Delay-aware power optimization model for mobile edge computing systems

Article *in* Personal and Ubiquitous Computing · May 2017

CITATIONS
0

READS
14

5 authors, including:

Yaser Jararweh
Jordan University of Science and Technology
172 PUBLICATIONS **986** CITATIONS
SEE PROFILE

Mahmoud Al-Ayyoub
Jordan University of Science and Technology
157 PUBLICATIONS **1,071** CITATIONS
SEE PROFILE

Muneera Al-Quraan
Jordan University of Science and Technology
4 PUBLICATIONS **5** CITATIONS
SEE PROFILE

Elhadj Benkhelifa
Staffordshire University
63 PUBLICATIONS **299** CITATIONS
SEE PROFILE

Pers Ubiquit Comput
DOI 10.1007/s00779-017-1032-2

ORIGINAL ARTICLE

Delay-aware power optimization model for mobile edge computing systems

Yaser Jararweh[1] · Mahmoud Al-Ayyoub[1] · Muneera Al-Quraan[1] · Lo'ai A. Tawalbeh[1,2] · Elhadj Benkhelifa[3]

Received: 11 November 2016 / Accepted: 2 April 2017

Abstract Reducing the total power consumption and network delay are among the most interesting issues facing large-scale Mobile Cloud Computing (MCC) systems and their ability to satisfy the Service Level Agreement (SLA). Such systems utilize cloud computing infrastructure to support offloading some of user's computationally heavy tasks to the cloud's datacenters. However, the delay incurred by such offloading process lead the use of servers (called cloudlets) placed in the physical proximity of the users, creating what is known as Mobile Edge Computing (MEC). The cloudlet-based infrastructure has its challenges such as the limited capabilities of the cloudlet system (in terms of the ability to serve different request types from users in vast geographical regions). To cover the users demand for different types of services and in vast geographical regions, cloudlets cooperate among each other by passing user requests from one cloudlet to another. This cooperation affects both power consumption and delay. In this work, we present a mixed integer linear programming (MILP) optimization model for MEC systems with these two issues in mind. Specifically, we consider two types of cloudlets: local cloudlets and global cloudlets, which have higher capabilities. A user connects to a local cloudlet and sends all of its traffics to it. If the local cloudlet cannot serve the desired request, then the request is moved to another local cloudlet. If no local cloudlet can serve the request, then it is moved to a global cloudlet which can serve all service types. The process of routing requests through the hierarchical network of cloudlets increases power consumption and delay. Our model minimizes power consumption while incurring an acceptable amount of delay. We evaluate it under several realistic scenarios to show that it can indeed be used for power optimization of large-scale MEC systems without violating delay constraints.

Keywords Mobile edge computing · Cooperative cloudlets · Global cloudlet · Power consumption optimization · Delay

1 Introduction

Nowadays, mobile devices such as tablets and smart phones are becoming an essential part of our lives. They have provided us with many capabilities and benefits that facilitate our daily activities. However, they are suffering from many limitations, such as battery lifetime, processing capabilities, and storage capacity. These limitations prevent mobile users from performing certain tasks. On the other hand, the increasing acceptance of Cloud Computing (CC) systems provides an opportunity for resource limited mobile devices to perform compute intensive applications on the cloud giving rise to Mobile CC (MCC) systems. The basic MCC concepts depend on a network-based resource sharing to

✉ Yaser Jararweh
yijararweh@just.edu.jo

Mahmoud Al-Ayyoub
maalshbool@just.edu.jo

Lo'ai A. Tawalbeh
latawalbeh@uqu.edu.sa; tawalbeh@just.edu.jo

[1] Jordan University of Science and Technology, Irbid, Jordan

[2] Umm Al Qura University, Makkah, Saudi Arabia

[3] Mobile Fusion Applied Research Center, Staffordshire University, Stafford, UK

Published online: 23 June 2017

increase the availability of the resources and to reduce the costs of operation and management [1, 2].

MCC has been used as a practical solution to the inherent limitations of mobile computing. These limitations, as mentioned earlier, include battery lifetime, processing power, and storage capacity. By using MCC, the processing and storage demands of the mobile device applications are being satisfied by the cloud system. Thus, the required power and time to perform intensive jobs will be reduced. Now, the connection between mobile devices and the cloud system suffers from the main network problems which are the high latency and the huge transmission power consumption especially when using 4G/LTE connections. These days, the most common applications in mobile devices are the multimedia applications where these applications require high computing resources and more power consumption.

To address these challenges, the systems discussed in the previous paragraphs can offload some of user's computationally heavy tasks to nearby servers typically called cloudlets. Despite being a very appealing solution in terms of relieving the mobile devices from the burden of computationally heavy tasks, using cloudlets does have its own limitations. Specifically, the limited capabilities of the cloudlet system (in terms of the ability to serve different request types from users in vast geographical regions) represent serious challenges to achieve the system's objectives. To cover the users demand for different types of services in vast geographical regions, cloudlets cooperate among each other by passing user requests from one cloudlet to another creating what is called Mobile Edge Computing (MEC) systems.

This cooperation allows the system to avoid the SLA penalties incurred by rejecting the user requests that cannot be served by the local cloudlets. However, this comes with a price. First, the total power consumption per request might increase as it will include the power consumption to send the request from the user to its local cloudlet in addition to the power needed to pass the request to a remote cloudlet if needed. Another issue is the response time of any request, which might also increase. Such response time includes the transmission delay to send the request from the user to its local cloudlet in addition to the transmission time to pass the request from the local cloudlet to a remote cloudlet that is capable of serving this request. Another potential casue of increase in the response time is the queuing delay inside the cloudlet itself. Now, to decrease the queuing delay and to avoid the starvation to serve the user request, any cloudlet must receive a workload that is less than or equal to its capacity. Thus, the penalty from the SLA will be minimized.

In this paper, we consider two types of cloudlets: local cloudlets and global cloudlets. The global cloudlets are a special kind of local cloudlets but with higher capabilities. The user connects to its closest local cloudlet and sends all of its traffics to it. If the local cloudlet cannot serve the desired request, then the request is moved to another local cloudlet capable of serving the request. If no local cloudlet can serve the request, then it is moved to a global cloudlet in which it can serve all service types. We adopt this view and present our efforts to optimize the power consumption in large-scale MEC systems while taking delay constraints and cloudlet capacities into account. Specifically, for this setting, we present a mixed integer linear programming (MILP) optimization model for MEC system power optimization. Our work can be viewed as an extension of [3] to introduce the concept of global cloudlets into the MILP formulation with the delay constraints.

This paper is organized as follows. Section 2 contains a literature review for the MCC concept. Section 3 introduces our proposed model. The following section shows the experimental results and illustrates the benefits of using the new model. Finally, we conclude the paper in Section 5.

2 Related work

With the rapid use of the mobile devices and the processing limitations they have such as batteries lifetime and memory space, offloading certain tasks over to MCC has become an appealing solution. MEC came as a solution of the problems that may come from the centralized processing resources in MCC by dynamically integrating the surrounding devices. MEC is a new technological model characterized by placing the compute and storage resources at the internet's edge to be closed to the mobile devices and sensors [4]. This model is proposed to reduce the response time of requests by reducing the delay time that comes from the distance between the computing resources and the end users and the queuing time inside the computing resources themselves [5]. The computing resources may refer to cloudlets, edge servers, micro-datacenters, or fog nodes [6, 7].

The work in [8] described how MCC has emerged from the fields of cloud computing and mobile computing. The work also described the MCC scope, developments, and current research area challenges. The authors proposed the MobiCloud system, which was developed at Arizona State University to simplify the study and the analysis of MCC. The authors of [9] gave a survey of MCC's definitions, advantages, architectures, and applications (Mobile Commerce, Mobile Learning, Mobile Healthcare, and Mobile Gaming). They also described MCC issues (low bandwidth, availability, heterogeneity, computing offloading, context-aware mobile cloud services, security and enhancing the efficiency of data access) and listed the existing solutions. At the end, they presented the future works in this field.

The impact of using cloudlets with respect to cloud mobile computing in interactive applications (file editing,

video streaming and collaborative chatting) is analyzed in [10]. The authors discussed the advantages of using cloudlets over using typical CC systems in terms of system throughput and data transfer delay. The paper results showed that the use of cloudlet-based model has outperformed the typical CC model in most cases. In [11], the authors introduced a new architecture called MOCHA for face recognition applications. The purpose of this architecture is to reduce the response time during the face recognition process. MOCHA integrates mobile device, cloudlet and cloud servers. A large-scale cloudlet-based MCC system deployment was introduced in [12] aiming at reducing the power consumption and the network delay of multimedia applications. In [13], the authors discussed the benefits MEC can offer to the internet of things (IoT) category and the orchestration of the applications in MEC to ensure the efficiency of the operations of the network and the delivery of the service.

Admission control and resource allocation problems for the running mobile applications in the cloudlet are discussed in [14]. To solve these problems, the authors formulate them as a semi-Markov decision process (SMDP). The proposed model in [14] provides a QoS for different classes of mobile users.

In [15], the technical obstacles of using cloudlet in mobile computing have been discussed. A new architecture has been proposed to deal with these obstacles. This new architecture manages the sessions opened by mobile users inside the cloudlet. The management is based on VM instantiation for each mobile user.

The authors of [16] discussed the key performance metrics of using VM to manage jobs execution inside the cloudlets. These metrics include overhead of VM life cycle when deployed for execution at cloudlet, cloudlet allocation to VM, and scheduling of VMs. The authors used the CloudSim as a platform environment and concluded that it is so important to efficiently deploy and manage VMs in CC to reduce the amount of execution time because of the previously mentioned performance metrics.

The authors of [17] present a prototype implementation of cloudlet architecture and show the advantages of this architecture for the real-time applications. The proposed architecture in [17] uses cloudlet to manage the running application on the component model, where the cloudlet can be chosen dynamically from any resource rich device inside the LAN and not as the traditional concept where the cloudlet is fixed near to the wireless access points.

In [18], authors analyzed the critical factors that affect the power consumption of mobile clients when using CC. They also provided an example on how to save mobile client power. To define the balance between using local mobile computing and remote CC, they presented their own measurements of the main characteristics of modern mobile devices. As for [19], the authors reviewed existing work in energy consumption of MCC and propose a system whereby user applications may be profiled for their resource consumption locally and then if augmentation is required, they may negotiate with an external cloud for optimum energy consumption.

The authors in [20] present a survey of the intended usages of MCC. They discuss three existing architectures of MCC, which are the traditional centralized cloud, cloudlet and peer-based ad hoc mobile cloud, and provide their visions for the future MCC architecture. Also, they discuss the main contributions of using clouds in mobile computing as (i) computation offloading and (ii) capability extension in terms of computation, networking, storage, etc. The works in [21, 22] discuss mainly on the feasibility of using MCC for multimedia applications and data collection in large scale networks whereas [23, 24] focus on using MCC for healthcare systems.

In [25], the authors proposed CloudAware, a context adaptive mobile middleware that is responsible for automatically the changing of the context configuration by linking the distribution features of mobile middleware with context-aware self-adaptation techniques. The authors showed their evaluation by using real usage data supporting from Nokia Mobile Data Challenge (MDC) dataset.

The authors of [26] discussed the benefits of the offloading to the cloudlets as a solution to minimize the delay time and to increase the quality of the service with comparing to the offloading into a remote cloud. They proposed a cloudlet selection strategy based on power and latency for multi cloudlet environment.

The authors in [27] discussed the waste of energy and the delay problems of using MCC in the dynamic network environment. They proposed a dynamic energy-aware cloudlet-based MCC model to solve the mentioned problems by using the benefits of the dynamic cloudlets.

The authors of [28] also discussed the offloading to the remote processing elements and how such offloading leads to the increasing in the power consumptions than performing the task on the mobile devices. The authors mentioned that the waste in power may come from the wireless communications to a remote servers or heterogeneous core processors. The authors proposed energy-aware heterogeneous resource management model which depends on the optimal task assignment to heterogeneous cores and mobile clouds which implies to minimize the energy cost of mobile heterogeneous embedded system.

The authors in [29] consider the effect of massive applications (also called hungry applications) on the network bandwidth. They showed how it is beneficial to introduce additional datacenters as an edge layer in mobile edge network infrastructure. They showed that the amount of cost saving of such hungry applications can go up to 67%.

They also recommended to start building DC edge layer to serve the bandwidth hungry applications and to reduce the operational cost.

In [30–32], the authors used the power of the fog computing to solve the problem of selective forwarding method in mobile wireless sensor networks to detect different kinds of intrusions. The authors built a model that provided a global monitoring capability based on the infrastructure of the fog computing to trace the movement of sensors and catch malicious ones. They also performed experiments to prove the validity of their model.

In [33], the authors examined the Fog Gateway (FG) as an intermediate component in the Long Term Evolution (LTE) to reduce the communication time in mobile networks among users in the same area. FG can prevent all traffic of specific services from reaching the core network by analyzing the inner destination IP address of the traffic using tunneling protocol (GTP) and then determining whether to route the packet to the fog network or to forward it to the destination Service Gateway (SGW). The authors optimized this proposed method in realistic environment and the result showed its effectiveness in reducing the delay based on the deployment area.

3 System model

In this section, we discuss the proposed model for the problem at hand. We explain the constraints we place and justify our choices. We note that this model is an extension of the model proposed in [3]. So, we start by introducing a slightly refined version of [3]'s model in Section 3.1. We then present two major extensions for this model as follows. In Section 3.2, we discuss an extension of the model to support global cloudlets, whereas, in Section 3.3, we extend the model to allow more generic cloudlets in addition to adding constraints related to the delay and the capacity of the cloudlets.

3.1 Basic model

Given a set of users, U, each requesting different types of services and a set of cloudlets, C, each capable of serving different types of requests, the goal is to assign each user request to a specific cloudlet capable of serving it while minimizing the overall power consumption. The users are assumed to be operating battery-powered devices and the cloudlets are assumed to be stationary, plugged into continuous power sources and connected to each other through backbone network.

This problem is formulated as a mixed integer linear programming (MILP) problem with the following assumptions. The time is assumed to be divided into a discrete set of time slots, T. Another assumption is that there are $|R|$ service types and, for each $r \in R$, the binary variable $a_i^{r,t}$ is used to represent whether cloudlet i can serve r-type requests at time slot t or not. Similarly, the binary variable $b_i^{r,t}$ represents whether user i requests service of type r at time slot t or not. Finally, the binary variable $x_{i,j}^{r,t}$ is used to represent whether user i's r-type requests are assigned to cloudlet j at time slot t or not.

The goal of this model is to assign each request type from each user to exactly one cloudlet at a time, which means that the following condition must be satisfied.

$$\sum_{j \in C} x_{i,j}^{r,t} = b_i^{r,t}, \quad \forall\, i \in U, r \in R, t \in T \tag{1}$$

To simplify things, in each time slot $t \in T$, the user is assumed to be able to generate at most one service request of each type, which means that it can make up to $|R|$ requests.

The following constraint is to ensure that each user $i \in U$ is connected to one cloudlet $j \in C$ and sends all of its requests to it.

$$x_{i,j}^{r_1,t} = x_{i,j}^{r_2,t}, \quad \forall\, r_1, r_2 \in R$$

To put the above constraint in linear form, the following constraint is used.

$$x_{i,j}^{r_1,t} - x_{i,j}^{r_2,t} = 0, \quad \forall\, r_1, r_2 \in R \tag{2}$$

The binary variable $v_{i,j}^t$ is used to represent whether user i is in the coverage region of cloudlet j at time slot t or not. This depends mainly on the distance between user i and cloudlet j at time slot t, which is denoted by $d_{i,j}^t$. Also, to ensure that the user $i \in U$ is in the coverage region of at least one cloudlet $j \in C$ at time slot $t \in T$, we add the following constraint.

$$\sum_{j \in C} v_{i,j}^t \geq 1, \quad \forall\, i \in U, t \in T \tag{3}$$

Users outside the coverage regions of all cloudlets are considered as disconnected from the network and, thus, can be safely disregarded. For user i to be connected to cloudlet j at time slot t, it must be in j's coverage region, which is ensured by the following condition.

$$x_{i,j}^{r,t} \leq v_{i,j}^t, \quad \forall\, i \in U, j \in C, r \in R, t \in T \tag{4}$$

Cloudlets are assumed to be connected to each other. Moreover, they use this connectivity to delegate the execution of service requests to each other. This is important because a cloudlet is not necessarily assumed to be capable of serving all request types. However, Constraint 2 might force requests of certain type to be sent to a cloudlet that is not capable of serving them. Such requests are forwarded

to other cloudlets capable of serving them. The forwarding process is governed by the use of "service catalogs." To be specific, a cloudlet i, that is not capable of serving requests of type r at time slot t (i.e., $a_i^{r,t} = 0$), should have an entry in its service catalog listing another cloudlet through which i can serve requests of type r. This is modeled using the binary variable $c_{i,j}^{r,t}$, which represents whether cloudlet i forwards/re-directs/delegates all of the type-r requests it receives to cloudlet j at time slot t or not. Since each cloudlet's service catalog should include a single entry for each service type r not offered by the cloudlet, the following condition must be satisfied.

$$\sum_{j \in C} c_{i,j}^{r,t} = 1 - a_i^{r,t}, \quad \forall\, i \in C, r \in R, t \in T \tag{5}$$

We now turn our attention to modeling the cost function which represents the total power consumption of the system. This model does not discard any request it receives and it makes sure that each request is served by a cloudlet customized to serve it. This means that the amount of computation power required for each service request is the same regardless of when and where it is served. Hence, the entire computation power can be disregarded in this model and the attention can be focused only on minimizing the communication power.

Let $w_{i,j}^{r,t}$ be the communication power required to deliver user i's r-type request, that is to be served in time slot t, to cloudlet j. $w_{i,j}^{r,t}$ depends on the size of the request as well as the distance between i and j. Let $s_i^{r,t}$ be the size of user i's r-type request to be served in time slot t. According to the path loss model, the signal strength drops significantly with the increase in the distance between the transmitter and the receiver [34]. Thus, we have the following equation.

$$w_{i,j}^{r,t} = s_i^{r,t} \times (d_{i,j}^t)^{\alpha}, \quad \forall\, i \in U, j \in C, r \in R, t \in T \tag{6}$$

where α is a constant to account for the decay in signal strength. Similarly, we use the notation, $l_{j,k}^t$, to denote the communication cost between cloudlets j and k at time slot t and the notation, $\widetilde{w}_{i,j,k}^{r,t}$, to denote the total communication power required to move user i's r-type request, that is to be served in time slot t, from cloudlet j to cloudlet k.

$$\widetilde{w}_{i,j,k}^{r,t} = s_i^{r,t} \times l_{j,k}^t, \quad \forall\, i \in U, j,k \in C, r \in R, t \in T \tag{7}$$

Thus, the total power required to serve user i's r-type request at time slot t by first sending it to cloudlet j is given by the following equation.

$$W_{i,j}^{r,t} = w_{i,j}^{r,t} + \sum_{k \in C} (c_{j,k}^{r,t} \times \widetilde{w}_{i,j,k}^{r,t}), \quad \forall\, i \in U, j \in C, r \in R, t \in T \tag{8}$$

The objective is formulated as follows.

$$\text{Minimize } \sum_{i \in U, j \in C, r \in R, t \in T} (x_{i,j}^{r,t} \times W_{i,j}^{r,t})$$
$$\text{Subject to} \quad \text{Constraints} \quad 1 - 8$$

3.2 Introducing global cloudlets

In this extension, we consider the more recent trend of adopting mobile edge computing (MEC) and fog computing (FC). The basic idea is to use a mixture of "local cloudlets" and "global cloudlets." The local cloudlets are distributed so that they can be near the end users. Also, they have moderate communication and computation capabilities. As for the global cloudlets, they are closer to the core of the network, more powerful and more generic (in the sense that they are capable of serving all request types). Hence, in addition to the set of local cloudlets, C, we consider a set of global cloudlets, G. Both C and G cloudlets are deployed at fixed locations and are connected together through backbone network. Similar to [3], we formulate this problem as a MILP problem with the goal of assigning each user request to a specific cloudlet capable to serving it while minimizing the overall power consumption.

Constraint (3) forced each user to be in the coverage region of at least one cloudlet so that its service requests can be received and processed by the network of cloudlets. With the introduction of global cloudlets that are assumed to have much wider coverage regions than those of the local cloudlets, some users can be connected directly to a global cloudlets. Thus, Constraint (3) is modified as follows.

$$\sum_{j \in (C \cup G)} v_{i,j}^t \geq 1, \quad \forall\, i \in U, t \in T \tag{9}$$

Moreover, Constraint (1) is modified as follow.

$$\sum_{j \in (C \cup G)} x_{i,j}^{r,t} = b_i^{r,t}, \quad \forall\, i \in U, r \in R, t \in T \tag{10}$$

The model of the previous subsection allows local cloudlets to delegate the requests of types it cannot handle to another local cloudlet capable of serving these request types. This is achieved through the use of service catalogs. Introducing global cloudlets that are capable of serving any request type gives each local cloudlet the additional option of delegating the request of types it cannot handle to a global cloudlet. Thus, in the service catalog of a local cloudlet i, if no other local cloudlet $c \in C$ can serve r-type request in a certain time slot, then cloudlet i must re-direct all r-type requests it receives to a global cloudlet $g \in G$. Thus, Constraint (5) is updated as follows.

$$\sum_{j \in (C \cup G)} c_{i,j}^{r,t} = 1 - a_i^{r,t}, \quad \forall\, i \in C, r \in R, t \in T \tag{11}$$

Accordingly, two more constraints (Constraint (7) and Constraint (8)) are updated as follows.

$$\tilde{w}^{r,t}_{i,j,k} = s^{r,t}_i \times l^t_{j,k},\ \forall\, i \in U, j \in C, k \in (C \cup G), r \in R, t \in T \tag{12}$$

$$W^{r,t}_{i,j} = w^{r,t}_{i,j} + \sum_{k \in (C \cup G)} (c^{r,t}_{j,k} \times \tilde{w}^{r,t}_{i,j,k}), \quad \forall\, i \in U, j \in C, r \in R, t \in T \tag{13}$$

Finally, the objective is updated as follows.

Minimize $\sum_{i \in U, j \in C, r \in R, t \in T} (x^{r,t}_{i,j} \times W^{r,t}_{i,j})$
Subject to Constraints $2, 5, 6, 9 - 13$

3.3 Delay and capacity

In this subsection, we extend the model to allow more generic cloudlets in addition to adding constraints related to the delay and the capacity of the cloudlets. Let us start with the delay and the different types of delay incurred while processing the users requests.

The term delay in a network refers to the time it takes a bit to travel from one place to another. There are multiple types of delay; however, we only consider in our model two types, which are transmission delay and queuing delay. The transmission delay is the amount of time that is required to transmit the request from the user to the cloudlet. Since we have cooperative cloudlets, then we have another type of transmission delay, which is incurred when transmitting a request from one cloudlet to another. As for the queuing delay, it is the time from the arrival to the cloudlet to the time of the completing of the request. The queuing delay is the most important type of delay and hardest to compute [35, 36].[1]

To consider the delay types in our model, we define the variable $e^{r,t}_{i,j}$ to represent the delay incurred to transmit user i's r-type request to cloudlet j at time slot t and the variable $\tilde{e}^{r,t}_{i,j,k}$ to denote the delay incurred to transmit user i's r-type request from cloudlet j to cloudlet k at time slot t. Also, we denote the queuing delay in the cloudlet j to serve user i's r-type request in time slot t as $q^{r,t}_{i,j}$. The values of $e^{r,t}_{i,j}$ and $\tilde{e}^{r,t}_{i,j,k}$ are computed as follows [37].

$$e^{r,t}_{i,j} = \frac{s^{r,t}_i}{F^t_{i,j}} + \frac{d^t_{i,j}}{PS^t_{i,j}} \tag{14}$$

$$\tilde{e}^{r,t}_{i,j,k} = \frac{s^{r,t}_i}{\widetilde{F}^t_{j,k}} + \frac{d^t_{j,k}}{\widetilde{PS}^t_{j,k}} \tag{15}$$

where $F^t_{i,j}$ is the link bandwidth between user i and cloudlet j at time slot t, $\widetilde{F}^t_{j,k}$ is the link bandwidth between cloudlet j and cloudlet k at time slot t, $PS^t_{i,j}$ is the propagation speed in medium between user i and cloudlet j at time slot t, and $\widetilde{PS}^t_{j,k}$ is the propagation speed in medium between cloudlet j and cloudlet k at time slot t.

Using Little's theorem [38], the average number of packets in one cloudlet j in time slot t, which is denoted by N^t_j, is computed using the following equation.

$$N^t_j = \lambda^t_j \times Y^t_j \tag{16}$$

where λ^t_j is the arrival rate of packets into cloudlet j at time slot t, which is derived from Poisson distribution, and Y^t_j is the average amount of time the packet spends in the cloudlet j at time slot t. Based on this, the queuing delay $q^{r,t}_{i,j}$ is computed as follows.

$$q^{r,t}_{i,j} = Y^t_j - \frac{1}{\mu^{r,t}_{i,j}} \tag{17}$$

where $\mu^{r,t}_{i,j}$ is the number of user i's r-type packets in cloudlet j at time slot t and $1/\mu^{r,t}_{i,j}$ is the time to serve user i's r-type request in cloudlet j at time slot t. Also, the average amount of time the packet spends at cloudlet j at time slot t, Y^t_j can be computed as follows.

$$Y^t_j = \frac{1}{\mu^{r,t}_{i,j} - \lambda^t_j} \tag{18}$$

The total delay to serve user i's r-type request in time slot t is denoted by $Z^{r,t}_i$.

$$Z^{r,t}_i = e^{r,t}_{i,j} + \tilde{e}^{r,t}_{i,j,k} + q^{r,t}_{i,j} a^{r,t}_j + q^{r,t}_{i,k}(1 - a^{r,t}_j) \quad \forall\, i \in U, \{j,k\} \in C, r \in R, t \in T \tag{19}$$

Note that $Z^{r,t}_i$ should not exceed the upper bound limit $E^{r,t}_i$.

$$Z^{r,t}_i \le E^{r,t}_i,\ \ \forall\, i \in U, r \in R, t \in T \tag{20}$$

In addition to considering delay, we also consider the limitations of the cloudlets' capacity in this extension. Specifically, no cloudlet should receive an average set of requests exceeding its processing capacity in each time slot. This is important in order to avoid the starvation at any cloudlet and to minimize the penalties of the SLA.

Each cloudlet j has a specific capacity (denoted by $M^{r,t}_j$) for the amount of r-type requests it can process in each time slot t.

$$\sum_{i \in U} s^{r,t}_i x^{r,t}_{i,j} a^{r,t}_j + \sum_{i \in U} \sum_{k \in C} s^{r,t}_i x^{r,t}_{i,j} (1 - a^{r,t}_k) c^{r,t}_{k,j} \le M^{r,t}_j, \quad \forall\, j \in C, r \in R, t \in T \tag{21}$$

In our model, there are $|R|$ request types, where each request type has different processing requirements. To simplify things, we consider the request type with minimum requirements as the "unit" type, which means that each job of this type requires one unit of processing power. The processing requirements of the remaining types are measured

[1] http://faculty.ycp.edu/~dhovemey/fall2005/cs375/lecture/9-7-2005.html

in terms of this unit type. Specifically, for any request type r, we define $\delta_r \geq 1$ to be the amount of processing units required by each job of type r. This formulation allows the model to support both dedicated cloudlets (capable of processing specific types of requests) as well as generic cloudlets. The following constraint is to ensure that the total work done by a certain cloudlet j at a certain time slot t does not exceed its maximum capacity, which is denoted by M_j^t.

$$\sum_{r \in R} s_i^{r,t} \times \delta_r \leq M_j^t, \quad \forall\, j \in C, i \in U, t \in T \tag{22}$$

4 Experimentation and results

In this section, we examine our model under several realistic scenarios. The goal is to prove that our model works and it can be used to optimize power consumption in large-scale MEC systems. To achieve this, we conduct three sets of experiments. In the first one, we examine the effect of increasing the number of users while, in the second one, we examine the effect of increasing the number of cloudlets. For these two sets, we consider two types of users: heavily loaded users and lightly loaded users. Finally, in the third one, we consider a finer grained division of users based on their loads.

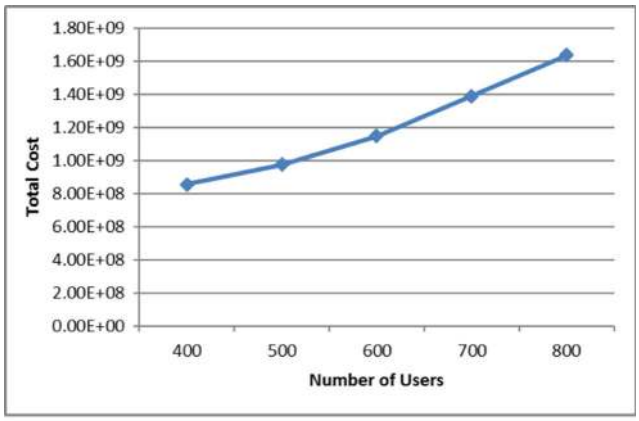
Before discussing the details of each experiment set, we briefly go over the general assumptions/settings used in all of them, which are listed in Table 1. We assume that there are five different service types. Each local cloudlet can serve any number of service types between one and three types. On the other hand, the global cloudlet can serve all request types. Similarly, each user can request any number of the existing types between one and five types. From a time slot to the next, the sizes and types of users' requests may differ.

Table 1 Experiment parameters (default values are in *italic* font)

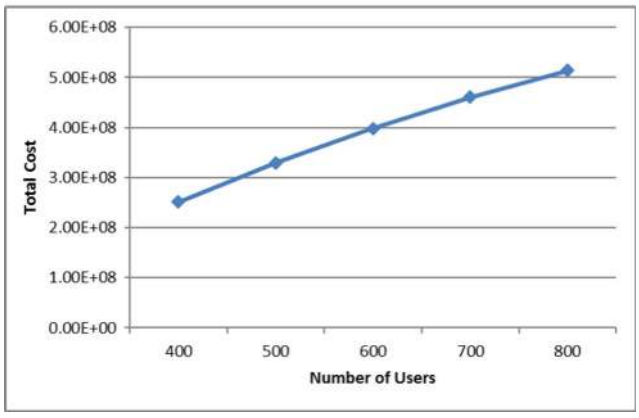
Input parameter	Value
Deployment area	1000×1000 m^2
Number of users	400, 500, *600*, 700, 800
Number of local cloudlets	5, 6, *7*, 8, 9
Number of global cloudlets	1
$\|R\|$	5
$\|T\|$	3
α	2
$F_{i,j}^t$	100 Mb/s
$\widetilde{F}_{j,k}^t$	10 Gb/s
PS, $\widetilde{PS}$	2×10^8 m/s
$E_i^{r,t}$	0.01 s
M_j^t	40 K
M_j^t	1 M

We are interested in computing two main parameters: the total cost (i.e., power consumption) and the average delay. The total cost is the value of the objective function discussed at the end of Section 3.2 while respecting the capacity and delay constraints of Section 3.3. As for the average delay, it is computed by averaging the total delay defined by (19).

As discussed in the previous section, each user is connected to a single cloudlet and sends all of its requests to it. The cost of moving any user's request to its designated cloudlet is given by (13). In our experiments, we assume α to be equal to two, which means that the cost is a quadratic function of the distance between the user and the cloudlet. Now, in the case that the cloudlet is unable to service the user's request, the request is re-directed to another cloudlet according to the catalog. We assume that the cost of moving a request from a cloudlet to another is a linear function of the distance between the two cloudlets. To reduce the cost, the choice of catalog entries is based only on the distances between local cloudlets. If no local cloudlet can serve the desired request, then the catalog entry for this service type is a global cloudlet.

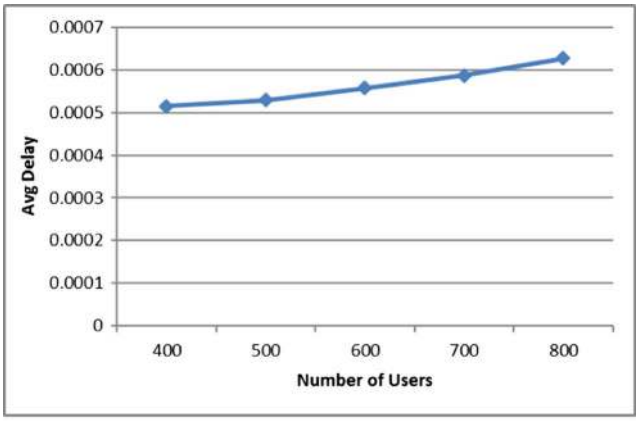


(a) Heavily Loaded Users

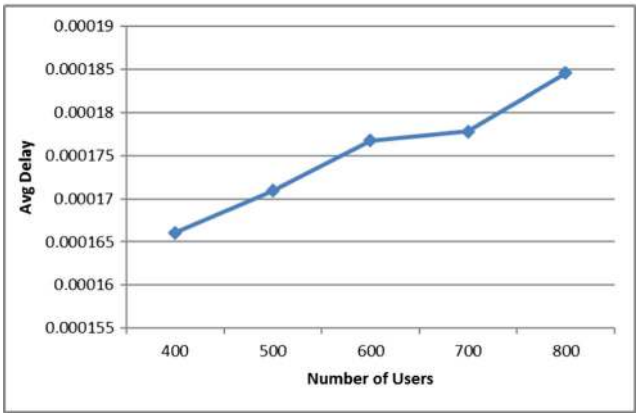


(b) Lightly Loaded Users

Fig. 1 The effect of increasing the number of users on the total power consumption



(a) Heavily Loaded Users



(b) Lightly Loaded Users

Fig. 2 The effect of increasing the number of users on the delay

To solve our model, we use the GNU Linear Programming Kit (GLPK) tool.[2]. We run our experiments on a Dell Inspiron laptop equipped with a 6th Generation Intel Core i7-6700HQ Processor (with 6MB Cache) and 16GB RAM (in addition to 16GB swap space).

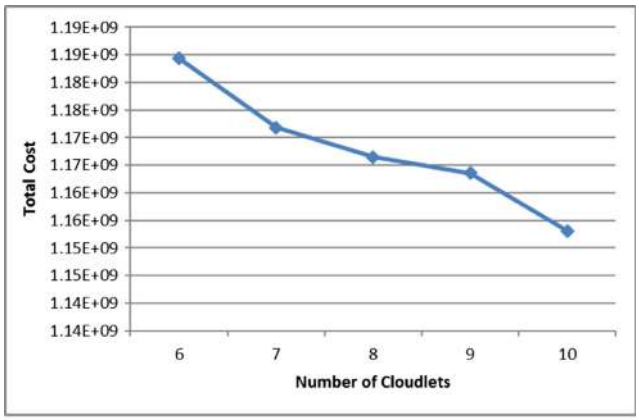
In the first set of experiment, we increase the number of users and compute the total cost of serving their requests as well as the average delay. For this set, we consider two types of users and conduct separate experiments for each type. The main difference between the two types is in the amount of load generated by each user (specifically, the request sizes s_i that are being generated). For the two types, the sizes are uniformly distributed over the intervals [1,50] for the first one (which is called lightly loaded users) and (50,100] for the second one (which is called heavily loaded users). The experimental setting that we use is discussed in the following paragraph.

We consider a simple topology of seven local cloudlets and one global cloudlet. The local cloudlets are randomly distributed over a deployment region of size 1000×1000 m^2 while the global cloudlet is assumed to be placed at the center of the deployment area with a 15 m height from the land surface. For each experiment, we distribute a number of users randomly. The number of users ranges between 400 users and 800 users.

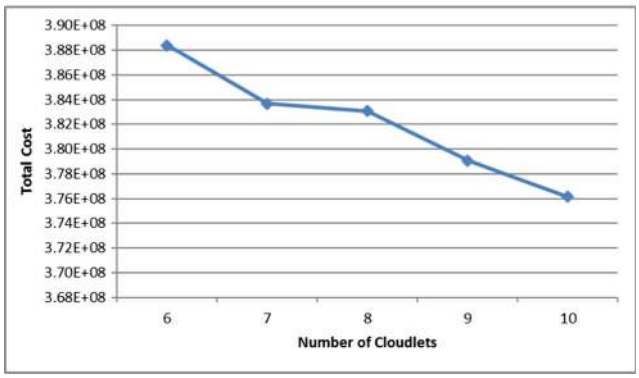
Figure 1a shows the effect if increasing the number of users on the total power consumption for the first experiments with only heavily loaded users. The figure shows a simple trend: a linear increase in the number of users corresponds to linear increases in the total power consumption. The same trend can be observed about the delay in the same experiment which is shown in Fig. 2a. However, the increase in the average delay is small compared with the increase in the total power consumption.

Figures 1b and 2b show the results of the same experiments set for lightly loaded users. The same trend appears here as well. However, the increase in the average delay (as shown in Fig. 2b) is much significant here than what appeared in Fig. 2a.

For the second set of experiments, we consider increasing the number of local cloudlets. For all settings, we consider a single global cloudlet positioned in the center of the



(a) Heavily Loaded Users



(b) Lightly Loaded Users

Fig. 3 The effect of increasing the number of cloudlets on the total power consumption

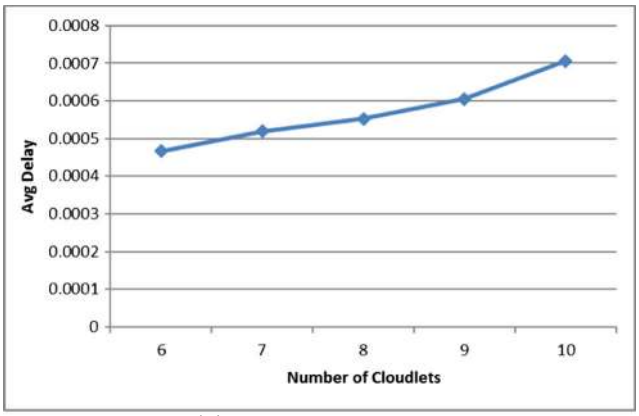
[2] https://www.gnu.org/software/glpk/

deployment area. We vary the number of local cloudlets from five to nine. The local cloudlets are randomly distributed across the deployment area.

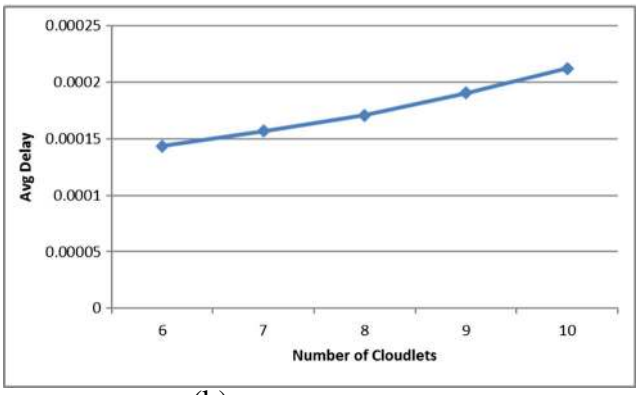
Figures 3a and 4a show the results of the second experiments set for heavily loaded users. Similar to Figs. 1a and 2a, the figures show a simple trend: a linear increase in the number of cloudlets corresponds to a linear decrease in the total cost. As for the average delay, increasing the number of cloudlets corresponds to an increase in the average delay. This delay can be justified by the fact that increasing the number of local cloudlets means that the dependence on the global cloudlet to serve the requests that cannot be handled by the local cloudlets is reduced.

Figures 3b and 4b show the results of the same experiments set for lightly loaded users. The same trend of Figs. 3a and 4a appears here as well.

For the final experiment, we consider a finer grained division of users based on their load. Instead of considering heavily loaded users and lightly loaded users, we consider four types of user loads: very light, moderately light, moderately heavy, and very heavy, where the sizes are uniformly distributed over the intervals [1,25], (25,50], (50,75], and (75,100], respectively.

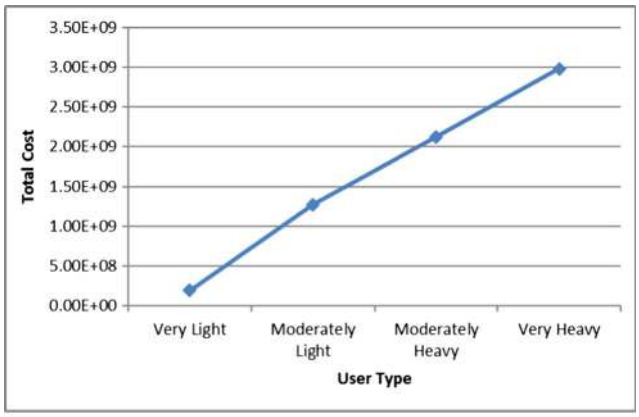


(a) Heavily Loaded Users

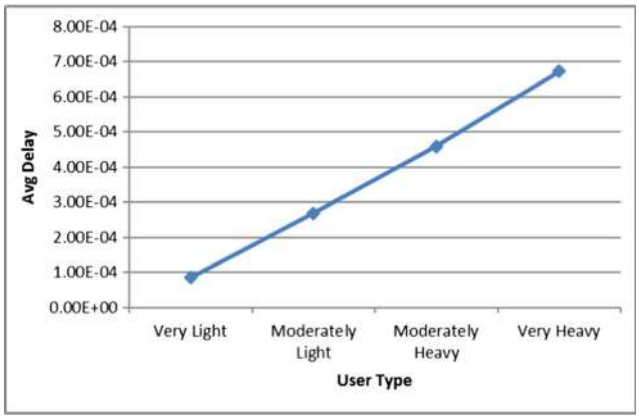


(b) Lightly Loaded Users

Fig. 4 The effect of increasing the number of cloudlets on the delay



(a) Total Cost



(b) Average Delay

Fig. 5 The effect of increasing the users' loads

Figure 5 shows the results of the third experiment set. The figure shows a simple trend: a linear increase in the loads generated by the users corresponds to linear increases in the total cost and the average delay.

5 Conclusion

In this work, we dealt with the problem of optimizing power consumption of MEC systems while adhering to delay and capacity constraints. We modeled this problem as a MILP optimization problem. Despite the wide consideration of the problem and the heavy usage of such an approach in both the CC literature and the mobile networking literature, to the best of our knowledge, very limited number of works have been done on this problem with such an optimization problem formulation for MEC systems. Moreover, we considered the recent trend that shy away from completely relying on WiFi-equipped local cloudlets with limited capabilities and add another layer of servers called global cloudlets. To prove that the developed model works well and generates reasonable results, we tested it using several realistic scenarios.

Acknowledgments This work was supported in part by the Jordan University of Science and Technology (Project Number 20160081) and supported financially by the Deanship of Scientific Research at Umm Al-Qura University to Dr. Lo'ai Tawalbeh (Grant Code: 15-COM-3-1-0017).

References

1. Shuja J, Gani A, ur Rehman MH, Ahmed E, Madani SA, Khan MK, Ko K (2016) Towards native code offloading based mcc frameworks for multimedia applications: a survey. J Netw Comput Appl 75:335–354
2. Shuja J, Gani A, Naveed A, Ahmed E, Hsu C-H (2016) Case of ARM emulation optimization for offloading mechanisms in mobile cloud computing. Futur Gener Comput Syst. ISSN 0167-739X
3. Al-Ayyoub M, Jararweh Y, Tawalbeh L, Benkhelifa E, Basalamah A (2015) Power optimization of large scale mobile cloud computing systems. In: 3rd international conference on future internet of things and cloud (FiCloud), 2015, IEEE, pp 670–674
4. Satyanarayanan M (2017) The emergence of edge computing. Computer 50(1):30–39
5. Jararweh Y, Doulat A, AlQudah O, Ahmed E, Al-Ayyoub M, Benkhelifa E (2016) The future of mobile cloud computing: integrating cloudlets and mobile edge computing. In: 23rd international conference on telecommunications (ICT), 2016, IEEE, pp 1–5
6. Shuja J, Bilal K, Madani SA, Othman M, Ranjan R, Balaji P, Khan SU (2016c) Survey of techniques and architectures for designing energy-efficient data centers. IEEE Syst J 10(2):507–519
7. Shuja J, Gani A, Shamshirband S, Ahmad RW, Bilal K (2016d) Sustainable cloud data centers: a survey of enabling techniques and technologies. Renew Sust Energ Rev 62:195–214
8. Huang D et al. (2011) Mobile cloud computing. IEEE COMSOC Multimedia Communications Technical Committee (MMTC) E-Letter 6(10):27–31
9. Dinh HT, Lee C, Niyato D, Wang P (2013) A survey of mobile cloud computing: architecture, applications, and approaches. Wirel Commun Mob Comput 13(18):1587–1611
10. Fesehaye D, Gao Y, Nahrstedt K, Wang G (2012) Impact of cloudlets on interactive mobile cloud applications. In: Enterprise distributed object computing conference (EDOC), 2012 IEEE 16th international, IEEE, pp 123–132
11. Soyata T, Muraleedharan R, Funai C, Kwon M, Heinzelman W (2012) Cloud-vision: real-time face recognition using a mobile-cloudlet-cloud acceleration architecture. In: IEEE Symposium on computers and communications (ISCC), 2012, IEEE, pp 000,059–000,066
12. Tawalbeh L, Jararweh Y, Ababneh F, Dosari F (2015) Large scale cloudlets deployment for efficient mobile cloud computing. J Networks 10(01)
13. Hegyi A, Flinck H, Ketyko I, Kuure P, Nemes C, Pinter L (2016) Application orchestration in mobile edge cloud: placing of iot applications to the edge. In: IEEE 1st international workshops on foundations and applications of self* systems (FAS*W), IEEE, pp 230–235
14. Hoang DT, Niyato D, Wang P (2012) Optimal admission control policy for mobile cloud computing hotspot with cloudlet. In: Wireless communications and networking conference (WCNC), 2012, IEEE, IEEE, pp 3145–3149
15. Satyanarayanan M, Bahl P, Caceres R, Davies N (2009) The case for vm-based cloudlets in mobile computing. IEEE Pervasive Comput 8(4):14–23
16. Shiraz M, Gani A (2012) Mobile cloud computing: critical analysis of application deployment in virtual machines. In: Proceedings of the international conference on information and computer networks (ICICN'12), vol 27
17. Verbelen T, Simoens P, De Turck F, Dhoedt B (2012) Cloudlets: bringing the cloud to the mobile user. In: Proceedings of the third ACM workshop on mobile cloud computing and services, ACM, pp 29–36
18. Miettinen AP, Nurminen JK (2010) Energy efficiency of mobile clients in cloud computing. In: Proceedings of the 2nd USENIX conference on hot topics in cloud computing, USENIX association, pp 4–4
19. Benkhelifa E, Welsh T, Tawalbeh L, Jararweh Y, Basalamah A (2015) User profiling for energy optimisation in mobile cloud computing. Procedia Computer Science 52:1159–1165
20. Liu F, Shu P, Jin H, Ding L, Yu J, Niu D, Li B (2013) Gearing resource-poor mobile devices with powerful clouds: architectures, challenges, and applications. IEEE Wirel Commun 20(3):14–22
21. Wang S, Dey S (2013) Adaptive mobile cloud computing to enable rich mobile multimedia applications. IEEE Trans Multimedia 15(4):870–883
22. Karadimce A, Davcev D (2013) Adaptive multimedia learning delivered in mobile cloud computing environment. In: CLOUD COMPUTING 2013, the fourth international conference on cloud computing, GRIDs, and virtualization, pp 62–67
23. Quwaider M, Jararweh Y (2016) A cloud supported model for efficient community health awareness. Pervasive Mob Comput 28:35–50
24. Althebyan Q, Yaseen Q, Jararweh Y, Al-Ayyoub M (2016) Cloud support for large scale e-healthcare systems. Ann Telecommun 71(9-10):503–515
25. Orsini G, Bade D, Lamersdorf W (2016) Cloudaware: a context-adaptive middleware for mobile edge and cloud computing applications. In: IEEE 1st international workshops on foundations and applications of self* systems (FAS*W), IEEE, pp 216–221
26. Mukherjee A, De D, Roy DG (2016) A power and latency aware cloudlet selection strategy for multi-cloudlet environment. In: IEEE transactions on cloud computing, vol PP, no 99, p 1
27. Gai K, Qiu M, Zhao H, Tao L, Zong Z (2016) Dynamic energy-aware cloudlet-based mobile cloud computing model for green computing. J Netw Comput Appl 59:46–54
28. Gai K, Qiu M, Zhao H, Liu M (2016) Energy-aware optimal task assignment for mobile heterogeneous embedded systems in cloud computing. In: IEEE 3rd international conference on cyber security and cloud computing (CSCloud), 2016, IEEE, pp 198–203
29. Mehta A, Tarneberg W, Klein C, Tordsson J, Kihl M, Elmroth E (2016) How beneficial are intermediate layer data centers in mobile edge networks? In: IEEE 1st international workshops on foundations and applications of self* systems (FAS*W), IEEE, pp 222–229
30. Yaseen Q, AlBalas F, Jararweh Y, Al-Ayyoub M (2016) A fog computing based system for selective forwarding detection in mobile wireless sensor networks. In: IEEE 1st international workshops on foundations and applications of self* systems (FAS*W), IEEE, pp 256–262
31. Yaseen Q, Albalas F, Jararwah Y, Al-Ayyoub M (2017) Leveraging fog computing and software defined systems for selective forwarding attacks detection in mobile wireless sensor networks. Trans Emerging Tel Tech e3183. doi:10.1002/ett.3183
32. Yaseen Q, Jararweh Y, Al-Ayyoub M, AlDwairi M (2017) Collusion attacks in internet of things: detection and mitigation using

a fog based model. In: Sensors applications symposium (SAS), 2017 IEEE, IEEE, pp 1–5
33. Garcia-Perez CA, Merino P (2016) Enabling low latency services on lte networks. In: IEEE International workshops on foundations and applications of self* systems, IEEE, pp 248–255
34. Al-Ayyoub M (2010) Dynamic spectrum allocation in cellular networks. PhD Thesis, State University of New York at Stony Brook
35. Modiano E, Wieselthier JE, Ephremides A (1996) A simple analysis of average queueing delay in tree networks. IEEE Trans Inf Theory 42(2):660–664
36. Kurose JF (2005) Computer networking: a top-down approach featuring the internet. 3/e. Pearson Education India
37. Tanenbaum AS et al (2003) Computer networks, 4-th edition. ed: Prentice Hall
38. Little J (1961) A proof of the queueing disciplines. Oper Res 9:383–387