The First International Workshop on Smart Edge Computing and Networking 2017
Part of IEEE PerCom 2017

Latency Aware Mobile Task Assignment and Load Balancing for Edge Cloudlets

Vinay Chamola[1], Chen-Khong Tham[1] and G. S. S Chalapathi[2]
[1]Department of Electrical and Computer Engineering, National University of Singapore, Singapore
[2] Department of EEE, BITS-Pilani, Pilani, Rajasthan, India

***Abstract*—With the various technological advances, mobile devices are not just being used as a means to make voice calls; but are being used to accomplish a variety of tasks. Mobile devices are being envisioned to practically accomplish any task which could be done on a computer. This is hurdled by the limited computational resources available with the mobile devices due to their portable size. With the mobile devices being connected to the Internet, leveraging cloud services is being seen as a promising solution to overcome this hurdle. Computationally intensive tasks can be offloaded to the Cloud servers. However, owing to the latency and cost associated with using cloud services, edge devices (termed cloudlets) stationed near the mobile devices are being seen as a prospective alternative to replace/assist the Cloud services. The mobile devices have an easier access to the cloudlets being situated in their vicinity and can offload their task requests to them to be served at a lower cost. This paper considers a network of such connected cloudlets which provide service to the mobile devices in a given area. We address the issue of task assignment in such a scenario (i.e. which cloudlet serves which mobile device) aimed towards improving the quality of service experienced by the mobile devices in terms of minimizing the latency. Through numerical simulations we demonstrate the performance gains of the proposed task assignment scheme showing lower latency as compared to the traditional scheme for task assignment.**

I. Introduction

With the rapid advances being made in mobile computing, mobile devices are now able to perform a wider variety of tasks as compared to that they could do a decade back. In fact, many of the tasks which could be done earlier on computer can now be performed on the mobile devices (like making video calls, playing games etc.). However, powerful applications necessitate the need of high computational power owing to their requirement of processing data and involving other computationally intensive operations. The limitation on the size of the mobile device (and hence the allowable storage, power and computational resources) comes forth as a challenge in running such powerful applications on the mobile device. However, since last few years, Cloud services have started gaining popularity owing to their facilitating offloading the computation-intensive applications of various the mobile devices and allowing them to leverage many diverse applications [1]-[3]. Many cloud service providers like Amazon and Microsoft have come up with tailor-made facilities which cater to the diverse needs of such mobile devices for variety of computational tasks. The large storage and computational capability at the cloud servers of such cloud service providers enables them to quickly execute the computational tasks offloaded by the mobile devices. Many of these tasks if performed locally on the device would had taken hours for the mobile device, not just reserving its computational resources but draining their battery too. Note that the process of mobile device offloading its request to the cloud server and receiving the result back is generally done over an internet connection (like that accessed by the mobile device through the 3G internet, Wi-Fi etc.). Having mentioned the various benefits which cloud services offer, there are several challenges and factors which discourage its use, some of which are mentioned as follows. Leveraging cloud services requires the mobile devices connectivity to the Internet which may be at times limited. Additionally, the cost-associated with the use of such services is also at times unprofitable for pursuing them. In absence of dedicated Wi-Fi, the user may have to incur the cost of sending the request to cloud and receiving the results over cellular network adding to his mobile billing in addition to the charges for using the cloud services. Additionally, the applications face a high latency while being served by such services due to the cloud servers being positioned very far. The above mentioned concerns have thus given birth to a novel concept popularly known in the recent literature as Edge or Fog Computing, where the above mentioned tasks which were done by the cloud are rather carried out on a computationally powerful device present in the vicinity of the mobile device. As compared to the Cloud servers, these edge/fog devices (commonly known as cloudlets [4]) are smaller in size, cheaper, easy to install, serve a fewer user requests and are computationally less powerful. These cloudlets provide an additional layer between the mobile devices and the Cloud services and can alleviate the various problems which using the Cloud services as standalone poses.

In this paper we consider a network of such edge/fog devices which are termed as cloudlets (small clouds) throughout this paper. These cloudlets are networked together to provide an infrastructure for serving tasks offloaded by nearby mobile devices [5]. The computational resources of the cloudlets are used to serve the requests offloaded by the users. Note that in a traditional cloudlet setting, the service requests offloaded by the mobile device is served by the nearest cloudlet. However, such an approach is certainly not so efficient for it may be possible that at times few cloudlets may be overloaded whereas the others may be very lightly loaded. Users associated with a cloudlet having very high load thus face poor quality of service in terms of a high latency for fulfilment of the requested tasks [6]. However, this problem can be overcome if the

978-1-5090-4338-5/17/$31.00 ©2017 IEEE

cloudlets are connected to each other through SDN (Software defined network) switches, enabling them to process the tasks offloaded by the mobile devices in a cooperative fashion, thus better utilizing the available computational resource pool. This paper addresses the problem of task assignment in such a setup where the task assignment to the cloudlets is done with the aim to reduce the latency experienced by the mobile devices for processing of the offloaded tasks. We propose an optimal scheme for task assignment and compare its performance against that achieved using the traditionally existing scheme.

The remainder of this paper is organized as follows. Section II discusses related work. In section III, we describe the system model. In Section IV, we present the problem formulation followed by the solution methodology which is presented in Section V. Section VI presents the simulation results. We conclude in Section VII.

II. Literature Review

Due to the various limitations and challenges associated with using cloud services as discussed in the previous section, the concept of replacing/assisting them with cloudlets has been proposed in existing literature [4]. The concept of cloudlets has found its applicability in solving a wide variety of real time problems which were initially introspected to be addressed by cloud services. Recent papers consider the use of cloudlets in variety of scenarios like in the area of smart grid where the cloudlets process data generated by grid devices and sensors to generate useful insights and operational actions, in smart traffic light operations etc. [7]. Cloudlet networks implementing software defined networks (SDN) in a vehicular network scenario are discussed in [8]. Software defined networks separate the control and communication layers. In SDN the centralized server takes care of the control and the nodes decide their communication path as dictated by the server.

An application area where the use of cloudlets has found potential applicability is in the area of mobile device task offloading where computation intensive tasks on mobile devices can be offloaded to nearby cloudlets. Recent works on task offloading from a mobile device to cloudlets have focused on applications like decision making, language processing, voice/image recognition and mission planning. Examples of such applications include the popular face recognition task offloading studied in [9]. Another example is of Apple's Siri that facilitates a user to use his voice for making phone calls, to send messages etc. Cloudlets can be used in such tasks for voice recognition and translation to appropriate actions. Cloudlets can also coordinate with each other with a goal of improving a particular system wide metric [9]. Cloudlet placement issues have also been explored in recent works like [10] and [11]. An another class of work in the area of task offloading to cloudlets have focused on developing algorithms for addressing vital questions centring around whether a mobile device should offload or not, which cloudlet should it offload to, how much of the task should it offload etc. The need to address these questions in context of task offloading arise due to various factors like device mobility,

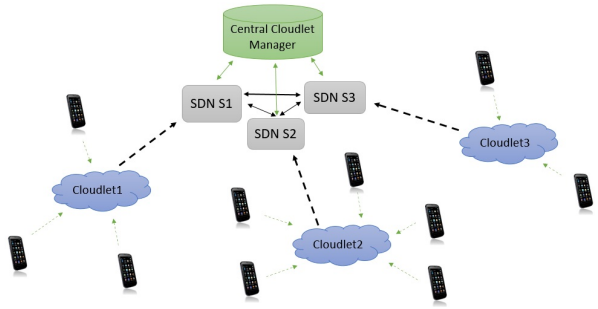

Fig. 1. System model consisting of the various network elements.

connectivity options, energy limitation, cost constraints and the need for task distribution [12]. Some early works addressing these issues include [13] and [14]. The authors in [15] propose dynamic offloading decision model for mobile devices where the application could be partitioned into phases and to offload it in parts to nearby devices/cloudlets to process at the earliest. [16] extends that work with an aim of minimizing the processing cost by using a dynamic opportunistic offloading algorithm that accounts for the device mobility also. However, these works consider cloudlets which are more or less standalone and are capable of primarily serving the mobile devices in their vicinity. However, this paper considers a network of cloudlets interconnected through SDN switches where they can cooperatively serve the task offload requests of the mobile devices. By doing so, the mobile devices can be offered a lower latency, improving their quality of service experience.

III. System Description

Fig. 1 shows the system considered in this paper. We assume that all the cloudlets are connected through a SDN switched network. Thus in our system model, if a cloudlet is overloaded, the tasks offloaded to it by a nearby mobile device can be processed on another cloudlet which is not so heavily loaded. The decision of which clouldet serves the task offloaded by a mobile device is made by a central cloudlet manager which is present at the SDN network core. We consider a network of cloudlets in a geographical region $\mathcal{R}$. We denote the set of the cloudlets as $\mathcal{C}$ with $\mathcal{C} = \{C_1, C_2, \cdots, C_j, \cdots, C_{|\mathcal{C}|}\}$ where C_j refers to the j-th cloudlet. We denote the maximum service rate which can be offered by the j-th cloudlet as S_j^{max}. The mobile device locations are denoted by $x \in \mathcal{R}$. We assume that mobile offload task requests at location x arrive following a Poisson point process with arrival rate $\lambda(x)$ per unit area and an average file size of $\tau(x)$. We define the mobile task offload density at the location x as $\gamma(x) = \lambda(x)\tau(x)$. Note that $\gamma(x)$ captures the spatial task offload variability. The mobile devices offload their service requests to the nearby cloudlets via the wifi connection. The offloaded request can be served either at the very same cloudlet or some other cloudlet as decided by the central cloudlet controller. Note that the latency experienced by the mobile device for the execution of the service request offloaded by the mobile device depends on various factors like a. the maximum service rate offered by the cloudlet processing its request, b. the load at the cloudlet serving its

request (which captures the queuing delay experienced by the mobile device request) and c. the distance of the mobile device from the cloudlet serving its request (which captures the communication latency and other network overheads). To simplify our analysis, we consider the service rate offered by a cloudlet j to the mobile device at location x as

$$s_j(x) = \frac{S_j^{max}}{1 + \beta(dis(x, C_j))^{\alpha}} \tag{1}$$

where $dis(x, C_j)$ is the Cartesian distance between the mobile device at location x and the j-th cloudlet. α and β are parameters which give the flexibility to adjust the service rate to accommodate a wide variety of network scenarios (in terms of the effect of service rate as a function of the distance between the mobile device and the cloudlet serving its offloaded service request). The intuition behind the above formulation is as follows. The service rate which a cloudlet is able to offer to the mobile device is proportional to its maximum service rate (owing to the particular cloudlet's hardware configuration). Also, it is inversely proportional to the distance of the cloudlet from the mobile device. Next, let us introduce a task assignment indicator function $u_j(x)$ that specifies task assignment relationship between the cloudlets and the mobile devices. This value is 1 if cloudlet j serves the mobile device at location x, and is 0 otherwise. We now define the cloudlet load ρ_j, which denotes the fraction of time the cloudlet j is busy serving its traffic requests and is given by [17]

$$\rho_j = \int_{\mathcal{R}} \frac{\gamma(x)}{s_j(x)} u_j(x) dx. \tag{2}$$

Definition 1: The feasible set of the cloudlet loads $\boldsymbol{\rho} = (\rho_1, \cdots, \rho_{|\mathcal{C}|})$ is denoted by $\mathcal{V}$ and is defined as

$$\mathcal{V} = \Big\{ \boldsymbol{\rho} \mid \rho_j = \int_{\mathcal{R}} \frac{\gamma(x)}{s_j(x)} u_j(x) dx, \ \ 0 \le \rho_j \le 1 - \epsilon, \ \ \forall j \in \mathcal{C}, \\ u_j(x) \in \{0, 1\}, \sum_{j=1}^{|\mathcal{C}|} u_j(x) = 1, \ \ \forall j \in \mathcal{C}, \ \ \forall x \in \mathcal{R} \Big\},$$

where ϵ is an arbitrarily small positive constant.

We assume that a task offloaded by a mobile device is entirely served by one among the different cloudlets in the network. The mobile devices attach to the closest cloudlet, however the cloudlet on which its request is served is based on the scheme described later in the paper in Section V. Since task offload arrivals are Poisson processes, the sum of task offload transfer arrival flows at the cloudlet can also be concluded to be a Poisson process. Since the service process at a cloudlet follows a general distribution, we model the cloudlets as a M/G/1-PS(processor sharing) queue. The average number of flows at cloudlet j can thus be given by $\frac{\rho_j}{1-\rho_j}$ [17]. From Little's law, we know the latency experienced by a traffic flow to be proportional to the average number of flows in the system [18]. Hence, we consider the total number of flows at a given cloudlet as the latency indicator of the j-th cloudlet, $\mathcal{L}_j(\rho_j)$, which is given by [17]

$$\mathcal{L}_j(\rho_j) = \frac{\rho_j}{1 - \rho_j}. \tag{3}$$

Note that as the ρ_j value in the above expression increases, the latency increases exponentially approaching ∞ when ρ_j tends to 1. It is also to be noted that the above mentioned metric does not have any units but is just a relative indicator of the system latency. The indicator above has been widely used in several contemporary studies like [19] and [20] to quantify the system latency performance.

IV. Problem Formulation

We consider the problem, [P1], which seeks to minimize the total network wide latency during a given time instant. The problem can be formulated as

$$\textbf{[P1]} \ \underset{\rho}{\text{minimize}} \quad \mathcal{M}(\boldsymbol{\rho}) = \sum_{j=1}^{|\mathcal{C}|} \mathcal{L}_j(\rho_j) \\ \text{subject to:} \quad \boldsymbol{\rho} \in \mathcal{V}$$

This optimization problem is solved by the central cloudlet manager. It decides task assignment for the offloaded service requests from the mobile devices i.e. which cloudlet will serve which mobile device, so as to improve the overall quality of service experience of the mobile devices. Note that solving the above problem ensures best overall latency performance for the network. The problem is solved by performing load balancing among the cloudlets as discussed in the next section (i.e. adjusting the cloudlet loads (ρ)).

V. Optimal task assignment policy

This section presents our proposed task assignment policy which is aimed at achieving the optimal solution for the problem [P1] formulated in the previous section.

Since $u_j(x) \in \{0, 1\}$, the set $\mathcal{V}$ is not convex. Thus, in order to transform problem [P1] to a convex optimization problem, we begin with relaxing this constraint to $0 \le u_j(x) \le 1$. Note that with such a relaxation, $u_j(x)$ could be thought of as the probability of the task offloaded by mobile device at location x being served by cloudlet j. The relaxed set of cloudlet loads, $\tilde{\mathcal{V}}$, is thus as follows

$$\tilde{\mathcal{V}} = \Big\{ \boldsymbol{\rho} \mid \rho_j = \int_{\mathcal{R}} \frac{\gamma(x)}{s_j(x)} u_j(x) dx, \ \ 0 \le \rho_j \le 1 - \epsilon, \ \ \forall j \in \mathcal{C}, \\ 0 \le u_j(x) \le 1, \sum_{j=1}^{|\mathcal{C}|} u_j(x) = 1, \ \ \forall j \in \mathcal{C}, \ \ \forall x \in \mathcal{R} \Big\}.$$

Note that the above-mentioned set $\tilde{\mathcal{V}}$ is convex. The authors in [21] have proved its convexity. With the above relaxation applied to problem [P1], we obtain the transformed problem [P2] which is as follows

$$\textbf{[P2]} \ \underset{\boldsymbol{\rho} \in \tilde{\mathcal{V}}}{\text{minimize}} \quad \mathcal{M}(\boldsymbol{\rho}) = \sum_{j=1}^{|\mathcal{C}|} \mathcal{L}_j(\rho_j).$$

The optimization problem [P2] is formulated using $\tilde{\mathcal{V}}$. However, the task assignment algorithm that is proposed in later part of this section gives deterministic task assignment (which belongs to $\mathcal{V}$). Theorems 1 and 2 show the same.

Next, we describe the working of the task assignment algorithm. The cloudlets periodically evaluate their traffic loads which is used by them to estimate a variable (termed resistance index in our work) which is advertised to the central cloudlet manager. The central cloudlet manager chooses which cloudlet to assign the task with the aim of minimizing the value of the objective function in [P2]. For convergence of the proposed scheme, it is assumed that the time scale on which the cloudlets broadcast their resistance indicators is slower as compared to the scale of the traffic arrival and departure processes. This assumption makes sure that the central cloudlet manager is able to make the task offload decisions for the resistance indexes currently broadcast by the cloudlets before the next set of resistance index is broadcast from the cloudlets. The cloudlets are assumed to be synchronized, broadcasting the resistance indexes to the central manager at the same time.

Now, we describe the central cloudlet manager and the cloudlet side algorithms which carry out the above described task offloading scheme.

1) **Task assignment algorithm at cloudlet manager:** In our algorithm, the time between two subsequent resistance index updates is defined as a time slot. At the beginning of the k-th time slot, the cloudlets broadcast their resistance index to the cloudlet manager. The mobile devices at location x are assigned to the cloudlets based on the resistance indexes broadcast by the cloudlets and their offered service rate. We use superscript k to represent a particalar variable's value at the start of time slot k. The resistance index broadcast by cloudlet j is given by

$$\psi_j^k = \frac{\partial \mathcal{L}_j^k(\boldsymbol{\rho})}{\partial \rho_j^k} = \frac{1}{(1-\rho_j^k)^2} \tag{4}$$

The mobile devices are assigned to the cloudlets based on the function below

$$u_j^k(x) = \begin{cases} 1 & \text{if } j = \arg\max_{j \in \mathcal{C}} \frac{s_j(x)}{\psi_j^k} \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

As described in Section III, the association indicator $u_j(x)$ captures the information whether the cloudlet j serves the request offloaded by the mobile device at location x, and $s_j(x)$ indicates the service rate offered by the j-th cloudlet to the mobile device at location x. Note that the computational complexity associated with each task assignment is $O(|\mathcal{C}|)$.

2) **Cloudlet side algorithm:** The cloudlets evaluate their load at the end of the k-th time slot, $T_j(\rho_j^k)$, which is given by

$$T_j(\rho_j^k) = \min\left(\int_{\mathcal{R}} \frac{\gamma(x)}{s_j(x)} u_j(x) dx, 1-\epsilon\right). \tag{6}$$

After measuring $T_j(\rho_j^k)$, the cloudlet updates its cloudlet load to be used for evaluating the subsequent resistance index to be broadcast to the central cloudlet manager as

$$\rho_j^{k+1} = \sigma \rho_j^k + (1-\sigma) T_j(\rho_j^k) \tag{7}$$

where $0 < \sigma < 1$ is an averaging factor.

Algorithm 1 The Task Assignment Algorithm

1: Central cloudlet manager: At the k-th iteration, the central cloudlet manager evaluates the service rate $s_j(x)$ ($\forall j \in \mathcal{C}, \forall x \in \mathcal{R}$) and receives the cloudlet resistance index broadcast by the cloudlets. The mobile devices are assigned the cloudlet $u^k(x)$ according to Equation (5).

2: Cloudlets: At the end of the k-th iteration, each cloudlet estimates its load and updates its traffic load $\rho_j^{(k+1)}$ to be used for evaluating the resistance index to be broadcast for the next iteration.

Next, we prove the optimality and convergence of the above mentioned task assignment algorithm. First of all, we show the objective function $\mathcal{M}$ to be convex in $\rho \in \tilde{\mathcal{V}}$ which ensures that there is an unique optimal task assignment that leads to objective function minimization.

Lemma 1: *The objective function $\mathcal{M}(\boldsymbol{\rho})$ is convex in $\boldsymbol{\rho}$ when $\boldsymbol{\rho}$ is defined on $\tilde{\mathcal{V}}$.*

Proof. We prove this by showing that $\nabla^2 \mathcal{M}(\boldsymbol{\rho}) > 0$. The objective function written in terms of $\boldsymbol{\rho}$ is given as

$$\mathcal{M}(\boldsymbol{\rho}) = \sum_{j=1}^{|\mathcal{C}|} \mathcal{L}_j(\rho) = \sum_{j=1}^{|\mathcal{C}|} \frac{\rho_j}{1-\rho_j} \tag{8}$$

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

$$\nabla \mathcal{M}(\boldsymbol{\rho}) = \sum_{j=1}^{|\mathcal{C}|} \frac{1}{(1-\rho_j)^2} \tag{9}$$

$$\nabla^2 \mathcal{M}(\boldsymbol{\rho}) = \sum_{j=1}^{|\mathcal{C}|} \frac{2}{(1-\rho_j)^3} \tag{10}$$

The derivatives evaluated above are non-negative due to $\frac{2}{(1-\rho_j)^3}$ being non-negative for all cloudlets. This proves the convexity of the objective function. ■

As the objective function is convex, a natural implication from the same is that there exists an unique optimal task assignment corresponding to the optimal load $\boldsymbol{\rho}^* \in \tilde{\mathcal{V}}$ which minimizes the objective function $\mathcal{M}(\boldsymbol{\rho}) = \sum_{j=1}^{|\mathcal{C}|} \mathcal{L}_j(\rho_j)$. We next prove convergence property of the proposed task assignment algorithm. We begin with proving that $T_j(\boldsymbol{\rho}^k)$ gives a descent direction for $\mathcal{M}(\boldsymbol{\rho}^k)$ at $\boldsymbol{\rho}^k$ (shown in Lemma 2). Thus after some iterations the cloudlet load converges which is proved in Theorem 1. Further, in Theorem 2 we prove that the cloudlet load thus obtained minimizes the objective function $\mathcal{M}(\boldsymbol{\rho})$.

Lemma 2: *When $\boldsymbol{\rho}^k \neq \boldsymbol{\rho}^*$, then $T_j(\boldsymbol{\rho}^k)$ provides a descent direction for $\mathcal{M}(\boldsymbol{\rho}^k)$ at $\boldsymbol{\rho}^k$.*

Proof. As the function $\mathcal{M}(\boldsymbol{\rho})$ is a convex function of ρ when ρ is defined in $\tilde{\mathcal{V}}$, this lemma can be easily proved by showing $< \nabla\mathcal{M}(\boldsymbol{\rho}^k), T(\boldsymbol{\rho}^k) - \boldsymbol{\rho}^k > \leq 0$ (with $< l, m >$ denoting the inner product of the vectors l and m). [22]. Let $u_j(x)$ and $u_j^T(x)$ be task assignment indicators that result in the cloudlet load ρ_j^k and $T(\rho_j^k)$. Then the inner product is given by

$$\begin{aligned}
&< \nabla\mathcal{M}(\boldsymbol{\rho}^k), T(\boldsymbol{\rho}^k) - \boldsymbol{\rho}^k > \\
&= \sum_{j=1}^{|\mathcal{C}|} \frac{1}{(1-\rho_j^k)^2} \left(T_j(\rho_j^k) - \rho_j^k\right) \\
&= \sum_{j=1}^{|\mathcal{C}|} \frac{1}{(1-\rho_j^k)^2} \left(\int_{\mathcal{R}} \frac{\gamma(x)(u_j^T(x) - u_j(x))}{s_j(x)} dx \right) \\
&= \int_{\mathcal{R}} \gamma(x) \sum_{j=1}^{|\mathcal{C}|} \left(\frac{\frac{1}{(1-\rho_j^k)^2}}{s_j(x)} (u_j^T(x) - u_j(x)) \right) dx.
\end{aligned}$$

Note that

$$\sum_{j=1}^{|\mathcal{C}|} \frac{\frac{1}{(1-\rho_j^k)^2}(u_j^T(x) - u_j(x))}{s_j(x)} \leq 0$$

holds because $u_j^T(x)$ from (5) maximizes the value of $\frac{s_j(x)}{\frac{1}{(1-\rho_j^k)^2}}$. Thus as a result we can claim that $< \nabla\mathcal{M}(\boldsymbol{\rho}^k), T(\boldsymbol{\rho}^k) - \boldsymbol{\rho}^k > \leq 0$ which proves the lemma. ■

In Theorem 1 and 2, we prove the convergence and optimality of the proposed task assignment scheme, respectively.

Theorem 1: *The cloudlet load vector $\boldsymbol{\rho}$ converges to the cloudlet load vector $\boldsymbol{\rho}^* \in \mathcal{V}$.*

Proof. To prove this, we show that $\boldsymbol{\rho}^{k+1} - \boldsymbol{\rho}^k$ is also a descent direction of $\mathcal{M}(\boldsymbol{\rho}^k)$. Considering the following expression we have

$$\begin{aligned}
\rho_j^{k+1} - \rho_j^k &= \sigma\rho_j^k + (1-\sigma)T_j(\rho_j^k) - \rho_j^k \\
&= (1-\sigma)(T(\rho_j^k) - \rho_j^k). \qquad (11)
\end{aligned}$$

In Lemma 2, we have already shown that $(T(\boldsymbol{\rho}^k) - \boldsymbol{\rho}^k)$ is the descent direction of $\mathcal{M}(\boldsymbol{\rho}^k)$ and furthermore we have $(1-\sigma) > 0$ due to $0 < \sigma < 1$. Therefore even $\boldsymbol{\rho}^{k+1} - \boldsymbol{\rho}^k$ gives the direction for descent of $\mathcal{M}(\boldsymbol{\rho}^k)$. Furthermore, as $\mathcal{M}(\boldsymbol{\rho}^k)$ has been shown to be convex, the convergence of $\mathcal{M}(\boldsymbol{\rho}^k)$ to $\boldsymbol{\rho}^*$ can be guaranteed. Suppose $\mathcal{M}(\boldsymbol{\rho}^k)$ converges not to $\mathcal{M}(\boldsymbol{\rho}^*)$, but some other point; then $\boldsymbol{\rho}^{k+1}$ again gives a descent direction so as to decrease $\mathcal{M}(\boldsymbol{\rho}^k)$. This is a contradiction to the convergence assumption. Additionally, as $\boldsymbol{\rho}^k$ is derived based on (5) where $u_j(x) \in \{0,1\}$, $\boldsymbol{\rho}^*$ belongs to set $\mathcal{V}$. ■

Theorem 2: *If the set $\mathcal{V}$ is non-empty and the cloudlet load vector $\boldsymbol{\rho}$ converges to $\boldsymbol{\rho}^*$, the task assignment corresponding to $\boldsymbol{\rho}^*$ minimizes $\mathcal{M}(\boldsymbol{\rho})$.*

Proof. Let $u^* = \{u_j^*(x) | u_j^*(x) \in \{0,1\}, \forall j \in \mathcal{C}, \forall x \in \mathcal{R}\}$ and $u = \{u_j(x) | u_j(x) \in \{0,1\}, \forall j \in \mathcal{C}, \forall x \in \mathcal{R}\}$ be the task assignment which corresponds to $\boldsymbol{\rho}^*$ and $\boldsymbol{\rho}$, with $\boldsymbol{\rho}$ being some cloudlet load vector such that $\boldsymbol{\rho} \in \mathcal{V}$. $\mathcal{M}(\boldsymbol{\rho})$ has been shown to be a convex function over $\boldsymbol{\rho}$, thus we show $< \nabla\mathcal{M}(\boldsymbol{\rho}^*), \Delta\boldsymbol{\rho} - \boldsymbol{\rho}^* > \geq 0$ to prove this theorem. Note that in the proof, we substitute $\frac{\partial\mathcal{M}(\boldsymbol{\rho}^*)}{\partial\rho_j^*}$ as $\psi_j(\rho_j^*)$ for notational clarity.

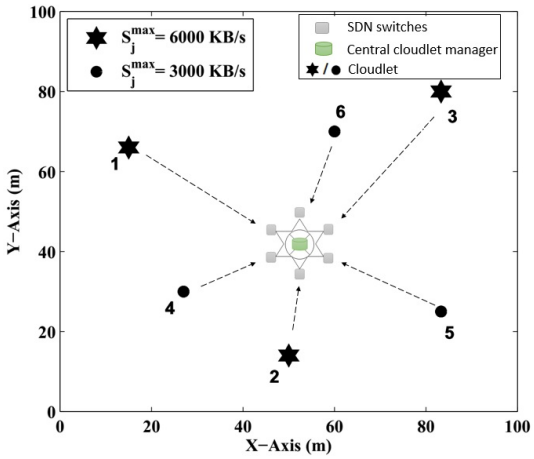

Fig. 2. Network topology considered for simulations.

$$\begin{aligned}
< \nabla\mathcal{M}(\boldsymbol{\rho}^*), \boldsymbol{\rho} - \boldsymbol{\rho}^* > &= \sum_{j=1}^{|\mathcal{C}|} \psi_j(\rho_j^*) (\boldsymbol{\rho} - \boldsymbol{\rho}^*) \\
&= \sum_{j=1}^{|\mathcal{C}|} \left(\int_{\mathcal{R}} \frac{\gamma(x)(u_j(x) - u_j^*(x))}{s_j(x)\psi_j^{-1}(\rho_j^*)} dx \right) \\
&= \int_{\mathcal{R}} \gamma(x) \sum_{j=1}^{|\mathcal{C}|} \frac{(u_j(x) - u_j^*(x))}{s_j(x)\psi_j^{-1}(\rho_j^*)} dx.
\end{aligned}$$

But as the optimal task assignment is decided according to

$$u_j^*(x) = \begin{cases} 1, & \text{if } j = \arg\max_{j \in \mathcal{C}} \frac{s_j(x)}{\psi_j(\rho_j^*)}, \\ 0, & \text{otherwise.} \end{cases},$$

we can say that,

$$\sum_{j=1}^{|\mathcal{C}|} \frac{u_j^*(x)}{s_j(x)\psi_j^{-1}(\rho_j^*)} \leq \sum_{j=1}^{|\mathcal{C}|} \frac{u_j(x)}{s_j(x)\psi_j^{-1}(\rho_j^*)}. \qquad (12)$$

Hence, $< \nabla\mathcal{M}(\boldsymbol{\rho}^*), \boldsymbol{\rho} - \boldsymbol{\rho}^* > \geq 0$ which proves the theorem. ■

VI. Simulation Results

For numerical simulations, we consider a cloudlet topology (shown in Fig. 2) with 6 randomly positioned clouldets providing service in a 100 m x 100 m area. These cloudlets are connected to each other through a SDN switched network with the central cloudlet manager in the SDN core. Note that we have considered three cloudlets which have a maximum service rate of 6000 KB/s (cloudlets 1, 2 and 3 in Fig. 2) whereas the other three cloudlets (cloudlets 4, 5 and 6) have a maximum service rate of 3000 KB/s. We use a homogeneous Poisson point process to generate the mobile task offload requests. We do performance analysis for different mobile device task offload arrival rates for the Poisson process with smallest number of task offload requests having an average of 20 task offload requests per second in the given coverage area and the largest number of offload requests with an average

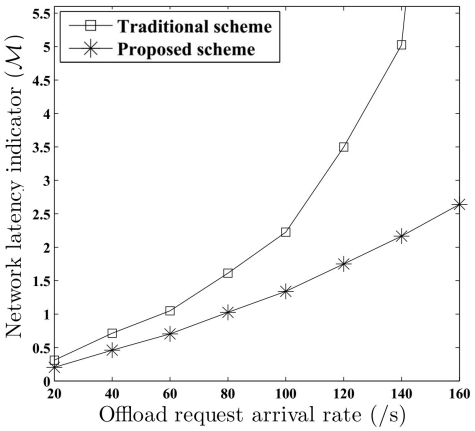

Fig. 3. Latency behaviour with different offload request arrival rates.

of 160 requests. For analytical simplicity, each task offload request is assumed to be associated with processing of 25 KB of data. The model discussed in section III is used to calculate the location based task offload density. We take the numerical values of α and β for Eqn. (1) as 1 and 10 respectively. The function $dis(.)$ gives the distance between the cloudlet and mobile device in km. The averaging factor for the cloudlet side algorithm in section V, σ, is taken to be 0.95. With this σ, the proposed task assignment algorithm is observed to converge to optimal solution in 10 iterations. For comparing the performance improvement, we compare the performance against the traditional scheme in which the tasks offloaded by the mobile devices are processed at the nearest cloudlet.

A. Network latency performance

In this section, we study how the network latency performance varies for different offload request arrival rates. The offload request arrival rates are varied from 20 to 160 offload requests/s and the network wide latency indicator ($\mathcal{M}$) is evaluated. Note that this metric does not have any units as discussed in Section III. From the results we can see that the performance of the proposed task assignment scheme is better than the traditional scheme offering a lower latency. Note that as the arrival rate of the task offload requests increases, the performance gain of the proposed scheme as compared to the traditional scheme becomes even more prominent. The reasoning behind such a behaviour is as follows. At higher offload request arrival rates, as the traditional scheme allocates the task processing to the nearest cloudlet; certain cloudlets having too many users in their vicinity get overloaded. This increases the overall network latency. However in the proposed task assignment scheme, the load at the cloudlets are also considered while assigning the tasks, thus giving better latency performance due to load balancing among the cloudlets.

VII. Conclusion

This paper proposed a framework for task assignment in a setup where a set of cloudlets offer its services to mobile devices which can offload their computationally intensive tasks to be executed on the cloudlets. The proposed task assignment framework allows the offloaded tasks to be served at appropriate cloudlets so as to ensure reduced latency in the network, thereby improving the quality of service experienced by the mobile device. Simulation results show the performance gains of the proposed framework in terms of reducing the network latency indicator showing a superior performance over the existing traditional approach.

VIII. Acknowledgement

This research/project is supported by the National Research Foundation, Prime Ministers Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme.

References

[1] E. F. Nakamura, A. A. Loureiro, and A. C. Frery, "Information fusion for wireless sensor networks: Methods, models and classifications," *ACM Comput. Surv.,* vol. 39, no. 3, p. 9–14, 2007.

[2] L. Xie, Y. Shi, Y. T. Hou, W. Lou, H. D. Sherali and S. F. Midkiff, "Bundling mobile base station and wireless energy transfer: Modeling and optimization," in Proc. *IEEE INFOCOM,* 2013.

[3] B. Chun, S. Ihm, P. Maniatis, M. Naik and A. Patti, "Clonecloud: elastic execution between mobile device and cloud," *Proc. EuroSys,* 2011.

[4] M. Satyanarayanan, P. Bahl, R. Caceres and N. Davies, "The case for vm-based cloudlets in mobile computing," *Proc. IEEE Pervasive Computing,* vol. 8, no. 4, pp. 14–23, 2009.

[5] S. Kosta, A. Aucinas, P. Hui, R. Mortier and X. Zhang, "Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading," *Proc. IEEE INFOCOM*, 2012.

[6] M. Jia, W. Liang, Z. Xu and M. Huang, "Cloudlet load balancing in wireless metropolitan area networks," *INFOCOM*, 2016.

[7] F. Bonomi et.al., "Fog computing and its role in the internet of things," *ACM MCC Workshop on Mobile Cloud Computing,* pp. 13–16. 2012.

[8] K. Liu, J.K.Y. Ng, V.C.S. Lee, S.H. Son and Ivan Stojmenovic I., "Cooperative Data Dissemination in Hybrid Vehicular Networks: VANET as a Software Defined Network," *IEEE/ACM Trans. on Netw.*, vol. 24, iss 3, pp. 1759–1773, 2016.

[9] T. Soyata, R. Muraleedharan, C. Funai, M. Kwon and W. Heinzelman, "Cloud-Vision: Real-time face recognition using a mobile-cloudlet-cloud acceleration architecture," *Proc. IEEE Symp. on Computers and Communications (ISCC)*, 2012.

[10] A. Manjhi et. al. "Tributaries and deltas: Efficient and robust aggregation in sensor network streams,"*Proc. ACM SIGMOD*, 2005.

[11] H. Luo, Y. Liu and S. K. Das, "Distributed algorithm for en route aggregation decision in wireless sensor networks," *IEEE Trans. Mobi. Comput.,* vol. 8, no. 1, pp. 1–13, 2009.

[12] M. Satyanarayanan, "Fundamental Challenges in Mobile Computing," *Proc. PODC*, Philadelphia, PA, USA, May 1996.

[13] G. Huerta-Canepa and D. Lee, "A Virtual Cloud Computing Provider for Mobile Devices," *Proc. MCS*, San Francisco, CA, USA, Jun. 2010.

[14] G. Kirby, A. Dearle, A. Macdonald and A. Fernandes, "An Approach to Ad hoc Cloud Computing," *CoRR*, vol. abs/1002.4738, 2010.

[15] Y. Zhang, D. Niyato, P. Wang and C. K. Tham, "Dynamic Ofoading Algorithm in Intermittently Connected Mobile Cloudlet Systems," *Proc. ICC*, Sydney, Australia, Jun. 2014.

[16] T. T. Huu, C. K. Tham and D. Niyato "To Offload or to Wait: An Opportunistic Offloading Algorithm for Parallel Tasks in a Mobile Cloud," *Proc. IEEE Cloudcom*, 2014.

[17] D Liu, Y. Chen, K. K. Chai and T. Zhang, "Distributed latency-energy aware user association in 3-tier HetNets with hybrid energy sources," *Proc. IEEE GLOBECOM Workshops,* Austin, TX, 2014.

[18] L. Kleinrock, *Queueing Systems, vol. II: Computer applications*, Wiley-Interscience, New York, 1976.

[19] T. Han and N. Ansari, "Powering mobile networks with green energy," *IEEE Wireless Commun. Mag.*, vol. 21, no. 1, pp. 90-96, Feb. 2014.

[20] V. Chamola, B. Krishnamachari and B. Sikdar, "An Energy and Delay Aware Downlink Power Control Strategy for Solar Powered Base Stations," *IEEE Communications Letters*, vol. 20.5, pp. 954-957, 2016.

[21] H. Kim, D. G. Veciana G, X. Yang and M. Venkatachalam, "Distributed-optimal user association and cell load balancing in wireless networks.," *IEEE/ACM Trans. Netw.*, pp. 177-90, Feb. 2012.

[22] S. Boyd and L. Vandenberghe, *Convex Optimization,* Cambridge University Press, 2004.