\ & \text{and computes it locally} \\ 0, & \text{otherwise} \end{cases}$$

$$\mu(i,j,q) = \begin{cases} 1, & \text{if user } i \text{ chooses localization model } q \\ & \text{and computes it on MEC server } j \\ 0, & \text{otherwise} \end{cases}$$

$$\beta(i,q) = \begin{cases} 1, & \text{if user } i \text{ chooses localization model } q \\ & \text{and computes it on cloud server} \\ 0, & \text{otherwise} \end{cases}$$

$$\min\ L = \Big(\sum_{q=1}^{Q}\sum_{i=1}^{N}\alpha(i,q)L_l(i,q) + \sum_{q=1}^{Q}\sum_{i=1}^{N}\beta(i,q)L_C(i,q) + \sum_{q=1}^{Q}\sum_{i=1}^{N}\sum_{j=1}^{k}\mu(i,j,q)L_M(i,j,q)\Big) \tag{28}$$

subject to

$$\text{C1:}\quad \sum_{q=1}^{Q}\sum_{i=1}^{N}\mu(i,j,q) \le \mathcal{R}(j), \quad \forall j \in \mathbb{K} \tag{29}$$

$$\text{C2:}\quad \alpha(i,q)L_l(i,q) \le \Upsilon(i) \quad \forall q \in \mathbb{Q},\ \forall i \in \mathbb{N} \tag{30}$$

$$\sum_{j=1}^{k}\mu(i,j,q)L_M(i,j,q) \le \Upsilon(i) \quad \forall q \in \mathbb{Q},\ \forall i \in \mathbb{N} \tag{31}$$

$$\beta(i,q)L_C(i,q) \le \Upsilon(i) \quad \forall q \in \mathbb{Q},\ \forall i \in \mathbb{N} \tag{32}$$

$$\text{C3:}\quad \alpha(i,q)\Lambda_l(i,q) \le H(i) \quad \forall q \in \mathbb{Q},\ \forall i \in \mathbb{N} \tag{33}$$

$$\sum_{j=1}^{k}\mu(i,j,q)\Lambda_M(i,j,q) \le H(i) \quad \forall q \in \mathbb{Q},\ \forall i \in \mathbb{N} \tag{34}$$

$$\beta(i,q)\Lambda_C(i,q) \le H(i) \quad \forall q \in \mathbb{Q},\ \forall i \in \mathbb{N} \tag{35}$$

$$\text{C4:}\quad \Big(\sum_{q=1}^{Q}\alpha(i,q)E_l(i,q) + \sum_{q=1}^{Q}\beta(i,q)E_C(i,q)\Big) + \sum_{q=1}^{Q}\sum_{j=1}^{k}\mu(i,j,q)E_M(i,j,q) \le \exists(i) \quad \forall i \in \mathbb{N} \tag{36}$$

$$\text{C5:}\quad \sum_{q=1}^{Q}\alpha(i,q) + \sum_{q=1}^{Q}\sum_{j=1}^{k}\mu(i,j,q) + \sum_{q=1}^{Q}\beta(i,q) \le \psi, \quad \forall i \in \mathbb{N} \tag{37}$$

$$\text{C6:}\quad \alpha(i,q) \in \{0,1\}, \quad \forall q \in \mathbb{Q},\ \forall i \in \mathbb{N} \tag{38}$$

$$\mu(i,j,q) \in \{0,1\}, \quad \forall q \in \mathbb{Q},\ \forall i \in \mathbb{N},\ \forall j \in \mathbb{K} \tag{39}$$

$$\beta(i,q) \in \{0,1\}, \quad \forall q \in \mathbb{Q},\ \forall i \in \mathbb{N} \tag{40}$$

The objective function of the proposed problem that is presented in (28) is to minimize the total time L that all devices will take in order to find their accurate location. It is divided into three terms: the first term represents the latency of the user device if the localization is done locally, the second term represents the total latency if the localization process is computed on any MEC server and finally, the last term represents the total time when the localization is done on the cloud server. Each device will choose to compute the localization process locally or in remote servers according to capacity, latency, energy and accuracy constraints.

Constraint C1 in (29) limits the number of requests that can be served by a given MEC j to maximum of $\mathcal{R}(j)$. Where $\mathcal{R}(j)$ is the maximum number of user devices that can offload their localization process to be computed on the MEC server j. Constraints C2 in (30-32) guarantee that the total delay taken for finding the location of a given user device i does not exceed the latency requirements of this user device i, where $\Upsilon(i)$ denotes the maximum tolerable latency for user device i. Constraints C3 in (33-35) are for satisfying the accuracy constraints of each user device i. $\Lambda_l(i,q)$, $\Lambda_M(i,j,q)$ and $\Lambda_C(i,q)$ are the accuracy calculated for local, MEC and cloud computing respectively. The localization process that will be applied must deliver accuracy that is smaller than or equal to the maximum accuracy that each user device i can tolerate $H(i)$. Constraint C4 in (36) guarantees that the user device i will not compute the localization process if it does not have enough battery, where $\exists(i)$ denotes the battery lifetime for each user device i. Constraint C5 in (37) ensures that the localization process will be done either locally or on the MEC server or on the cloud server. ψ is a binary vector that indicates whether the user device i can calculate its localization or not, i.e. the user device i receives more than three beacons from different anchors. Finally, Constraints C6 in (38-40) ensure the binary value of the offloading decision variables.

B. COLLABORATIVE COMPUTATION OFFLOADING SOLUTION OVER MEC

1) INDOOR LOCALIZATION LATENCY OPTIMAL OFFLOADING (ILLOO)

In ILLOO as shown in Algorithm 1, a unique optimal solution is obtained by checking on all possible combinations of offloading decisions for all devices and then choosing the solution that delivers the least latency for the whole network while satisfying the system constraints (C1 - C6) as well. Although, this algorithm is very effective as it guarantees that there is no better solution than the delivered one, but it has a very high computational complexity $O((2+K)M)^N$ due to enumerating on all possible offloading decisions. Therefore, this algorithm is used only as a benchmark when the number of devices is not very large.

2) INDOOR LOCALIZATION LATENCY CENTRALIZED OFFLOADING (ILLCO)

The main aim of proposing this algorithm is to handle a large number of users to overcome the high complexity of ILLOO. In ILLCO as shown in Algorithm 2, a sub optimal solution is obtained. The algorithm iterates over all N devices, for each user device i the latency is computed for all combination of localization techniques and transmission technologies three times: locally $L_l(i,q)$, on MEC servers $L_M(i,j,q)$ and

Algorithm 1 Indoor Localization Latency Optimal Offloading (ILLOO)

Input: $Q, N, K, G, \Upsilon(i), \Lambda(i), \exists(i), \omega_{l,M}, \omega_{M,C}, \omega_{l,C}, \varrho, \Gamma_{l,M}, \Gamma_{l,C}, \Gamma_{M,C}, \upsilon, \rho_M, \rho_l, \rho_C, f_l, f_M(j), f_C, \chi_r(g), c, \chi_o(i), \chi_p(i), \chi_s(i)$

Output: a) optimal offloading decision for each user device i $\alpha(i, q)$, $\mu(i, j, q)$, $\beta(i, q)$. b) Total Latency of the system L

1. Find all possible combination of $\alpha(i, q)$, $\mu(i, j, q)$, $\beta(i, q)$.
2. Discard all the solutions that do not satisfy the constraints.
3. Calculate the total latency for all applicable solutions.
4. Choose the solution that gives the least latency L.
5. Output $\alpha(i, q)$, $\mu(i, j, q)$, $\beta(i, q)$ and L.

on cloud server $L_C(i, q)$. The solution that satisfies all the system constraints (C1 - C6) and has the least latency will be chosen. Thus, the computational complexity of ILLCO algorithm is $O(NMK)$, as it is still an exhaustive search but with respect to a given user at a given network condition. It can be noticed that this is a greedy algorithm as each user device i decides the best offloading decision according to its minimum latency. Hence, it neglects minimizing the overall latency of the network. In this algorithm, the center station is the cloud server. The cloud server collects all the local parameters of all devices in order to be able to make the best offloading decision for each user device i. As a result, no privacy is preserved and all computational burden is done on the central controller which may lead to the problem of a single point of failure.

Algorithm 2 Indoor Localization Latency Centralized Offloading (ILLCO)

Input: $Q, N, K, G, \Upsilon(i), \Lambda(i), \exists(i), \omega_{l,M}, \omega_{M,C}, \omega_{l,C}, \varrho, \Gamma_{l,M}, \Gamma_{l,C}, \Gamma_{M,C}, \upsilon, \rho_M, \rho_l, \rho_C, f_l, f_M(j), f_C, \chi_r(g), c, \chi_o(i), \chi_p(i), \chi_s(i)$

Output: a) sub optimal offloading decision for each user device i

$\alpha(i, q)$, $\mu(i, j, q)$, $\beta(i, q)$. b) Total Latency of the system L

1. **for** $i \in \mathbb{N}$ **do**
2. **for** $q \in \mathbb{Q}$ **do**
3. compute $L_l(i, q), L_C(i, q)$.
4. **for** $j \in \mathbb{K}$ **do**
5. compute $L_M(i, j, q)$.
6. **end for**
7. **end for**
8. **end for**
9. **for** $i \in \mathbb{N}$ **do**
10. choose min $(L_l(i, q), L_M(i, j, q), L_C(i, q))$ that satisfies all system constraints.
11. update $\alpha(i, q)$, $\mu(i, j, q)$, $\beta(i, q)$ according to the minimum latency.
12. update the system parameters according to the offloading decisions.
13. **end for**
14. Output $\alpha(i, q)$, $\mu(i, j, q)$, $\beta(i, q)$ and L.

3) INDOOR LOCALIZATION LATENCY GAME-THEORETIC OFFLOADING (ILLGO)

a: GAME FORMULATION

The main aim of using a decentralized algorithm is to reduce the complexity of the system and to protect the privacy of each user, where each user device i computes for itself the total time taken to process the localization model. This avoids the problem of a single point of failure and reduces the burden on the single center station. For simplification, we denote the computing offloading decision for user device i by $a(i)$, on the other hand $a(-i)$ describes the offloading decision of all other devices except user device i. Each user device i acts as a player that competes with other players to compute the localization task with its minimum latency. There are three offloading strategies: local computing, MEC offloading and cloud offloading. The payoff is the total time taken to find the location of the user device i. The objective function is reformulated in (41- 42) with the same system constraints (C1 - C6) as follows:

$$\min_{a(i)=\{1,2,3\}} u(a(i), a(-i)) \tag{41}$$

$$u(a(i)) = \begin{cases} L_l(i, q), & \text{if } a(i) = 1, \forall q \in \mathbb{Q} \\ L_M(i, j, q), & \text{if } a(i) = 2, \forall q \in \mathbb{Q}, \forall j \in \mathbb{K} \\ L_C(i, q), & \text{if } a(i) = 3, \forall q \in \mathbb{Q}, \end{cases} \tag{42}$$

b: EXISTENCE OF NASH EQUILIBRIUM

In order to prove that the problem will reach Nash Equilibrium (NE), we first prove that the problem is a potential game with a potential function.

Definition 1: A game is called a potential game if it has a potential function $\Theta(a)$ and it is called an exact potential game if:

$$u(a(i), a(-i)) - u(a(i)', a(-i)) = \Theta(a(i), a(-i)) - \Theta(a(i)', a(-i)) \tag{43}$$

where $a(i)'$ is set to be an improvement step for user i if $u(a(i)', a(-i)) \leq u(a(i), a(-i))$. Every ordinal potential game with finite strategy sets owns at least one pure-strategy NE and has a finite improvement step. So in order to prove that the ILLGO is an exact potential game, a potential function must be constructed and then (43) must be proved. The potential function of the proposed game is defined in (44) as follows:

$$\begin{aligned} \Theta(a) = & \sum_{q=1}^{Q}\sum_{i=1}^{N} L_l(i, q) I_{(a(i)=1)} + \sum_{q=1}^{Q}\sum_{i=1}^{N}\sum_{j=1}^{K}(2.L_{M,d} \\ & + L_{M,s} + L_{M,r} + L_{M,p}(j) + L_{l,p}(i) \\ & + L_{M,z}(j, q) I_{(a(i)=2)}) + \sum_{q=1}^{Q}\sum_{i=1}^{N}(L_{C,d_1} + L_{C,d_2} + L_{C,d_3} \\ & + L_{C,s} + L_{C,r} + L_{C,p} + L_{l,p}(i) + L_{C,z}(q) I_{(a(i)=3)}) \end{aligned} \tag{44}$$

$I_{(event)}$ is an indicator function that represents the offloading decision for each user device i, where $I_{(a(i)=1)} = 1$, if the user device i computes the localization process locally. $I_{(a(i)=2)} = 1$, if the user device i computes the localization process on MEC servers. $I_{(a(i)=3)} = 1$, if the user device i computes the localization process on the cloud server. This potential function must be applicable for every change in the offloading decision the user device i may take. All possible cases are justified as follows:

Case 1: User i changes the offloading decision from locally to MEC server.

$$\begin{aligned}
&\Theta(1, a(-i)) - \Theta(2, a(-i))\\
&\quad = \sum_{q=1}^{Q}\sum_{i=1}^{N} L_l(i, q) - (\sum_{q=1}^{Q}\sum_{i=1}^{N}\sum_{j=1}^{K}(2.L_{M,d} + L_{M,s}\\
&\qquad + L_{M,r} + L_{M,p}(j) + L_{l,p}(i) + L_{M,z}(j, q))\\
&\quad = u(1, a(-i)) - u(2, a(-i))
\end{aligned}$$

Case 2: User i changes the offloading decision from locally to cloud server.

$$\begin{aligned}
&\Theta(1, a(-i)) - \Theta(3, a(-i))\\
&\quad = \sum_{q=1}^{Q}\sum_{i=1}^{N} L_l(i, q) - (\sum_{q=1}^{Q}\sum_{i=1}^{N}(L_{C,d_1} L_{C,d_2} + L_{C,d_3}\\
&\qquad + L_{C,s} + L_{C,r} + L_{C,p} + L_{l,p}(i) + L_{C,z}(q))\\
&\quad = u(1, a(-i)) - u(3, a(-i))
\end{aligned}$$

Case 3: User i changes the offloading decision from MEC server to locally.

$$\begin{aligned}
&\Theta(2, a(-i)) - \Theta(1, a(-i))\\
&= \sum_{q=1}^{Q}\sum_{i=1}^{N}\sum_{j=1}^{K}(2.L_{M,d} + L_{M,s} + L_{M,r} + L_{M,p}(j) + L_{l,p}(i)\\
&\quad + L_{M,z}(j, q) - \sum_{q=1}^{Q}\sum_{i=1}^{N} L_l(i, q)\\
&= u(2, a(-i)) - u(1, a(-i))
\end{aligned}$$

Case 4: User i changes the offloading decision from MEC server to cloud server.

$$\begin{aligned}
&\Theta(2, a(-i)) - \Theta(3, a(-i))\\
&= \sum_{q=1}^{Q}\sum_{i=1}^{N}\sum_{j=1}^{K}(2.L_{M,d} + L_{M,s} + L_{M,r} + L_{M,p}(j) + L_{l,p}(i)\\
&\quad + L_{M,z}(j, q) - (\sum_{q=1}^{Q}\sum_{i=1}^{N}(L_{C,d_1} + L_{C,d_2} + L_{C,d_3} + L_{C,s}\\
&\quad + L_{C,r} + L_{C,p} + L_{l,p}(i) + L_{C,z}(q))\\
&= u(2, a(-i)) - u(3, a(-i))
\end{aligned}$$

Case 5: User i changes the offloading decision from cloud server to locally.

$$\begin{aligned}
&\Theta(3, a(-i)) - \Theta(1, a(-i))\\
&\quad = (\sum_{q=1}^{Q}\sum_{i=1}^{N}(L_{C,d_1} + L_{C,d_2} + L_{C,d_3} + L_{C,s} + L_{C,r} + L_{C,p}\\
&\qquad + L_{l,p}(i) + L_{C,z}(q)) - \sum_{q=1}^{Q}\sum_{i=1}^{N} L_l(i, q)\\
&\quad = u(3, a(-i)) - u(1, a(-i))
\end{aligned}$$

Case 6: User i changes the offloading decision from cloud server to MEC.

$$\begin{aligned}
&\Theta(3, a(-i)) - \Theta(2, a(-i))\\
&\quad = (\sum_{q=1}^{Q}\sum_{i=1}^{N}(L_{C,d_1} + L_{C,d_2} + L_{C,d_3} + L_{C,s} + L_{C,r} + L_{C,p}\\
&\qquad + L_{l,p}(i) + L_{C,z}(q)) - (\sum_{q=1}^{Q}\sum_{i=1}^{N}\sum_{j=1}^{K}(2.L_{M,d} + L_{M,s}\\
&\qquad + L_{M,r} + L_{M,p}(j) + L_{l,p}(i) + L_{M,z}(j, q))\\
&\quad = u(3, a(-i)) - u(2, a(-i))
\end{aligned}$$

Accordingly, this means that the proposed problem is an exact potential game and an equilibrium state will be reached after finite number of iterations and all users' requirements will be satisfied.

Algorithm 3 Indoor Localization Latency Game-Theoretic Offloading (ILLGO)

Input: $Q, N, K, G, \Upsilon(i), H(i), \exists(i), \omega_{l,M}, \omega_{M,C}, \omega_{l,C}, \varrho, \Gamma_{l,M}, \Gamma_{l,C}, \Gamma_{M,C}, \upsilon, \rho_M, \rho_l, \rho_C, f_l, f_M(j), f_C, \chi_r(g), c, \chi_o(i), \chi_p(i), \chi_s(i)$

1. Initialization:
set the offloading decisions to all devices to be locally
the initial step $\sigma_o = 0$
2. **for** $i \in \mathbb{N}$ **do**
3. find the optimal offloading decision of each user device i
4. store the devices that want to update their current offloading decision into $\bar{V}$.
5. **end for**
6. **for** each step σ **do**
7. **if** $\bar{V}$ is not empty
8. choose one user device from $\bar{V}$ to win the opportunity to update the offloading decision.
9. update system parameters according to the new offloading decision.
10. **for** $i \in \mathbb{N}$ **do**
11. find the optimal offloading decision of each device.
12. store the devices that want to update their current offloading decision into $\bar{V}$.
13. **end for**
14. **end if**
15. **end for**

output: a) optimal offloading decision for each user device $a(i)$, b) total latency of the system L.

c: INDOOR LOCALIZATION GAME-THEORETIC COMPUTATION OFFLOADING (ILLGO) ALGORITHM

The steps of the code are briefly illustrated in ILLGO as shown in Algorithm 3. We set a specific number that describes the value of the finite improvement steps σ_{max}. First, it is assumed that all devices choose to compute the localization task locally. Second, the optimal offloading decision for each user device will be calculated, and then the devices that want to update their decisions from locally to remote servers will be stored in $\bar{V}$. Third, in each step σ, if vector $\bar{V}$ is not empty, one device from vector $\bar{V}$ will win the opportunity to update the offloading decision. Fourth, all system parameters will be updated according to the new offloading decision. Fifth, we will loop on all users, find the new optimal offloading decisions of them and store only the devices that want to change their offloading decisions in $\bar{V}$. Finally, after σ_{max} iterations, which it is called finite improvement steps, when no device needs to update its offloading decision anymore, the system reaches an equilibrium state which is the NE state. The system outputs the offloading decisions of all devices and the total latency of the system. The algorithm guarantees sub optimum solution with lower complexity than the previous algorithms where its computational complexity is $O(N\sigma_{max})$.

V. PERFORMANCE EVALUATION

In this Section, the simulation setup and then the numerical results of running the three proposed algorithms are demonstrated.

A. SIMULATION SETUP

To evaluate the performance of the proposed techniques by numerical studies, Matlab software is used. The optimal solution of ILLOO algorithm is obtained using CVX solver. We consider a multi-user multi-MEC heterogeneous network with different number of devices, MEC servers, anchors, and a remote cloud server. The simulation area was defined within a range of 50m × 50m. Each user device requests a single task. The indoor localization application is considered as our computational task. The objective of our proposed schemes is to find the optimum offloading decision with the minimum latency without constraint violation. Table 1 summarizes all the simulation parameters that are used in evaluating the proposed algorithms. Table 2 shows the comparison between three performance metrics of processing all combinations of the localization techniques and transmission technologies. These performance metrics are total latency, power consumption and accuracy. The are 6 localization models, It is found that BLE has lower latency than WiFi but higher latency than IEEE 802.15.4. Although, BLE has the lowest energy consumption, its accuracy is worse than the others. Thus, the solution should guarantee that we choose the optimum localization model concerning the requirements of each user. These results are obtained from real-life experiments that are conducted in [51].

TABLE 1. Simulation parameters.

Symbol	Description	Value
Q	No. of Localization models	6
N	No. of user Devices	$1-80$
K	No. of MEC servers	$1-25$
G	No. of Anchors	10
$R(j)$	Maximum capacity of each MEC server	$1-10$
$\Upsilon(i)$	Latency requirement for each user device i	$2-10s$
$H(i)$	Accuracy requirement for each user device i	$5-10m$
$\exists(i)$	Battery lifetime of each user device i	$1-10$
δ	a constant value that represents the average workload of the localization process	$200\ kB$
ϱ	Size of localization request	$152\ bits$ [48]
υ	Size of localization response	$152\ bits$ [48]
$\omega_{l,M}$	Distance between the user device and the MEC server	$0.05\ km$ [52]
$\omega_{l,C}$	Distance between the user device i and the cloud server	$1000\ km$ [52]
$\omega_{M,C}$	Distance between the MEC server and cloud	$1000\ km$ [52]
c	Speed of light	$3.10^8\ m/s$
$\Gamma_{l,M}$	Transmission rate between the user device and the MEC server	$5-10Mbits/s$ [53]
$\Gamma_{M,C}$	Transmission rate between the MEC server and cloud	$10-25Mbits/s$ [53]
$\Gamma_{l,C}$	Transmission rate between the user device and the cloud server	$25-50Mbits/s$ [53]
ρ_l	No. of CPU cycles to process one bit on user device server	1000 [54]
ρ_M	No. of CPU cycles to process one bit on MEC server	1000 [54]
ρ_C	No. of CPU cycles to process one bit on cloud server	1000 [54]
f_l	computational capability of the user device	$2.5\ GHz$ [55]
$f_M(j)$	Computational capability of each MEC server	$5\ GHz$ [55]
f_C	computational capability of the cloud server	$5\ GHz$ [55]
$\chi_p(i)$	power consumed of each user device due to processing	$0.3\ W$ [56]
$\chi_s(i)$	power consumed of each user device due to sending	$0.2\ W$ [56]
$\chi_r(i)$	power consumed of each user device due to receiving	$0.2\ W$ [56]
$\chi_o(i)$	power consumed of each user device when being idle	$0.03\ W$ [56]
$\chi_r(g)$	Power received by Anchors	$-120-90\ db$

TABLE 2. Comparison between performance metrics of each combination of localization technique and technology.

Combination of techniques and technologies	Accuracy (m)	Power Consumption (mW)	Latency (sec)
1) WiFi FP	1.81	324	5.56
2) IEEE 802.15.4 FP	2.03	74.16	1.82
3) BLE FP	3.72	30	3.04
4) WiFi RSSI	3.65	324	3
5) IEEE 802.15.4 RSSI	4.06	74.16	0.5
6) BLE RSSI	3.85	30	2.5

B. NUMERICAL RESULTS

1) AVERAGE LATENCY AND ENERGY CONSUMPTION OF TWO CONVENTIONAL OFFLOADING SCHEMES

To evaluate the performance of the optimal offloading scheme, ILLOO algorithm is compared with other two schemes:

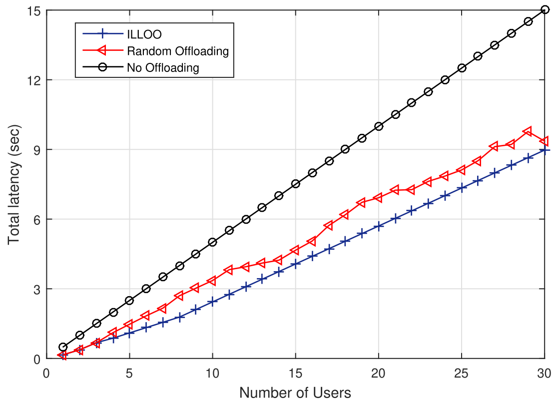

FIGURE 4. Total latency versus different number of users for different offloading schemes.

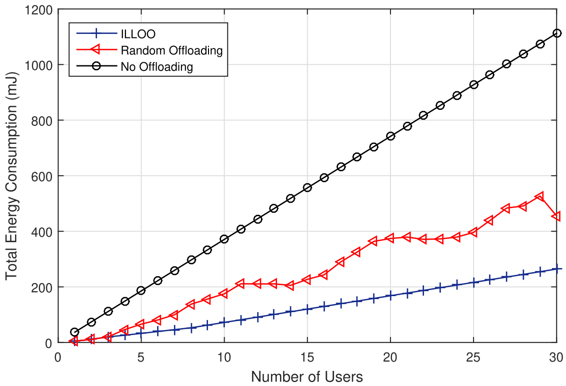

FIGURE 5. Total energy consumption versus different number of users for different offloading schemes.

Local Computing (No Offloading): In this scheme, the user always chooses to process the localization task on his own device. In this case, the offloading decision variable $a(i)$ is always 1. The user always chooses the localization model that gives minimum latency which in this case IEEE 802.15.4 RSSI where $L = 0.5\ s$, due to the large tolerance of accuracy and battery lifetime that is assumed in the simulation parameters. Thus, the results of local computing are linear as shown Figure.4 and Figure.5.

Random Offloading Scheme: In this scheme, each user chooses randomly to compute the localization task on his own device or on the MEC and the cloud servers. Thus, $a(i) = rand(1, 2, 3)$. In each case the constraints of the problem must be satisfied, if not, another random selection will be done on the remaining two choices. This algorithm is done once for each user. The results in Figure.4 showed the improvement of the total latency by 70.36% when it is compared with local computing. This is due to the difference of the computational capability of the user device and the remote servers. Latency here depends on both processing the task and offloading it. Thus, if a certain task is processed on the user device, it will take much more time when compared to if it is transmitted and processed on other remote servers with a higher processing capability. Hence, users can reduce

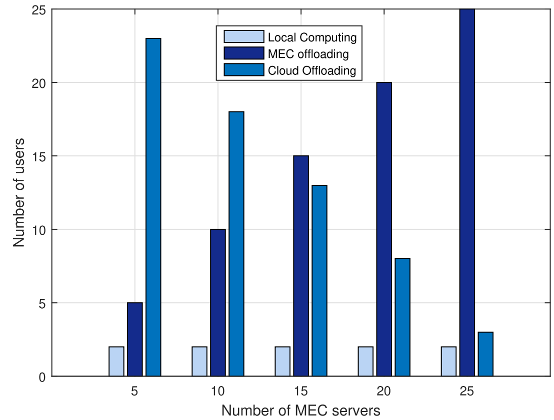

FIGURE 6. Offloading decision of all users with different number of MEC servers.

more latency and energy by the proposed offloading schemes than by local computing and random offloading. Accordingly, we can prove the necessity of the proposed computation offloading algorithms.

2) OFFLOADING DECISION OF ALL USERS WITH DIFFERENT NUMBER OF MEC SERVERS

Figure.6 represents the optimal offloading decision that is obtained from ILLOO algorithm. The simulation runs on $N = 30$ and K ranges from $5 - 25$. The capacity of each MEC server is assumed to be 1. As shown in the figure, there are three offloading decisions: Local Computing, MEC offloading and Cloud offloading. According to our simulation parameters, it is shown that at any number of MEC servers, all MEC servers will be utilized. This proves the necessity of MEC offloading.

3) AVERAGE LATENCY AND ENERGY CONSUMPTION OF PROPOSED ALGORITHMS WITH DIFFERENT NUMBER OF USERS

In order to evaluate the efficiency of ILLCO and ILLGO algorithms, the total latency and the total energy consumption of the system of each algorithm are compared to the optimal solution obtained from the benchmark ILLOO algorithm. As shown in Figure. 7 and Figure. 8 for different number of devices, ILLCO and ILLGO can reach a sub optimal solution. These results prove the effectiveness of the proposed algorithm, as although the objective function is to minimize the total latency of the system, this did not have an adverse impact on the total energy consumption. It is found that at $N = 30$ ILLGO has an optimality gap equal to 17.18% in terms of latency and equal to 16.68% in terms of energy consumption. The optimality gap means the difference between the optimal solution and the best possible obtained solution from a certain algorithm.

4) AVERAGE LATENCY AND OF PROPOSED ALGORITHMS WITH DIFFERENT MEC CAPACITY

Figure. 9 represents the impact of the MEC capacity on the total latency. The simulation runs on $N = 25$ and $k = 5$.

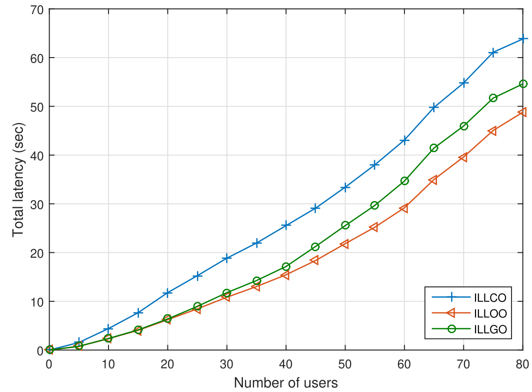

FIGURE 7. Comparison of total latency of the system for ILLOO, ILLCO and ILLGO versus number of users.

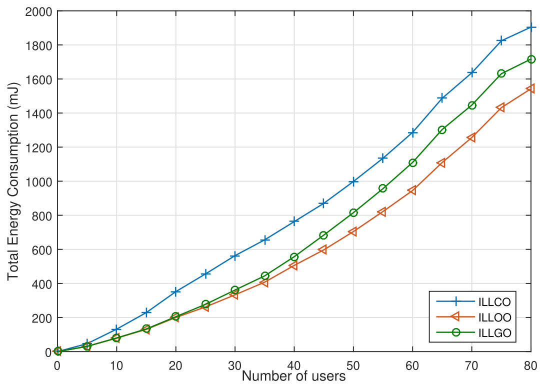

FIGURE 8. Comparison of total energy consumption of the system for ILLOO, ILLCO and ILLGO versus number of users.

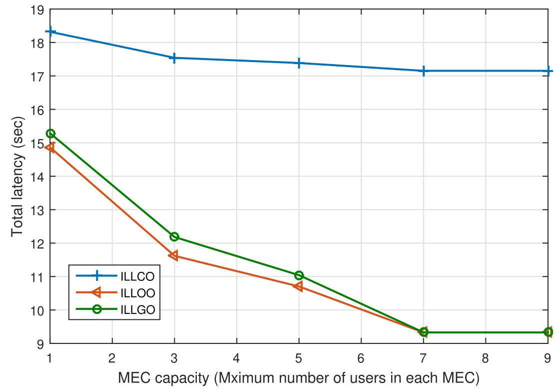

FIGURE 9. Comparison of total latency of the system for ILLOO, ILLCO and ILLGO versus MEC capacity (number of tasks).

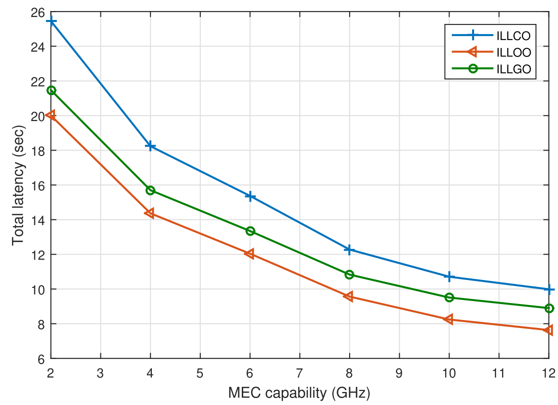

FIGURE 10. Comparison of total latency of the system for ILLOO, ILLCO and ILLGO versus MEC capability (CPU cycles per second).

As the capacity of the MEC servers increases, the total latency decreases. Capacity here means the maximum number of tasks that each MEC server can process. It is assumed that each user has a single task. It is shown that when the capacity of the MEC server reaches 7, ILLGO succeeds in reaching the optimal solution. The reason behind this that we have only 25 users in our simulation, so having 5 MEC servers and each with capacity 7 means that the total capacity on MEC servers is 35. Accordingly, the optimal solution of each user is to offload the task on MEC servers, thus ILLGO succeeds to give same results as ILLOO with higher capacity.

5) AVERAGE LATENCY OF PROPOSED ALGORITHMS WITH DIFFERENT MEC CAPABILITY

Figure. 10 shows the effect of the computational capability of MEC server on the total latency. Capability here means the processing capability which is the number of CPU cycles per second. The simulation runs on $N = 25$ and $k = 5$. The capability of the MEC servers ranges from 2 to 12 GHz. It is shown that as the MEC capability increases, the total latency decreases. It is found that at $f_M(j) = 7\ GHz$ ILLGO has an optimality gap only equal to 9.09%. This gap remains nearly constant with different MEC capabilities.

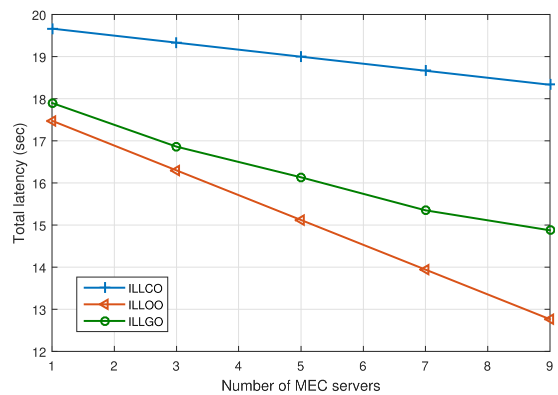

FIGURE 11. Comparison of total latency of the system for ILLOO, ILLCO and ILLGO versus number of MEC servers.

6) AVERAGE LATENCY OF PROPOSED ALGORITHMS WITH DIFFERENT NUMBER OF MEC SERVERS

Figure. 11 illustrates the effect of increasing the number of MEC servers on the total latency. The simulation runs on $N = 25$. As the number of MEC servers increases, the total latency decreases. It is clear that the worst latency appears in the case of a single MEC. Although the latency is improved by deploying more MEC servers, this will increase the total

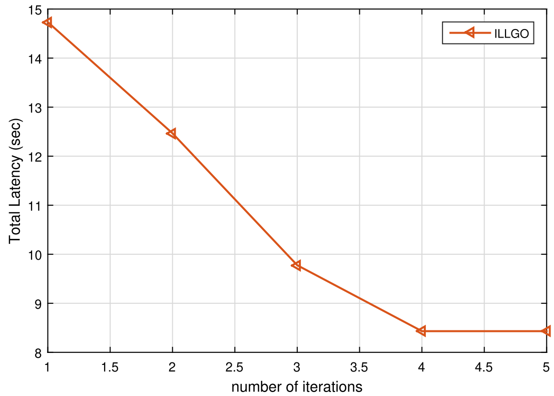

FIGURE 12. Convergence behavior of ILLGO in terms of the latency.

cost of the system. Thus, a trade-off between the cost of the system and the latency must be considered.

7) CONVERGENCE ANALYSIS OF ILLGO

The system is considered to be in an equilibrium state if the offloading decisions of all users remain constant for more than 1 iteration. The simulation runs on $N = 21$ and $k = 3$. As shown in Figure. 12 the proposed ILLGO algorithm converges to a stable point after 4 iterations. This point represents the minimum value of the total latency of the system. Hence, ILLGO proved to have the ability to reach NE after finite improvement steps.

VI. CONCLUSION AND FUTURE WORK

In this paper, the problem of collaborative computation offloading under the MEC environment is formulated to find the accurate location of each user device with the minimum latency. Each user device chooses to compute the localization technique with the transmission technology locally or on a remote server (the MEC server or the cloud server). Three proposed algorithms are implemented. a) ILLOO: It acts as a benchmark and delivers the optimal solution. b) ILLCO: it is a heuristic centralized technique that delivers a sub optimal solution with a lower complexity. c) ILLGO: it is a decentralized game theory technique that guarantees an equilibrium state for all devices and has the least complexity. Simulation results proved that offloading the localization process to remote servers improves the performance of the system in terms of latency reduction and energy saving. Moreover, it is shown that ILLGO can reach a sub optimal solution after finite improvement steps. For future work, we may consider more indoor localization techniques and different transmission technologies. We plan to study the impact of partial offloading, where the computational task will be considered as a group of sub-tasks, some of them can be processed locally and the others are offloaded to remote servers with higher computational capability.

REFERENCES

[1] S. N. Ghorpade, M. Zennaro, and B. S. Chaudhari, "GWO model for optimal localization of IoT-enabled sensor nodes in smart parking systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 2, pp. 1217–1224, Feb. 2021, doi: 10.1109/TITS.2020.2964604.

[2] V. Bianchi, P. Ciampolini, and I. De Munari, "RSSI-based indoor localization and identification for ZigBee wireless sensor networks in smart homes," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 2, pp. 566–575, Feb. 2019, doi: 10.1109/TIM.2018.2851675.

[3] H. Flatt, N. Koch, C. Rocker, A. Gunter, and J. Jasperneite, "A context-aware assistance system for maintenance applications in smart factories based on augmented reality and indoor localization," in *Proc. IEEE 20th Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2015, pp. 1–4, doi: 10.1109/ETFA.2015.7301586.

[4] K. Lin, M. Chen, J. Deng, M. M. Hassan, and G. Fortino, "Enhanced fingerprinting and trajectory prediction for IoT localization in smart buildings," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 3, pp. 1294–1307, Jul. 2016, doi: 10.1109/TASE.2016.2543242.

[5] G. Paolini, D. Masotti, F. Antoniazzi, T. S. Cinotti, and A. Costanzo, "Fall detection and 3-D indoor localization by a custom RFID reader embedded in a smart e-health platform," *IEEE Trans. Microw. Theory Techn.*, vol. 67, no. 12, pp. 5329–5339, Dec. 2019, doi: 10.1109/TMTT.2019.2939807.

[6] M. K. Huynh and D. A. Nguyen, "A research on automated guided vehicle indoor localization system via CSI," in *Proc. Int. Conf. Syst. Sci. Eng. (ICSSE)*, Jul. 2019, pp. 581–585, doi: 10.1109/ICSSE.2019.8823560.

[7] A. Sciarrone, C. Fiandrino, I. Bisio, F. Lavagetto, D. Kliazovich, and P. Bouvry, "Smart probabilistic fingerprinting for indoor localization over fog computing platforms," in *Proc. 5th IEEE Int. Conf. Cloud Netw. (Cloudnet)*, Oct. 2016, pp. 39–44, doi: 10.1109/CloudNet.2016.43.

[8] C. Liu, C. Wang, J. Luo, and Q. He, "A cooperative indoor localization enhancement framework on edge computing platforms for safety-critical applications," in *Proc. 15th Int. Conf. Mobile Ad-Hoc Sensor Netw. (MSN)*, Dec. 2019, pp. 372–377, doi: 10.1109/MSN48538.2019.00077.

[9] X. Zhu, W. Qu, T. Qiu, L. Zhao, M. Atiquzzaman, and D. O. Wu, "Indoor intelligent fingerprint-based localization: Principles, approaches and challenges," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 4, pp. 2634–2657, 4th Quart., 2020, doi: 10.1109/COMST.2020.3014304.

[10] S. Sadowski and P. Spachos, "RSSI-based indoor localization with the Internet of Things," *IEEE Access*, vol. 6, pp. 30149–30161, 2018, doi: 10.1109/ACCESS.2018.2843325.

[11] J. Shen, A. F. Molisch, and J. Salmi, "Accurate passive location estimation using TOA measurements," *IEEE Trans. Wireless Commun.*, vol. 11, no. 6, pp. 2182–2192, Jun. 2012, doi: 10.1109/TWC.2012.040412.110697.

[12] A. Makki, A. Siddig, M. Saad, J. R. Cavallaro, and C. J. Bleakley, "Indoor localization using 802.11 time differences of arrival," *IEEE Trans. Instrum. Meas.*, vol. 65, no. 3, pp. 614–623, Mar. 2016, doi: 10.1109/TIM.2015.2506239.

[13] Y. Zheng, M. Sheng, J. Liu, and J. Li, "Exploiting AoA estimation accuracy for indoor localization: A weighted AoA-based approach," *IEEE Wireless Commun. Lett.*, vol. 8, no. 1, pp. 65–68, Feb. 2019, doi: 10.1109/LWC.2018.2853745.

[14] K. Wu, J. Xiao, Y. Yi, D. Chen, X. Luo, and L. M. Ni, "CSI-based indoor localization," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 7, pp. 1300–1309, Jul. 2013, doi: 10.1109/TPDS.2012.214.

[15] C. Wu, Z. Yang, Y. Liu, and W. Xi, "WILL: Wireless indoor localization without site survey," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 4, pp. 839–848, Apr. 2013, doi: 10.1109/TPDS.2012.179.

[16] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 37, no. 6, pp. 1067–1080, Nov. 2007, doi: 10.1109/TSMCC.2007.905750.

[17] A. Motroni, A. Buffi, and P. Nepa, "A survey on indoor vehicle localization through RFID technology," *IEEE Access*, vol. 9, pp. 17921–17942, 2021, doi: 10.1109/ACCESS.2021.3052316.

[18] W. Bing, L. Yanyan, L. Qingquan, and Z. Yan, "A high-precision dynamic indoor localization algorithm based on UWB technology," in *Proc. Ubiquitous Positioning, Indoor Navigat. Location-Based Services (UPINLBS)*, Mar. 2018, pp. 1–7, doi: 10.1109/UPINLBS.2018.8559902.

[19] J. Schmid, M. Volker, T. Gadeke, P. Weber, W. Stork, and K. D. Müller-Glaser, "An approach to infrastructure-independent person localization with an IEEE 802.15.4 WSN," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat.*, Sep. 2010, pp. 1–9, doi: 10.1109/IPIN.2010.5646831.

[20] S. Tao, M. Kudo, B.-N. Pei, H. Nonaka, and J. Toyama, "Multiperson locating and their soft tracking in a binary infrared sensor network," *IEEE Trans. Human-Mach. Syst.*, vol. 45, no. 5, pp. 550–561, Oct. 2015, doi: 10.1109/THMS.2014.2365466.

[21] J.-W. Yoon and T. Park, "Maximizing localization accuracy via self-configurable ultrasonic sensor grouping using genetic approach," *IEEE Trans. Instrum. Meas.*, vol. 65, no. 7, pp. 1518–1529, Jul. 2016, doi: 10.1109/TIM.2016.2534319.

[22] R. Li, Z. Fang, B. Hao, and F. Yang, "Research on indoor wireless localization system for radioactive sources based on ZigBee," in *Proc. Int. Conf. Comput., Control Ind. Eng.*, Jun. 2010, pp. 359–362, doi: 10.1109/CCIE.2010.208.

[23] C. Basri and A. El Khadimi, "Survey on indoor localization system and recent advances of WiFi fingerprinting technique," in *Proc. 5th Int. Conf. Multimedia Comput. Syst. (ICMCS)*, Sep. 2016, pp. 253–259, doi: 10.1109/ICMCS.2016.7905633.

[24] M. A. Al-Ammar, S. Alhadhrami, A. Al-Salman, A. Alarifi, H. S. Al-Khalifa, A. Alnafessah, and M. Alsaleh, "Comparative survey of indoor positioning technologies, techniques, and algorithms," in *Proc. Int. Conf. Cyberworlds*, Oct. 2014, pp. 245–252, doi: 10.1109/CW.2014.41.

[25] A. Yassin, Y. Nasser, M. Awad, A. Al-Dubai, R. Liu, C. Yuen, R. Raulefs, and E. Aboutanios, "Recent advances in indoor localization: A survey on theoretical approaches and applications," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 1327–1346, 2nd Quart., 2016, doi: 10.1109/COMST.2016.2632427.

[26] F. Zafari, A. Gkelias, and K. K. Leung, "A survey of indoor localization systems and technologies," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2568–2599, 3rd Quart., 2017, doi: 10.1109/COMST.2019.2911558.

[27] D. Vasisht, S. Kumar, and D. Katabi, "Decimeter-level localization with a single WiFi access point," in *Proc. 13th USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, Mar. 2016, pp. 165–178. [Online]. Available: https://www.usenix.org/conference/nsdi16/technical-sessions/presentatio%n/vasisht

[28] K. Liu, X. Liu, and X. Li, "Guoguo: Enabling fine-grained smartphone localization via acoustic anchors," *IEEE Trans. Mobile Comput.*, vol. 15, no. 5, pp. 1144–1156, May 2016, doi: 10.1109/TMC.2015.2451628.

[29] P. Kriz, F. Maly, and K. Tomás, "Improving indoor localization using Bluetooth low energy beacons," *Mobile Inf. Syst.*, vol. 2016, pp. 1–11, Apr. 2016, doi: 10.1155/2016/2083094.

[30] J. Gjengset, J. Xiong, G. McPhillips, and K. Jamieson, "Phaser: Enabling phased array signal processing on commodity WiFi access points," in *Proc. 20th Annu. Int. Conf. Mobile Comput. Netw.*, Sep. 2014, pp. 153–164, doi: 10.1145/2639108.2639139.

[31] J. T. Biehl, M. Cooper, G. Filby, and S. Kratz, "LoCo: A ready-to-deploy framework for efficient room localization using Wi-Fi," in *Proc. ACM Int. Joint Conf. Pervas. Ubiquitous Comput.*, Sep. 2014, pp. 183–187, doi: 10.1145/2632048.2636083.

[32] E. Martin, O. Vinyals, G. Friedland, and R. Bajcsy, "Precise indoor localization using smart phones," in *Proc. Int. Conf. Multimedia (MM)*, 2010, pp. 787–790, doi: 10.1145/1873951.1874078.

[33] A. A. N. Shirehjini, A. Nazari, A. Yassine, and S. Shirmohammadi, "An RFID-based position and orientation measurement system for mobile objects in intelligent environments," *IEEE Trans. Instrum. Meas.*, vol. 61, no. 6, pp. 1664–1675, Jun. 2012, doi: 10.1109/TIM.2011.2181912.

[34] L. M. Ni, Y. Liu, Y. Cho Lau, and A. P. Patil, "LANDMARC: Indoor location sensing using active RFID," in *Proc. 1st IEEE Int. Conf. Pervas. Comput. Commun. (PerCom)*, Mar. 2003, pp. 407–415, doi: 10.1109/PERCOM.2003.1192765.

[35] H. Guo, J. Liu, and J. Zhang, "Efficient computation offloading for multi-access edge computing in 5G HetNets," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–6, doi: 10.1109/ICC.2018.8422238.

[36] K. Li, "Computation offloading strategy optimization with multiple heterogeneous servers in mobile edge computing," *IEEE Trans. Sustain. Comput.*, early access, Mar. 12, 2019, doi: 10.1109/TSUSC.2019.2904680.

[37] P. Porambage, J. Okwuibe, M. Liyanage, M. Ylianttila, and T. Taleb, "Survey on multi-access edge computing for Internet of Things realization," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2961–2991, 4th Quart., 2018, doi: 10.1109/COMST.2018.2849509.

[38] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 450–465, Feb. 2018, doi: 10.1109/JIOT.2017.2750180.

[39] Y. Chen, E. Sun, and Y. Zhang, "A weight factors localization algorithm in fog-supported wireless sensor networks," in *Proc. IEEE 11th Int. Conf. Commun. Softw. Netw. (ICCSN)*, Jun. 2019, pp. 268–274, doi: 10.1109/ICCSN.2019.8905271.

[40] W. Li, Z. Chen, W. Liu, J. Wang, and X. Gao, "Multimodel framework for indoor localization under mobile edge computing environment," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4844–4853, Jun. 2019, doi: 10.1109/JIOT.2018.2872133.

[41] 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, doi: 10.1109/COMST.2017.2682318.

[42] A. U. R. Khan, M. Othman, A. N. Khan, J. Shuja, and S. Mustafa, "Computation offloading cost estimation in mobile cloud application models," *Wireless Pers. Commun.*, vol. 97, no. 3, pp. 4897–4920, Dec. 2017, doi: 10.1007/s11277-017-4757-3.

[43] E. Ahmed, A. Ahmed, I. Yaqoob, J. Shuja, A. Gani, M. Imran, and M. Shoaib, "Bringing computation closer toward the user network: Is edge computing the solution?" *IEEE Commun. Mag.*, vol. 55, no. 11, pp. 138–144, Nov. 2017, doi: 10.1109/MCOM.2017.1700120.

[44] T. Zheng, J. Wan, J. Zhang, C. Jiang, and G. Jia, "A survey of computation offloading in edge computing," in *Proc. Int. Conf. Comput., Inf. Telecommun. Syst. (CITS)*, Oct. 2020, pp. 1–6, doi: 10.1109/CITS49457.2020.9232457.

[45] X. Shan, H. Zhi, P. Li, and Z. Han, "A survey on computation offloading for mobile edge computing information," in *Proc. IEEE IEEE 4th Int. Conf. Big Data Secur. Cloud (BigDataSecurity) Int. Conf. High Perform. Smart Comput., (HPSC) Int. Conf. Intell. Data Secur. (IDS)*, May 2018, pp. 248–251, doi: 10.1109/BDS/HPSC/IDS18.2018.00060.

[46] J. Shuja, S. Mustafa, R. W. Ahmad, S. A. Madani, A. Gani, and M. K. Khan, "Analysis of vector code offloading framework in heterogeneous cloud and edge architectures," *IEEE Access*, vol. 5, pp. 24542–24554, 2017, doi: 10.1109/ACCESS.2017.2713818.

[47] M. Zamzam, T. El-Shabrawy, and M. Ashour, "Game theory for computation offloading and resource allocation in edge computing: A survey," in *Proc. 2nd Novel Intell. Lead. Emerg. Sci. Conf. (NILES)*, Oct. 2020, pp. 47–53, doi: 10.1109/NILES50944.2020.9257921.

[48] A. Guidara, G. Fersi, and F. Derbel, "Lookup service for fog-based indoor localization platforms using chord protocol," in *Proc. Int. Wireless Commun. Mobile Comput. (IWCMC)*, Jun. 2020, pp. 345–350, doi: 10.1109/IWCMC48107.2020.9148348.

[49] S. P. Khare, J. Sallai, A. Dubey, and A. Gokhale, "Short paper: Towards low-cost indoor localization using edge computing resources," in *Proc. IEEE 20th Int. Symp. Real-Time Distrib. Comput. (ISORC)*, May 2017, pp. 28–31, doi: 10.1109/ISORC.2017.23.

[50] M. Zamzam, T. El-Shabrawy, and M. Ashour, "Energy-efficient computation offloading for indoor localization based on game theory," in *Proc. 2nd Novel Intell. Lead. Emerg. Sci. Conf. (NILES)*, Oct. 2020, pp. 270–275, doi: 10.1109/NILES50944.2020.9257948.

[51] P. Crombez, J. Gesquiere, G. Glorioso, C. Villoute, W. Yves, N. Wilstrom, T. V. Haute, E. D. Poorter, V. Handziski, F. Lemic, and A. Wolisz, "Evaluation of RF-based indoor localization solutions for the future internet," in *Proc. Future Netw. Mobile Summit Conf.*, Jul. 2013, pp. 3–5.

[52] 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, doi: 10.1109/TVT.2018.2790421.

[53] H. Sun, H. Yu, G. Fan, and L. Chen, "Energy and time efficient task offloading and resource allocation on the generic iot-fog-cloud architecture," *Peer Peer Netw. Appl.*, vol. 13, pp. 548–563, Mar. 2020, doi: 10.1007/s12083-019-00783-7.

[54] D. Nowak, T. Mahn, H. Al-Shatri, A. Schwartz, and A. Klein, "A generalized Nash game for mobile edge computation offloading," in *Proc. 6th IEEE Int. Conf. Mobile Cloud Comput., Services, Eng. (MobileCloud)*, Mar. 2018, pp. 95–102, doi: 10.1109/MobileCloud.2018.00022.

[55] C. Wang, C. Dong, J. Qin, X. Yang, and W. Wen, "Energy-efficient offloading policy for resource allocation in distributed mobile edge computing," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jun. 2018, pp. 366–372, doi: 10.1109/ISCC.2018.8538612.

[56] H. Wu, Y. Sun, and K. Wolter, "Energy-efficient decision making for mobile cloud offloading," *IEEE Trans. Cloud Comput.*, vol. 8, no. 2, pp. 570–584, Apr. 2020, doi: 10.1109/TCC.2018.2789446.

MARWA ZAMZAM (Graduate Student Member, IEEE) received the B.Sc. and M.Sc. degrees in networking engineering from the Faculty of Information Engineering and Technology, German University in Cairo, New Cairo, Egypt, in 2016 and 2018, respectively, where she is currently pursuing the Ph.D. degree with the Department of Networks, Faculty of Information Engineering and Technology. She is currently an Assistant Lecturer with the Department of Networks, Faculty of Information Engineering and Technology, German University in Cairo. Her research interests include mobile edge computing, indoor localization, and artificial intelligence techniques.

TALLAL ELSHABRAWY received the B.Sc. degree in electrical engineering from Ain Shams University, Cairo, Egypt, in 1998, the M.A.Sc. degree in electrical and computer engineering from Concordia University, Montreal, Canada, in 2000, the Ph.D. degree in electrical and computer engineering from McGill University, Montreal, in 2006, and the M.B.A. degree from The American University in Cairo, in June 2014, with a focus on finance. He has also complemented his technical expertise with academic business experience from The American University in Cairo. Since 2006, he has been with the Department of Networking, Faculty of Information Engineering and Technology, German University in Cairo. His current research interests include development and implementation of joint physical and MAC layer techniques for low-power wide area networks.

MOHAMED ASHOUR received the B.Sc. and M.Sc. degrees in electrical engineering from Ain Shams University, Cairo, Egypt, and the Ph.D. degree in electrical and computer engineering from McGill University, Montreal, Canada. He is currently working as a Lecturer with the Faculty of Information Engineering and Technology, German University in Cairo. He worked for Hughes Networks, General Dynamics, and Egyptian Cabinet Information and Decision Support Center (IDSC), as a Professional Telecommunications and Network Engineer. He has a rigorous professional experience. He has contributed to research related to planning and optimization of indoor wireless networks, resource allocation in WiMAX networks, power and frequency allocation in wireless mesh networks, satellite-based internet, cross layer wireless *ad-hoc* routing, traffic engineering of MPLS networks, and multi-scale queuing analysis network protocol design. This research included cooperation with Canadian Institute for Telecommunication Research, Ericsson Research Canada, Vodafone Egypt, and Egyptian Information Technology Industry Development Agency (ITIDA). He has more than 60 publications in the field of wired and wireless telecommunication networks.