Simulation Modelling Practice and Theory 118 (2022) 102531

Contents lists available at ScienceDirect

Simulation Modelling Practice and Theory

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

Energy and latency aware mobile task assignment for green cloudlets

G. S.S. Chalapathi [a,b,*], Vinay Chamola [b], Wafa Johal [c], Jagannath Aryal [d], Rajkumar Buyya [a]

[a] Cloud Computing and Distributed Systems (CLOUDS) Lab, School of Computing and Information Systems, The University of Melbourne, VIC, 3010, Australia
[b] Department of EEE, BITS-Pilani, Pilani, Rajasthan, 333031, India
[c] School of Computing and Information Systems, The University of Melbourne, VIC, 3010, Australia
[d] Faculty of Engineering and IT, Department of Infrastructure Engineering, The University of Melbourne, VIC, 3010, Australia

ARTICLE INFO

Keywords:
Edge computing
Green cloudlet
Task offloading
Mobile edge computing

ABSTRACT

Edge computing places cloudlets with high computational capabilities near mobile devices to reduce the latency and network congestion encountered in cloud server-based task offloading. However, many cloudlets are required in such an edge computing network, leading to a tremendous increase in carbon emissions of computing networks globally. This increase in carbon emission envisages the need to employ green energy resources to power these cloudlets. This need has led to the concept of Green Cloudlet Networks (GCNs). But GCNs must deal with the problem of the unpredictability of green energy available to them while optimizing the performance (in terms of latency) delivered to the mobile user. This paper proposes a novel task-assignment called Green Energy and Latency Aware Task Assignment (*Ge-LATA*) for GCNs to address this issue. The primary aim of Ge-LATA is to optimize the latency and the green energy consumed in processing the offloaded tasks from the mobile devices. In this GCN, the cloudlets are connected in a network to process the incoming tasks cooperatively to ensure load-balancing at the cloudlets. Ge-LATA considers various factors like the current load, available green energy, service rate offered by cloudlets, and the distance from the mobile user, leading to optimal decisions in terms of latency and green energy consumed. Simulations are performed using the actual solar insolation data taken from the NREL database. Ge-LATA is tested with other offloading schemes for latency in processing the offloaded tasks and green-energy consumed under different solar insolation scenarios in these simulations. Simulation results show that Ge-LATA achieves up to 31.87% of reduction in the latency while ensuring up to 50.15% of reduction in the energy consumption than other comparable task-assignment schemes.Thus, Ge-LATA suggests that it leads to an optimal task assignment by considering the various factors mentioned above during the task assignment process. Thus, Ge-LATA considers the above-mentioned extensive set of parameters during the task allotment process. It also proposes an efficient green energy allotment scheme that adapts itself to actual weather and network conditions, leading to optimal task assignment decisions in GCNs.

* Corresponding author at: Department of EEE, BITS-Pilani, Pilani, Rajasthan, 333031, India.
E-mail addresses: gssc@pilani.bits-pilani.ac.in (G.S.S. Chalapathi), vinay.chamola@pilani.bits-pilani.ac.in (V. Chamola), wafa.johal@unimelb.edu.au (W. Johal), jagannath.aryal@unimelb.edu.au (J. Aryal), rbuyya@unimelb.edu.au (R. Buyya).

https://doi.org/10.1016/j.simpat.2022.102531
Received 25 January 2022; Received in revised form 15 March 2022; Accepted 16 March 2022
Available online 27 March 2022

1. Introduction

Advancements made in processor, memory, and network technologies in the last two decades have fueled the domain of mobile computing tremendously. As a result, we use mobile devices like smartphones and tablet PCs for executing applications like image processing, speech processing, augmented reality, and teleconferencing. However, it is difficult to execute these applications on these mobile devices because of the limited processing power, energy availability, and memory of these devices. This has led to the evolution of cloud computing [1–3]. The cloud computing technology and its applications in servicing multiple stakeholders at the time of emergency to various other decision support systems are widely recognized [4,5].

Conceptual cloud-based solution framework and translational research of them for practical use cases, for example; health care systems, bushfire emergencies, flood emergencies, robotics assisted monitoring and rescue, optimal resource allocation, intelligent and precise information extraction from geo-locations, and punctual routing of operational resources have broader impact on social, economic and environmental fronts. Using cloud infrastructure in conjunction with other technologies such as Internet of Things(IoT) networks, swarm of robots, fog and edge computing has tremendous capabilities in serving the businesses [6], protecting lives [7], developing resilient infrastructure [8], and creating a cyber–physical safe society [9].

In cloud computing, powerful cloud servers execute computationally intensive tasks offloaded by mobile devices in remote locations. These mobile devices offload the data of computationally intensive tasks onto these cloud servers through a network connection (usually the internet). Then the cloud servers execute these tasks and return the results to the mobile devices. This solution helps mobile devices perform computationally intensive tasks. However, there are many drawbacks to using cloud services. The first drawback is that there is a significant latency involved in offloading the tasks and then receiving the results from the cloud servers. A major factor contributing to this delay is the large distance between the mobile devices and the cloud servers, which significantly increases the propagation delays involved in the offloading process.

Additionally, there could be significant delays because of the traffic in the network involved in the communication. The second drawback of offloading to the cloud is that frequent communication between the mobile devices and the cloud server increases the traffic in the network, thus reducing the bandwidth available to each mobile device. The drawbacks mentioned above reduce the Quality of Service (QoS) offered to the mobile device while offloading its tasks. Apart from latency, usage of cloud servers involves a high cost in terms of the subscription charges levied by the cloud service providers like Amazon Web Services (AWS), Microsoft Azure. Further, the users of these cloud services should also pay for the Internet connectivity to access these cloud servers. All these factors have led to a new domain called *Edge Computing* [10–14].

Edge Computing targets to bring the computation from the cloud servers to the edge of the network [15]. Instead of offloading the tasks to the cloud servers, the mobile devices offload their tasks to edge devices placed at geographically closer locations (than the cloud servers). These edge devices, which have high computational, memory, and networking capabilities, are called *cloudlets* [16,17]. It is to be noted that the capabilities of these cloudlets in terms of computational power, memory, etc., are lesser than that of the cloud servers. Since the cloudlets are close to the mobile devices, the latency in offloading the tasks and delivering the output to the mobile devices is significantly reduced. Also, these cloudlets can be connected to mobile devices through Wi-Fi or Bluetooth in a local network doing away with the need for the internet. This solution significantly reduces the monetary costs compared to the scenario where cloud services were used.

1.1. Motivation

Although there are several benefits of using edge computing as discussed above, its rise is expected only to increase the energy consumption of the entire mobile edge computing (MEC) system [18] which comprises cloud and cloudlets catering to the needs of the mobile devices. This increase in energy consumption is because MEC systems require the dense deployment of many cloudlets in the close vicinity of the mobile devices, thus increasing the system's energy consumption. In addition, there is a growing concern because of the increasing energy consumption of grid energy produced by traditional non-renewable energy resources like coal-fire, which is the primary source used for generating electricity [19].

The cloud data centers have been a significant consumer of electricity consuming 200–250 Terra Watt Hour of electricity globally in 2020, which is around 1% of the annual electricity usage of the world [20].

However, each cloudlet individually consumes very little energy. This proves to be a silver lining to this problem. These cloudlets can thus be powered by energy resources like solar or wind energy (which we refer to as green energy sources in the rest of the paper), reducing the strain on non-renewable energy resources. Therefore, there has been an increasing interest in ''greening'' the cloudlets in a MEC network. Green cloudlet networks have been proposed in [21,22]. In these networks, the cloudlets are powered by green energy sources like solar energy.

In cloudlets powered by grid energy, the focus of the task offloading algorithms is to optimize the energy consumption while achieving the performance demanded or required by the user. However, the focus of the task offloading process in Green Cloudlets Networks (GCN) should be changed to optimize the performance delivered to the user, subject to the green energy available in the GCN [18]. Also, the unpredictability in both — the amount of available green energy in a day and the workload from the users make the task-assignment schemes of traditional cloudlet networks unsuitable to GCNs. Thus, GCNs require specific task-assignment schemes to handle the tasks offloaded by mobile users.

1.2. Problem addressed by this work

This work proposes a task-assignment scheme called Green Energy and Latency Aware Task Assignment (Ge-LATA) to assign the tasks offloaded by the mobile users onto a GCN. Ge-LATA considers a network of cloudlets that cooperate among themselves in processing the offloaded tasks. Therefore, the task offloaded onto one cloudlet can be processed on another cloudlet in the network. This task assignment scheme makes task assignment decisions so that the overall latency and the green energy consumed by the GCN are optimized. While making the task assignment decisions, the energy harvested at cloudlets at each hour of the day, their current load, and the estimated hourly workload from the mobile users are considered. Simulations were performed using the actual solar insolation data taken from the National Renewable Energy Laboratory, USA (NREL) database [23]. In addition, Ge-LATA was tested with other offloading schemes for latency in processing the offloaded tasks and green-energy consumed under different solar insolation scenarios in these simulations. Simulation results show that Ge-LATA achieves lower latency while ensuring lower energy consumption than other comparable task-assignment schemes.

1.3. Organization of this paper

The rest of the paper is organized as follows: Section 2 discusses the state-of-the-art [SOTA] works related to this paper. Section 3 discusses the system model considered for the operations of the proposed Ge-LATA scheme. We elaborate on the energy allocation used in this work in Section 4. Section 5 presents the details of the optimization problem that is solved by the Ge-LATA scheme. We discuss the Ge-LATA scheme and mathematically prove its optimality in Section 6. We discuss the simulation results performed to evaluate Ge-LATA and compare it with the state-of-the-art [SOTA] schemes in Section 7. We offer our conclusions and future directions in Section 8.

2. Related work

dge computing grew to reduce latency and costs in executing the offloaded tasks. Edge computing does so by placing computationally powerful devices called cloudlets close to mobile devices.

Though the physical proximity of these cloudlets to the mobile devices favors a reduction in the latency involved in carrying out these computations, the resources at these cloudlets must be efficiently managed to optimize the latency. Many works aim at reducing the latency for cloudlets. Latency optimization or latency reduction in task offloading applications has been well studied recently. Many works like [24–27] to cite a few, have reported algorithms which perform latency aware task offloading.

In [25], a simple task assignment scheme which they call as Location awarE workloAd offloadiNg (LEAN) strategy has been mentioned. In LEAN, the task offloaded by the mobile device to a cloudlet is processed at the nearest cloudlet without considering the current task load at that cloudlet. Thus, a cloudlet having many offloading mobile devices near it will be heavily loaded. As a result, the mobile users near such heavily loaded cloudlets experience high latency in receiving the results of their offloaded tasks.

Latency has been optimized in [24,27] using optimal task allocation algorithm. This task assignment algorithm considers the cloudlets' current load, distance, and service rate while assigning the offloaded tasks to the cloudlets.

Kao et al. in [26], focussed on a specific case where the tasks depend on each other and can be represented using serial task graphs. Each application comprises many routines that are assigned to different computing nodes (the cloudlets).

Ali et al. in [28], consider a game theoretic approach to select a cloudlet for the offloading while constraining the latency and workload on each cloudlet. The cloudlet selection process is formulated as a matching game with externalities. However, all these works do not consider energy optimization during the offloading process.

Along with latency, the MEC network must also consider the energy consumption of the cloudlets. We must deploy a large number of cloudlets at the edge in the MEC network to reduce latency [18]. Such deployment will increase the energy consumption of the network. This increase in energy consumption will increase the carbon emissions because of data centers and the newly deployed cloudlets. A study estimates that by 2040, carbon emission because of data centers will account for 14% of the total emission of the world [29,30]. Analysts also predict that by 2025, the data centers will consume one-fifth of the electricity of the world [29,30]. Coal-fired thermal power plants are still the largest source of electricity globally, producing 38% of the world's electricity [31]. Natural gas-based plants are the next prominent source of electricity accounting for 23% of global electricity production [31]. However, both thermal power plants and gas-based power plants emit a significant amount of greenhouse gases which underlines the importance of using green energy sources to power computing nodes in the MEC. Since cloudlets have low energy requirements of the cloudlets, it is possible to cater to their energy requirements using green energy sources like solar and wind energies [18]. Therefore, there is a need to explore the possibility of powering the cloudlets using green energy sources.

Gai et al. [17] is one of the first works to consider "green computing" using the cloudlets. In this work, the authors propose a dynamic energy-aware model to manage the operations in the cloudlet-based MEC network. The authors of this work [17] use dynamic programming to manage the cloudlet resources. However, this work does not consider energy harvesting either at the cloudlets or the mobile devices.

However, this direction of powering the cloudlets using green energy sources opens up newer problems that the researchers must be address to make GCNs viable. The most important aspect of the task allocation strategy to be adopted in GCNs is to focus on optimizing the performance (especially the latency) subject to the green energy available at the cloudlets. Thus, we need to optimize both latency and green energy consumed in processing the tasks offloaded by the mobile devices.

Researchers recently have to some extent explored this problem of using energy harvesting in the systems involving task-offloading from mobile devices to a cloudlet. Most of these works consider energy harvesting only at the mobile device, while grid energy powers the cloudlet. For example, in [32], the execution cost of the offloaded tasks considers latency and task failure rate. The optimization problem jointly decides transmitting power while offloading, frequency of the CPU, offloading decision. Thus, this work neither considers greening the cloudlets nor optimizes the energy consumed at the cloudlets (which this works calls MEC server).

In [33], reinforcement learning (RL) is used to offload the tasks at the mobile device. Mobile devices employ energy harvesting to power themselves. The offloading algorithm uses reinforcement learning to predict the harvested energy available. It uses this information along with parameters like battery energy level and past radio transmission rate to decide the cloudlet on to which task is offloaded and the offloading rate. This work also does not consider energy harvesting at the cloudlet, one of the biggest energy consumers in the MEC system. Also, this work does not optimize the latency and energy consumption at the cloudlet.

Similar to [33,34] also uses reinforcement learning to make offloading decisions. In this work, the authors use a Markov Decision Process (MDP) before using reinforcement learning to model the offloading problem. It also considers energy harvesting at the mobile devices only and does not optimize the energy and latency at the cloudlets. Further, it does not consider the interconnection of the cloudlets, leading to an imbalance in the loads at the cloudlets.

In [35], the authors propose an algorithm for Artificial Intelligence (AI) based tasks offloaded to the cloudlets. They propose a Multiple algorithm service model (MASM) in which devises different strategies according to the data sizes and computational needs of the offloaded tasks. However, this work does not consider the energy harvesting of the cloudlets. Also, the authors do not consider a network of cloudlets that cooperatively handles the offloaded tasks.

Wang et al. propose a Stackelberg Game-based energy latency minimization algorithm in [36]. The authors, in this work, consider the mobile devices as leaders and the MEC server as the follower in their system model. The leaders try to optimize the energy consumed in the task offloading while the follower tries to optimize the latency in processing the tasks. Further, the authors derive Karush–Kuhn–Tucker (KKT) conditions for this problem to derive optimal solutions for latency and energy consumption. However, this work considers a single cloudlet (MEC server) only. Also, it does not consider any energy harvesting either at the cloudlet or the mobile device.

The authors of [37], propose a task allocation algorithm in Vehicular Fog computing (VFC) which optimizes energy and latency experienced by the tasks offloaded by the edge devices. The cloudlet layer in this work considers a multi-tier network comprising Road-side units (RSUs), gateway routers, etc. which are termed as cloudlet nodes. This work also considers vehicular nodes (which have computation and communication resources). The vehicular nodes are also a part of the cloudlet layer. This work uses a three phase strategy to identify the optimal tasks on the overloaded cloudlet node, which can be offloaded to a vehicular node to optimize the energy and latency costs. However, this work also does not consider energy harvesting at the cloudlets.

In [38], energy harvesting is considered at the cloudlets. However, the authors of [38] aim to maximize the revenue earned by the cloudlets from task offloading, and they do not optimize the delay and energy consumed in the offloading process. Also, they consider the cloudlets discretely. This means that a cloudlet which is closest to a mobile device services the task offloaded by the mobile device. This system does not consider cloudlets networked together.

A scheme in which the cloudlets are powered by a green energy source (which they refer to as energy harvesting) is presented in [39]. In this scheme, the mobile devices, as well as the cloudlets, use energy harvesting. First, the mobile device evaluates the green energy available at the nearest cloudlet. Then the mobile decides whether to offload the task to the cloudlet. The task is offloaded to the cloudlet only if it has sufficient green energy to process the task. This work, however, does not consider the scenario where the cloudlets can be networked among themselves to process the offloaded tasks. Second, [39] uses a Markov Decision Process (MDP) to arrive at an optimal solution for making offloading decisions. The system state in this work is taken to be observable, and this incurs tremendous communication cost [40]. Also, in [39], the mobile devices solve the problem online as offline solving is not possible. However, online solving of the optimization problem, that too at every time slot, is not workable for energy-constrained mobile devices [40]. Because of the reasons mentioned above, we feel that [39] is not suited for practical GCNs in which the cloudlets are energy-constrained. This work also neglects load balancing at the cloudlets by considering the cloudlets to be isolated. We have not compared our work with [39] because of the reasons mentioned above.

The focus of the present work is to optimize the latency and the green energy consumed in processing the tasks offloaded by mobile devices onto the cloudlet. We consider a network of cloudlets that are connected. This network ensures a task offloaded onto a cloudlet can be executed on any cloudlet in the network to minimize the latency and the energy consumed in processing the task. Our work also ensures load-balancing among the cloudlets, ensuring that a particular cloudlet is not excessively loaded. Note that if a cloudlet is excessively loaded, it will significantly increase the execution time of the tasks processed by that cloudlet and drain the harvested energy.

2.1. Reasons for the works chosen for comparison of Ge-LATA

Existing work that can be aptly compared with our work must have three aspects to make a fair comparison. First, the task allocation scheme should optimize both energy (or power) and latency. The second aspect is that the task allocation scheme should consider a network of cloudlets. Third, the cloudlets must be powered by a green energy source. We do not find an existing work that fits all these criteria to the best of our knowledge. Thus, we choose a state-of-the-art that is reasonably comparable to the proposed scheme in this paper. Mukherjee et al. [41], is a state-of-the-art task allocation scheme that optimizes both power and latency and considers a network of cloudlets that are connected. However, it does not consider powering the cloudlets through green

Table 1
Summary of the proposed work with other related works.

References	Optimizes latency	Optimizes power or energy	Considers a network of cloudlets for load balancing	Green energy powered cloudlets	Energy harvesting considered at
G.S.S. Chalapathi et al. [24]	✓	✘	✓	✘	–
Sun and Ansari [25]	✓	✘	✘	✘	–
Kao et al. [26]	✓	✘	✘	✘	–
Ali et al. [28]	✓	✘	✓	✘	–
Mao et al.[32]	✓	✘	✘	✘	Mobile devices only
Min et al.[33]	✘	✘	✘	✘	Mobile devices only
Xu et al. [39]	✓	✓	✘	✓	–
Zhang et al. [35]	✓	✓	✘	✘	–
Yadav et al. [37]	✓	✓	✓	✘	–
Wei et al.[34]	✘	✘	✘	✘	Mobile devices only
Chen et al.[38]	✘	✘	✘	✓	–
Gai et al. [17]	✘	✓	✓	✘	–
Wang et al. [36]	✓	✓	✘	✘	–
Mukherjee et al [41]	✓	✓	✓	✘	–
Ge-LATA (this work)	✓	✓	✓	✓	Energy At the cloudlets

energy sources. In this scheme, [41], the mobile device offloads the task to the nearest cloudlet. If the nearest cloudlet can process the offloaded task within a specified temporal deadline, it executes the task on behalf of the mobile device. Suppose the nearest cloudlet cannot process the task within the deadline. In that case, it acts like a proxy server to negotiate with other cloudlets in the network to process the task with minimum power and minimum latency. Thus, a task offloaded onto a cloudlet can be processed on any other cloudlet to minimize power and latency. However, this scheme uses the nearest cloudlet as a proxy server, which increases the delay during the negotiation process [42]. When many mobile devices are present near a cloudlet, that cloudlet will be heavily loaded for the negotiation operation, delaying both the tasks already executing on that cloudlet and the incoming tasks.

We have also compared our work with LEAN mentioned in [25]. The reason for comparing this work with LEAN is that choosing the nearest cloudlet (as done by LEAN) to process the offloaded task appears to be the easiest and most obvious way to achieve minimal latency in processing the offloaded task. We have compared and reviewed our work with other existing works, which are summarized and presented by Table 1.

2.2. Contributions

The major contributions of this work are:

(1) We propose a task allocation algorithm named *Ge-LATA* for Green energy powered Cloudlets Network (GCN), which optimizes the latency of executing the offloaded tasks while optimizing the green energy consumed.
(2) The proposed scheme considers a network of cloudlets and achieves load balancing while serving the task requests.
(3) The task allocation scheme considers an extensive set of parameters like current load at each cloudlet, distance of the cloudlet from the mobile device, available green energy, and maximum service rate capability of the network while making task allocation decisions.
(4) This scheme uses an energy allocation scheme to allot a portion of expected green energy during a day to each hour based on the expected network traffic and weather. This energy allocation is revisited every hour, making the algorithm adaptive to actual weather and network traffic conditions.
(5) The proposed scheme achieves the lowest latency and higher levels of battery energy availability (thus optimizing battery energy) compared to other comparable schemes.

3. System model

3.1. Cloudlet load and latency indicator

The GCN considered in this work is depicted in Fig. 1. The notations used in this paper as summarized in Table 2. Each cloudlet in this network is powered by a green energy source (in our case, through solar energy). Each cloudlet has photovoltaic (PV) panels to capture the solar irradiance and generate the energy used to power the cloudlet. The cloudlets are also equipped with lead–acid batteries to store the energy harvested by them. The cloudlets are connected through a network, and there is a central cloudlet supervisor to coordinate the task assignment (its role will be explained later in Section 6.1).

Let us represent the cloudlets considered in such a system using the set $\mathcal{N} \triangleq \{\mathcal{N}_1, \mathcal{N}_2, \ldots, \mathcal{N}_j, \ldots, \mathcal{N}_{|\mathcal{N}|}\}$. Note that $\mathcal{N}_j$ refers to the jth cloudlet in the system. We assume that these cloudlets cater to the task offloading requests of the mobile users in a geographical region $\mathcal{R}$. The mobile users at a location $z \in \mathcal{R}$ request the cloudlets to process the tasks offloaded by it. The arrival rate of these requests is assumed to follow a Poisson point process as assumed in [24,43,44]. The arrival rate of these requests per

Table 2
Notation summary.

Notation	Meaning
$\mathcal{N}$	Set of all cloudlets considered in the system
z	Mobile user's location
$\lambda(z)$	Arrival rate of the offloaded tasks per unit area
$\frac{1}{\mu(z)}$	Average size of the offloaded task
$\tau(z)$	Density of the tasks offloaded per unit area
S_j^{mx}	Maximum service rate of the cloudlet $\mathcal{N}_j$
$s_j(z)$	The service rate offered by jth cloudlet at z
$u_k(y)$	Task assignment indicator of kth cloudlet at y
$\psi_{h,j}$	Load at cloudlet $\mathcal{N}_j$ during the hth hour
$\mathcal{Y}$	Feasible set of loads at the cloudlets
$\tilde{\mathcal{Y}}$	Relaxed feasible set of loads at the cloudlets
E_j^s	Static energy consumption of the cloudlet $\mathcal{N}_j$
ζ	Scaling factor that relates the load and the dynamic energy
$E_{h,j}$	Energy consumption at $\mathcal{N}_j$ in hth hour
$B_{ini,j}$	Initial battery energy level at the cloudlet $\mathcal{N}_j$ at the beginning of the day
$B_{cr,j}$	The critical battery energy level
$B_{cap,j}$	The maximum energy capacity of the battery at $\mathcal{N}_j$
W_j	Total energy available at $\mathcal{N}_j$
$\beta_{h,j}$	Deficiency factor
$\mathcal{I}_{h,j}$	Latency indicator for $\mathcal{N}_j$ during the h_{th} hour
η	Trade-off factor between latency and energy optimization
k	Index of a period
$\chi_{h,j}^k$	Aversion factor of a cloudlet $\mathcal{N}_j$ at the beginning of the kth period in the hth hour
$v_{h,j}^k(z)$	Task assignment indicator in the kth period of cloudlet $\mathcal{N}_j$
$U_{h,j}(\psi_{h,j}^k)$	Current load at the end of the kth period of the hth hour at $\mathcal{N}_j$
$B_{h,j}^{ex}$	Energy left in the battery at $\mathcal{N}_j$ at the end of hth hour

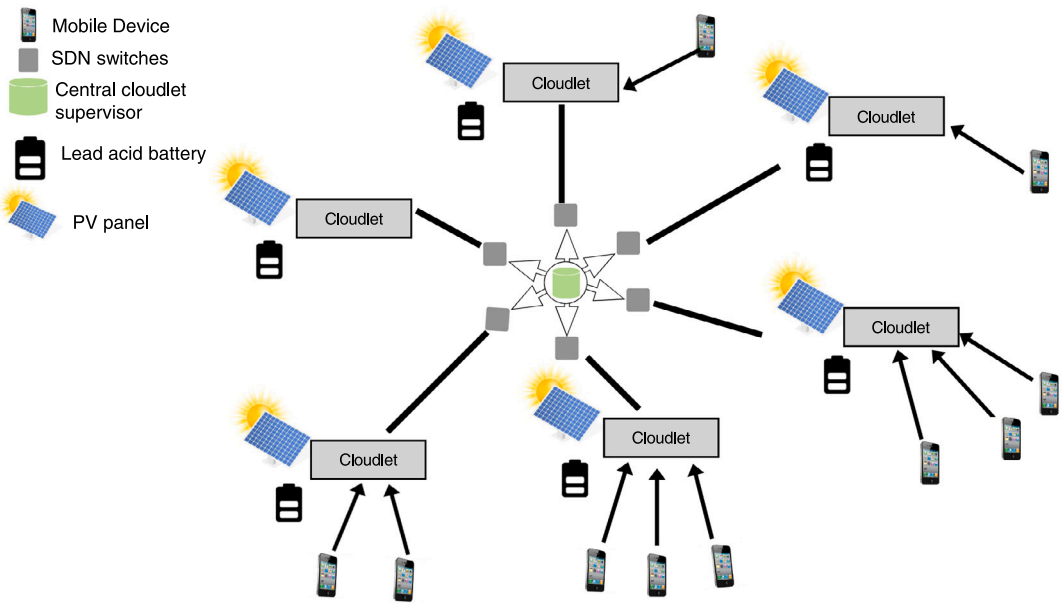

Fig. 1. GCN network considered.

unit area is denoted by $\lambda(z)$. The average size of the offloaded tasks is assumed to be $\frac{1}{\mu(z)}$. Thus the density of the tasks offloaded per unit area denoted by $\tau(z)$ is given by $\tau(z) = \frac{\lambda(z)}{\mu(z)}$. Let S_j^{mx} be the maximum service rate of the cloudlet $\mathcal{N}_j$ and let $d(\mathcal{N}_j, z)$ represent the Cartesian distance between the mobile user at z ($z \in \mathcal{R}$) and cloudlet $\mathcal{N}_j$.

Then, by using the service-rate model used in [24], we evaluate $s_j(z)$, the service rate offered by cloudlet $\mathcal{N}_j$ at z as

$$s_j(z) = \frac{S_j^{mx}}{1 + (\frac{d(\mathcal{N}_j,z)}{d_0})^{\alpha}}, \tag{1}$$

where d_0 is a factor used to normalize the distance and α adjusts the service rate according to the network scenarios. This model has been proved in [24].

We use a indicator function $\upsilon_j(z)$ to indicate whether $\mathcal{N}_j$ serves the request of the mobile user at z or not. The value of this function is 1 if $\mathcal{N}_j$ serves the mobile user at z and its value is 0 otherwise. The load of a $\mathcal{N}_j$ is indicated by ψ_j which is given by

$$\psi_j = \int_{\mathcal{R}} \frac{\tau(z)}{s_j(z)} \upsilon_j(z) dz. \tag{2}$$

The term ψ_j represents the fraction of its time $\mathcal{N}_j$ is busy with handling the offload requests from the mobile users and is used in [45] as well. Let $\boldsymbol{\psi}$ represent the set of cloudlet loads, i.e., $\boldsymbol{\psi} = \{\psi_1, \psi_2, \ldots \psi_{|\mathcal{N}|}\}$.

We further define a set $\mathcal{Y}$ which contains all the feasible cloudlet loads, i.e.,

$$\begin{aligned} \mathcal{Y} = \Big\{ \boldsymbol{\psi} \mid \psi_j = \int_{\mathcal{R}} \frac{\tau(z)}{s_j(z)} \upsilon_j(z) dz, \;\; 0 \le \psi_j \le 1 - \epsilon, \;\; \forall \mathcal{N}_j \in \mathcal{N}, \\ \upsilon_j(z) \in \{0,1\}, \sum_{j=1}^{|\mathcal{N}|} \upsilon_j(z) = 1, \;\; \forall \mathcal{N}_j \in \mathcal{N}, \;\; \forall z \in \mathcal{R} \Big\}, \end{aligned} \tag{3}$$

where ϵ is a constant which has an arbitrarily small positive value.[1] From the above definition in Eq. (3) and the definition of the indicator function, it must be noted that a request from a mobile user is served by a single cloudlet only. However, as stated before, the mobile user offloads the task to the closest cloudlet, which then transfers the task to the cloudlet which will serve the request. The cloudlet that actually serves the request will be determined by the task assignment policy described in this work. As mentioned before, the task arrivals from the mobile users follow a Poisson process and each request is served by a single cloudlet, i.e., there is a single server. Further, the service at the cloudlets follows a general distribution. Thus, the cloudlets can be modeled using M / G / 1 sharing model and the average number of flows at a cloudlet is given by $\frac{\psi_j}{1-\psi_j}$ [45]. As per Little's law, the latency experienced by a traffic flow is directly proportional to this average number of flows. Thus, this total number of flows is taken as the latency indicator of a cloudlet. The latency indicator of $\mathcal{N}_j$ is denoted as $\mathcal{I}_j$ and is given by

$$\mathcal{I}_j = \frac{\psi_j}{1-\psi_j}. \tag{4}$$

This latency indicator has been used to analyze and quantify the network latency performance in many works as [45–47]. This indicator is a unitless quantity, and it captures both computational and queuing delays.

3.2. Cloudlet energy consumption

The energy consumption of a cloudlet $\mathcal{N}_j$ in a given hour h is given as:

$$E_{h,j} = E_j^s + \zeta \; \psi_{h,j}, \tag{5}$$

where E_j^s is the static energy consumption of the cloudlet $\mathcal{N}_j$ (when there is no load), $\psi_{h,j}$ is the load at this cloudlet at hth hour and ζ is a scaling factor that relates the load and the dynamic energy consumption.

3.3. Solar energy and batteries

We assume the cloudlets are placed at appropriate locations with sufficient sunshine exposure during the day to harvest solar energy.

As mentioned earlier, every cloudlet has a rechargeable lead–acid battery that stores the excess energy generated during the day, which can be used during night or in bad weather periods. They are also equipped with PV panels to capture solar energy. The PV panels considered in this implementation are 20 W, 12 V panels. The reason for choosing this rating is that the cloudlets considered in this work are Raspberry Pi processor-based boards [48] which work at 1.5 GHz. They typically consume about 700 mA for their operation at a 5 V power supply [48]. Their small form factor and reduced energy requirements enable them to be deployed in a tiny place where sufficient sunlight is available. The lead–acid batteries considered for this work are 12 V 10Ah batteries which will be sufficient to power the Raspberry Pi-based cloudlets even if they receive solar energy for 5 h. Each cloudlet has two such batteries to store the excess energy.

4. Energy allocation

Before choosing the cloudlets for executing the tasks offloaded by the mobile users, an energy allocation phase is carried out. In this phase, the cloudlet allots the harvested energy in a day to each hour in a day to serve the requests from the mobile users. We

[1] By introducing ϵ we ensure that the cloudlet is not fully loaded. If the cloudlet is 100% loaded, the latency indicator (introduced later in Eq. (4)) will become infinity. To avoid this we ensure that ψ_j is slightly lesser than 1.

Algorithm 1 Energy Allocation Phase at the beginning of the day

```
for j = 1 : |𝒩| do
    𝒲_j = B_{ini,j} − B_{cr,j} + Σ_{t=1}^{24} ℋ_{h,j}
    for h = 1 : 24 do
        E_{h,j} = E_j^s + ζ ψ_{h,j},
        G_{h,j} = 𝒲_j E_{h,j} / Σ_{u=1}^{24} E_{u,j}.
        if G_{h,j} > B_{cap,j} then
            G_{h,j} = B_{cap,j}
        end if
    end for
end for
```

assume that we have partial information on the traffic profile that helps us in making this decision. Note that this information can be easily obtained by noting the traffic pattern for the trial run for a week or two. Also, note that such traffic is strongly influenced by factors that can be easily estimated. For example, a cloudlet installed in the cafeteria area on a university campus is expected to be heavily loaded during breakfast and lunch hours. Also, on weekends and holidays, it is expected to be lightly loaded because of very few students. Further, the effect of the month of the year can also be observed for some time to have more accurate predictions.

To make the energy allocation decision, we also need to know the expected harvested solar energy profile from the weather forecast for a given day. Thus, given the expected traffic and the expected harvested solar energy profile, we can calculate the energy allocated to a cloudlet for a given hour of the day. Let $B_{ini,j}$ be the initial battery energy level at the cloudlet $\mathcal{N}_j$ at the beginning of the day. Let $B_{cr,j}$ be the critical battery energy level below which the battery should not be discharged. Let $\mathcal{H}_{h,j}$ be the energy harvested at hth hour of the day. Then the total energy available at $\mathcal{N}_j$ is given by

$$\mathcal{W}_j = B_{ini,j} - B_{cr,j} + \sum_{h=1}^{24} \mathcal{H}_{h,j}, \quad \forall j \in \mathcal{N}; \tag{6}$$

In the above equation (Eq. (6)), in the third term, we considered the energy harvested for every hour rather than for a smaller time duration. The reason for this is that later in the simulations in Section 7, we will use the actual solar insolation data taken from the NREL database to evaluate W_j using Eq. (6). However, solar insolation data in this database is available only at a frequency of every hour in a day (and not lesser than this). Thus in Eq. (6), we added up the solar insolation for each hour of the day to get the total harvested energy available in a day.

From Eq. (5), we know the energy required in a given hour h at $\mathcal{N}_j$ to serve its requests. We allocate a portion of the total harvested energy available, i.e., $\mathcal{W}_j$ for serving the requests for the hth hour in proportion to the energy required during that hour. The energy allocated to $\mathcal{N}_j$ in a given hour h is denoted by $G_{h,j}$ and is given by

$$G_{h,j} = \mathcal{W}_j \frac{E_{h,j}}{\sum_{u=1}^{24} E_{u,j}}. \tag{7}$$

Note that this allocation of the harvested energy is done at the beginning of the day. In addition, at the end of every hour, the excess energy, i.e., the green energy that was allotted but not used, is accessed. If the excess energy is positive, this energy is distributed to the remaining hours of the day. We shall discuss this excess handing algorithm in detail in Section 6.3.

The maximum energy capacity of the battery at $\mathcal{N}_j$ is denoted as $B_{cap,j}$. If the energy allotted for $\mathcal{N}_j$ using Eq. (7) is greater than $B_{cap,j}$, then $G_{h,j}$ is taken as $B_{cap,j}$ as the battery cannot be charged more than $B_{cap,j}$. The steps in energy allocation algorithm are summarized in Algorithm 1.

The energy deficiency for a particular cloudlet $\mathcal{N}_j$ at a given hour h is indicated by a deficiency factor $\beta_{h,j}$ which is given as follows.

$$\beta_{h,j} = \frac{E_{h,j}}{G_{h,j}}. \tag{8}$$

If $\mathcal{N}_j$ is energy-deficient, then $E_{h,j} > G_{h,j}$ and thus $\beta_{h,j} > 1$. Thus, the task allocation problem would aim to reduce the deficiency factor as much as possible so that the task requests are served with the green energy available in the GCN. We will use this deficiency factor in defining our optimization problem in the next section.

5. Problem formulation

The aim of this task-assignment policy is to minimize the network-wide latency and the energy deficiency. The latency indicator defined in Eq. (4) is for a cloudlet $\mathcal{N}_j$. We will denote the latency indicator for $\mathcal{N}_j$ during the h_{th} hour as $\mathcal{I}_{h,j}$ which is given by

$$\mathcal{I}_{h,j} = \frac{\psi_{h,j}}{1 - \psi_{h,j}}. \tag{9}$$

Thus we will use the latency indicator specified in Eq. (9) and the deficiency factor defined in Eq. (8) to formulate our optimization problem. This optimization problem denoted as **[Pb1]** can be formulated as

$$\begin{aligned}[\textbf{Pb1}]\ \underset{\psi}{\text{minimize}}\ Y(\boldsymbol{\psi}) &= \sum_{h=1}^{24}\sum_{j=1}^{|\mathcal{N}|}\left(\mathcal{I}_{h,j}(\psi) + \eta\frac{P\,\beta_{h,j}}{Q-\beta_{h,j}}\right) \\ \text{subject to: } \boldsymbol{\psi} &\in \mathcal{Y}\end{aligned} \tag{10}$$

In the above problem, η refers to the *trade-off factor* which helps us trade-off between latency and energy optimization. When we want to optimize both latency and green-energy equally, the value of η is taken as unity (i.e., 1). Also in the above problem, P and Q are positive constants (P and $Q \geq 1$) chosen such that as the deficiency factor $\beta_{h,j}$ approaches the value Q the value of the optimization function increases sharply. Thus if a cloudlet is energy deficient, i.e., $\beta_{h,j} > 1$, any further allocation of the mobile users to this cloudlet $\mathcal{N}_j$ will bring about a sharp increase in the objective function. This sharp increase prevents mobile users from being allocated to the energy-deficient cloudlet. We want to avoid allocating mobile users (or equivalently offloaded tasks) to energy-deficient cloudlets. If the energy-deficient cloudlets are assigned any more offloaded tasks, latency will increase tremendously, which is undesirable.

6. Green energy and Latency Aware Task Assignment (Ge-LATA)

GE-LATA aims to optimize the objective function $Y(\psi)$ such that both the network-wide latency and the green energy deficiency are minimized. To facilitate this optimization, each cloudlet periodically evaluates a parameter called *aversion factor* which will be formally defined shortly in this section. The aversion factor reflects the current state of the load and the energy deficiency factor of the cloudlet. The values of these aversion factors are sent to the central cloudlet supervisor, who will then make the task assignment decisions whenever the mobile user offloads a task to a cloudlet in the network. The task assignment decisions minimize the objective function $Y(\psi)$. It is assumed that all the cloudlets send the values of their aversion factor to the central cloudlet supervisor periodically in a synchronized way. Also, the time period of this unicast is chosen such that it is larger than the time interval between traffic arrival and departure. This is done to ensure that the value of the aversion factor remains valid for the time duration between request arrival to completion. Two different algorithms are executed to arrive at the optimal task assignment decisions, each at the central cloudlet supervisor and the cloudlet. We describe each of these algorithms in the following subsections.

6.1. Algorithm at the central cloudlet supervisor

The optimization problem **[Pb1]** is a function of the cloudlet load $\boldsymbol{\psi}$ where $\boldsymbol{\psi} \in \mathcal{Y}$. However, $\mathcal{Y}$ is not convex because the indicator function $\upsilon_j(z) \in \{0,1\}$. Thus this constraint is relaxed to $0 \leq \upsilon_j(z) \leq 1$ so that the optimization problem becomes convex. The modified feasible set with this relaxation on the indicator function $\upsilon_j(z)$ is denoted as $\tilde{\mathcal{Y}}$ and is denoted as

$$\begin{aligned}\tilde{\mathcal{Y}} = \Big\{ \boldsymbol{\psi} \ \mid\ \psi_j = \int_{\boldsymbol{\mathcal{R}}} \frac{\tau(z)}{s_j(z)}\upsilon_j(z)dz,\ \ 0 \leq \psi_j \leq 1-\epsilon,\ \ \forall \mathcal{N}_j \in \boldsymbol{\mathcal{N}}, \\ 0 \leq \upsilon_j(z) \leq 1, \sum_{j=1}^{|\mathcal{N}|}\upsilon_j(z) = 1, \forall \mathcal{N}_j \in \boldsymbol{\mathcal{N}}, \forall z \in \boldsymbol{\mathcal{R}}\ \Big\},\end{aligned} \tag{11}$$

It has been proven in [46] that the relaxed feasible set $\tilde{\mathcal{Y}}$ is convex. Thus optimization problem **[Pb1]** after the above relaxation can be now written as **[Pb2]** can be formulated as

$$\begin{aligned}[\textbf{Pb2}]\ \underset{\psi}{\text{minimize}}\ Y(\boldsymbol{\psi}) &= \sum_{h=1}^{24}\sum_{j=1}^{|\mathcal{N}|}\left(\mathcal{I}_{h,j}(\psi)\ + \eta\frac{P\,\beta_{h,j}}{Q-\beta_{h,j}}\right) \\ \text{subject to: } \boldsymbol{\psi} &\in \tilde{\mathcal{Y}}\end{aligned} \tag{12}$$

As mentioned earlier, the cloudlets send their aversion factor values to the central cloudlet supervisor periodically. At the beginning of each period, the cloudlet supervisor receives the values of aversion factor of all the cloudlets. It then assigns the offloaded tasks to the cloudlets based on these aversion factors values and the effective service rates offered by the cloudlets to a mobile user. We denote the aversion factor of a cloudlet $\mathcal{N}_j$ at the beginning of the kth period in the hth hour as $\chi^k_{h,j}$ and it is defined as follows:

$$\begin{aligned}\chi^k_{h,j} &= \frac{\partial Y(\boldsymbol{\psi})}{\partial \psi^k_{h,j}} \\ &= \frac{1}{(1-\psi^k_{h,j})^2} + \frac{\eta P\zeta G_{h,j}Q}{(G_{h,j}Q - E^s_j - \zeta\psi^k_{h,j})^2} \\ &= \frac{1}{(1-\psi^k_{h,j})^2} + \frac{\eta P\zeta Q}{G_{h,j}(Q-\beta_{h,j})^2}\end{aligned} \tag{13}$$

Algorithm 2 Algorithm at the Central Cloudlet Supervisor

1: At the beginning of every period k of the hth hour, the aversion factor $\chi^k_{h,j}$, $\forall j \in \mathcal{N}$, i.e., all the cloudlets in the network is sent to the Central cloudlet supervisor.
2: When a request for task offloading is received at location z evaluate $s_{h,j}(z)$, $\forall j \in \mathcal{N}$ using Eq. (1).
3: The task offloaded at location z is allocated by evaluating $\upsilon^k_{h,j}(z)$ using Eq. (14).

It must be noted that the aversion factor of a cloudlet considers its current load and its deficiency of green energy. Thus, it can be observed that as the load at the cloudlet and/or its deficiency factor increases, the aversion factor's value increases. Thus the cloudlet will be *averse* to accept more requests from the incoming mobile users. Thus this factor derives its name from this phenomenon.

The offloaded tasks at a location z are assigned to the cloudlets based on the following function:

$$\upsilon^k_{h,j}(z) = \begin{cases} 1 & \text{if } j = \arg\max\limits_{j \in \mathcal{N}} \frac{s_{h,j}(z)}{\chi^k_{h,j}} \\ 0 & \text{otherwise.} \end{cases} \tag{14}$$

The above function specifies whether a cloudlet $\mathcal{N}_j$ will serve the requests at location z in the kth time period or not. Since the cloudlet supervisor has to decide on the task assignment based on the aversion factor values of all the cloudlets, the computation complexity of this algorithm is of the order of $O(|\mathcal{N}|)$. This algorithm used by the cloudlet supervisor is summarized in Algorithm 2.

6.2. Algorithm at the cloudlet

The main function of the cloudlets is to enable the central cloudlet supervisor to make the task assignment decision. It does so by evaluating their aversion factor periodically. Each hour in a day is further divided into a fixed number of *periods*. To calculate the aversion factor, the cloudlet evaluates its current load at the end of the kth period of the hth hour denoted by $U_{h,j}(\psi^k_j)$ using the following equation:

$$U_{h,j}(\psi^k_{h,j}) = \min\left(\int_{\mathcal{R}} \frac{\tau(z)}{s_{h,j}(z)} \upsilon^k_{h,j}(z) dz, 1-\epsilon\right), \tag{15}$$

where ϵ is a constant which has an arbitrarily small positive value as mentioned in Eq. (3). The intuition behind the above equation is that the load at the cloudlet which represents the fraction of time of the cloudlet engaged in serving the requests cannot be more than 1. In the scenario when the cloudlet is heavily loaded, the value of this load almost approaches unity. The cloudlets then updates the load for the next period using the following equation:

$$\psi^{k+1}_{h,j} = \omega\ \psi^k_{h,j} + (1-\omega)\ U_{h,j}(\psi^k_{h,j}), \tag{16}$$

where ω is used for averaging and $\omega \in (0,1)$.

We now prove the optimality of the Ge-LATA scheme discussed above. To prove the optimality, we will first prove that the objective function $Y(\boldsymbol{\psi})$ convex for $\psi \in \tilde{\mathcal{Y}}$.

Theorem 1. *The objective function $Y(\boldsymbol{\psi})$ is convex with respect to $\psi_{h,j}$ when $\psi_{h,j} \in \tilde{\mathcal{Y}}$.*

Proof. To prove this, we will show that the second-order derivative of the objective function with respect to $\psi_{h,j}$ is positive. We re-write the objective function as

$$\begin{aligned} Y(\boldsymbol{\psi}) &= \sum_{h=1}^{24} \sum_{j=1}^{|\mathcal{N}|} \left(I_{h,j}(\psi) + \eta \frac{P\,\beta_{h,j}}{Q - \beta_{h,j}} \right) \\ &= \sum_{h=1}^{24} \sum_{j=1}^{|\mathcal{N}|} \left(\frac{\psi_{h,j}}{1-\psi_{h,j}} + \eta \frac{P(E^s_j + \zeta\ \psi_{h,j})}{G_{h,j}Q - E^s_j - \zeta\ \psi_{h,j}} \right) \end{aligned}$$

The first order and second order derivatives of $Y(\boldsymbol{\psi})$ with respect to $\psi_{h,j}$ are

$$\nabla Y(\boldsymbol{\psi}) = \sum_{h=1}^{24} \sum_{j=1}^{|\mathcal{N}|} \left(\frac{1}{(1-\psi^k_{h,j})^2} + \frac{\eta P \zeta G_{h,j} Q}{(G_{h,j}Q - E^s_j - \zeta\psi^k_{h,j})^2} \right) \tag{17}$$

$$\nabla^2 Y(\boldsymbol{\psi}) = \sum_{h=1}^{24} \sum_{j=1}^{|\mathcal{N}|} \left(\frac{2}{(1-\psi^k_{h,j})^3} + \frac{2\eta P \zeta G_{h,j} Q}{(G_{h,j}Q - E^s_j - \zeta\psi^k_{h,j})^3} \right) \tag{18}$$

It can be easily concluded that both the fractions on the right hand side of Eq. (18) are positive (i.e., >0). This is because the numerators in both the fractions are positive and $\psi^k_{h,j}$ is always <1. Likewise, $Q > \beta^k_{h,j}$ as mentioned in Section 5 and thus the

denominator of the second fraction which can be written as $Q-\beta_{h,j}^{k}$ will be >0. Thus, $\nabla^2 Y(\boldsymbol{\psi})$ proving the convexity of the function $Y(\boldsymbol{\psi})$ which proves the theorem. □

Now that we have proven that the function $Y(\boldsymbol{\psi})$ is convex, $\exists\ \boldsymbol{\psi}^{\dagger} \in \tilde{\mathcal{Y}}$ (where $\boldsymbol{\psi}^{\dagger}$ represents a load vector of all the cloudlets) such that $Y(\boldsymbol{\psi})$ is minimized for a particular task assignment. With the help of Lemmas 1 and 2, we will prove that Ge-LATA scheme converges to an optimal value of load, i.e., $\boldsymbol{\psi}^{\dagger}$. We will prove in Lemmas 1 and 2 that $U_j(\psi_{h,j}^{k})$ and $(\psi_{h,j}^{k+1}-\psi_{h,j}^{k})$ will give a descent direction for $Y(\boldsymbol{\psi})$ at $\psi_{h,j}^{k}$.

Lemma 1. *$U_{h,j}(\psi_{h,j}^{k})$ gives the descent direction for $Y(\boldsymbol{\psi})$ at $\psi_{h,j}^{k}$ when $\psi_{h,j}^{k} \neq \psi_{h,j}^{\dagger}$ (where $\psi_{h,j}^{\dagger}$ is the load of $\mathcal{N}_j$ in hth hour when $Y(\boldsymbol{\psi})$ is optimal.)*

Proof. It was proved in Theorem 1 that for $\psi \in \tilde{\mathcal{Y}}$, the function $Y(\boldsymbol{\psi})$ is convex. By showing the inner product $\langle \nabla Y(\boldsymbol{\psi}), U_{h,j}(\psi_{h,j}^{k}) - \psi_{h,j}^{k} \rangle \leq 0$ we can prove this lemma [49]. Let the indicator function for task assignment for the cloudlet load $U_{h,j}(\psi_{h,j}^{k})$ and $\psi_{h,j}^{k}$ be $\upsilon_{h,j}^{U}(z)$ and $\upsilon_{h,j}^{k}(z)$ respectively. Then the inner product can be solved as the following:

$$
\begin{aligned}
&\langle \nabla Y(\boldsymbol{\psi}), U_{h,j}(\psi_{h,j}^{k}) - \psi_{h,j}^{k} \rangle \\
&= \sum_{h=1}^{24} \sum_{j=1}^{|\mathcal{N}|} \Big(\frac{1}{(1-\psi_{h,j}^{k})^2} + \frac{\eta P \zeta G_{h,j} Q}{(G_{h,j} Q - E_j^{s} - \zeta \psi_{h,j}^{k})^2} \Big) \\
&\times \left(\int_{\mathcal{R}} \frac{\tau(z)}{s_{h,j}(z)} (\upsilon_{h,j}^{U}(z) - \upsilon_{h,j}^{k}(z)) dz \right) \\
&= \int_{\mathcal{R}} \tau(z) \sum_{h=1}^{24} \sum_{j=1}^{|\mathcal{N}|} \chi_{h,j}^{k} \frac{(\upsilon_{h,j}^{U}(z) - \upsilon_{h,j}^{k}(z))}{s_{h,j}(z)} dz.
\end{aligned}
$$

From Eq. (14) we observe that $\upsilon_{h,j}^{U}(z)$ maximizes $\frac{s_{h,j}(z)}{\chi_{h,j}^{k}}$. Thus we can say that

$$
\sum_{h=1}^{24} \sum_{j=1}^{|\mathcal{N}|} \chi_{h,j}^{k} \frac{(\upsilon_{h,j}^{U}(z) - \upsilon_{h,j}^{k}(z))}{s_{h,j}(z)} \leq 0.
$$

Thus, $\langle \nabla Y(\boldsymbol{\psi}), U_{h,j}(\psi_{h,j}^{k}) - \psi_{h,j}^{k} \rangle \leq 0$ and this proves this lemma. □

Lemma 2. *$(\boldsymbol{\psi^{k+1}} - \boldsymbol{\psi^{k}})$ gives the direction of descent to $Y(\boldsymbol{\psi})$.*

Proof. $\boldsymbol{\psi^{k}}$ represents the load vector of all the cloudlets at the end of kth period. We will prove this lemma by first considering the term $(\psi_{h,j}^{k+1} - \psi_{h,j}^{k})$. From Eq. (16), we know that

$$
\begin{aligned}
(\psi_{h,j}^{k+1} - \psi_{h,j}^{k}) &= \omega\ \psi_{h,j}^{k} + (1-\omega)\ U_{h,j}(\psi_{h,j}^{k}) - \psi_{h,j}^{k} \\
&= (1-\omega)(U_{h,j}(\psi_{h,j}^{k}) - \psi_{h,j}^{k}).
\end{aligned}
$$

From Lemma 1, it has been proven that $(U_{h,j}(\psi_{h,j}^{k}) - \psi_{h,j}^{k})$ provides a direction of descent to $Y(\boldsymbol{\psi})$. Also because $0 < \omega < 1$, it is clear that $(1-\omega) > 0$. Therefore $(\psi_{h,j}^{k+1} - \psi_{h,j}^{k})$ also provides a direction of descent to $Y(\boldsymbol{\psi})$. Further, $(\boldsymbol{\psi^{k+1}} - \boldsymbol{\psi^{k}})$ will also provide a direction of descent to $Y(\boldsymbol{\psi})$ which proves this lemma. □

Theorem 2. *The vector ψ which represents the load vector of the cloudlets converges to $\boldsymbol{\psi}^{\dagger} \in \mathcal{Y}$, where $\boldsymbol{\psi}^{\dagger}$ represents the optimal load vector of the cloudlets.*

Proof. It was proved in Theorem 1 that $Y(\boldsymbol{\psi})$ is convex and it was also proved in Lemma 2 that $(\boldsymbol{\psi^{k+1}} - \boldsymbol{\psi^{k}})$ will give a direction of descent to $Y(\boldsymbol{\psi})$. Therefore, we can be sure that $Y(\boldsymbol{\psi})$ converges to $\boldsymbol{\psi}^{\dagger}$. We will prove it through contradiction. Let us assume that $Y(\boldsymbol{\psi})$ does not converge to $\boldsymbol{\psi}^{\dagger}$ and it converges to another point. Then, by Lemma 2, it can be concluded $\boldsymbol{\psi^{k+1}}$ will give the further descent direction to $Y(\boldsymbol{\psi})$ and this contradicts the assumption. Thus, $Y(\boldsymbol{\psi})$ converges to $\boldsymbol{\psi}^{\dagger}$. Since $\boldsymbol{\psi^{k}}$ has been derived from Eq. (14), in which $\upsilon_{h,j}^{k}(z) \in \{0,1\}$, the optimal load vector $\boldsymbol{\psi}^{\dagger} \in \mathcal{Y}$. □

Now we will prove the optimality of Ge-LATA using Lemmas 1 and 2.

Theorem 3. *The objective function $Y(\boldsymbol{\psi})$ is minimized by the task assignment which corresponds to $\boldsymbol{\psi}^{\dagger}$ when it is given that the set $\mathcal{Y}$ is non-empty and the cloudlet load vector $\boldsymbol{\psi}$ converges to $\boldsymbol{\psi}^{\dagger}$*

Proof. Let the task assignment which corresponds to the optimal load vector $\boldsymbol{\psi}^{\dagger}$ be $\upsilon^{\dagger} = \{\upsilon_{h,j}(z) | \upsilon_{h,j}(z) \in \{0,1\}, \forall j \in \mathcal{N}, \forall z \in \mathcal{R}\}$ and that corresponding to load vector $\boldsymbol{\psi}$ be $\upsilon = \{\upsilon_{h,j}(z) | \upsilon_{h,j}(z) \in \{0,1\}, \forall j \in \mathcal{N}, \forall z \in \mathcal{R}\}$. From Theorem 1, it is proven that $Y(\boldsymbol{\psi})$ is

Algorithm 3 Algorithm at the Cloudlet

1: The current load for the kth period in the hth hour, i.e., $U_{h,j}(\psi^k_{h,j})$ is evaluated using Eq. (15).
2: The load for the next period $\psi^{k+1}_{h,j}$ using Eq. (16).
3: The aversion factor $\chi^{k+1}_{h,j}$ using the Eq. (13)
4: The value of $\chi^{k+1}_{h,j}$ is sent to the Central Cloudlet Supervisor at the beginning of the next period.

Algorithm 4 Excess energy handling algorithm

for $j = 1 : |\mathcal{N}|$ **do**
  $E_{hr,j} = E^s_j + \zeta\ \psi_{hr,j}$,
  $B^{ex}_{hr,j} = G_{hr,j} - E_{hr,j}$
  for $h = hr + 1 : 24$ **do**
    $G_{h,j} = G_{h,j} + B^{ex}_{h-1,j} \frac{E_{h+1,j}}{\sum_{u=t+1}^{24} E_{u,j}}$
  end for
end for

convex. To prove that the $Y(\boldsymbol{\psi})$ is minimized, we will prove that $\langle \nabla Y(\boldsymbol{\psi}^\dagger), \boldsymbol{\psi} - \boldsymbol{\psi}^\dagger \rangle \geq 0$. For brevity and clarity, we use $\frac{\partial Y(\boldsymbol{\psi}^\dagger)}{\partial \psi^\dagger_{h,j}}$ as $\chi_{h,j}(\psi^\dagger_{h,j})$ in the following proof.

$$
\begin{aligned}
&\langle \nabla Y(\boldsymbol{\psi}^\dagger), \boldsymbol{\psi} - \boldsymbol{\psi}^\dagger \rangle \\
&= \sum_{h=1}^{24} \sum_{j=1}^{|\mathcal{N}|} \chi_{h,j}(\psi^\dagger_{h,j}) \left(\boldsymbol{\psi} - \boldsymbol{\psi}^\dagger\right) \\
&= \sum_{h=1}^{24} \sum_{j=1}^{|\mathcal{N}|} \left(\int_{\mathcal{R}} \frac{\tau(z)(\upsilon_{h,j}(z) - \upsilon^\dagger_{h,j}(z))}{s_{h,j}(z)\chi^{-1}_{h,j}(\psi^\dagger_{h,j})} dz \right) \\
&= \int_{\mathcal{R}} \tau(z) \sum_{h=1}^{24} \sum_{j=1}^{|\mathcal{N}|} \frac{(\upsilon_{h,j}(z) - \upsilon^\dagger_{h,j}(z))}{s_{h,j}(z)\chi^{-1}_{h,j}(\psi^\dagger_{h,j})} dz
\end{aligned}
$$

Since the optimal task assignment is given by the equation

$$
\upsilon^\dagger_{h,j}(z) = \begin{cases} 1, & \text{if } j = \arg\max\limits_{j \in \mathcal{N}} \frac{s_{h,j}(z)}{\chi_{h,j}(\psi^\dagger_{h,j})}, \\ 0, & \text{otherwise.} \end{cases}
$$

we can conclude that

$$
\sum_{h=1}^{24} \sum_{j=1}^{|\mathcal{N}|} \frac{\upsilon^\dagger_{h,j}(z)}{s_{h,j}(z)\chi^{-1}_{h,j}(\psi^\dagger_{h,j})} \leq \sum_{h=1}^{24} \sum_{j=1}^{|\mathcal{N}|} \frac{\upsilon_{h,j}(z)}{s_{h,j}(z)\chi^{-1}_{h,j}(\psi^\dagger_{h,j})}.
$$

This in turn proves that $\langle \nabla Y(\boldsymbol{\psi}^\dagger), \boldsymbol{\psi} - \boldsymbol{\psi}^\dagger \rangle \geq 0$ which proves this theorem. □

6.3. Carrying forward excess energy

The energy allocated at the cloudlet $\mathcal{N}_j$ for hth hour is denoted by $G_{h,j}$ and it is given by Eq. (7). Suppose the energy used at $\mathcal{N}_j$ during the hth hour is lesser than $G_{h,j}$. In that case, this remaining energy will be allocated to the remaining hours in the day proportional to the expected load at $\mathcal{N}_j$ for the remaining hours in the day. Let the energy left in the battery at $\mathcal{N}_j$ at the end of hth hour be $B^{ex}_{h,j}$.

The energy allocated for the $(h+1)$th hour is updated now by adding a portion of $B^{ex}_{h,j}$ proportional to the load expected at $\mathcal{N}_j$ during $(h+1)$th hour. The updation is performed using the equation

$$
G_{h+1,j} = G_{h+1,j} + B^{ex}_{h,j} \frac{E_{h+1,j}}{\sum_{u=h+1}^{24} E_{u,j}}. \tag{19}
$$

The above update is performed for all the hours of the day. This process is summarized in the Algorithm 4. Thus, using this algorithm, the energy allocation made during the energy allocation phase at the beginning of the day is fine-tuned and made more accurate. It also makes Ge-LATA adaptive to the changes in the traffic, which may be different from the expected traffic profile at the cloudlets. We present the results of the simulations of Ge-LATA in the next section.

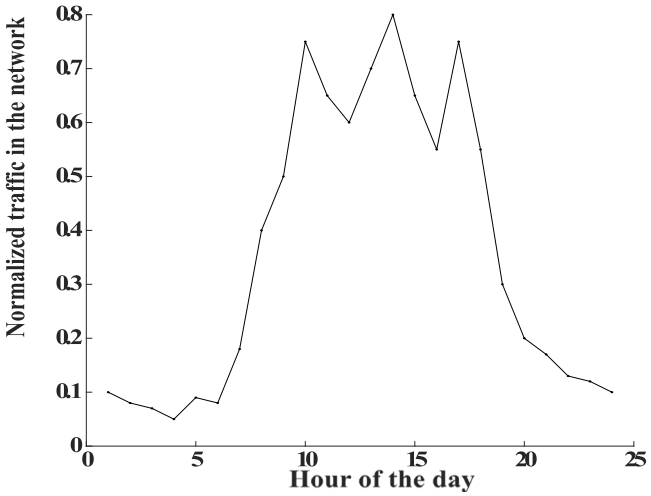

Fig. 2. Traffic load in the network during different hours in the day.

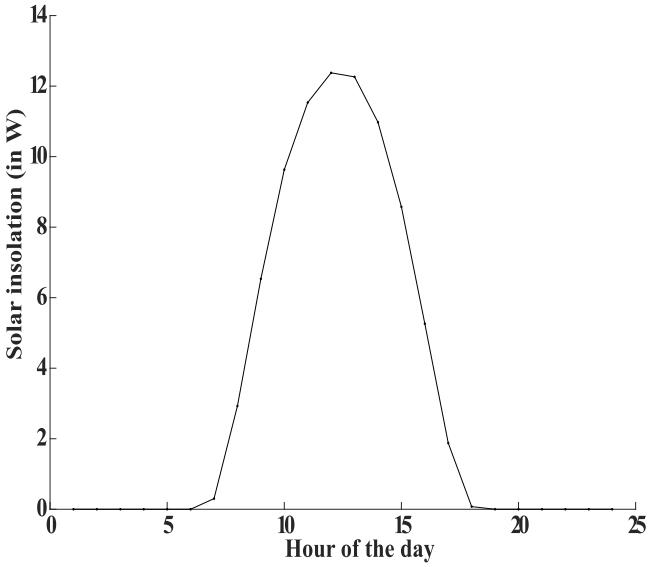

Fig. 3. Solar Insolation profile for a sunny day on a 20 W solar panel.

7. Performance evaluation

The performance of Ge-LATA was evaluated through simulations performed on MATLAB R2019a. For these simulations, the value of the trade-off factor η is taken as 1 to give equal weight to optimizing the green energy and latency. As mentioned in Section 3.3, each cloudlet has a PV panel of 12 V and 20 W. Every cloudlet also has two lead–acid batteries of 12 V and 10 Ah. The task offloading request follows a Poisson point process with an average of 200 requests per unit area. There were six cloudlets deployed in an area of 1000 m $\times$ 1000 m. The maximum service rate of three of the cloudlets is 3000 kB/s, and that of the other three cloudlets is 6000 kB/s. This area of deployment area was divided into 1600 locations. Each hour was divided into 200 slots, i.e., the cloudlets evaluate their aversion factor every 18 s.

The critical battery level was chosen to be 30%, i.e., the batteries are not discharged beyond this level. The solar insolation data was taken from the NREL database for London [23]. The results of Ge-LATA are compared with the LEAN [25] and the scheme proposed in [41]. The reasons for choosing LEAN [25,41] has been mentioned in Section 2. To have a fair comparison of Ge-LATA with [41] and LEAN, we assume that the cloudlets use only green energy resources to process the incoming task requests for all three task assignment schemes.

We have plotted the task offload request traffic for different hours of a day in Fig. 2. The solar insolation for different hours in a day as received on the 20 W PV panel is shown in Fig. 3. Fig. 4 shows the latency behavior of Ge-LATA, LEAN [25], and [41] for

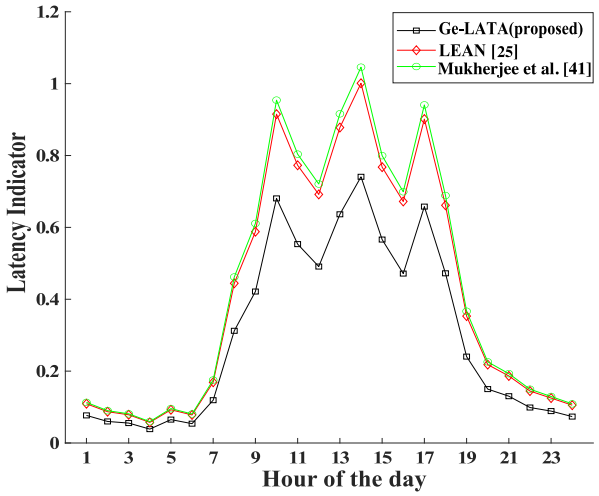

Fig. 4. Latency variation during a sunny day.

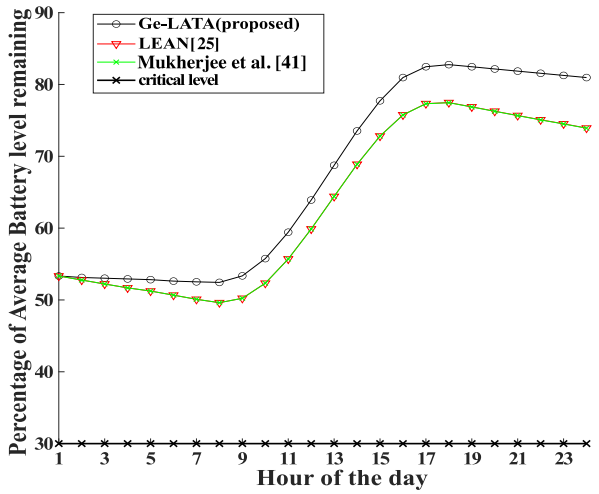

Fig. 5. Average battery level during one day where solar insolation is high (sunny day).

different hours of a day when the solar insolation was high, i.e., when it was a sunny day. As expected, the latency for servicing the offloaded tasks increases as the load in the network increases. This trend is valid for all three task assignment schemes analyzed in these simulations (Ge-LATA, LEAN and [41]). Latency reduces when the traffic decreases in the last quarter of the day (after 18th hour). We also observe in Fig. 4 that Ge-LATA shows a reduction of up to 29.025% compared to LEAN and 31.87% compared to [41] (observed at the 12th hour of the day). Ge-LATA has lower latency because it considers the current load of the cloudlet to which the incoming requests are offloaded. However, as mentioned in Section 2, LEAN over-burdens the cloudlet nearest to the mobile devices, leading to higher latency. On the other hand, [41] offloads the incoming tasks to the nearest cloudlet if it can complete the request within a specified deadline. If the nearest cloudlet cannot process the task on time, it offloads the task to a cloudlet that can complete the request within the deadline.

Another parameter that Ge-LATA optimizes is the green energy consumed in processing the task requests. We evaluate the performance of the three task assignment schemes by considering the average battery levels at the cloudlets for each hour in a day. An energy-efficient task assignment scheme will consume lesser green energy to process the offloaded tasks. The average battery level for processing the offloaded task requests for Ge-LATA, LEAN, and [41] on a sunny day is shown in Fig. 5. We see that Ge-LATA manages the green energy level efficiently. Thus, the batteries have higher green energy levels throughout the day compared to LEAN, and [41]. More specifically we observe from Fig. 5 that Ge-LATA has 9.5% higher average battery level remaining than LEAN and [41] at the end of the day. Since LEAN does not optimize the energy consumption for serving the requests, the average battery level for this scheme will be lower than that of Ge-LATA. Though [41] aims to optimize the power consumed in processing the tasks, it prefers the nearest cloudlet to process the incoming task if it can be completed before the deadline. As mentioned before,

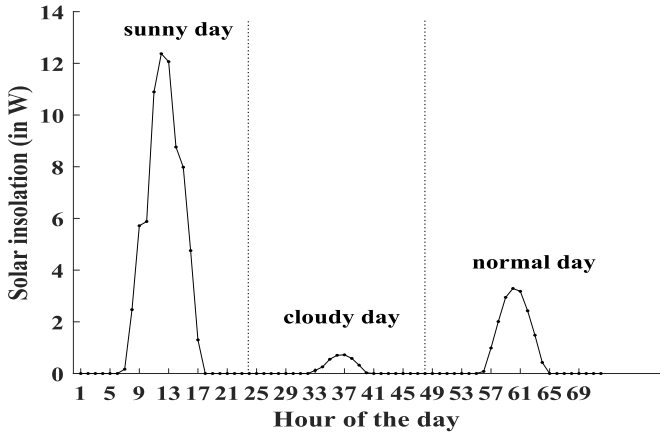

Fig. 6. Solar insolation profile on a 20 W solar panel for sunny, cloudy and normal days on consecutive days considered for the simulation.

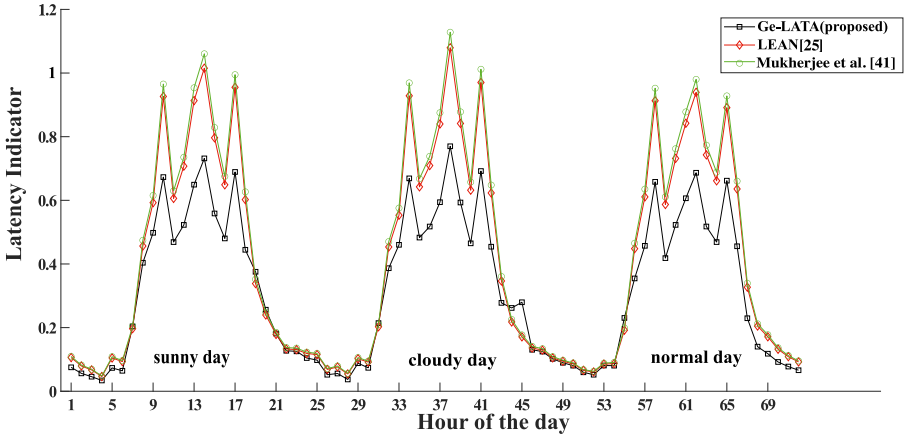

Fig. 7. Latency variation for three consecutive days where solar profile is sunny, cloudy and normal.

since [41] does not consider the load on the cloudlet comprehensively, it consumes higher amounts of green energy resources of the cloudlets.

Ge-LATA optimizes latency and energy consumption and has robust performance even when solar insolation is low. To demonstrate this, we investigated a situation how low solar insolation affects Ge-LATA compared to other schemes. We considered a scenario where the solar insolation was different on three consecutive days.

The first day was when the solar insolation was very good, i.e., it was a sunny day, and the solar insolation data was taken from the NREL database for a day in March. The second day had lower solar insolation than the first day, i.e., it was a cloudy day, and the solar insolation data for this day was taken from the database for a day in February. The third day was a normal day (i.e., neither sunny nor cloudy) where the data corresponds to a day in October taken from the database. The solar insolation for these three days as received on the PV panel is shown in Fig. 6. The traffic profile for the incoming task requests was considered to be the same as in Fig. 2. Fig. 7 shows the latency performance for three consecutive days for the proposed Ge-LATA, LEAN, and [41]. We see from Fig. 7 that the latency for Ge-LATA is much lesser than the LEAN and [41]. Ge-LATA shows a maximum of 31.7% and 28.64% lower latency than [41] and LEAN respectively, which was observed on a cloudy day. This behavior is attributed to the better consideration of cloudlet load by the Ge-LATA scheme while offloading the requests, as mentioned before. Fig. 8 shows the average battery levels of the cloudlets for these three days. Again, we can see a clear advantage of Ge-LATA over the other two schemes showing its superior ability to manage green energy resources. Quantitatively. Ge-LATA shows a maximum of 50.15% higher average battery levels than the other two schemes.

8. Conclusions and future work

We have proposed a novel task-assignment scheme named Ge-LATA, which optimizes the green energy consumed and the latency experienced by the mobile users in offloading their requests to the green energy-powered cloudlets. These cloudlets form a network to serve the request cooperatively such that the request offloaded onto one cloudlet can also be served by other cloudlets. Ge-LATA

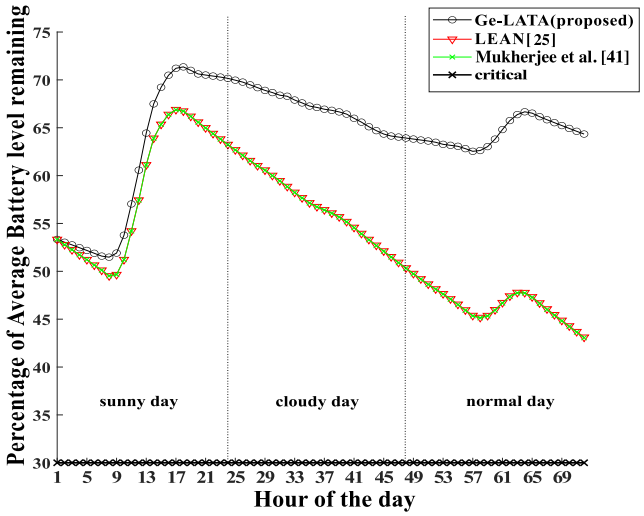

Fig. 8. Average battery level for three consecutive days where solar profile is sunny, cloudy and normal.

considers network elements such as current cloudlet load, the green energy available with each cloudlet, its maximum service rate, and its distance from the mobile user while assigning the offloaded tasks. Thus, it leads to optimal task-assignment, which minimizes the latency and green energy deficiency, leading to better performance than competing task-assignment schemes. The optimality of Ge-LATA was proved mathematically. The performance of Ge-LATA was compared to other task-assignment schemes through simulations that used the actual solar insolation data taken from the NREL database. In these simulations, different solar insolation scenarios were considered to analyze the efficiency (in terms of latency and consumed green energy) and robustness of Ge-LATA compared to other candidate/competing task-assignment schemes. These performance evaluations show that Ge-LATA achieves lower latency while consuming lesser green energy than other comparable task-assignment schemes. Thus, these results prove that Ge-LATA is suitable for making task-assignment decisions in Green Cloudlet Network (GCNs).

8.1. Limitations

In this work, we have assumed that the mobile device offloading the task at cloudlet will be associated with the same cloudlet till the offloaded task is processed by the GCN. However, to support rapid mobility-oriented applications such as smart transport systems involving vehicles, we need a dynamic approach wherein requests are directed to nearest cloudlets. Therefore, results of earlier offloaded tasks (processed by previously assigned cloudlet) need to be routed to the current cloudlet. The work on incorporating such dynamic routing mechanisms will be explored as part of the future work.

8.2. Future work

We envisioned the future work as a succession of this work in two domains. First, is the theoretical domain and the second is the real world application domain. In the theoretical domain, as mentioned above, we will consider the mobility of the users from the start of the request to the end of the process dynamically. Such consideration will factor in the cloudlet associated with the mobile user when offloading the task request is completed which may differ from the associated cloudlet at the beginning of the request. In such scenarios, we have to consider additional mechanisms to route the results of the offloaded tasks to the cloudlet with which the mobile user is currently associated.

In the real-world applications domain, the insights developed from these optimal system models and dynamic granularity can be employed in the disaster risk reduction scenarios where the multiple users of mobile can be serviced by providing precise notification of disaster such as flood or bushfire. In robotics, to perform path planning and navigation in real time, robots such as mobile service robots or drones need to process a big quantity of data (i.e. video, laser) in real time. However, the computational power embarked on these robots is limited due to several constraints (weight, power supply). Besides, these mobile robots have often a limited battery life and need to process data until they reach their next charging station. If a robot fails, human operators also need to be informed of the robot's last location and status. In these scenarios, Green Cloudlet Networks can be helpful to rely on green computing. It can also provide a secure way for the remote operator to access the robot's last logs even in the instance of a robot not responding anymore.

Future work also includes validation of Ge-LATA on a discrete event simulator like CloudSim [50] and practical realization using systems such as FogBus [51]. This will give more insights about the limitations and validity conditions of Ge-LATA.

Acknowledgments

This work has been funded by the grants received from Melbourne-Chindia Cloud Computing (MC3) Research Network and Australian Research Council (ARC). This work was also supported by DST-SERB [Science and Engineering Research Board (SERB)] funding under Grant ECR/2018/0001479 and OPERA Award, BITS-Pilani (funding reference FR/SCM/10-Jun-19/EEE).

References

[1] R. Buyya, C.S. Yeo, S. Venugopal, J. Broberg, I. Brandic, Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility, Future Gener. Comput. Syst. 25 (6) (2009) 599–616.
[2] K. Kumar, Y. Lu, Cloud computing for mobile users: Can offloading computation save energy? Computer 43 (4) (2010) 51–56.
[3] C. Stergiou, K.E. Psannis, B.-G. Kim, B. Gupta, Secure integration of IoT and cloud computing, Future Gener. Comput. Syst. 78 (2018) 964–975.
[4] U. KC, S. Garg, J. Hilton, J. Aryal, N. Forbes-Smith, Cloud computing in natural hazard modeling systems: Current research trends and future directions, Int. J. Disaster Risk Reduct. 38 (2019) 101188.
[5] S. Garg, J. Aryal, H. Wang, T. Shah, G. Kecskemeti, R. Ranjan, Cloud computing based bushfire prediction for cyber–physical emergency applications, Future Gener. Comput. Syst. 79 (2018) 354–363.
[6] A. Khayer, M.S. Talukder, Y. Bao, M.N. Hossain, Cloud computing adoption and its impact on SMEs' performance for cloud supported operations: A dual-stage analytical approach, Technol. Soc. 60 (2020) 101225.
[7] U. K.C., S. Garg, J. Hilton, J. Aryal, N. Forbes-Smith, Cloud computing in natural hazard modeling systems: Current research trends and future directions, Int. J. Disaster Risk Reduct. 38 (2019) 101188.
[8] M. Qiu, Z. Ming, J. Wang, L.T. Yang, Y. Xiang, Enabling cloud computing in emergency management systems, IEEE Cloud Comput. 1 (4) (2014) 60–67.
[9] S. Mehrban, M.W. Nadeem, M. Hussain, M.M. Ahmed, O. Hakeem, S. Saqib, M.L.M. Kiah, F. Abbas, M. Hassan, M.A. Khan, Towards secure FinTech: A survey, taxonomy, and open research challenges, IEEE Access 8 (2020) 23391–23406.
[10] M. Satyanarayanan, The emergence of edge computing, Computer 50 (1) (2017) 30–39.
[11] W. Shi, J. Cao, Q. Zhang, Y. Li, L. Xu, Edge computing: Vision and challenges, IEEE Internet Things J. 3 (5) (2016) 637–646.
[12] P. Mach, Z. Becvar, Mobile edge computing: A survey on architecture and computation offloading, IEEE Commun. Surv. Tutor. 19 (3) (2017) 1628–1656.
[13] G. Muhammad, M.F. Alhamid, M. Alsulaiman, B. Gupta, Edge computing with cloud for voice disorder assessment and treatment, IEEE Commun. Mag. 56 (4) (2018) 60–65.
[14] L. Xiao, X. Lu, T. Xu, X. Wan, W. Ji, Y. Zhang, Reinforcement learning-based mobile offloading for edge computing against jamming and interference, IEEE Trans. Commun. 68 (10) (2020) 6114–6126.
[15] M.R. Hossain, M. Whaiduzzaman, A. Barros, S.R. Tuly, M.J.N. Mahi, S. Roy, C. Fidge, R. Buyya, A scheduling-based dynamic fog computing framework for augmenting resource utilization, Simul. Model. Pract. Theory 111 (2021) 102336.
[16] 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.
[17] K. Gai, M. Qiu, H. Zhao, L. Tao, Z. Zong, Dynamic energy-aware cloudlet-based mobile cloud computing model for green computing, J. Netw. Comput. Appl. 59 (2016) 46–54.
[18] 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.
[19] International Energy Agency, World gross electricity production, by source, 2017, 2019, [Online]. Available: https://www.iea.org/data-and-statistics/charts/world-gross-electricity-production-by-source-2017.
[20] E. Masanet, A. Shehabi, N. Lei, S. Smith, J. Koomey, Recalibrating global data center energy-use estimates, Science 367 (6481) (2020) 984–986, [Online]. Available: https://www.science.org/doi/abs/10.1126/science.aba3758.
[21] X. Sun, N. Ansari, Green cloudlet network: A sustainable platform for mobile cloud computing, IEEE Trans. Cloud Comput. 8 (1) (2020) 180–192.
[22] X. Sun, N. Ansari, Green cloudlet network: A distributed green mobile cloud network, IEEE Netw. 31 (1) (2017) 64–70.
[23] N.R.E.L. (NREL), Solar resource data and tools, 2015, [Online]. Available: http://www.nrel.gov/rredc/solar_data.html.
[24] S.C. G., V. Chamola, C.-K. Tham, G. S., N. Ansari, An optimal delay aware task assignment scheme for wireless sdn networked edge cloudlets, Future Gener. Comput. Syst. 102 (2020) 862–875.
[25] X. Sun, N. Ansari, Latency aware workload offloading in the cloudlet network, IEEE Commun. Lett. 21 (7) (2017) 1481–1484.
[26] 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.
[27] V. Chamola, C.K. Tham, G.S.S. Chalapathi, Latency aware mobile task assignment and load balancing for edge cloudlets, in: 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), 2017, pp. 587–592.
[28] M. Ali, N. Riaz, M.I. Ashraf, S. Qaisar, M. Naeem, Joint cloudlet selection and latency minimization in fog networks, IEEE Trans. Ind. Inf. 14 (9) (2018) 4055–4063.
[29] C. world, Why data centres are the new frontier in the fight against climate change, 2019, [Online]. Available: https://www.computerworld.com/article/3431148/why-data-centres-are-the-new-frontier-in-the-fight-against-climate-change.html.
[30] J. Vidal, 'Tsunami of data' could consume one fifth of global electricity by 2025, 2017, [Online]. Available: https://www.climatechangenews.com/2017/12/11/tsunami-data-consume-one-fifth-global-electricity-2025/.
[31] W.C. Association, Coal & electricity, 2020, [Online]. Available: https://www.worldcoal.org/coal/uses-coal/coal-electricity.
[32] 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.
[33] M. Min, L. Xiao, Y. Chen, P. Cheng, D. Wu, W. Zhuang, Learning-based computation offloading for IoT devices with energy harvesting, IEEE Trans. Veh. Technol. 68 (2) (2019) 1930–1941.
[34] Z. Wei, B. Zhao, J. Su, X. Lu, Dynamic edge computation offloading for internet of things with energy harvesting: A learning method, IEEE Internet Things J. 6 (3) (2019) 4436–4447.
[35] W. Zhang, Z. Zhang, S. Zeadally, H.-C. Chao, V.C.M. Leung, Masm: A multiple-algorithm service model for energy-delay optimization in edge artificial intelligence, IEEE Trans. Ind. Inf. 15 (7) (2019) 4216–4224.
[36] K. Wang, Z. Ding, D.K.C. So, G.K. Karagiannidis, Stackelberg game of energy consumption and latency in MEC systems with NOMA, IEEE Trans. Commun. 69 (4) (2021) 2191–2206, http://dx.doi.org/10.1109/TCOMM.2021.3049356.
[37] R. Yadav, W. Zhang, O. Kaiwartya, H. Song, S. Yu, Energy-latency tradeoff for dynamic computation offloading in vehicular fog computing, IEEE Trans. Veh. Technol. 69 (12) (2020) 14198–14211.
[38] W. Chen, D. Wang, K. Li, Multi-user multi-task computation offloading in green mobile edge cloud computing, IEEE Trans. Serv. Comput. 12 (5) (2019) 726–738.

[39] J. Xu, L. Chen, S. Ren, Online learning for offloading and autoscaling in energy harvesting mobile edge computing, IEEE Trans. Cogn. Commun. Netw. 3 (3) (2017) 361–373.
[40] M. Alfaqawi, M.H. Habaebi, M.R. Islam, M.U. Siddiqi, Energy harvesting network with wireless distributed computing, IEEE Syst. J. 13 (3) (2019) 2605–2616.
[41] 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.
[42] D.G. Roy, D. De, A. Mukherjee, R. Buyya, Application-aware cloudlet selection for computation offloading in multi-cloudlet environment, J. Supercomput. 73 (4) (2017) 1672–1690.
[43] 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.
[44] V. Chamola, B. Krishnamachari, B. Sikdar, Green energy and delay aware downlink power control and user association for off-grid solar-powered base stations, IEEE Syst. J. 12 (3) (2018) 2622–2633.
[45] 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.
[46] 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.
[47] V. Chamola, B. Krishnamachari, B. Sikdar, An energy and delay aware downlink power control strategy for solar powered base stations, IEEE Commun. Lett. 20 (5) (2016) 954–957.
[48] Raspberry Pi Foundation, Raspberry Power Supply, [Online]. Available: https://www.raspberrypi.org/documentation/hardware/raspberrypi/power/README.md.
[49] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge Univ. Press, Cambridge, U.K., 2004.
[50] R.N. Calheiros, R. Ranjan, A. Beloglazov, C.A.F. De Rose, R. Buyya, Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms, Softw. - Pract. Exp. 41 (1) (2011) 23–50.
[51] S. Tuli, R. Mahmud, S. Tuli, R. Buyya, Fogbus: A blockchain-based lightweight framework for edge and fog computing, J. Syst. Softw. 154 (2019) 22–36.