TELKOMNIKA Telecommunication Computing Electronics and Control
Vol. 21, No. 4, August 2023, pp. 836~845
ISSN: 1693-6930, DOI: 10.12928/TELKOMNIKA.v21i4.24350

Power consumption and energy management for edge computing: state of the art

Tawfeeq E. Abdoulabbas[1], Sawsan M. Mahmoud[2]
[1]Department of Electrical Engineering, Faculty of Engineering, University of Mustansiriyah, Baghdad, Iraq
[2]Department of Computer Engineering, Faculty of Engineering, University of Mustansiriyah, Baghdad, Iraq

Article Info

Article history:

Received Aug 04, 2022
Revised Dec 13, 2022
Accepted Feb 16, 2023

Keywords:

Edge computing
Energy efficiency
Offloading manager
Power management

ABSTRACT

Edge computing aims to make internet-based services and remote computing power close to the user by placing information technology (IT) infrastructure at the network edges. This proximity provides data centers with low-latency and context-aware services. Edge computing power consumption is mainly caused by data centers, network equipment, and user equipment. With edge computing (EC), energy management platforms for residential, industrial, and commercial sectors are built. Energy efficiency is considered to be one of the key aspects of edge power constraints. This paper provides the state of the art of power consumption and energy management for edge computing, the computation offloading methods, and more important highlights the power efficiency of edge computing systems. Furthermore, renewable energy and related concepts will also be explored and presented since no human participation is required in replacing or recharging batteries when using such energy sources. Based on such study, a recommendation is to develop a dynamic system for energy management in real-time with the assessment of local renewable energy so that the system be reliable with minimum power consumption. Also, regarding energy management, we recommend providing backup energy sources (or using more than one energy source) or (a hybrid technique).

Corresponding Author:

Sawsan M. Mahmoud
Department of Computer Engineering, Faculty of Engineering, University of Mustansiriyah
Baghdad, Iraq
Email: sawsan.mahmoud@uomustansiriyah.edu.iq

1. INTRODUCTION

Using edge computing (EC) over cloud computing (CC) provides some improvements and extension capabilities and also there is ensuring of low latency within the service. The Internet of Things (IoT) uses a variety of devices to collect real-time data, exchange information, store data, make computations, and provide services. IoT technology is developing rapidly, but on the other hand, it is constrained by its limited power sources with the devices of power demanded that must be reduced to ensure sustainable operation. It is good to use special technologies such as compressed sensing and transmitting data to provide devices of low-power consumption. From an architectural point of view, these devices must be energy efficient and require treatments such as energy management with low energy [1]. Energy management can be viewed as a set of techniques for controlling or managing multiple power sources to provide efficient power delivery in networked systems and to ensure continuous operating conditions in spite of the available limited power sources. Power management schemes can be divided into two aspects; energy supplying and energy consuming aspects. Energy supply related to the characteristics of the sources required for supplying and transferring checks, e.g., sensor nodes are often powered by connected batteries to ensure energy-efficient operation of the network, but these batteries have limited harvesting capacity and sometimes need to be replaced or recharged, so the management is required with this situation to provide an alternative option for this power supply [2]. On the other hand, the energy consumption should be determined so that a suitable energy management method could be specified. For example, smart mobile devices (SMDs) typically have limited capacity processing and fixed battery energy. But there is a lot of power consumption needed for tasks computing in the mobile edge computing (MEC) server [3]. For that reason, a method like computation offloading is needed to minimize the overall energy consumption. Our contributions to this paper are stated as:

- Studying and highlighting the performance of energy efficiency (EE) in EC and MEC to infer the optimization of EC systems.
- Discuss various energy management mechanisms and make an overall insight classification upon them.
- Studying and explaining the offloading technique and the main computations of offloading techniques.
- Introducing the concepts of renewable energy and green energy as energy harvesting (EH) for mobile along with challenges like resource allocation for green energy and computation offloading.

This paper is structured as: in the next section, we discuss the EE and its enhancement methods according to different aspect layers involving the discussion of the green energy and renewable energy powered MEC System. Section 3 will present the fundamentals of the offloading techniques in detail. Finally, section 5 concludes and lists some recommendations from the paper.

2. ENERGY EFFICIENCY

In general, EE is defined as the ability of a system to perform a specified task like transferring a bulk of data traffic with saving power to meet the quality of services needed for a given service i.e., performing the same task to produce the same result but with less energy needed. With the edge computing the total amount of data traversing the network is reduced so the energy consumption is decreased which improve EE. An EE metric is expressed as [4]:

$$\eta = \frac{output\ power}{consumed\ power};\ or\ \eta = \frac{power}{traffic};\ or\ EE_{UE} = \frac{P_{UE}}{T_{UE}}\ [W/bps] or [J/bit] \quad (1)$$

The meaning of EE depends on the way of considering of the power of each part. For example, P_{UE} is the power of the mobile terminal(s) and T_{UE} is the power of traffic (at the user level). With the increasing services required at the edge network of EC infrastructure, this will lead to the growth of energy demand. For this reason, EE is one of the most important things in the designing of the EC system [5]. It can be seen that the EE of a data center in the cloud is greater than that with the edge data centers (DCs). Also, in IoT applications, sending computational tasks to the edge of the server have EE better than in the cloud. But on the other side, the network latency of the cloud system is greater with any workloads assigned (far users from DCs). These facts were investigated in [6] by introducing a way of edge/cloud energy management for wide area network (WAN) using a network algorithm based on a software-defined network (SDN) with studying the interchange of energy and the challenge for both types (edge/cloud). It has been shown that edge traffic can handle the latency problem considering the short paths of the infrastructure of the edge network dependent.

2.1. Energy efficiency enhancement on EC

Much research is done to solve the difficulties associated with the limited battery capacity and enhance the EE on EC. A survey in [1] presented a way of multiple batteries source-driven schemes of energy consumption are used to ensure more EE network operation and get continuous energy preservation. EH can be used as a conservation power source with battery supplying. From that sturdy, the wireless sensor network (WSN) is classified to be a high level of energy management, and a wireless transferring energy is introduced as another feeding to ordinary batteries [1]. EC can be studied as three-layers as shown in Figure 1 [2]. The figure illustrates the layout of the process of EE enhancement on EC.

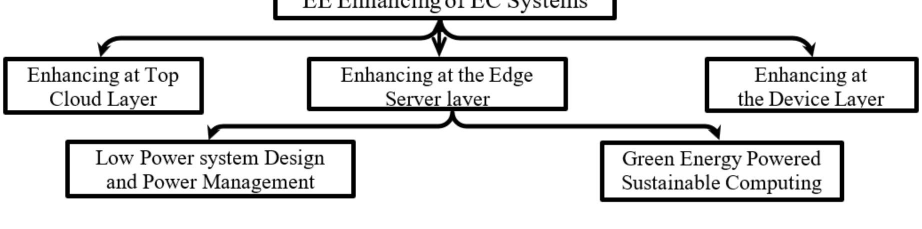

Figure 1. Energy efficiency enhancement layout

2.1.1. Enhancing at the top cloud layer

The remote computation capacity at the edge of MEC networks is very close to the SMDs work nature of bounded capacity processing and limited battery power. Many computations and jobs consume a lot of energy so the solution is to offload these jobs to the MEC server especially when the battery is run out. The following issues should be taken into account when the offloading is chosen [2]: wireless network state, the number of SMDs, radio resources accessibility, and local battery power availability. A multi-user MEC system is proposed in [3] where an increasing demand for computation offloading by multiple SMDs was well-noted. The authors studied the overall energy consumption reduction by introducing an optimization criterion, taking into account the battery’s lifetime. The optimization decision depends on some parameters such as offloading, radio, and local resource computational allocation. Resource partitioning and a heuristic approach are suggested to minimize the overall energy. In [7], it has been shown that the distribution of tasks and IoT devices over multi-computation elements need more energy consumption compared with the centralized scheme. A scheme which was proposed by Jalali *et al*. [8] studied the nano data centers (nDCs) models (or peer to peer P2P scheme) energy consumption by dividing the network energy consumption according to the flow and time of shared and unshared equipment usage. Energy analyzing has been done for applications that running over DC either under cloud centralized DC or fog mode decentralized nano DCs. The results demonstrate that a higher EE may be in the Fog mode. In addition, the applications which generate and distribute huge data near users (i.e., edge networks) can facilitate the way for saving energy. Moreover, the number of hobs from the user to the DC servers has less affecting on the whole energy demand. It has been shown that using nano servers can give the best energy saving with the applications like video surveillance. For this case, it is preferred to allocate data near the requested user because it needs a low rate of access data. So, in cloud applications, the centralized DCs and nano servers together can give the best EE for content storage and distribution [8]. For the energy consumed by equipment versus accessing data according to different locations, it is shown that the lower power of (core/edge) will be for core-local DC than that of other types due to less distance from the user. Also, for the energy consumption of IoT applications running from the cloud /fog with various computations, the IoT needs more energy (transmission and device) where it will be lower for fog and edge of decentralized type. This happened due to the centralized data center with the cloud system [9].

2.1.2. Enhancement at the edge server layer

Energy is considered to be an essential factor in this middle layer of the EC approach, servers might be located in a private room or run on battery packs. Consequently, several power management systems have been suggested to reduce the energy consumption of edge servers while still assuring their performance in order to give a greater level of availability. We provide an overview of the following key tactics employed recently in EC systems’ edge server layer.

a) The low-power management system strategies

The low-power management system strategies are mainly used at the edge servers in EC recently [2]. Such a system type is called cloudlets which are a small data center deployed usually away from mobile devices at one wireless hop. For instance, small clouds at the edge called tactical cloudlet are proposed in [10] where the energy consumption is analyzed via a virtual machine (VM) approach. Then it was appreciated the provision of some cloudlet mechanisms. The maximum rate of energy depends on (VM compounding and the provision of the on-demand VM). It was shown from the results that high EE can be found. Due to the combination of a cloudlet pushed with a cached VM in a cloudlet environment, suitable resources can be provided of offload computing tasks for multiple users [10]. Any IoT devices produce information that will be processed in the extreme device, so the centralized servers will be released from the computational load, this leads to decreasing overloading in the network traffic, and also minimizing the applications response time of IoT devices. For wireless communication (which uses rechargeable sensor networks) there is a problem of high energy consumption with these networks. Network utility maximization is another problem associated with the dynamic-routing, so combined optimization is necessary to be done for routing and rate control with energy management [11]. In a service-related to fog/edge architecture computing, enormous energy is saved by performing data mining with an embedded system. Criteria of raw-data set orders so that the transmission data will be reduced and eventually reduce the energy required [1].

b) Power management techniques with sustainable green energy power system

With the embedding of IoT technology, green energy management is developed with widespread monitoring and reasonable communications in smart cities and sustainable energy. In [9] energy management model based on IoT is implemented using deep reinforcement learning with EC. This model can improve the performance of energy management and also decrease the execution time. The architecture includes mainly three components: energy devices, energy edge servers, and energy cloud servers. The proposed model supported with software contains four layers: sensing layer, network layer, cognition layer, and application layer. Fog computing dual power sources are utilized in [12] to supply the system where the primary power source is solar energy to supply the fog nodes; the secondary power source is the backup battery. In that work, a framework, which is involved with the overall analytical aspect, is proposed to get minimum energy consumption of long-term cost. The framework can provide a high-quality service by using offloading data with an energy-efficient mechanism for fog/cloud nodes. Also, in [13], a powered edge infrastructure is used with a green energy of the rack-range type which is implemented with the help of "in-situ server systems using renewable energy" InSURE system to pre-process data at the edge. InSURE source power is supplied by (solar or wind) standalone power supply supported by a backup source like batteries. For this reason, efficient energy flow control could be insured with the power supply of the edge server that is due to the buffering scheme of energy, and a joint spatial-temporal power management mechanism is applied.

Green energy scheduling and other aspects such as task allocation, and VM migration joint are used in [14] in minimizing the energy cost. To handle the complexity of the computation (with and transmission data) and get the optimal solution, a heuristic algorithm approximating is proposed and attains a much more suitable optimal solution and shows actively more reduction in energy consumption. In that algorithm, the energy consumption might also be affected by the location of the VM by considering the location-related attenuation ratio during transmission [14]. The distribution of the resources and devices (e.g., base stations (BS)) will be managed at a network edge. In this pattern, the applications that need low latency can be implemented at the network edge. Devices topology will make EC an ideal EE platform, so it has an impact on the green energy distribution of EE computing. It is necessary to increase the available green energy so that to get minimum brown energy consumption. This will need concentration job allocation and careful energy scheduling to combine the energy provision with the required demand.

As we know, MEC servers have data centers that are small-scale compared with the data centers in the cloud. So that, each server consumes less energy. Computation workload patterns may have a problem with resource management and it needs some optimization and assessment approaches. So, to achieve green MEC, different approaches are developed [15]. For example, dynamic right sizing, geographical load balancing, and designing MEC use renewable energy.

c) Dynamic right-sizing

The energy consumption at MEC servers can be computed as [15]:

$$E_s = \alpha . E_{max} + (1 - \alpha)u \quad (2)$$

Where, E_{max} is the peak demand server energy; α: the percentage of the inactive system energy; u: central processing unit (CPU) utilization. It can be shown from the above equation when there is a light load; the servers are worked in an inactive mode for EE in MEC. The computation loads are consolidated into a few numbers in active servers. The design was forwarded to energy-proportional servers [16]. Thus, the energy consumption would be proportional to the computation load. The dynamic right-sizing process is one of the methods that are used to realize energy proportional servers by switching the speed of the servers at the edge and the computation loads. The alternating server modes between the active and sleep could cause prejudice as long as potential energy savings. It could tolerate the turning energy cost and application data transmission latency [15].

d) Geographical load balancing for MEC

Here, the idea of balance is coordinating MEC servers with user requisitions. Thus, the processes are managed by the edge server with a high dataflow location to a nearby edge server with a low dataflow location. Hence; the EE of the edge servers will be improved. In addition, the battery life of mobile devices will be extended. For instance, the tasks can be offloaded to a closed server and the energy that is transferred could be saved in this case. Generally, at edge servers, applications like VM management and dynamic right-sizing require effective techniques for resource management [15].

e) Renewable energy-powered MEC systems

Renewable energy is an affirmative feature that can be beneficial for MEC. Mobile devices can use renewable energy as EH can protract their battery life [15]. The use of renewable energy sources makes human participation not required in replacing or recharging batteries. Some challenges are introduced like resource allocation for green energy and computation offloading. One of the limitations in designing renewable energy-powered MEC systems is the fulfilling performance, as renewable energy is mostly coming free. Also, renewable energy has a randomness feature that may affect the offloading reliability and risks of failure. The following possible solutions to handle the indicated issues can be considered [15]:

- Deploying the renewable energy-powered edge servers densely (more offloading will be available).
- It is conceivable to reduce the chance of energy shortage by choosing renewable energy sources.
- Hybrid energy sources are used to power MEC servers so reliability is improved. Harvested energy, electric power and uninterrupted power are equipped to supply units at the edge servers [10].
- A technique of wireless power transfer (WPT) also is adopted, so the battery life will be longer the energy harvested by each user E_{ui} in the download (DL) [17].

$$E_{ui} = \zeta_i P_A h_i \tau_o \tag{3}$$

Where, ζ_i = energy harvesting efficiency at each receiver; P_A = transmit power at the hybrid access point (H-AP); h_i = DL channel power gains; τ_o = time portions in each block allocated to the H-AP. The microgrid is used to provide the renewable energy sources and supplying green energy to the edge computing. To achieve the sustainable conditions operation the edge computing energy usage as well as the power supply of the system should be minimized. So with suitable energy management [18] the power demand can be satisfied according to the power supply from local or external energy sources through direct or indirect load control strategy as shown in Figure 2, where four energy sources are provided i.e., renewable energy generation profile(s), storage profiles (battery), available nearby renewable energy, and grid energy. To maximize the renewable energy utilization with the edge computing side the scheduling techniques (time-sensitive decision) and application programs (energy efficiency and IoT controlling) are used. Accordingly, the required renewable energy of microgrid is provided according the load profiles demand from the edge computing. The red energy is supplied from the traditional electricity supply network.

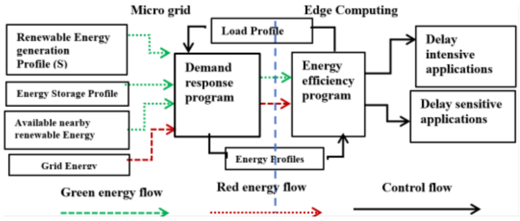

Figure 2. A renewable-energy-driven edge computing system

2.1.3. Enhancement at the device layer (end or bottom layer)

The energy-saving system is to be adopted at this end layer (edge device layer) of the EC diagram. But the short life of the battery needed by IoT used with EC systems constitutes one of the constrained solutions of power consumption, especially for a big operation scale. Cuervo *et al.* [19] propose an architecture named mobile assistance using infrastructure (MAUI), to solve the battery life challenges using mobile code offloading and remote execution. They proved that the energy consumption could be decreased when adopting the MAUI program partitioning mechanism and minimizes. It should be necessary to remember that sometimes power can be consumed by data transmission of user equipment (UE) which will be added to power server consumption where it is shown that the power consumption is conversely proportional (trade-off) with changing of average response time. Meanwhile, with all MECs, the processing times for the offloaded tasks are reduced. The only way to increase the speed of the UE with a little decrease in the average response time is by making a decrement in the energy consumption of data transferring [20]. The average power consumption for computation (UE -server power P_S, switching dynamic power consumption P_d, an activity factor ρ, loading capacitance C, the supply voltage V, and the clock frequency f):

$$P = P_d + P_S = \rho C V^2 f + P_S \tag{4}$$

A new type of communication network had been described in [21] called data and energy integrated communication networks (DEINs) which provide compromises wireless information transfer with wireless energy transfer, to achieve the co-transmission of data and energy, especially for EH the energy transmission using radio frequency instead of information decoding. The force side of the advent of DEINs is the big data, which comes from sensors that produce a large number of small pieces of data. EH has arisen as a technology for charging batteries in wireless communication. Huq *et al.* [22] develop a wireless system that treats the energy efficient in an orthogonal frequency division multiple access. It has been performed to make transmissions by coordinated multi-point between the small cell base stations (BSs) in a heterogeneous network with a technique of 3rd generation partnership project-long term evolution advanced to meet international mobile telecommunications (IMT)-advanced targets. The power consumption in the network is confined to the sum of all BS's power consumption. The power consumption in mobile networks is usually causing overlooked in comparison to radio network BSs [22]. Another model was proposed in [23] to improve EE in mobile computing. In that model, a three-node MEC system is considered: a user node, a helper node, and an access point node in contact with a MEC server. The optimization is made between the following aspects subject to constraints related to the user's computation latency (task partition to overcome most difficulties. Time allocation that makes control for resource allocation, transmit power for the offloading system and CPU frequencies of local processing or (local computing element) in the network at the user) [23].

3. OFFLOADING MANAGER

Generally, implementing applications that require intensive calculation in terms of UE is restricted by a battery capacity and energy-consuming of that equipment. Hence, making the battery life longer is demanded to offload such applications to a common central cloud. However, the process of delivering and back causes a considerable execution delay in addition to calculated time in the cloud. Such a delay is undesirable, especially for real-time applications [24]. Offloading is the way of moving a resource-heavy task from the mobile devices to a nearby device or network or any controller to complete this task and then retrieving the resulting task to the mobile devices so that the energy consumption minimization, quality of service (QoS) guarantee, and quality experiences enhancement [25].

3.1. Computation offloading division

The offloading decision relies on the application model to decide whether it is offloaded or not, whether full or partial offload will be used, and what and how the calculation could be offloaded. Briefly, UE needs to be collected from all aspects (i.e., network and management controller) to manage the offloading process [26]. A computation offloading could be included in the four areas [27].

- The UE computation offloading determination is related to energy-consuming and running delay.
- Determining the computing resources allocation within MEC in terms of efficiency.
- Reducing the computation delay and managing the load distributed around resources and network nodes.
- The management of tasks offloaded should ensure the availability of supplemented service with all users.

A ternary decision maker (TDM) offloading framework was presented to reduce the response time and energy consumption as well. The required execution task can be treated with local processors and the cloud which are combined to provide multiple ways of application execution for mobile. A sophisticated decision model was developed in [28]. Depending on the hierarchical structure and the elementary decision model which face the offloading challenges in edge computing by using online scheduling to deal with the requests. To reduce mobile energy consumption the offloading ratio joint optimization and the power transmission should be reduced. A queening theory in [29] is utilized in MEC systems to investigate the energy consumption, delay in execution, and payment cost of the offloading tasks. In [30], conduct an in-depth study of the power consumption, execution latency, and payment cost of offloading processes in fog computing systems using three queuing models to the MD, fog, and cloud centers, respectively, and explicitly consider the data rate and power consumption of the wireless link. A multi-objective optimization problem is formulated with the common goal of minimizing energy consumption, execution delay and payment cost by finding and delivering optimal offloading probabilities vitality. The energy consumption to offload a task t_i is [3]:

$$E_i^o = E_i^T + E_i^s = \frac{p_i d_i}{R_i} + k_i f_i^2 l_i \quad (5)$$

Where, E_i^T = the energy consumption of the communication process; E_i^s = the energy consumption at the MEC server to execute task t_i ; p_i = the transmission undertaken power; d_i = the amount of transfer bits (data + codes) between the user's device and the MEC server; R_i = achievable uplink rate for SMD_i; k_i = the energy coefficient depending on the chip architecture of SMD_i; f_i^2 = CPU frequency of SMD_i; l_i = computation amount (cycles) needed to execute the task. The computation offloading in edge computing faces many challenges: application partitioning, task allocation, and task execution Figure 3 presents a summary of these challenges [30].

3.2. Offloading server strategy

The computation offloading can be classified into four categories [31]:

- Server-to-cloud server offloading: e.g., EC server offloads deliver computation tasks to a cloud server.
- Server-to-another EC server offloading: the process is similar for both CC and EC. It is more challenging in EC due to the different features of EC, EH energy harvest, and EC server's wireless link.
- From end device to EC server: (similar to cloud computing offloading), the computation tasks are offloaded from end devices to cloud server, but offloading from end devices to the EC servers should consider new factors, e.g., energy harvesting and mobility.
- End host-to-server1 offloading (3-tier): the end host decides the process whether it is locally using an EC server or remotely using the cloud server.

The possible solution to the energy challenge is by enabling cooperation between the adjacent mobile (or any controller) and offloading the higher energy-consuming tasks. Besides, it is conceivable to model the offloading problem by investigating the management of energy-efficient resource consumption (e.g., multiuser mobile edge computation offloading models). The study has made with data partition (scheduling) and time division of communication then an optimal resource management is made. It has been found that optimal time-sharing strategies tend to balance the effective computing power of multiple offloading devices by helping time-sharing handsets [32].

Multiple-input multiple-output (MIMO) multi-cell system was considered in [33] where multi-mobile users make a query to offload the computation to a cloud server. Hence, the offloaded problem formed as the linked optimization of the wireless resources assigned to each mobile user by the cloud to overcome consuming energy. Meanwhile, the meeting latency is a constraint. So, the problem was modeled as non-convex in the objective function and constraints. In [34], an algorithm has been performed for the offloading process dynamically using the Lyapunov optimization. Thus, the offloaded parts of running the active program are determined with changing rates along 3G network locations. The optimization is made for the offloading assigning the related task, and the running time with the resource hardware needed. A heuristic broach was applied in [35] using two stages. The first stage uses semi-definite relaxation and randomization techniques to specify the offloading parts between users. The second stages optimize and manage the resource demands of users. Meanwhile, a scenario of optimizing is offloaded from a single mobile device to multiple edge devices. Thus, the total tasks' running latency and the mobile device's energy consumption were minimized by jointly optimizing the task allocation decision and CPU frequency (network speed scaling) of the mobile device. As a result, the improvement in performance is achieved.

3.2.1. Offloading device-to-device

Device-to-device (D2D) communication is proposed in [36], [37] links in MEC and corporate computing sharing between multi-users so the mobility problems will be overcome. Also, the computation offload is enabled for users to surrounding users exploiting powerful computation abilities. Due to the short distance between the two devices, the transmission energy consumption of data will be minimized, but there are new challenges are arising. These are [36]:

- Utilization way of D2D and cellular communications.
- The desertion of choosing surrounded users to ensure optimum offloading taking into account (users' device information, dynamic links and computing power of heterogeneous users).
- Suffering from numerous D2D channels in terms of interference for a reliable link.

3.2.2. Device-to-MEC

The energy consumption and running time can be decreased by offloading computation from a UE to a MEC server [38], the burden of the computation of mobile devices is lessened, the performance of applications is increased, the energy consumption is minimized, and the lifetime of the battery of mobile UEs is also increased. The power and time allocations were optimized in [18] to minimize offload power consumption in non-orthogonal multiple access (NOMA) multi-user MEC networks. NOMA is applied to multi-user MEC networks, where multiple users can simultaneously offload their tasks to the MEC server on the same frequency band. Any user can split computing tasks into offload computing and local computing parts by applying partial offloading. Thus, the energy consumption, offloading power, and task completion time limitations were considered as non-convex problems. There is a method [26] in which the task type is determined and then making partial offloading by dividing the program into local (mobile) and remote execution (cloud), to overcome the resource needed and make energy serving, the task-input data is randomly partitioned. Meanwhile, a system in [39] has been designed so that a server at the MEC and several applications runs by a mobile device with considering three processes: computation offloading, task scheduling, and energy consumption. Another offloading decision strategy was proposed in [40] for the multi-UE case. The strategy reduces the energy consumption at the UE while allowing the maximum execution delay. In each time slot, a decision on the computation offloading was often taken for the period where the most UE were divided into two groups: in the first group, the computation is offloaded to the MEC while in the second group the computation is locally achieved since the computation resources are inaccessible at the MEC. A systematic optimization procedure was proposed in [41] using a mobile edge environment. Partition can be made with running tasks and then, swapped out is processed (offloading) in parallel for distributing edge nodes. Therefore, by simultaneously optimizing edge node selection, quality of results (QoR) level, and task assignment to all edges, response time and power consumption can be reduced. Mainly there are two energy consumption types for the computation offloading of mobile devices: task offloading and downloading computational energy consumption. Sheng *et al*. [42] have considered the optimization of computation offloading strategy using different servers in MEC. For UE, they use a queuing model and multiple servers. The average response time (of offloaded and non-offloaded tasks for UE) and rate value of power consumption with the cost-performance ratio were minimized. Furthermore, an efficient numerical method was developed using a variety of efficient numerical algorithms [42]. The energy consumption by the mobile device (only that needed when the task is uploaded) or (task is offloaded) [42] is:

$$E_{UP} = T_{up} \times P_{up} = \frac{D_n}{W \log_2(1+\frac{P_{up} \times G_{OS}}{N})} \times P_{up} \qquad (6)$$

Where T_{up} = the transmission time required to upload a task; P_{up} = upload power; D_n = the amount of data needed to upload tasks; N = the Gauss noise power in the channel; W = the channel bandwidth, and G_{os} is the channel gain. The computations that are used in cloudlets with this approach can decrease the overload of wearable devices by about 30% of energy [42]. The closer device will be the more tasks it can be offloading (due to the lesser bandwidth loss of wireless transmissions). This also shows that EC is more suitable for computation offloading than centralized cloud computing. The performing offloading computations show that the offloading computation can save at least 50% of the power for mobile devices, which means that mobile devices have longer endurance [42].

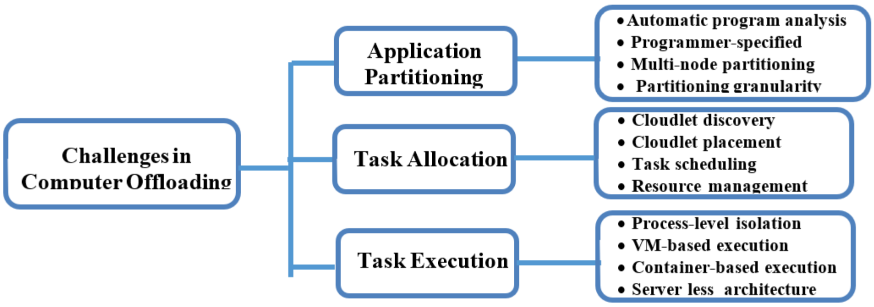

Figure 3. Challenges for computation offloading due to edge computing

3.3. Computation offloading of MEC

Different approaches were developed in relation to the efficient computation offloading mechanism of MEC. For example, in MEC, a framework was introduced in [43] where the Markov decision is used based on a sequential offloading decision to address the problem of dynamic service migration. Also, Zhang *et al*. [44], have tried to trade-off between the tasks offloading computation of the cloud and keeping hold of them in the mobile edge cloud. In [45] a small cell cluster formation and a scheme was used for load balancing on the edge of the clouds in a 5G network where dense deployments were implemented. Regarding the feature of the MEC that concerns the uniqueness of the wireless task offloading, network virtualization in the context of MEC networks was presented in [46]. Two privacy issues are proposed named location privacy and usage pattern privacy. There is a challenge to tackle the proposed privacy issues and at the same time preserve the best performance of the delay and energy consumption. In [47], this challenge is handled by suggesting a scheme for the MEC system to minimize the delay and the cost of energy consumption and even maintain the privacy of the user above a specific level. The problem is considered a Markov decision process and an offloading scheme based on privacy-aware tasks.

4. CONCLUSION

In this paper, different EE mechanisms in EC for MEC with their enhancement strategies for performance consideration and technologies are studied and highlighted. It also discusses different energy management techniques with their classification of each type. Moreover, some offloading approaches according to different categories are explained and studied in detail. As mentioned earlier the EE of a data center in the cloud is greater than that with the edge DCs. Also, in IoT applications, sending computational tasks to the edge of the server have EE better than in the cloud. It can be concluded that, in EC systems, to achieve minimum power consumption, some approaches cannot be simply applied because of the variability and intermittency of renewable energy which adds other challenges to energy management. Also, as explained earlier renewable energy is an affirmative feature that can be useful for MEC. It can be used by mobile devices since EH can prolong their battery life.

It is recommended that designing a dynamic system for energy management in real-time is important so as to respond to the availability of local renewable energy, and to the kind of IoT application. For edge devices, it is recommended to use alternative ways for supply and not only depend on the single source but also enabling using more than one energy source or a hybrid method to provide new energy sources. It is also recommended to use an approach based on privacy-aware tasks for computation offloading tasks to minimize the delay and the cost of energy consumption and even maintain the privacy of the user above a specific level for the MEC system. Furthermore, we advocate making evaluations on the proposed management modeling to be reliable for more than one system type with minimum power consumption.

REFERENCES

[1] J. A. Khan, H. K. Qureshi, and A. Iqbal, "Energy management in wireless sensor networks: A survey," *Computers & Electrical Engineering*, vol. 41, pp. 159-176, 2015, doi: 10.1016/j.compeleceng.2014.06.009.

[2] F. Liu, G. Tang, Y. Li, Z. Cai, X. Zhang, and T. Zhou, "A Survey on Edge Computing Systems and Tools," in *Proceedings of the IEEE*, 2019, vol. 107, no. 8, pp. 1537-1562, doi: 10.1109/JPROC.2019.2920341.

[3] Y. Hmimz, M. El Ghmary, T. Chanyour, and M. O. C. Malki, "Computation Offloading to a Mobile Edge Computing Server with Delay and Energy Constraints," *2019 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS)*, 2019, pp. 1-6, doi: 10.1109/WITS.2019.8723733.

[4] ETSI, "Mobile Edge Computing Market Acceleration MEC Metrics Best Practice and Guidelines," *ETSI GS MEC-IEG 006 V1.1.1*, Mobile Edge Computing (MEC) ETSI Industry Specification, (2017-01). [Online]. Available: https://www.etsi.org/deliver/etsi_gs/mec-ieg/001_099/006/01.01.01_60/gs_mec-ieg006v010101p.pdf

[5] X. Li, J. Wan, H. -N. Dai, M. Imran, M. Xia, and A. Celesti, "A Hybrid Computing Solution and Resource Scheduling Strategy for Edge Computing in Smart Manufacturing," in *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 4225-4234, 2019, doi: 10.1109/TII.2019.2899679.

[6] Y. Zhao, W. Wang, Y. Li, C. C. Meixner, M. Tornatore, and J. Zhang, "Edge Computing and Networking: A Survey on Infrastructures and Applications," in *IEEE Access*, vol. 7, pp. 101213-101230, 2019, doi: 10.1109/ACCESS.2019.2927538.

[7] A. Feldmann, A. Gladisch, M. Kind, C. Lange, G. Smaragdakis, and F. -J. Westphal, "Energy trade-offs among content delivery architectures," *2010 9th Conference of Telecommunication, Media and Internet*, 2010, pp. 1-6, doi: 10.1109/CTTE.2010.5557700.

[8] F. Jalali, K. Hinton, R. Ayre, T. Alpcan, and R. S. Tucker, "Fog Computing May Help to Save Energy in Cloud Computing," in *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 5, pp. 1728-1739, 2016, doi: 10.1109/JSAC.2016.2545559.

[9] F. Jalali, A. Vishwanath, J. de Hoog, and F. Suits, "Interconnecting Fog computing and microgrids for greening IoT," *2016 IEEE Innovative Smart Grid Technologies-Asia (ISGT-Asia)*, 2016, pp. 693-698, doi: 10.1109/ISGT-Asia.2016.7796469.

[10] G. Lewis, S. Echeverría, S. Simanta, B. Bradshaw, and J. Root, "Tactical Cloudlets: Moving Cloud Computing to the Edge," *2014 IEEE Military Communications Conference*, 2014, pp. 1440-1446, doi: 10.1109/MILCOM.2014.238.

[11] B. Liu, H. Xu, and X. Zhou, "Resource Allocation in Wireless-Powered Mobile Edge Computing Systems for Internet of Things Applications," *Electronics*, vol. 8, no. 2, 2019, doi: 10.3390/electronics8020206.

[12] R. -I. Ciobanu, C. Negru, F. Pop, C. Dobre, C. X. Mavromoustakis, and G. Mastorakis, "Drop computing: Ad-hoc dynamic collaborative computing," *Future Generation Computer Systems*, vol. 92, pp. 889-899, 2019, doi: 10.1016/j.future.2017.11.044.

[13] C. Li, Y. Hu, L. Liu, J. Gu, M. Song, X. Liang, J. Yuan, and T. Li, "Towards sustainable in-situ server systems in the big data era," *ISCA '15: Proceedings of the 42nd Annual International Symposium on Computer Architecture*, 2015, pp. 14-26, doi: 10.1145/2749469.2750381.

[14] M. V. Barbera, S. Kosta, A. Mei, and J. Stefa, "To offload or not to offload? The bandwidth and energy costs of mobile cloud computing," *2013 Proceedings IEEE INFOCOM*, 2013, pp. 1285-1293, doi: 10.1109/INFCOM.2013.6566921.

[15] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A Survey on Mobile Edge Computing: The Communication Perspective," in *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322-2358, 2017, doi: 10.1109/COMST.2017.2745201.

[16] M. Lin, A. Wierman, L. L. H. Andrew, and E. Thereska, "Dynamic right-sizing for power-proportional data centers," 2011 *Proceedings IEEE INFOCOM*, 2011, pp. 1098-1106, doi: 10.1109/INFCOM.2011.5934885.

[17] H. Ju and R. Zhang, "Throughput Maximization in Wireless Powered Communication Networks," in *IEEE Transactions on Wireless Communications*, vol. 13, no. 1, pp. 418-428, 2014, doi: 10.1109/TWC.2013.112513.130760.

[18] W. Li *et al*., "On Enabling Sustainable Edge Computing with Renewable Energy Resources," in *IEEE Communications Magazine*, vol. 56, no. 5, pp. 94-101, 2018, doi: 10.1109/MCOM.2018.1700888.

[19] E. Cuervo *et al*., "MAUI: Making Smartphones Last Longer with Code Offload," *MobiSys '10: Proceedings of the 8th international conference on Mobile systems, applications, and services*, 2010, pp. 49-62, doi: 10.1145/1814433.1814441.

[20] K. Li, "Computation Offloading Strategy Optimization with Multiple Heterogeneous Servers in Mobile Edge Computing," in *IEEE Transactions on Sustainable Computing*, doi: 10.1109/TSUSC.2019.2904680.

[21] K. Yang, Q. Yu, S. Leng, B. Fan, and F. Wu, "Data and Energy Integrated Communication Networks for Wireless Big Data," in *IEEE Access*, vol. 4, pp. 713-723, 2016, doi: 10.1109/ACCESS.2016.2526622.

[22] K. M. S. Huq, S. Mumtaz, J. Bachmatiuk, J. Rodriguez, X. Wang, and R. L. Aguiar, "Green HetNet CoMP: Energy Efficiency Analysis and Optimization," in *IEEE Transactions on Vehicular Technology*, vol. 64, no. 10, pp. 4670-4683, 2015, doi: 10.1109/TVT.2014.2371331.

[23] A. Fahim, A. Mtibaa, and K. A. Harras, "Making the case for computational offloading in mobile device clouds," *MobiCom '13: Proceedings of the 19th annual international conference on Mobile computing & networking*, 2013, pp. 203–205, doi: 10.1145/2500423.2504576.

[24] P. Mach and Z. Becvar, "Mobile Edge Computing: A Survey on Architecture and Computation Offloading," in *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1628-1656, 2017, doi: 10.1109/COMST.2017.2682318.

[25] C. Jiang, X. Cheng, H. Gao, X. Zhou, and J. Wan, "Toward Computation Offloading in Edge Computing: A Survey," in *IEEE Access*, vol. 7, pp. 131543-131558, 2019, doi: 10.1109/ACCESS.2019.2938660.

[26] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. O. Wu, "Energy-Optimal Mobile Cloud Computing under Stochastic Wireless Channel," in *IEEE Transactions on Wireless Communications*, vol. 12, no. 9, pp. 4569-4581, 2013, doi: 10.1109/TWC.2013.072513.121842.

[27] M. Z. Nayyer, I. Raza, and S. A. Hussain, "A Survey of Cloudlet-Based Mobile Augmentation Approaches for Resource Optimization," *ACM Computing Surveys*, vol. 51, no. 5, pp. 1–28, 2018, doi: 10.1145/3241738.

[28] S. Anitha and T. Padma, "Adptive Proximate Computing Framework for Mobile Resource Augmentation," *Mobile Networks and Applications*, vol. 25, pp. 553-564, 2020, doi: 10.1007/s11036-019-01278-8.

[29] Y. Wang, M. Sheng, X. Wang, L. Wang, and J. Li, "Mobile-Edge Computing: Partial Computation Offloading Using Dynamic Voltage Scaling," in *IEEE Transactions on Communications*, vol. 64, no. 10, pp. 4268-4282, 2016, doi: 10.1109/TCOMM.2016.2599530.

[30] L. Lin, X. Liao, H. Jin, and P. Li, "Computation Offloading Toward Edge Computing," in *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1584-1607, 2019, doi: 10.1109/JPROC.2019.2922285.

[31] L. Liu, Z. Chang, X. Guo, S. Mao, and T. Ristaniemi, "Multiobjective Optimization for Computation Offloading in Fog Computing," in *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 283-294, 2018, doi: 10.1109/JIOT.2017.2780236.

[32] C. You, Y. Zeng, R. Zhang, and K. Huang, "Asynchronous Mobile-Edge Computation Offloading: Energy-Efficient Resource Management," in *IEEE Transactions on Wireless Communications*, vol. 17, no. 11, pp. 7590-7605, 2018, doi: 10.1109/TWC.2018.2868710.
[33] X. Cao, F. Wang, J. Xu, R. Zhang, and S. Cui, "Joint computation and communication cooperation for mobile edge computing," *2018 16th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, 2018, pp. 1-6, doi: 10.23919/WIOPT.2018.8362865.
[34] D. Huang, P. Wang, and D. Niyato, "A Dynamic Offloading Algorithm for Mobile Computing," in *IEEE Transactions on Wireless Communications*, vol. 11, no. 6, pp. 1991-1995, 2012, doi: 10.1109/TWC.2012.041912.110912.
[35] J. Kwak, Y. Kim, J. Lee, and S. Chong, "DREAM: Dynamic Resource and Task Allocation for Energy Minimization in Mobile Cloud Systems," in *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 12, pp. 2510-2523, 2015, doi: 10.1109/JSAC.2015.2478718.
[36] M. -H. Chen, M. Dong and B. Liang, "Joint offloading decision and resource allocation for mobile cloud with computing access point," *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 3516-3520, doi: 10.1109/ICASSP.2016.7472331.
[37] Y. Li, L. Sun, and W. Wang, "Exploring device-to-device communication for mobile cloud computing," 2014 *IEEE International Conference on Communications (ICC)*, 2014, pp. 2239-2244, doi: 10.1109/ICC.2014.6883656.
[38] T. Q. Dinh, J. Tang, Q. D. La, and T. Q. S. Quek, "Offloading in Mobile Edge Computing: Task Allocation and Computational Frequency Scaling," in *IEEE Transactions on Communications*, vol. 65, no. 8, pp. 3571-3584, 2017, doi: 10.1109/TCOMM.2017.2699660.
[39] Y. Tao, C. You, P. Zhang, and K. Huang, "Stochastic Control of Computation Offloading to a Helper With a Dynamically Loaded CPU," in *IEEE Transactions on Wireless Communications*, vol. 18, no. 2, pp. 1247-1262, 2019, doi: 10.1109/TWC.2018.2890653.
[40] W. Labidi, M. Sarkiss, and M. Kamoun, "Joint multi-user resource scheduling and computation offloading in small cell networks," *2015 IEEE 11th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2015, pp. 794-801, doi: 10.1109/WiMOB.2015.7348043.
[41] Y. Xiao and M. Krunz, "QoE and power efficiency tradeoff for fog computing networks with fog node cooperation," *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, 2017, pp. 1-9, doi: 10.1109/INFOCOM.2017.8057196.
[42] J. Sheng, J. Hu, X. Teng, B. Wang, and X. Pan, "Computation Offloading Strategy in Mobile Edge Computing," *Information*, vol. 10, no. 6, 2019, doi: 10.3390/info10060191 .
[43] U. Drolia *et al*., "The Case for Mobile Edge-Clouds," *2013 IEEE 10th International Conference on Ubiquitous Intelligence and Computing and 2013 IEEE 10th International Conference on Autonomic and Trusted Computing*, 2013, pp. 209-215, doi: 10.1109/UIC-ATC.2013.94.
[44] K. Zhang *et al*., "Energy-Efficient Offloading for Mobile Edge Computing in 5G Heterogeneous Networks," in *IEEE Access*, vol. 4, pp. 5896-5907, 2016, doi: 10.1109/ACCESS.2016.2597169.
[45] S. Wang, R. Urgaonkar, M. Zafer, T. He, K. Chan, and K. K. Leung, "Dynamic service migration in mobile edge-clouds," *2015 IFIP Networking Conference (IFIP Networking)*, 2015, pp. 1-9, doi: 10.1109/IFIPNetworking.2015.7145316.
[46] J. Oueis, E. C. Strinati, and S. Barbarossa, "The Fog Balancing: Load Distribution for Small Cell Cloud Computing," *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, 2015, pp. 1-6, doi: 10.1109/VTCSpring.2015.7146129.
[47] T. Taleb and A. Ksentini, "An analytical model for Follow Me Cloud," *2013 IEEE Global Communications Conference (GLOBECOM)*, 2013, pp. 1291-1296, doi: 10.1109/GLOCOM.2013.6831252.

BIOGRAPHIES OF AUTHORS

Tawfeeq E. Abdoulabbas received the B.Sc in Electrical Engineering from University of Mustansiriyah /Baghdad, Iraq in 1992. In 1996, he obtained his M.Sc. from the University of Technology/Baghdad, Iraq in Electrical and Engineering science. He joined the academic staff of Mustansiriyah University/College of Engineering in 2012. His research interests are on power management systems, smart homes, and smart grids. He can be contacted: tawfikenad@uomustansiriyah.edu.iq.

Sawsan M. Mahmoud received her B.Sc. in Computer Science from University of Technology/Baghdad, Iraq in 1994. She obtained her M.Sc. from University of Baghdad n 1998. Her Ph.D. in Computational Intelligence is obtained from Nottingham Trent University, Nottingham, UK in 2012. Sawsan joined the academic staff of Mustansiriyah University/Engineering College in 1994. Her research interests include computational intelligence, ambient intelligence (smart home and intelligent environment), wireless sensor network, data mining, and health monitoring. She can be contacted: sawsan.mahmoud@uomustansiriyah.edu.iq.