Future Generation Computer Systems 102 (2020) 862–875

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

An optimal delay aware task assignment scheme for wireless SDN networked edge cloudlets

G.S.S. Chalapathi [a,1], Vinay Chamola [a,*], Chen-Khong Tham [b], S. Gurunarayanan [a], Nirwan Ansari [c]

[a] Department of Electrical and Electronics Engineering, BITS-Pilani, Pilani, Rajasthan 333031, India
[b] Department of Electrical and Computer Engineering, National University of Singapore, 117583 Singapore
[c] Advanced Networking Laboratory, Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102, USA

ARTICLE INFO

Article history:
Received 15 January 2019
Received in revised form 24 June 2019
Accepted 6 September 2019
Available online 14 September 2019

Keywords:
Edge computing
Load balancing
Quality of service
Task assignment
Wireless SDN

ABSTRACT

Over the past decade, there has been an increasing demand for mobile devices to perform computationally intensive tasks. However, the computational capability of these devices is limited due to memory, power and portability constraints. One of the feasible and attractive ways to enhance the performance of the resource-limited mobile devices is to offload their computationally intensive tasks on to the cloud servers when internet connectivity is available. However, when cloud servers are involved in processing, the latency and cost of computation increases. To mitigate these problems, devices with high computational resources, called cloudlets, can be deployed in the locations close to the mobile users/devices. The mobile devices can then offload their computationally intensive tasks on to them. Due to easier access and nearness of the cloudlets, the cost and latency in processing the tasks decreases. In this work, we focus on task assignment problem in a multi-cloudlet network connected via a wireless SDN network, which services the task offload requests from mobile devices in a given locality. The aim of the proposed solution is to minimize latency and thus enhance the quality of service for mobile devices. We prove the optimality of the proposed solution mathematically and employ an admission control policy to maintain this optimality even in heavily loaded networks. We also perform numerical simulations for two scenarios of small and large networks and evaluate the performance for varying traffic and network parameters. The results demonstrate that the proposed task assignment method offers reduced latency compared to state-of-the-art task assignment approaches and hence improves the quality of service offered to mobile devices.

1. Introduction

Rapid advances made in mobile computing have enabled proliferation of mobile devices in a variety of tasks like video calling, playing games, image processing applications, etc. However, such capabilities have also come with new challenges. Many of these applications being computationally intensive require high computational power. Mobile devices like smartphones, tablet PCs, etc. cannot handle such intensive computations due to their memory, power and portability constraints. Cloud services have been seen as a solution to this problem and have become quite popular as the cloud devices are capable of performing diverse kind of computationally intensive tasks offloaded from these mobile devices on to them [1,2]. They perform these tasks and provide the results to the mobile devices. These cloud services have features like large data-storage capacity, high computation capabilities and can cater to a variety of computational demands of the mobile devices. Thus, the computationally intensive tasks that have been offloaded from mobile devices on to them can be executed very quickly. In the absence of these cloud services, trying to execute such tasks on the mobile device itself leads not only to the slowing down of the device (due to consumption of large fraction of processing resources of the mobile device) but also to quicker battery energy drainage.

Along with the above-mentioned advantages of using the cloud services, there are certain drawbacks of using this option. Firstly, the mobile device requires an internet connection to offload its computation request on to the cloud server and receive the results of its computation. Internet connectivity is limited and in the absence of Wi-Fi hotspots, the user has to rely on

* Corresponding author.
E-mail addresses: gssc@pilani.bits-pilani.ac.in (S.S.C. G.), vinay.chamola@pilani.bits-pilani.ac.in (V. Chamola), eletck@nus.edu.sg (C.-K. Tham), sguru@pilani.bits-pilani.ac.in (S. G.), nirwan.ansari@njit.edu (N. Ansari).

[1] This work is based in part on our previous paper titled "Latency Aware Mobile Task Assignment and Load Balancing for Edge Cloudlets", Presented at PerCom2017 workshop.

https://doi.org/10.1016/j.future.2019.09.003
0167-739X/

cellular networks for offloading and receiving the results. This could incur data usage charges. Secondly, usage of cloud services would incur additional subscription charges levied by the cloud service providers on their users. Also, when the user is using the cloud services from far off locations, high computation tasks like face recognition and image processing based applications can experience high latency in getting the results. These factors envisage the need for devising a new solution to this problem of mobile computing and this has led to *edge* or *fog computing* in the recent years.

In edge computing, the computation is performed on a computationally powerful device, which is present at a location near the offloading mobile device. Such computationally powerful devices are called *cloudlets* [3,4]. These cloudlets have advantages like the small size, easy installation and lower cost. However, they are computationally less powerful than cloud servers. The cloudlets can mitigate the problems involved in using the cloud servers by introducing an additional layer that lies in between cloud servers and mobile devices.

A network of fog/edge devices termed as cloudlets has been considered in this paper. These cloudlets form a network and their computational resources are used to serve the tasks offloaded by mobile devices in their vicinity [5]. Traditionally, the nearest cloudlet serves the request of a mobile device. However, this is an inefficient approach because the nearest cloudlet may be heavily loaded at a given time while another cloudlet in the network may be lightly loaded as illustrated in Fig. 1. In this figure, we see that there are more mobile devices in the vicinity of Cloudlet1 as compared to Cloudlet2 or Cloudlet3. Thus, as per the traditional task assignment scheme, Cloudlet1 will handle all the tasks offloaded by the mobile devices in its vicinity. Therefore, Cloudlet1 becomes heavily loaded (hence marked as red). The mobile devices served by such a heavily loaded cloudlet will thus experience high latency for completion of their computation tasks [6]. At the same time, since there are lesser mobile devices near Cloudlet3, it is lightly loaded (hence marked as green) and Cloudlet2 has a medium load (hence marked as yellow). We, therefore, observe that there is an imbalance in the load allocated to these cloudlets. We can solve this load imbalance among the cloudlets by networking these cloudlets through Software Defined Network (SDN) switches thus enabling the network load to be served in a cooperative manner. This ensures a better quality of service and better utilization of resources by load balancing. The work in this paper focuses on the problem of task assignment for wireless SDN networked cloudlets with an aim to reduce the latency in processing the tasks offloaded by the mobile devices. An optimal scheme is proposed here and its performance is compared with other state-of-the-art offloaded task assignment schemes discussed in [7,8].

The rest of the paper is organized as follows. Section 2 of this paper discusses the previous works related to this problem. The next section (Section 3) describes the system model considered in this work. The problem formulation is discussed in Section 3.2 and the solution methodology in Section 4. The simulation results are discussed in Section 5 and the conclusion is presented in Section 6.

2. Literature review

Mobile computing has made information and data processing possible everywhere. Yet, by its very nature, mobile hardware is inferior to stationary hardware in terms of performance because of limited computing power, storage and battery capacity of the former [9]. Cloud technology has found its application in recent years in mobile computing commonly referred to as *Mobile Cloud Computing* (MCC). It is used especially for offloading computationally intensive tasks from the mobile devices on to the cloud

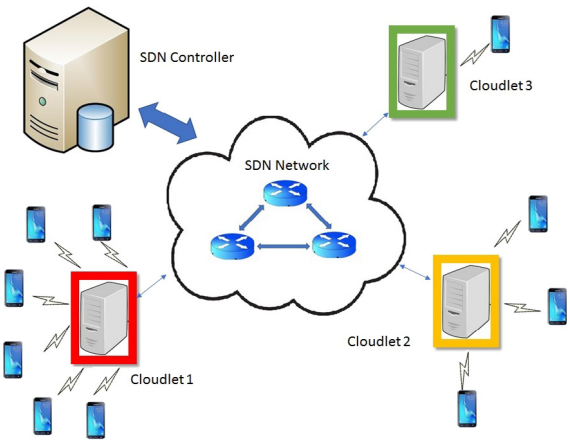


Fig. 1. A cloudlet network with the different participating entities. Cloudlet1 is heavily loaded, Cloudlet2 is having medium-load and Cloudlet3 is lightly loaded.

servers [10,11]. Apart from the challenges of a heterogeneous network [12], the MCC technology has associated limitations of high network latency and high transmission power involved in connecting with the cloud [13]. To mitigate these shortcomings, researchers have explored the efficient use of a network of supporting devices called cloudlets [3,14] also referred to as fog devices [15,16] or edge devices [17].

Cloudlets have been used to support the cloud services to mobile devices in many real time applications. An architecture based on edge computing has been proposed in [18] to achieve ultra-low latency and network congestion reduction for the upcoming 5G mobile systems for application in smart cities. A novel method to manage data streams at the mobile edge to enhance scalability in IoT architecture has been presented in [19]. Recently, a vehicle control system has been proposed in [20] where resources are allocated dynamically. In this system, computation is switched between the edge devices and the cloud according to the network conditions to overcome instability in vehicle control caused by long latencies in the absence of cloudlets. For widely used mobile device applications like gaming, face recognition, etc., that involve heavy computation and require low latencies, the use of cloudlets has been demonstrated to be technically feasible and beneficial compared to direct communication with cloud servers [21]. Surveys on edge computing have been presented in [22–29] while [30] presents an early version of a survey on computational offloading in mobile systems. A survey on computational offloading in Mobile Edge Computing (MEC) especially in terms of the current state of standardization and current work on various aspects of offloading like mobility management, decision making in offloading, and resource allocation on the cloudlets has been presented in [25].

Recently, researchers have also explored the effect of combining task-offloading decisions with several other network parameters. Multi-objective resource allocation and joint task offloading schemes have been presented by authors in [31–33]. Wang et al. [34] have addressed the problem of task offloading along with strategies employed for content caching in cellular networks having edge computing. The same authors have focused on managing task offloading along with interference management in [35]. Many works like [3] have propagated the concept of Virtual Machines (VMs), which will be invoked on the cloud/cloudlet devices to execute the offloaded tasks from the mobile users. Plachy et al. [36] have presented a strategy to find an optimal placement of these VMs while optimizing the communication costs in an MEC environment where the mobile devices are not stationary. They tried to find an optimal trade-off between VM

migration cost and reducing the cost of communication from the VM and the mobile user. Another important aspect is to deal with dynamic mobility of the mobile users in a cloudlet network. This opens up other interesting problems namely — where should the services requested by a mobile device be deployed/run in an MEC network and where should the service be migrated to cope up with user mobility and/or network changes. Wang et al. [37] explored the problem of service placement in MEC, coined as Mobile-micro-cloud (MMC).

In offloading the resource intensive tasks on to the cloudlets, one of the primary considerations is to minimize latency by managing the network traffic to provide better QoS to the users. The traditional way of task offloading is to offload the task from the mobile device to the nearest available cloudlet with an intention to minimize the communication delay (the Round-Trip-Time (RTT)) between the mobile device and destined cloudlet. This approach however does not take into consideration of the current workload at the nearest cloudlet and thus leads to poor latency during heavy traffic conditions. In an MEC environment, the devices are mobile, energy limited, and multiple cloudlets are available with possibly distributed specialized resources. These factors create a need for specific decisions regarding choice of cloudlet to offload the task, proportion of computation to be offloaded, and task distribution [9]. These details have been discussed in [38]. More recently, [39–43] have dealt with some or all of these issues for mobile cloud computing in different directions. Zhang et al. [39] tried to minimize energy of both computation and data transmission with latency as constraint while Sardellitti et al. [40] addressed a similar problem in Multi-Input–Multi-output (MIMO) multicell systems. Fan and Ansari [44] proposed a workload allocation scheme to assign user tasks among cloudlets in a hierarchical cloudlet network to minimize their response time. Muñozet al. [41] jointly optimized energy consumption and latency by delivering a framework in Femto Access Point (FAP) network using MIMO radios. You et al. [42] also tried to optimize energy consumption using Time-division multiple access (TDMA) and orthogonal frequency-division multiple access (OFDMA) based resource allocation to the mobile users. A recent task assignment scheme [45] focuses on optimization of energy and delay for artificial intelligence(AI) based tasks. The authors in this work proposed a multiple algorithm approach which selects an appropriate algorithm for a given AI task to satisfy its real-time requirements and thus optimize latency and energy for task computation.

Kao et al. [43] addressed a special scenario where the application can be represented as a serial tree task graphs. They focused on optimizing the latency in computing the offloaded applications composed of many tasks/routines by mapping different tasks on to multiple computing nodes/devices. However, they focused on a scenario where the application can be expressed as serial trees, which may not be always possible. Mahmud et al. [46] have focused on fog applications which can be decomposed into modules which can be executed independently. The authors proposed a policy for managing these application modules to meet diverse latency and signal processing rate specifications. Zhang et al. [47] proposed strategies to maximize the benefit of mobile cloud computing resources and the utilities on mobile devices (which in this case are vehicles) with latency constraints. The authors in [48] proposed a task assignment scheme for vehicular edge computing network. They have first proposed an architecture for vehicular edge computing where resource rich vehicles become the edge devices. These vehicular edge devices collect similar computationally intensive tasks and then they employ a task offloading strategy to offer low-latency guarantees. Mao et al. [49] proposed an online algorithm to jointly decide CPU cycle frequencies of mobile devices, transmit power and offloading decision for minimizing latency and task failure. A generic problem of task offloading for intermittent connection between mobile device and cloudlet has been considered in [50]. This intermittent connection is modeled and solved to minimize the communication and computation costs. Here the application is divided into phases and the decision is made whether to execute each application phase locally or to offload on to a nearby cloudlet. A different case of [50] has been discussed in [51], where cost constraints are similar, but now a set of parallel tasks are to be processed on cloudlets. It is to be noted that these works have considered cloudlets that are stand-alone and primarily serve mobile devices in their service area.

Tiwary et al. [52] proposed a task assignment scheme for a special case of cloudlet network where all the cloudlets are smart phones, i.e., energy constrained devices. This scheme uses a game theory based approach to allocate the offloaded tasks to the cloudlets by optimizing the battery lives of the cloudlet devices. A task assignment scheme for a multi-cloudlet network named LEAN is discussed in [7]. In this scheme, the tasks are assigned to a cloudlet which is available in the closest proximity of the mobile device offloading the task. LEAN does not consider the current load on the cloudlet and thus leads to load imbalance which in turn leads to increase in the latency in processing the offloaded tasks. Yao and Ansari [53] offloaded the computing tasks to the fog node (i.e., cloudlet) at the edge in IoT networks and investigated the computing resource provisioning problem to minimize the network system cost constrained by user QoS requirements. They also included reliability in their resource provisioning problem in [54] by considering the failure and recovery probabilities of the computing resources.

There is another scheme that has been proposed by the Mukherjee et al. [8] to reduce the network latency in a multi-cloudlet network where the cloudlets are connected to each other. The tasks offloaded to one cloudlet can be served on any of the cloudlets in the network. The tasks to be offloaded in this scheme are offloaded to the nearest cloudlet and are served by them. However, when the nearest cloudlet is not able to serve this request, the task is processed on the cloudlet, which offers the minimum latency to process the request. However, this scheme does not account for the current load at the cloudlets which is a significant parameter that affects the overall network latency. Thus in our current paper, we try to improve the network latency by also considering the cloudlet load and develop a framework for optimal task offloading in a multi-cloudlet network. We consider a cloudlet network, which is connected by wireless SDN switches that enable the cloudlets to cooperatively serve tasks offloaded by the mobile devices. Networking the devices via SDN switches allows separation of the control and data pane and greater flexibility for routing of the offload requests among the clouldets.

This ensures reduced latency for mobile devices, thereby improving QoS experienced by them and also balancing the load at the cloudlets.

The main contributions of this paper can be summarized as follows:

1. We propose an optimal task assignment scheme for a multi-cloudlet environment in which a wireless SDN network is used to connect the cloudlets. We present the mechanics of this scheme and also establish its optimality and convergence mathematically.
2. We propose admission control for the proposed scheme to manage the admission of task assignment requests when the number of requests is greater than what the network can handle.
3. We also carry out a performance analysis for different sized networks- a small network consisting of three cloudlets and a large one consisting of ten cloudlets.

Table 1
Notation summary.

Notation	Meaning
y	Location of the mobile user
$\mathcal{W}$	Set of Cloudlets
k	Index of the cloudlet in $\mathcal{W}$
$\lambda(y)$	Task offloading arrival rate per unit area
$\tau(y)$	Average file size
$\gamma(y)$	$:= \lambda(y)\tau(y)$, Mobile task offload density
S_k^{max}	Maximum effective service rate offered by the kth cloudlet
$s_k(y)$	The effective service rate offered by kth cloudlet at y
α	Parameter capturing distance dependency of effective service rate
$u_k(y)$	Task assignment indicator of kth cloudlet at y
ρ_k	kth cloudlet's load
$\mathcal{L}_k(\rho_k)$	Latency indicator of kth cloudlet
$\mathcal{Z}$	Feasible set of cloudlet loads
$\tilde{\mathcal{Z}}$	Relaxed feasible set of cloudlet loads
$\Phi(\boldsymbol{\rho})$	Objective function which is sum of latency indicators of all clouldets
i	Index of the time slot
$u_k^i(y)$	Task assignment indicator of kth cloudlet at y in ith time slot
ψ_k^i	Resistance index of kth cloudlet in time slot i
ρ_k^i	Load sent by the kth cloudlet to the SDN controller in ith time slot
$T_k(\rho_k^i)$	Load of kth cloudlet in the ith time slot
θ	Admission control parameter

4. This work also includes a thorough performance analysis of latency for different requests rates for these different sized networks with varying network environment parameter.
5. Our simulations show that the proposed task assignment scheme reduces the latency experienced by offloaded tasks in a multi-cloudlet scenario as compared to the existing state-of-the-art methods.

3. System description

3.1. Network traffic and cloudlet load

We consider a system of interconnected cloudlet network in our paper, as depicted in Fig. 1. In our model, the tasks offloaded by a mobile device on an overloaded cloudlet can be served on some other lightly loaded cloudlet in the network. The wireless SDN controller present in the network takes care of the offloaded task assignment as per the scheme discussed in Section 4. Connecting the edge cloudlet devices using SDN network is extremely advantageous. Since the SDN controller has an overall view of the network, it can control the flow of packets dynamically according to the different parameters like current network traffic, available links, number of hops from a sender to a destination, etc. Thus, we can provision better Quality of Service (QoS) in an SDN network when compared to a traditional network because SDN can incorporate dynamic network feedback into its decisions.

Let us consider our cloudlet network as represented by the set of cloudlets $\mathcal{W}$ with $\mathcal{W} = \{G_1, G_2, \ldots, G_k, \ldots, G_{|\mathcal{W}|}\}$, where we refer to the kth cloudlet as G_k. Let us denote the maximum effective service rate offered by the kth cloudlet as S_k^{max}. It is assumed that the task offloading requests from the mobile devices located at $y \in \mathcal{R}$ ($\mathcal{R}$ being the geographical region) follows a Poisson Process that has an arrival rate of $\lambda(y)$. Poisson arrival process has been taken into account based on its wide acceptance for generic traffic for mobile networks [55]. The analysis presented later in our current work is not limited by the traffic's being Poisson in nature but holds valid for any other type of distribution as well. We denote the average file size offloaded at location y as $\tau(y)$. Thus, we get the mobile task offload density denoted by $\gamma(y)$ as $\gamma(y) = \lambda(y)\tau(y)$, where the spatial task offload variability is captured by $\gamma(y)$.

A number of factors determine the latency experienced by the tasks that the mobile devices offload on to the cloudlets for the execution. These factors are enumerated as: (*i*) the current load of the cloudlet in executing the offloaded task request, (*ii*) the maximum rate of service of the cloudlet processing its request offers, and (*iii*) the distance between the serving cloudlet serving and the corresponding mobile device. Without loss of generality, let us consider the effective service rate that the cloudlet k offers to the mobile device located at y as

$$s_k(y) = \frac{S_k^{max}}{1 + (dis(G_k, y)/d_0)^{\alpha}} \tag{1}$$

where $dis(G_k, y)$ is the Euclidean distance of the kth cloudlet from the mobile device that is at location y. In the above equation, we introduce the parameter α to helps us adjust the effective service rate according to different network scenarios. d_0 is the scaling factor for the distance. The proof of this expression is given in the appendix at the end of this paper.

As compared to the computational delay at the cloudlets, the transmission delay and the propagation delay for our network are negligible (especially due to the fact that the cloudlets are in close proximity of the mobile devices). To illustrate this point, we consider our wireless SDN network to support 10 Gbps links supported by state-of-the-art technology (e.g. [56]). Further let us consider the scenario of a mobile user whose offloaded tasks are being serviced on a cloudlet at a distance 100 m away and let the maximum computational service rate (S^{max}) of this cloudlet be 3000 KBps.. Assuming each packet to be of size 40 KB, it is observed that transmission delay and propagation delay (for a round trip) are 32 μs and 0.66 μs, respectively. The effective service rate observed by the mobile user at 100m from this cloudlet as per Eq. (1) is 272.72KBps (with α taken to be unity). The computational delay for a 40KB packet offloaded on to this cloudlet by this mobile user will be 146670.5 μs which is much higher than the propagation and transmission delays. Advances in the network technology have greatly reduced the forwarding delay at the switches to about 10 μs [57]. Thus, we can see that the computational delay is the most dominant delay and much greater than other delays. Therefore, in this work, we primarily consider the computational delay for latency evaluation.

To specify task assignment relationship between the mobile devices and the cloudlets, let us introduce a function $u_k(y)$ called task assignment indicator function. This function takes the value 1 if the mobile user located at y is served by the cloudlet k and 0 otherwise. Also, let us define the cloudlet load by ρ_k,

$$\rho_k = \int_{\mathcal{R}} \frac{\gamma(y)}{s_k(y)} u_k(y) dy. \tag{2}$$

As given by [58] the cloudlet load represents the time fraction for which the cloudlet k is engaged in serving its traffic requests.

Definition 1. $\mathcal{Z}$ denotes the feasible set of the cloudlet loads $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_{|\mathcal{W}|})$. We define $\mathcal{Z}$ as

$$\mathcal{Z} = \Big\{ \boldsymbol{\rho} \,|\, \rho_k = \int_{\mathcal{R}} \frac{\gamma(y)}{s_k(y)} u_k(y) dy,\ 0 \le \rho_k \le 1 - \epsilon,\ \forall k \in \mathcal{W},$$
$$u_k(y) \in \{0, 1\}, \sum_{k=1}^{|\mathcal{W}|} u_k(y) = 1,\ \ \forall k \in \mathcal{W},\ \ \forall y \in \mathcal{R} \Big\},$$

with ϵ being an arbitrarily small positive constant. The assumption here is that an offloaded task of a particular mobile device is served in its entirety by only one cloudlet in the network. Although a mobile device would be connected to the closest cloudlet in the network, the cloudlet that serves its offloaded task request is decided based on the task assignment algorithm discussed in Section 4. The arrivals of offloaded tasks follow Poisson process, and thus the sum of arrivals of offloaded tasks is also a Poisson process. There is only one server at a cloudlet and the service process follows a general distribution. Thus, the traffic arrivals at the cloudlets are modeled as an M/G/1 queue. At the cloudlet k, we can give the average traffic flow by the fraction $\frac{\rho_k}{1-\rho_k}$ [58]. From the Little's Law, the latency that a traffic flow experiences is proportional to the average number of flows in the system [59]. Hence, at a given cloudlet, the total number of flows is considered as the kth cloudlet's latency indicator i.e. $\mathcal{L}_k(\rho_k)$, which is given by [58]

$$\mathcal{L}_k(\rho_k) = \frac{\rho_k}{1-\rho_k}. \tag{3}$$

From the above equation, we can see that as the value of ρ_k tends to 1, the latency indicator, $\mathcal{L}_k(\rho_k)$ increases exponentially approaching ∞. It is to be noted that the latency indicator being a relative indicator of the overall latency of the system, is unitless. Several contemporary studies like [58,60,61] have used the above indicator to quantitatively analyze the performance of a system's latency, due to its ability to give a comprehensive view of the latency performance of the network by jointly capturing computational as well as queuing delay. It can be seen here that both these delays have been well captured in the expression for the latency in Eqn. (3) where ρ_k (given in Eqn. (2)) is a function of the data traffic size and computational rate offered, and $\frac{\rho_k}{1-\rho_k}$ is a well known expression for queuing delay.

3.2. Problem formulation

The problem [Pr1] denotes the problem to minimize the total network latency and is formulated as follows

$$[\textbf{Pr1}] \quad \underset{\rho}{\text{minimize}} \quad \Phi(\boldsymbol{\rho}) = \sum_{k=1}^{|\mathcal{W}|} \mathcal{L}_k(\rho_k)$$
$$\text{subject to:} \quad \boldsymbol{\rho} \in \mathcal{Z}$$

We propose a task assignment scheme called Latency Aware Task Assignment (LATA) in this paper. Using this scheme, the SDN controller aims to improve the overall QoS experience of the mobile devices by solving the above-mentioned optimization problem. It does the task assignment for the service requests, i.e., it maps offloaded task requests to the serving cloudlet, for ensuring best latency performance in the network. As described in Section 4, we arrive at the optimal solution to [Pr1] by balancing the current load of the cloudlets (ρ) in the network. A description of LATA is presented in the next section. The notations used in this work are summarized in Table 1.

4. The LATA optimal task assignment scheme

We present LATA, the proposed task assignment scheme in this section. This scheme aims to find the optimal solution of the problem [Pr1] that was formulated in the previous section. Let us now describe how the task assignment algorithm works briefly. The detailed description will follow in sequel.

Each cloudlet estimates a variable called "resistance index" (described later) and advertises it to the SDN controller. The resistance index is calculated for each individual cloudlet periodically by evaluating their respective traffic loads. The wireless SDN controller then decides which cloudlet should be assigned the offloaded task with the aim to minimize the value of the objective function formulated in [Pr1].

We make the following assumptions in this task assignment algorithm:

Algorithm 1 SDN Controller-side Algorithm

Input: At the beginning of time slot i, the resistance index ψ_k^i, $\forall k \in \mathcal{W}$ and the effective service rate $s_k(y)$ at location y, $\forall y \in \mathcal{R}$, $\forall k \in \mathcal{W}$

Output: Task assignment indicator $u_k^i(y)$

1: Receive the resistance indices (ψ_k^i) for all the cloudlets at the beginning of the time slot.
2: Upon receiving a task offloading request at location y evaluate $s_k(y)$, $\forall k \in \mathcal{W}$ using Eq. (1).
3: Assign a cloudlet for the request at location y, i.e., calculate $u_k^i(y)$ using Eq. (6).

1. The time scale of the traffic arrival and departure processes is faster than the scale at which the resistance indices of the cloudlets are unicasted (to the SDN controller), so that our proposed scheme converges. This assumption ensures that for the current set of resistance indices of the cloudlets, the SDN controller makes the task offloading decisions before the cloudlets send the next set of indices into the network.
2. The resistance indices are sent to the SDN controller by all the cloudlets at the same time, i.e. the cloudlets in our network are synchronized.

LATA consists of two algorithms, one running at the SDN controller and another at the cloudlet. We will now describe these two algorithms.

4.1. SDN controller side algorithm

As mentioned earlier, LATA aims at arriving at the optimal solution to the problem [Pr1] that has been formulated previously. The feasible set $\mathcal{Z}$ defined in Section 3 is not a convex set as $u_k(y) \in \{0, 1\}$. We modify the above constraint as $0 \le u_k(y) \le 1$, thus introducing convexity to the optimization problem [Pr1].

With such relaxation in place, $u_k(y)$ can be seen as the probability that task offloaded by a mobile device at location y is serviced by the cloudlet k. We can now define $\tilde{\mathcal{Z}}$, the set of relaxed cloudlet loads given as

$$\tilde{\mathcal{Z}} = \Big\{ \boldsymbol{\rho} \,|\, \rho_k = \int_{\mathcal{R}} \frac{\gamma(y)}{s_k(y)} u_k(y) dy,\ 0 \le \rho_k \le 1-\epsilon, \forall k \in \mathcal{W},$$
$$0 \le u_k(y) \le 1,\ \sum_{k=1}^{|\mathcal{W}|} u_k(y) = 1, \forall k \in \mathcal{W}, \forall y \in \mathcal{R} \Big\}. \tag{4}$$

The set $\tilde{\mathcal{Z}}$ mentioned above is convex and Kim et al. [62] proved its convexity of the same. For brevity, we have omitted the detailed proof, which can be found in [62]. By applying the above-obtained relaxation in problem [Pr1], we arrive at a modified problem [Pr2] which can be formulated as

$$[\textbf{Pr2}] \quad \underset{\boldsymbol{\rho} \in \tilde{\mathcal{Z}}}{\text{minimize}} \quad \Phi(\boldsymbol{\rho}) = \sum_{k=1}^{|\mathcal{W}|} \mathcal{L}_k(\rho_k).$$

The new optimization problem [Pr2] has been formulated using $\tilde{\mathcal{Z}}$. However, it can be seen from Theorems 1 and 2 presented further in this section that we get a deterministic task assignment (which belongs to $\mathcal{Z}$) based on the proposed task assignment algorithm presented later in the section.

We define a time slot as the time taken between any two consecutive updates of resistance index. In the beginning of the

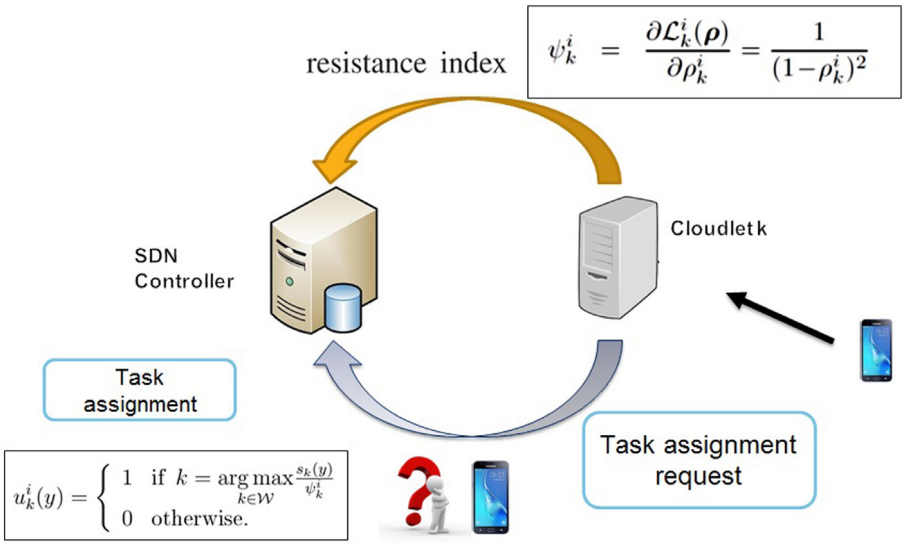


Fig. 2. Illustrative representation of the process of task offloading from the mobile user on to the cloudlet with intervention of the SDN Controller.

Algorithm 2 Algorithm at the Cloudlet-side

Input: Task assignment $u_k^i(y), \forall y \in \mathcal{R}$ at the beginning of time slot i

Output: The resistance index ψ_k^{i+1}

1: Calculate the current load $T_k(\rho_k^i)$ using Eq. (7)
2: Evaluate ρ_k^{i+1} using Eq. (8)
3: Calculate the resistance index ψ_k^{i+1} using the Eq. (5)
4: Advertise this ψ_k^{i+1} to the SDN Controller at the beginning of the next time slot.

ith time slot, the respective resistance indices are sent by the cloudlets to the SDN controller. The cloudlets are assigned the tasks offloaded by the mobile device based on the effective service rate they offer and the resistance indices that they have sent to the SDN controller. It is assumed that the resistance indices are sent by the cloudlets at the start of the time slot i. The term ψ_k^i denotes the resistance index sent by the cloudlet k at the beginning of time slot i which is defined as

$$\psi_k^i = \frac{\partial \mathcal{L}_k^i(\boldsymbol{\rho})}{\partial \rho_k^i} = \frac{1}{(1-\rho_k^i)^2}. \tag{5}$$

We use the function given below to assign the offloaded tasks of the mobile devices (for any device at location y) to the cloudlets.

$$u_k^i(y) = \begin{cases} 1 & \text{if } k = \arg\max_{k \in \mathcal{W}} \frac{s_k(y)}{\psi_k^i} \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

$u_k(y)$ is the task assignment indicator function (defined in Section 3), which indicates whether or not the kth cloudlet services the task requests offloaded by the device at y. The term $s_k(y)$ captures the effective service rate that the kth cloudlet offers at location y. Thus, the computation for each task assignment process is of the order $O(|\mathcal{W}|)$.

4.2. Algorithm at the cloudlet side

At the end of the ith time slot, load at the kth cloudlet is represented as $T_k(\rho_k^i)$. Each cloudlet evaluates its load defined as per the following equation

$$T_k(\rho_k^i) = \min\left(\int_{\mathcal{R}} \frac{\gamma(y)}{s_k(y)} u_k(y) dy, 1-\epsilon\right). \tag{7}$$

The cloudlets update their cloudlet load, $T_k(\rho_k^i)$, and use it to calculate the resistance index to be broadcasted to the SDN controller, using the load as per the below equation

$$\rho_k^{i+1} = \xi \rho_k^i + (1-\xi) T_k(\rho_k^i) \tag{8}$$

Here ξ is taken as the averaging factor with $0 < \xi < 1$.

The working of the algorithms on the SDN controller and at the cloudlet is depicted in Fig. 2.

4.3. Convergence of the proposed LATA scheme

This subsection and the next prove the convergence and optimality of the algorithm for task assignment that has been discussed above. Let us first show that our objective function Φ is convex in $\rho \in \tilde{\mathcal{Z}}$. This will ensure that we can have an optimal task assignment, given we minimize the objective function.

Lemma 1. *When the cloudlet load $\boldsymbol{\rho}$ is defined in $\tilde{\mathcal{Z}}$, the objective function $\Phi(\boldsymbol{\rho})$ is observed to be convex in $\boldsymbol{\rho}$.*

Proof. This can be proven by showing $\nabla^2 \Phi(\boldsymbol{\rho}) > 0$.

We can write the objective function as

$$\Phi(\boldsymbol{\rho}) = \sum_{k=1}^{|\mathcal{W}|} \mathcal{L}_k(\rho) = \sum_{k=1}^{|\mathcal{W}|} \frac{\rho_k}{1-\rho_k} \tag{9}$$

The 1st and 2nd order derivatives of the objective function evaluated with respect to $\boldsymbol{\rho}$ are as follows

$$\nabla \Phi(\boldsymbol{\rho}) = \sum_{k=1}^{|\mathcal{W}|} \frac{1}{(1-\rho_k)^2} \tag{10}$$

$$\nabla^2 \Phi(\boldsymbol{\rho}) = \sum_{k=1}^{|\mathcal{W}|} \frac{2}{(1-\rho_k)^3} \tag{11}$$

Since the value of $\frac{2}{(1-\rho_k)^3} > 0$, the above evaluated 2nd order derivative is also non-negative for all cloudlets. Thus, we have shown that our objective function is convex. ■

As proven earlier, the objective function is convex. Thus, we can find an optimal load $\boldsymbol{\rho}^* \in \tilde{\mathcal{Z}}$ which corresponds to a unique optimal task assignment, such that it minimizes the objective function, $\Phi(\boldsymbol{\rho}) = \sum_{k=1}^{|\mathcal{W}|} \mathcal{L}_k(\rho_k)$. Next step to prove is that the proposed task assignment algorithm is convergent. We will make

use of Lemma 2 to 3 (discussed further in the paper) for proving the same. We begin with proving that $T_k(\boldsymbol{\rho}^i)$, and in turn $(\boldsymbol{\rho}^{i+1} - \boldsymbol{\rho}^i)$, yields the direction of descent for $\Phi(\boldsymbol{\rho}^i)$ at $\boldsymbol{\rho}^i$ (as it will be proved in Lemmas 2 and 3). Further in Theorem 1, we will show that the cloudlet load will converge after a few iterations. It will be proved in Theorem 2 that the objective function Φ is minimized with the obtained cloudlet load.

Lemma 2. *Given $\boldsymbol{\rho}^i \neq \boldsymbol{\rho}^*$, $T_k(\boldsymbol{\rho}^i)$ provides the direction of descent for $\Phi(\boldsymbol{\rho}^i)$ at $\boldsymbol{\rho}^i$.*

Proof. From Lemma 1, we know that when ρ is defined in $\tilde{\mathcal{Z}}$, $\Phi(\boldsymbol{\rho})$ is a convex function of ρ. From this, we can easily prove Lemma 2 by showing $\langle \nabla\Phi(\boldsymbol{\rho}^i), T(\boldsymbol{\rho}^i) - \boldsymbol{\rho}^i \rangle \leq 0$ (where $\langle m, l \rangle$ denotes the inner product of the vectors m and l) [63]. For the cloudlet loads ρ_k^i and $T(\rho_k^i)$, let the task assignment indicators be $u_k(y)$ and $u_k^T(y)$. The inner product can be then given as

$$\begin{aligned}
&\langle \nabla\Phi(\boldsymbol{\rho}^i), T(\boldsymbol{\rho}^i) - \boldsymbol{\rho}^i \rangle \\
&= \sum_{k=1}^{|\mathcal{W}|} \frac{1}{(1-\rho_k^i)^2} \left(T_k(\rho_k^i) - \rho_k^i\right) \\
&= \sum_{k=1}^{|\mathcal{W}|} \frac{1}{(1-\rho_k^i)^2} \left(\int_{\mathcal{R}} \frac{\gamma(y)(u_k^T(y) - u_k(y))}{s_k(y)} dy \right) \\
&= \int_{\mathcal{R}} \gamma(y) \sum_{k=1}^{|\mathcal{W}|} \left(\frac{\frac{1}{(1-\rho_k^i)^2}}{s_k(y)} (u_k^T(y) - u_k(y)) \right) dy.
\end{aligned}$$

We can see that

$$\sum_{k=1}^{|\mathcal{W}|} \frac{\frac{1}{(1-\rho_k^i)^2}(u_k^T(y) - u_k(y))}{s_k(y)} \leq 0$$

holds because $u_k^T(y)$ from (6) will maximize the value of $\frac{s_k(y)}{\frac{1}{(1-\rho_k^i)^2}}$.

Therefore, $\langle \nabla\Phi(\boldsymbol{\rho}^i), T(\boldsymbol{\rho}^i) - \boldsymbol{\rho}^i \rangle \leq 0$, thus proving the lemma. ■

Lemma 3. *$(\boldsymbol{\rho}^{i+1} - \boldsymbol{\rho}^i)$ provides descent direction to $\Phi(\boldsymbol{\rho}^i)$.*

Proof. To prove this, we consider the expression given below.

$$\begin{aligned}
\rho_k^{i+1} - \rho_k^i &= \xi \rho_k^i + (1-\xi) T_k(\rho_k^i) - \rho_k^i \\
&= (1-\xi)(T(\rho_k^i) - \rho_k^i). \qquad (12)
\end{aligned}$$

We have earlier seen in Lemma 2 that $(T(\boldsymbol{\rho}^i) - \boldsymbol{\rho}^i)$ is a descent direction of $\Phi(\boldsymbol{\rho}^i)$. Since $0 < \xi < 1$, we can say $(1-\xi) > 0$. Thus, $\boldsymbol{\rho}^{i+1} - \boldsymbol{\rho}^i$ also gives the descent direction of $\Phi(\boldsymbol{\rho}^i)$. ■

Next, we prove that our proposed task assignment scheme is optimal and convergent in Theorems 1 and 2.

Theorem 1. *The cloudlet load vector $\boldsymbol{\rho}$ will converge to $\boldsymbol{\rho}^* \in \mathcal{Z}$ ($\boldsymbol{\rho}^*$ is the optimal cloudlet load vector).*

Proof. Earlier in Lemma 1, it has been shown that $\Phi(\boldsymbol{\rho}^i)$ is convex. Further, Lemma 3 shows that $(\boldsymbol{\rho}^{i+1} - \boldsymbol{\rho}^i)$ gives a descent direction for $\Phi(\boldsymbol{\rho}^i)$. Thus, the convergence of $\Phi(\boldsymbol{\rho}^i)$ to $\boldsymbol{\rho}^*$ can be guaranteed. Let us prove this by contradiction. Suppose that $\Phi(\boldsymbol{\rho}^i)$ does not converge to $\Phi(\boldsymbol{\rho}^*)$, but rather to a different point; then $\boldsymbol{\rho}^{i+1}$ will again give a direction of descent which decreases $\Phi(\boldsymbol{\rho}^i)$(as proven in Lemma 3), which contradicts the assumption of convergence that we began with. Additionally, as $\boldsymbol{\rho}^i$ is derived based on (6) where $u_k(y) \in \{0, 1\}$, $\boldsymbol{\rho}^*$ belongs to set $\mathcal{Z}$. ■

4.4. Optimality of the proposed LATA scheme

We establish the optimality of the proposed scheme through the following Theorem. Note that this theorem uses results of the lemmas proved in the previous subsection.

Theorem 2. *Given a non-empty set $\mathcal{Z}$, and given that the cloudlet load $\boldsymbol{\rho}$ has a convergence in $\boldsymbol{\rho}^*$, the task assignment corresponding to $\boldsymbol{\rho}^*$ minimizes the objective function $\Phi(\boldsymbol{\rho})$.*

Proof. Suppose that the task assignment corresponding to $\boldsymbol{\rho}^*$ is $u^* = \{u_k^*(y) | u_k^*(y) \in \{0, 1\}, \forall k \in \mathcal{W}, \forall y \in \mathcal{R}\}$ and the task assignment corresponding to $\boldsymbol{\rho}$ is $u = \{u_k(y) | u_k(y) \in \{0, 1\}, \forall k \in \mathcal{W}, \forall y \in \mathcal{R}\}$

with $\boldsymbol{\rho} \in \mathcal{Z}$ being the load vector of some cloudlet. We have already seen that $\Phi(\boldsymbol{\rho})$ is convex over $\boldsymbol{\rho}$, and now to prove this theorem, we show that $\langle \nabla\Phi(\boldsymbol{\rho}^*), \boldsymbol{\rho} - \boldsymbol{\rho}^* \rangle \geq 0$. Please note that for the purpose of clarity, we substitute $\frac{\partial\Phi(\boldsymbol{\rho}^*)}{\partial\rho_k^*}$ as $\psi_k(\rho_k^*)$ in the following proof.

$$\begin{aligned}
\langle \nabla\Phi(\boldsymbol{\rho}^*), \boldsymbol{\rho} - \boldsymbol{\rho}^* \rangle &= \sum_{k=1}^{|\mathcal{W}|} \psi_k(\rho_k^*) \left(\boldsymbol{\rho} - \boldsymbol{\rho}^*\right) \\
&= \sum_{k=1}^{|\mathcal{W}|} \left(\int_{\mathcal{R}} \frac{\gamma(y)(u_k(y) - u_k^*(y))}{s_k(y)\psi_k^{-1}(\rho_k^*)} dy \right) \\
&= \int_{\mathcal{R}} \gamma(y) \sum_{k=1}^{|\mathcal{W}|} \frac{(u_k(y) - u_k^*(y))}{s_k(y)\psi_k^{-1}(\rho_k^*)} dy.
\end{aligned}$$

However, we already know that the criterion to choose the optimal offloaded task assignment is as under

$$u_k^*(y) = \begin{cases} 1, & \text{if } k = \arg\max_{k \in \mathcal{W}} \frac{s_k(y)}{\psi_k(\rho_k^*)}, \\ 0, & \text{otherwise.} \end{cases},$$

and thus we can deduce the following equation using the optimal task assignment criterion,

$$\sum_{k=1}^{|\mathcal{W}|} \frac{u_k^*(y)}{s_k(y)\psi_k^{-1}(\rho_k^*)} \leq \sum_{k=1}^{|\mathcal{W}|} \frac{u_k(y)}{s_k(y)\psi_k^{-1}(\rho_k^*)}. \qquad (13)$$

Hence, $\langle \nabla\Phi(\boldsymbol{\rho}^*), \boldsymbol{\rho} - \boldsymbol{\rho}^* \rangle \geq 0$ proving this Theorem. ■

4.5. Admission control

To ensure optimality and convergence of the LATA scheme, the cloudlet assignment problem is required to be feasible. This means that the traffic loads of the cloudlets should lie in the feasible set that has been defined in Definition 1 in Section 3. Note that when the traffic in the network is beyond its capacity to serve, the cloudlet assignment problem ceases to be feasible. Thus, the optimality and convergence property of the LATA scheme does not hold. This necessitates an admission control policy to ensure that the above-mentioned properties still hold in presence of exceedingly high task offload requests in the network. For admission control, let $\theta(y)$ be the coefficient for admission control of user at location y, such that $0 \leq \theta(y) \leq 1$ is the probability of a mobile device at location y getting admittance into the network. The SDN controller assigns $\theta(y)$ for a location y. Note that $\theta(y)$ is not dependent on cloudlet selection. This ensures that the integration of admission control does not change cloudlet selection of the mobile users. The cloudlet serving the user is still evaluated based on Eq. (6). As a result of this admission control, the load of the jth cloudlet is updated in the following way

$$T_k(\rho_k^j) = \min\left(\int_{\mathcal{R}} \theta(y) \frac{\gamma(y)}{s_k(y)} u_k(y) dy, 1 - \epsilon \right), \qquad (14)$$

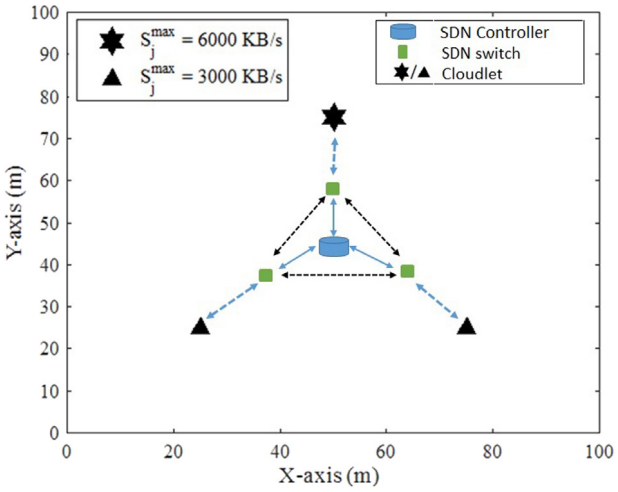


Fig. 3. The topology of the network with three cloudlets used in our simulations.

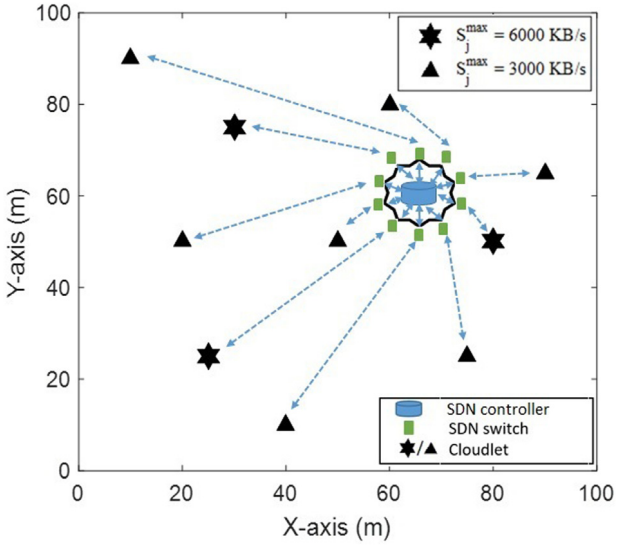


Fig. 4. The topology of the network with ten cloudlets used in our simulations.

with ϵ being an arbitrarily small positive constant.

The cloudlet updates its load based on Eq. (8). Thus, the SDN controller restricts the loads of the cloudlets to ensure that the cloudlet assignment problem remains feasible. For this admission control, the relaxed feasible set becomes

$$\tilde{\mathcal{Z}} = \left\{ \boldsymbol{\rho} | \rho_k = \int_{\mathcal{R}} \theta(y) \frac{\gamma(y)}{s_k(y)} u_k(y) dy,\ 0 \le \rho_k \le 1 - \epsilon,\ \forall k \in \mathcal{W},\right.$$
$$\left. 0 \le u_k(y) \le 1, \sum_{k=1}^{|\mathcal{K}|} u_k(y) = 1,\ \ \forall k \in \mathcal{W},\ \ \forall y \in \mathcal{R} \right\}.$$

As $0 \le \theta(y) \le 1$ is constant, Lemma 1 now also holds, and thus the set remains convex. Further as the integration of admission control does not modify the objective problem of clouldet assignment, thus Lemma 2 still stands true. This ensures that convergence and optimality proofs given previously still hold, thus enabling the traffic load to converge to the optimal solution even with admission control applied. The effect of variation in this parameter θ on the latency will be analyzed in Section 5. We now briefly summarize the salient differences between the approach taken by LATA and other approaches to reduce the latency in processing the offloaded tasks by the mobile devices.

4.6. Salient differences between LATA and other task assignment schemes

We summarize the features of LATA that help LATA achieve lower latency than the existing schemes.

- LATA considers a network of cloudlet devices to serve the offloaded request instead of a single cloudlet device considered by schemes like [7,64,65]. Thus, in LATA, any request offloaded onto a cloudlet device can be serviced by any other cloudlet device if the overall latency in servicing the requests is reduced by doing so.
- While identifying a cloudlet device to service the request, LATA considers the capabilities of a node, i.e., maximum effective service rate S_{max}, the distance of the cloudlet device from the mobile device and the file size of the offloaded data.
- Further, LATA also considers the current load of every cloudlet which is reflected by the value of ψ_k. The existing schemes like LEAN [7] and the scheme proposed by Mukherjee et al. [8] do not consider the current load, thus making non-optimal task offloading decisions which increases the latency. This will be shown in the simulations in the next section.

5. Numerical results

We consider two different cloudlet topologies (shown in Figs. 3 and 4) with three and ten cloudlets respectively, to carry out the performance analysis of the proposed scheme. Note that these two different topologies have been considered for the performance analysis for different network sizes. As it can be seen from Fig. 3, there are three cloudlets which provide service in an area of 100 m $\times$ 100 m whereas in Fig. 4 there are ten cloudlets providing service in the given area. The cloudlets are randomly placed within the network. It is to be noted that the SDN switched network with the SDN Controller at the core connects the different cloudlets in the network. Two kinds of cloudlets are considered here: one with its maximum effective service rate being 3000 KB/s and the other having maximum effective service rate as 6000 KB/s. For generating the mobile task offload requests, homogeneous Poisson point process (HPPP) is used. The performance is analyzed for various task arrival rates with the lowest traffic being 20 offload requests arriving in the network per second in the coverage area. The maximum traffic arrival rate has been taken to be 80 requests/s for the three-cloudlet network scenario and 160 requests/s for the 10 cloudlet network scenario. Note that we have taken the different maximum traffic arrival rates (of 80 and 160 requests/s) for these networks considering their traffic handling capabilities. Also, *dis*(.) denotes the distance of the cloudlet from the mobile device in *km*. The parameter d_0 has been taken as 15 m (i.e., 0.015 km). The offload requests are considered here to range from 10 KB/packet to 70 KB/packet by taking into account of the different computational demands of the users (e.g., 70 KB/packet traffic offloaded is from users which are requesting more computationally intensive tasks to be performed like 3D gaming whereas those with 10 KB/packet have lesser computationally intensive tasks offloaded like an application requesting to zip the streamed files into a folder). The location based task offload density ($\gamma(y)$) is calculated based on the model discussed in Section 3. The value of the averaging factor ξ (mentioned in (8)) is taken to be 0.95. Keeping this value of ξ, we note that the task assignment algorithm converges within 10 iterations. The performance of the proposed scheme has been

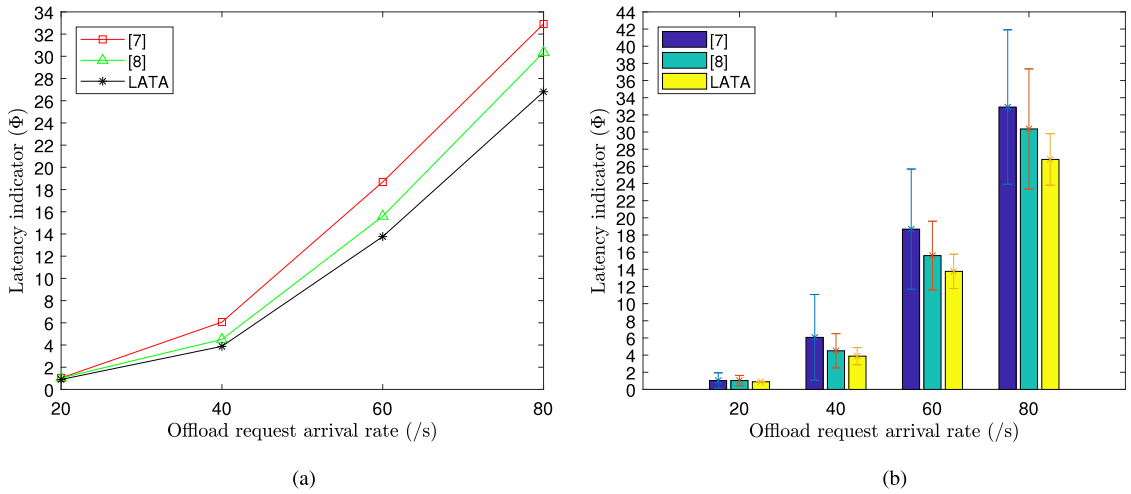


Fig. 5. (a) Latency variation for varying offload request arrival rates in a network of three cloudlets. (b) Latency variation for varying offload request arrival rates in a network of three cloudlets with confidence intervals of 95%.

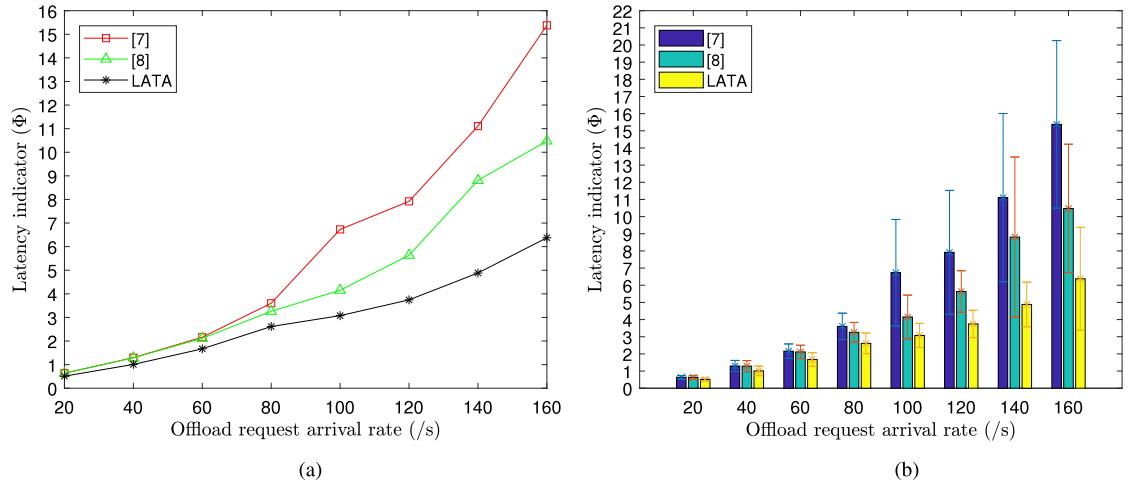


Fig. 6. (a) Latency variation for varying offload request arrival rates in a network of ten cloudlets. (b) Latency variation for varying offload request arrival rates in a network of ten cloudlets with confidence intervals of 95%.

compared to the state-of-the-art schemes, viz., LEAN [7] and a scheme proposed in [8], which have been briefly discussed in Section 2.

5.1. Latency performance with traffic variation

We study how the network latency performance changes when the request arrival rates of the offloaded traffic are varied. The arrival rates of the offloaded requests range from 20 to 80 offload request/s for the three-cloudlet network. We calculate the latency indicator for the whole network (Φ) corresponding to these arrival rates of requests. As stated in Section 3, this latency indicator is a unit-less quantity. The numerical values of α has been taken as 1 for the results in this section. However the effect of α on latency has been studied in the next subsection. The results pertaining to the network having three-cloudlets are shown in Fig. 5 whereas those pertaining to network of ten-cloudlets are shown in Fig. 6. For the three-cloudlets network, we observe from Fig. 5(a) that as the traffic load increases, the latency increases in all three schemes; this is due to a higher load on the cloudlets. It can be seen from the results that the LATA scheme proposed in this paper has better performance than the schemes proposed in [8] and [7] since it achieves lower latency. Further, we also see that performance gain of LATA over the schemes in [7] and [8] increases as the traffic is increased. For example, in the three-cloudlet scenario, for an average traffic arrival rate of 80 requests/s the latency for the scheme in [7] is 13% higher and for [8] is 10% higher as compared to LATA. We have estimated confidence intervals of 95% for the latency at each request arrival rate. The variation of latency of each scheme with confidence intervals of 95% with varying network traffic for the three-cloudlet network is shown in Fig. 5(b). From this figure, we observe that as the traffic increases in the network, the confidence intervals become longer. This is due to the large variation in the latencies during high traffic conditions. Confidence interval range for the latency of the LATA scheme is also observed to be smaller than those of other schemes, for each traffic arrival rate.

The behavior of the ten-cloudlet network has been analyzed similarly and the results are shown in Fig. 6. For the ten-cloudlet network, the arrival rates have been varied from 20 requests/s to 160 requests/s. From Fig. 5(a), we observe that the variation in the latency with increase in arrival rates in the ten-cloudlet network has a trend similar to that observed in the three-cloudlet network. In the ten cloudlet scenario, LATA achieves the lowest latency. In this network, for average traffic arrival rate of 160 requests/s, the latency is 140% higher for [7] and 64% higher for [8] as compared to the LATA scheme. The reasoning for this behavior is as follows. For the case of very high offload arrival rate, the schemes in [7]

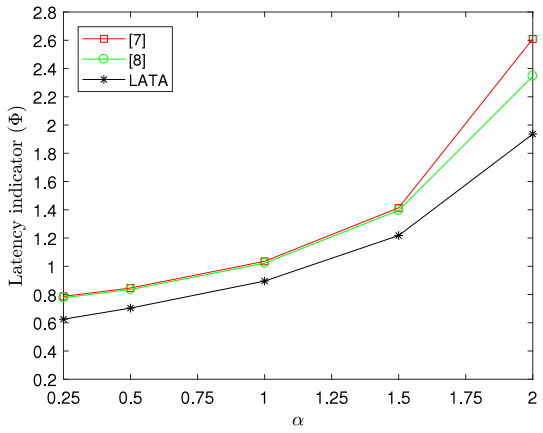


Fig. 7. Latency behavior with different alpha values for a network with three cloudlets and traffic of 20 requests/s.

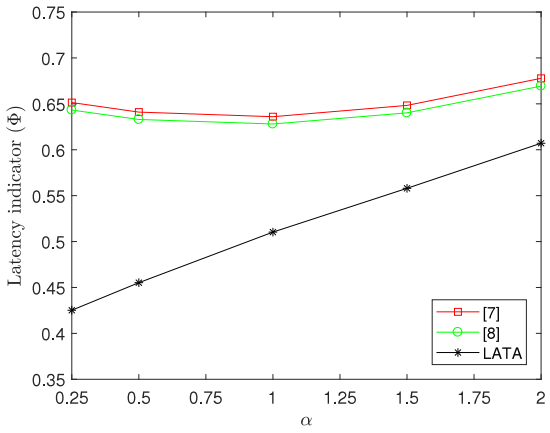


Fig. 9. Latency behavior with different alpha values for a network with ten cloudlets and traffic of 20 requests/s.

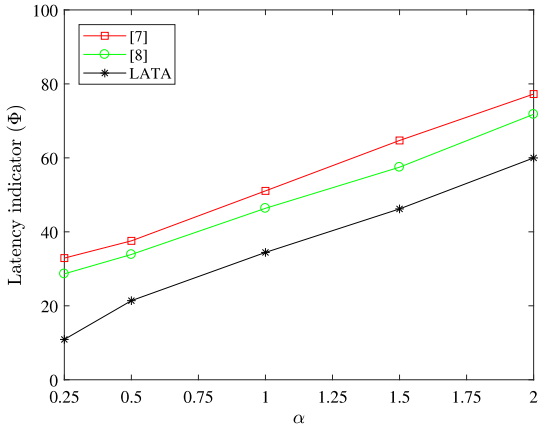


Fig. 8. Latency behavior with different alpha values for a network with three cloudlets and traffic of 80 requests/s.

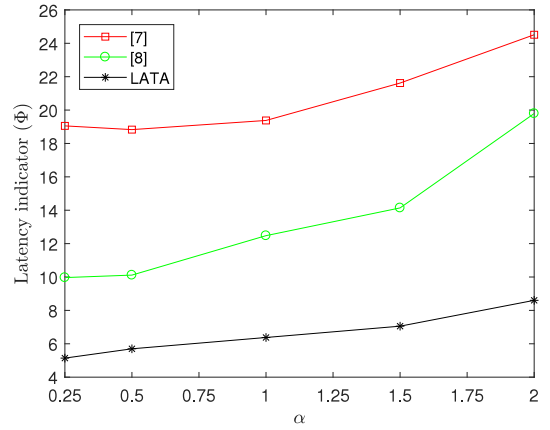


Fig. 10. Latency behavior with different alpha values for a network with ten cloudlets and traffic of 160 requests/s.

and [8] prefer offloading many of the requests to the cloudlets closest to them which leads to some of the cloudlets getting overloaded (which have more number of users near them). This causes an increase in the overall latency of the network. For the scheme proposed in [8], the task is assigned to the nearest cloudlet if it is not overloaded, else it assigns the task to the remote cloudlet which is able to serve the task with least latency. Note that the model proposed by them does not take into consideration of the current load on the remote cloudlet while making the task offload decisions, and thus is unable to optimize the network level latency. However, in case of LATA, in addition to the distance of the cloudlet from the request, the current load of the cloudlets is considered for making the task assignment decisions. This gives an improved latency on account of its load balancing nature.The variation of latency of each scheme with varying network traffic with confidence intervals of 95% for the ten-cloudlet network is shown in Fig. 6(b). From this figure, we observe that as the traffic increases in the network, the confidence intervals become longer just as we have observed for the three-cloudlet network. Also similar to the observation for the three-cloudlet network, the confidence interval range for the latency of the LATA scheme is smaller than those of other schemes, for each traffic arrival rate.

5.2. Latency performance with varying network parameters

In this section, we study the latency performance as the network parameter α is varied. In the formulation shown in (1), the parameter α captures the quality of the network in terms of the latency associated with the distance of the requests from the cloudlet. With other parameters fixed, a higher value of α would indicate that the cloudlet provides a lower effective service rate at the location where the request originated. We vary α from 0.25 to 2 for the two different network scenarios (three-cloudlet and ten-cloudlet networks) and study the performance of the system. The analysis is done considering the two extreme values of the request rate, i.e., the minimum and the maximum requests/second. We fix the value of traffic to a certain value and see the latency performance on varying the network parameter α. Note that a higher value of this parameter (α) indicates a slower system. In Figs. 7 and 8, we show the change in the latency as α is varied for an average traffic of 20 request arrivals/s and 80 request arrivals/s respectively for the three cloudlet network. Similarly, Figs. 9 and 10 show the change in the latency for an average traffic of 20 and 160 request arrivals/s respectively, for the ten-cloudlet network. Note that in the three-cloudlet network, for its maximum traffic (of 80 requests/s), the latency indicator's value for LATA increases from around 10 to nearly 60 (increase by 6 times) as α varies from 0.25 to 2. However, in the ten-cloudlet network for the maximum

traffic (of 160 requests/s), the latency indicator increases from 5 to 8.5 (increase by 1.7 times only). From this we conclude that the increase in the latency with an increase in α is less emphatic for the ten-cloudlet network as compared to the three-cloudlet network. This is because the requests generated in the network are served by a greater number of cloudlets in the ten-cloudlet network. Note that in all cases we find that LATA outperforms the schemes proposed in [7] and [8] because of making optimal cloudlet assignment based on the effective service rate, distance of the cloudlet from the user offloading the task as well as the current load on the cloudlet.

5.3. Latency performance with admission control

We analyze the latency performance when the admission control parameter θ is varied. As mentioned in the previous section (Section 4.5), the parameter $\theta(y)$ depicts the probability with which a mobile device at a location y is admitted into the network. For example, $\theta(y)$ =1 would indicate that all the offload requests arriving in the network at location y are admitted, whereas $\theta(y)$ = 0.7 would indicate that requests arriving at that location are allowed admittance into the network with a probability 0.7 (in-turn denied admittance with probability 0.3). We have considered this parameter to be the same for the entire area served by the cloudlets (i.e. for all y). We vary this parameter while keeping the other parameters constant. For this analysis, we keep the parameter α mentioned in (1) as unity. We performed the simulations for both three-cloudlet and ten-cloudlet networks for different arrival rates. For a three-node network, we simulated the LATA algorithm for different average arrival rates varying from 80 requests/s to 140 requests/s and the results are depicted in Fig. 11. Note that for higher arrival request rates (i.e. 120 requests/s and 140 requests/s), when the admission probability is high (> 0.7), the latency becomes unacceptably high. This indicates that the network cannot manage such high traffic. However, if the network has such high traffic, the admission control parameter can be reduced to enable lower latency for the served users (this is however at the cost of some of the users being dropped/denied admission into the network). For example, when the arrival rate is 140 requests/s and θ is set as 0.5, the latency indicator for the users being served is limited to 21 (at the cost of around 50% of the requests being dropped).

The latency performance for the ten cloudlet network for varying values of θ for average arrival rates varying from 160 requests/s to 400 requests/sec is depicted in Fig. 12. Note that when the offload request arrival rate is low, admission control is not required and the θ value can be kept as 1. However, when the traffic is high, admission control can be applied to reduce the latency experienced by the users served by the cloudlet network.

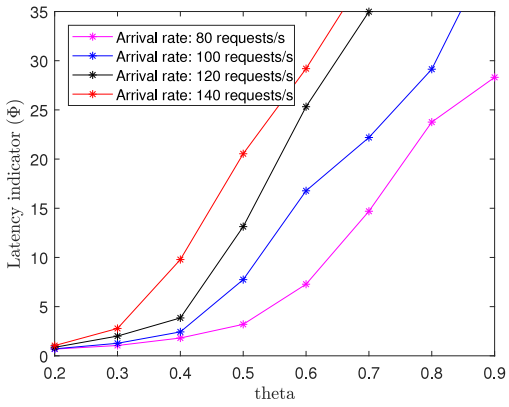


Fig. 11. Latency behavior for different admission control parameter values for a network with three cloudlets.

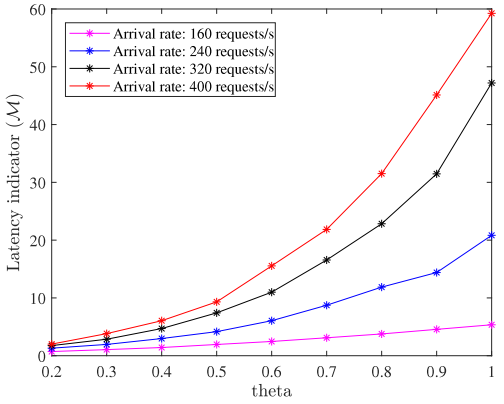


Fig. 12. Latency behavior for different admission control parameter values for a network with ten cloudlets.

6. Conclusion and future work

This paper proposes a novel task assignment scheme (named LATA) which makes task assignment decisions for a network of cloudlets, which serve computationally intensive tasks that have been offloaded by the mobile devices in their coverage area. The task assignment aims at reducing the network latency in processing the offloaded tasks thereby enhancing the QoS experienced by the mobile users served by the cloudlets. This scheme takes into account all the significant parameters which affect the latency in processing the task requests. Some of these parameters like current load of the cloudlet devices is often ignored in other existing schemes. Further, we have mathematically proved that LATA achieves optimal latency. Through simulation results, we showed how LATA gives a better performance in comparison to the existing schemes. We also proposed an admission control policy to ensure that the task assignment scheme remains optimal even when the traffic in the network is more than what it can handle. To the best of our knowledge, this is the first task assignment scheme for edge cloudlets which specifically handles extremely high task request rates and maintaining the optimality of the task assignment scheme.

As an extension of this work, end devices with different bounded delay requirements may be considered. We also plan to investigate how to apply LATA for recently proposed hierarchical mobile edge computing [66,67]. This work can be further extended to make latency aware and energy aware task assignment decisions especially for energy constrained cloudlet networks like the green edge cloudlet network proposed in [68]. The current work can also be extended to explore latency-aware task assignment in mobile edge cloudlet network in which the cloudlets are mobile. The current work has also assumed that the mobile devices offloading the data remain connected to the same cloudlet (on to which it has offloaded a task). We have to modify the current model when this assumption is not valid, i.e., when the mobile device gets connected to a different cloudlet before the task computations are completed.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This research/project was supported by DST-SERB, India funding ECR/2018/001479. It was also partially supported by the National Research Foundation, Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme.

Appendix

Let us consider the effective service rate that the cloudlet k offers to the mobile device located at y as $s_k(y)$. Note that the cloudlets serve the users through wireless links. The effective service rate experienced by the user is proportional to the Shannon's rate offered by the cloudlet. Thus, we have the following relationship

$$s_k(y) \propto B * log_2(1 + SNR) \tag{15}$$

$$\propto \frac{B}{ln\,2} * ln\left(1 + \frac{p_{max}}{N * (1 + (dis(G_k, y)/d_0)^{\alpha})}\right) \tag{16}$$

In the above equations, B is the bandwidth and p_{max} is the transmitted signal power. $dis(G_k, y)$ is the Euclidean distance of the kth cloudlet from the mobile device that is at location y. To accommodate path loss in the wireless media, the parameter α is used to model different network scenarios. d_0 is the scaling factor for the distance. Note that these expressions are customary in wireless networks. Note that when x is very small, $ln(1+x) \approx x$. Thus, (16) becomes

$$s_k(y) \propto B * \frac{p_{max}}{N * (1 + (dis(G_k, y)/d_0)^{\alpha})} \tag{17}$$

$$\implies s_k(y) \propto \frac{B * p_{max}}{N} * \frac{1}{1 + (dis(G_k, y)/d_0)^{\alpha}} \tag{18}$$

Further, the effective service rate is also proportional to the computational speed of the cloudlet. Let $\mathcal{C}(y)$ denote the computational speed of the cloudlet (given in kbps). Then, we can say that

$$s_k(y) \propto \mathcal{C}(y) \tag{19}$$

From (18) and (19), we can write the expression of the effective service rate experienced by the user as follows

$$s_k(y) = \mathcal{K} * \mathcal{C}(y) * \frac{B * p_{max}}{N} * \frac{1}{1 + (dis(G_k, y)/d_0)^{\alpha}} \tag{20}$$

where $\mathcal{K}$ is a constant. Further on re-arranging this equation, we get the following:

$$s_k(y) = \frac{B * p_{max} * \mathcal{K} * \mathcal{C}(y)}{N} * \frac{1}{1 + (dis(G_k, y)/d_0)^{\alpha}} \tag{21}$$

Denoting the expression $\frac{B*p_{max}*\mathcal{K}*\mathcal{C}(y)}{N}$ as $S_k^{max}(y)$, we get the expression for the effective service rate as follows

$$s_k(y) = \frac{S_k^{max}(y)}{1 + (dis(G_k, y)/d_0)^{\alpha}} \tag{22}$$

where $S_k^{max}(y)$ is the maximum effective service rate offered by cloudlet y.

References

[1] E.F. Nakamura, A.A.F. Loureiro, A.C. Frery, Information fusion for wireless sensor networks: methods, models, and classifications, ACM Comput. Surv. 39 (3) (2007) http://dx.doi.org/10.1145/1267070.1267073.

[2] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, A. Patti, Clonecloud: elastic execution between mobile device and cloud, in: Proceedings of the Sixth Conference on Computer Systems, in: EuroSys '11, ACM, New York, NY, USA, 2011, pp. 301–314, http://dx.doi.org/10.1145/1966445.1966473.

[3] M. Satyanarayanan, P. Bahl, R. Caceres, N. Davies, The case for VM-based cloudlets in mobile computing, IEEE Pervasive Comput. 8 (4) (2009) 14–23, http://dx.doi.org/10.1109/MPRV.2009.82.

[4] Z. Pang, L. Sun, Z. Wang, E. Tian, S. Yang, A survey of cloudlet based mobile computing, in: 2015 International Conference on Cloud Computing and Big Data, CCBD, 2015, pp. 268–275, http://dx.doi.org/10.1109/CCBD.2015.54.

[5] S. Kosta, A. Aucinas, P. Hui, R. Mortier, X. Zhang, ThinkAir: dynamic resource allocation and parallel execution in the cloud for mobile code offloading, in: 2012 Proceedings IEEE INFOCOM, 2012, pp. 945–953, http://dx.doi.org/10.1109/INFCOM.2012.6195845.

[6] M. Jia, W. Liang, Z. Xu, M. Huang, Cloudlet load balancing in wireless metropolitan area networks, in: IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications, 2016, pp. 1–9, http://dx.doi.org/10.1109/INFOCOM.2016.7524411.

[7] X. Sun, N. Ansari, Latency aware workload offloading in the cloudlet network, IEEE Commun. Lett. 21 (7) (2017) 1481–1484, http://dx.doi.org/10.1109/LCOMM.2017.2690678.

[8] A. Mukherjee, D. De, D.G. Roy, A power and latency aware cloudlet selection strategy for multi-cloudlet environment, IEEE Trans. Cloud Comput. 7 (1) (2019) 141–154, http://dx.doi.org/10.1109/TCC.2016.2586061.

[9] M. Satyanarayanan, Fundamental challenges in mobile computing, in: Proceedings of the Fifteenth Annual ACM Symposium on Principles of Distributed Computing, in: PODC '96, ACM, New York, NY, USA, 1996, pp. 1–7, http://dx.doi.org/10.1145/248052.248053.

[10] H. Qi, A. Gani, Research on mobile cloud computing: review, trend and perspectives, in: Digital Information and Communication Technology and it's Applications, DICTAP, 2012 Second International Conference on, 2012, pp. 195–202, http://dx.doi.org/10.1109/DICTAP.2012.6215350.

[11] M.V. Barbera, S. Kosta, A. Mei, V.C. Perta, J. Stefa, Mobile offloading in the wild: findings and lessons learned through a real-life experiment with a new cloud-aware system, in: IEEE INFOCOM 2014 - IEEE Conference on Computer Communications, 2014, pp. 2355–2363, http://dx.doi.org/10.1109/INFOCOM.2014.6848180.

[12] Z. Sanaei, S. Abolfazli, A. Gani, R. Buyya, Heterogeneity in mobile cloud computing: taxonomy and open challenges, IEEE Commun. Surv. Tutor. 16 (1) (2014) 369–392, http://dx.doi.org/10.1109/SURV.2013.050113.00090.

[13] S.S. Qureshi, T. Ahmad, K. Rafique, S. ul islam, Mobile cloud computing as future for mobile applications - Implementation methods and challenging issues, in: 2011 IEEE International Conference on Cloud Computing and Intelligence Systems, 2011, pp. 467–471, http://dx.doi.org/10.1109/CCIS.2011.6045111.

[14] Y. Jararweh, L. Tawalbeh, F. Ababneh, F. Dosari, Resource efficient mobile computing using cloudlet infrastructure, in: 2013 IEEE 9th International Conference on Mobile Ad-hoc and Sensor Networks, 2013, pp. 373–377, http://dx.doi.org/10.1109/MSN.2013.75.

[15] F. Bonomi, R. Milito, J. Zhu, S. Addepalli, Fog computing and its role in the internet of things, in: Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, in: MCC '12, ACM, New York, NY, USA, 2012, pp. 13–16, http://dx.doi.org/10.1145/2342509.2342513.

[16] I. Stojmenovic, Fog computing: A cloud to the ground support for smart things and machine-to-machine networks, in: 2014 Australasian Telecommunication Networks and Applications Conference, ATNAC, 2014, pp. 117–122, http://dx.doi.org/10.1109/ATNAC.2014.7020884.

[17] A. Ahmed, E. Ahmed, A survey on mobile edge computing, in: 2016 10th International Conference on Intelligent Systems and Control, ISCO, 2016, pp. 1–8, http://dx.doi.org/10.1109/ISCO.2016.7727082.

[18] T. Taleb, S. Dutta, A. Ksentini, M. Iqbal, H. Flinck, Mobile edge computing potential in making cities smarter, IEEE Commun. Mag. 55 (3) (2017) 38–43, http://dx.doi.org/10.1109/MCOM.2017.1600249CM.

[19] X. Sun, N. Ansari, EdgeIoT: mobile edge computing for the internet of things, IEEE Commun. Mag. 54 (12) (2016) 22–29, http://dx.doi.org/10.1109/MCOM.2016.1600492CM.

[20] K. Sasaki, N. Suzuki, S. Makido, A. Nakao, Vehicle control system coordinated between cloud and mobile edge computing, in: 2016 55th Annual Conference of the Society of Instrument and Control Engineers of Japan, SICE, 2016, pp. 1122–1127, http://dx.doi.org/10.1109/SICE.2016.7749210.

[21] T. Soyata, R. Muraleedharan, C. Funai, M. Kwon, W. Heinzelman, Cloud-Vision: real-time face recognition using a mobile-cloudlet-cloud acceleration architecture, in: 2012 IEEE Symposium on Computers and Communications, ISCC, 2012, pp. 000059–000066, http://dx.doi.org/10.1109/ISCC.2012.6249269.

[22] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, D. Sabella, On multi-access edge computing: a survey of the emerging 5G network edge cloud architecture and orchestration, IEEE Commun. Surv. Tutor. 19 (3) (2017) 1657–1681, http://dx.doi.org/10.1109/COMST.2017.2705720.

[23] Y. Mao, C. You, J. Zhang, K. Huang, K.B. Letaief, A survey on mobile edge computing: the communication perspective, IEEE Commun. Surv. Tutor. 19.4 (2017) 2322–2358, http://dx.doi.org/10.1109/COMST.2017.2745201.

[24] W. Yu, F. Liang, X. He, W.G. Hatcher, C. Lu, J. Lin, X. Yang, A survey on the edge computing for the internet of things, IEEE Access 6 (2018) 6900–6919, http://dx.doi.org/10.1109/ACCESS.2017.2778504.

[25] P. Mach, Z. Becvar, Mobile edge computing: a survey on architecture and computation offloading, IEEE Commun. Surv. Tutor. 19 (3) (2017) 1628–1656, http://dx.doi.org/10.1109/COMST.2017.2682318.

[26] W.Z. Khan, E. Ahmed, S. Hakak, I. Yaqoob, A. Ahmed, Edge computing: a survey, Future Gener. Comput. Syst. 97 (2019) 219–235, http://dx.doi.org/10.1016/j.future.2019.02.050, http://www.sciencedirect.com/science/article/pii/S0167739X18319903.

[27] Y. Mao, C. You, J. Zhang, K. Huang, K.B. Letaief, Mobile edge computing: survey and research outlook, 2017, arXiv preprint arXiv:1701.01090.

[28] Y. Mao, C. You, J. Zhang, K. Huang, K.B. Letaief, A survey on mobile edge computing: the communication perspective, IEEE Commun. Surveys Tuts. 19 (4) (2017) 2322–2358, http://dx.doi.org/10.1109/COMST.2017.2745201.

[29] X. Sun, N. Ansari, Mobile edge computing empowers internet of things, IEICE Trans. Commun. E101-B (3) (2018) 604–619, http://dx.doi.org/10.1587/transcom.2017NRI0001.

[30] K. Kumar, J. Liu, Y.-H. Lu, B. Bhargava, A survey of computation offloading for mobile systems, Mob. Netw. Appl. 18 (1) (2013) 129–140, http://dx.doi.org/10.1007/s11036-012-0368-0.

[31] Y. Mao, J. Zhang, S. Song, K.B. Letaief, Stochastic joint radio and computational resource management for multi-user mobile-edge computing systems, IEEE Trans. Wireless Commun. 16 (9) (2017) 5994–6009.

[32] X. Zhang, Y. Mao, J. Zhang, K.B. Letaief, Multi-objective resource allocation for mobile edge computing systems, in: 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications, PIMRC, IEEE, 2017, pp. 1–5.

[33] Y. Mao, J. Zhang, K.B. Letaief, Joint task offloading scheduling and transmit power allocation for mobile-edge computing systems, in: 2017 IEEE Wireless Communications and Networking Conference, WCNC, IEEE, 2017, pp. 1–6.

[34] C. Wang, C. Liang, F.R. Yu, Q. Chen, L. Tang, Computation offloading and resource allocation in wireless cellular networks with mobile edge computing, IEEE Trans. Wireless Commun. 16 (8) (2017) 4924–4938, http://dx.doi.org/10.1109/TWC.2017.2703901.

[35] C. Wang, F.R. Yu, C. Liang, Q. Chen, L. Tang, Joint computation offloading and interference management in wireless cellular networks with mobile edge computing, IEEE Trans. Veh. Technol. 66 (8) (2017) 7432–7445, http://dx.doi.org/10.1109/TVT.2017.2672701.

[36] J. Plachy, Z. Becvar, E.C. Strinati, Dynamic resource allocation exploiting mobility prediction in mobile edge computing, in: 2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications, PIMRC, 2016, pp. 1–6, http://dx.doi/org/10.1109/PIMRC.2016.7794955.

[37] S. Wang, R. Urgaonkar, T. He, K. Chan, M. Zafer, K.K. Leung, Dynamic service placement for mobile micro-clouds with predicted future costs, IEEE Trans. Parallel Distrib. Syst. 28 (4) (2017) 1002–1016, http://dx.doi.org/10.1109/TPDS.2016.2604814.

[38] G. Huerta-Canepa, D. Lee, A virtual cloud computing provider for mobile devices, in: Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond, in: MCS'10, ACM, New York, NY, USA, 2010, pp. 6:1–6:5, http://dx.doi.org/10.1145/1810931.1810937.

[39] K. Zhang, Y. Mao, S. Leng, Q. Zhao, L. Li, X. Peng, L. Pan, S. Maharjan, Y. Zhang, Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks, IEEE Access 4 (2016) 5896–5907, http://dx.doi.org/10.1109/ACCESS.2016.2597169.

[40] S. Sardellitti, G. Scutari, S. Barbarossa, Joint optimization of radio and computational resources for multicell mobile-edge computing, IEEE Trans. Signal Inf. Process. Netw. 1 (2) (2015) 89–103, http://dx.doi.org/10.1109/TSIPN.2015.2448520.

[41] O.M. noz, A. Pascual-Iserte, J. Vidal, Optimization of radio and computational resources for energy efficiency in latency-constrained application offloading, IEEE Trans. Veh. Technol. 64 (10) (2015) 4738–4755, http://dx.doi.org/10.1109/TVT.2014.2372852.

[42] C. You, K. Huang, H. Chae, B.H. Kim, Energy-efficient resource allocation for mobile-edge computation offloading, IEEE Trans. Wireless Commun. 16.3 (2017) 1397–1411, http://dx.doi.org/10.1109/TWC.2016.2633522.

[43] Y.H. Kao, B. Krishnamachari, M.R. Ra, F. Bai, Hermes: latency optimal task assignment for resource-constrained mobile computing, IEEE Trans. Mob. Comput. 16 (11) (2017) 3056–3069, http://dx.doi.org/10.1109/TMC.2017.2679712.

[44] Q. Fan, N. Ansari, Workload allocation in hierarchical cloudlet networks, IEEE Commun. Lett. 22 (4) (2018) 820–823, http://dx.doi.org/10.1109/LCOMM.2018.2801866.

[45] W. Zhang, Z. Zhang, S. Zeadally, H. Chao, V. Leung, Masm: a multiple-algorithm service model for energy-delay optimization in edge artificial intelligence, IEEE Trans. Ind. Inf. (2019) 1, http://dx.doi.org/10.1109/TII.2019.2897001.

[46] R. Mahmud, K. Ramamohanarao, R. Buyya, Latency-aware application module management for fog computing environments, ACM Trans. Internet Technol. 19 (1) (2018) 9:1–9:21, http://dx.doi.org/10.1145/3186592, http://doi.acm.org/10.1145/3186592.

[47] K. Zhang, Y. Mao, S. Leng, A. Vinel, Y. Zhang, Delay constrained offloading for Mobile Edge Computing in cloud-enabled vehicular networks, in: 2016 8th International Workshop on Resilient Networks Design and Modeling, RNDM, 2016, pp. 288–294, http://dx.doi.org/10.1109/RNDM.2016.7608300.

[48] G. Qiao, S. Leng, K. Zhang, Y. He, Collaborative task offloading in vehicular edge multi-access networks, IEEE Commun. Mag. 56 (8) (2018) 48–54, http://dx.doi.org/10.1109/MCOM.2018.1701130.

[49] Y. Mao, J. Zhang, K.B. Letaief, Dynamic computation offloading for mobile-edge computing with energy harvesting devices, IEEE J. Sel. Areas Commun. 34.12 (2016) 3590–3605, http://dx.doi.org/10.1109/JSAC.2016.2611964.

[50] Y. Zhang, D. Niyato, P. Wang, C.K. Tham, Dynamic offloading algorithm in intermittently connected mobile cloudlet systems, in: 2014 IEEE International Conference on Communications, ICC, 2014, pp. 4190–4195, http://dx.doi.org/10.1109/ICC.2014.6883978.

[51] T. Truong-Huu, C.K. Tham, D. Niyato, To offload or to wait: an opportunistic offloading algorithm for parallel tasks in a mobile cloud, in: 2014 IEEE 6th International Conference on Cloud Computing Technology and Science, 2014, pp. 182–189, http://dx.doi.org/10.1109/CloudCom.2014.33.

[52] M. Tiwary, D. Puthal, K.S. Sahoo, B. Sahoo, L.T. Yang, Response time optimization for cloudlets in mobile edge computing, J. Parallel Distrib. Comput. 119 (2018) 81–91, http://dx.doi.org/10.1016/j.jpdc.2018.04.004, http://www.sciencedirect.com/science/article/pii/S0743731518302430.

[53] J. Yao, N. Ansari, QoS-aware fog resource provisioning and mobile device power control in IoT networks, IEEE Trans. Netw. Serv. Manag. 16 (1) (2019) 167–175, http://dx.doi.org/10.1109/TNSM.2018.2888481.

[54] J. Yao, N. Ansari, Reliability-aware fog resource provisioning for deadline-driven IoT services, in: 2018 IEEE Global Communications Conference, GLOBECOM, 2018, pp. 1–6, http://dx.doi.org/10.1109/GLOCOM.2018.8647378.

[55] D. Lee, S. Zhou, X. Zhong, Z. Niu, X. Zhou, H. Zhang, Spatial modeling of the traffic density in cellular networks, IEEE Wirel. Commun. 21 (1) (2014) 80–88, http://dx.doi.org/10.1109/MWC.2014.6757900.

[56] Cisco catalyst 3650, 10 Gbps wireless SDN switch from cisco, https://www.cisco.com/c/en/us/products/collateral/switches/catalyst-3650-series-switches/data_sheet-c78-729449.html. (Accessed 02 November 2018).

[57] Cut-through and store-and-forward ethernet switching for low-latency environments, https://www.cisco.com/c/en/us/products/collateral/switches/nexus-5020-switch/white_paper_c11-465436.html. (Accessed 20 June 2019).

[58] D. Liu, Y. Chen, K.K. Chai, T. Zhang, Distributed delay-energy aware user association in 3-tier HetNets with hybrid energy sources, in: 2014 IEEE Globecom Workshops (GC Wkshps), 2014, pp. 1109–1114, http://dx.doi.org/10.1109/GLOCOMW.2014.7063581.

[59] L. Kleinrock, Queueing Systems vol. 2: Computer Applications, Wiley-Interscience, New York, U.S.A., 1976.

[60] T. Han, N. Ansari, Powering mobile networks with green energy, IEEE Wirel. Commun. 21 (1) (2014) 90–96, http://dx.doi.org/10.1109/MWC.2014.6757901.

[61] D. Liu, Y. Chen, K.K. Chai, T. Zhang, M. Elkashlan, Two-dimensional optimization on user association and green energy allocation for hetnets with hybrid energy sources, IEEE Trans. Commun. 63 (11) (2015) 4111–4124, http://dx.doi.org/10.1109/TCOMM.2015.2470659.

[62] H. Kim, G. de Veciana, X. Yang, M. Venkatachalam, Distributed *alpha* -optimal user association and cell load balancing in wireless networks, IEEE/ACM Trans. Netw. 20 (1) (2012) 177–190, http://dx.doi.org/10.1109/TNET.2011.2157937.

[63] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge Univ. Press, Cambridge, U.K., 2004.

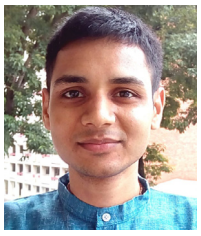
[64] J. Ren, G. Yu, Y. Cai, Y. He, Latency optimization for resource allocation in mobile-edge computation offloading, IEEE Trans. Wireless Commun. 17 (8) (2018) 5506–5519, http://dx.doi.org/10.1109/TWC.2018.2845360.

[65] Y. Liu, M.J. Lee, Y. Zheng, Adaptive multi-resource allocation for cloudlet-based mobile cloud computing system, IEEE Trans. Mob. Comput. 15 (10) (2016) 2398–2410, http://dx.doi.org/10.1109/TMC.2015.2504091.

[66] E.E. Haber, T.M. Nguyen, D. Ebrahimi, C. Assi, Computational cost and energy efficient task offloading in hierarchical edge-clouds, in: 2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC, 2018, pp. 1–6, http://dx.doi.org/10.1109/PIMRC.2018.8580724.

[67] A. Kiani, N. Ansari, Toward hierarchical mobile edge computing: an auction-based profit maximization approach, IEEE Internet Things J. 4 (6) (2017) 2082–2091, http://dx.doi.org/10.1109/JIOT.2017.2750030.
[68] X. Sun, N. Ansari, Green cloudlet network: A sustainable platform for mobile cloud computing, IEEE Trans. Cloud Comput. (2018) 1, http://dx.doi.org/10.1109/TCC.2017.2764463.

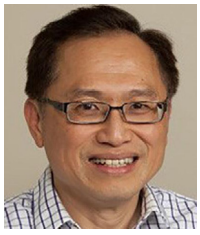
G S S Chalapathi obtained his B.E. and M.E. from Birla Institute of Technology and Science (BITS), Pilani, India in 2009 and 2011 respectively. He has completed his Ph.D. from BITS-Pilani, Pilani Campus in 2019. He is currently a visiting researcher at the Cloud Computing and Distributed Systems (CLOUDS) Lab, School of Computing and Information Systems, The University of Melbourne, Australia. His research interests include Wireless Sensor Networks, Edge Computing, Internet-of-Things (IoT).



Vinay Chamola received the B.E. degree in electrical and electronics engineering and master's degree in communication engineering from the Birla Institute of Technology and Science, Pilani, India, in 2010 and 2013, respectively, and the Ph.D. degree in electrical and computer engineering from the National University of Singapore, Singapore, in 2016. In 2015, he was a Visiting Researcher with the Autonomous Networks Research Group, University of Southern California, USA. He is currently an Assistant Professor in the Department of Electrical and Electronics Engineering, BITS-Pilani, Pilani Campus. His research interests include solar powered cellular networks, energy efficiency in cellular networks, internet of things, and networking issues in cyber–physical systems.

THAM Chen Khong is an Associate Professor at the Department of Electrical and Computer Engineering (ECE) of the National University of Singapore (NUS). His current research focuses on sensor network and machine learning architectures and sensor data analytics involving cyber–physical systems, wireless sensor networks, mobile cloud computing and participatory sensing. He obtained his Ph.D.* and M.A. degrees in Electrical and Information Sciences Engineering from the University of Cambridge, United Kingdom, and was an Edward Clarence Dyason Universitas Fellow at the University of Melbourne, Australia. He is a Senior Member of the IEEE, is in the editorial board of the International Journal of Network Management, was the general chair of the IEEE SECON 2014, IEEE AINA 2011 and IEEE APSCC 2009 conferences, and was the organizing chair of IEEE ICCS 2012.

S Gurunarayanan received his Ph.D. from BITS Pilani. He is currently a Professor, in the Department of Electrical and Electronics Engineering, BITS Pilani. He has about three decades of teaching experience at BITS-Pilani. His research interests are multi-core processor architectures, cache and memory architectures and embedded systems. He is also serving as the Dean (University-wide), Practice School Division, BITS Pilani.



Nirwan Ansari received the B.S.E.E. degree (summa cum laude) from the New Jersey Institute of Technology (NJIT), Newark, NJ, USA, in 1982, the M.S.E.E. degree from the University of Michigan, Ann Arbor, MI, USA, in 1983, and the Ph.D. degree from Purdue University, West Lafayette, IN, USA, in 1988. He is a Distinguished Professor of Electrical and Computer Engineering with NJIT. He has also been a Visiting (Chair) Professor with several universities. He has co-authored (with T. Han) Green Mobile Networks: A Networking Perspective (Wiley–IEEE, 2017) and two other books. He has also co-authored over 500 technical publications, over 200 published in widely cited journals/magazines. He has also been granted 35 U.S. patents. His current research interests include green communications and networking, cloud computing, and various aspects of broadband networks.

Prof. Ansari was a recipient of several Excellence in Teaching Awards, a few Best Paper Awards, the NCE Excellence in Research Award, the IEEE TCGCC Distinguished Technical Achievement Recognition Award, the ComSoc AHSN TC Technical Recognition Award, the ComSoc AHSN TC Outstanding Service Recognition Award, the NJ Inventors Hall of Fame Inventor of the Year Award, the Thomas Alva Edison Patent Award, and the Purdue University Outstanding Electrical and Computer Engineer Award. He has guest edited a number of special issues covering various emerging topics in communications and networking. He has served on the Editorial Board and Advisory Board of over ten journals and magazines including IEEE Communications Magazine (as a Senior Technical Editor). He was elected to serve on the IEEE Communications Society (ComSoc) Board of Governors as a Member-at-Large. He was recently selected to serve on the IEEE Fellow Committee. He has chaired ComSoc Technical Committees and has been actively organizing numerous IEEE international conferences/symposia/workshops. He has frequently delivered keynote addresses, distinguished lectures, tutorials, and invited talks. He holds a designation as a COMSOC Distinguished Lecturer.