Exploring Device-to-Device Communication for Mobile Cloud Computing

Yujin Li, Lei Sun, and Wenye Wang
Department of Electrical and Computer Engineering
North Carolina State University, Raleigh, NC, USA.

***Abstract*—With the popularity of smartphones and explosion of mobile applications, mobile devices become the prevalent computing platform for convenient communication and rich entertainment. Mobile cloud computing (MCC) is proposed to overcome the limited resources of mobile systems. However, when users access MCC through wireless networks, cellular network is likely to be overloaded and Wi-Fi connectivity is intermittent. Therefore, device-to-device (D2D) communication is exploited as an alternative for MCC. An important issue in exploring D2D communication for MCC is *how users can detect and utilize the computing resources on other mobile devices*. In this paper, we propose two mobile cloud access schemes: optimal and periodic access schemes, and study the corresponding performance of mobile cloud computing (i.e., mobile cloud size, node's serviceable time percentage, and task success rate). We find that optimally, node's serviceable time percentage and task success rate approach 1. Using more practical periodic access scheme, node's serviceable time percentage and task success rate are determined by the ratio of contact and inter-contact time between two nodes.**

I. Introduction

Mobile devices (such as smartphones and tablets) are becoming an inseparable part of our lives for convenient communication and entertainment. With the popularity of mobile devices, there is also an explosion of mobile applications in various categories, such as terrestrial navigation, email and web browsing, mobile games, mobile healthcare, mobile commerce, and social networking. This indicates that mobile devices are quickly becoming the dominant computing platform, which enables seamless work or entertainment for users regardless of user mobility.

Nonetheless, mobile systems are still limited in their resources (e.g., processor power, storage size, and battery life) and communications (e.g., bandwidth, connectivity, and security) [1]. Such resource scarceness significantly hinders the development of mobile applications and the improvement of mobile service qualities.

Recently, this problem has been addressed by researchers through mobile cloud computing (MCC). MCC provides services for resource constrained mobile devices to partition and offload their computationally intensive and storage demanding jobs to the cloud with vast computational resources [2]. In mobile cloud computing, computing-intensive mobile applications, such as video decoding, speech recognition, and augmented reality, can to be offloaded to the cloud for processing. Computation offloading can save energy and improve performance of mobile applications thereby overcoming limited resource capacities of mobile devices. MCC can also enable mobile users to store/access large data on the cloud through wireless networks, which can save data storage capacity and processing power on mobile devices.

In MCC, mobile users need access computing services on the cloud through high-speed and ubiquitous wireless connection. The computational resources in the cloud are feasible only if the information exchange between the cloud and mobile devices through wireless networks is fast, reliable, and secure. The most widely used network access technologies are cellular and Wi-Fi networks. Cellular network provides the near-ubiquitous coverage. But cellular network is under significant pressure and likely to be overloaded due to the increasing mobile data traffic [3], which may incur long latencies and slow data transfers. Although Wi-Fi has high data rate, Wi-Fi connections are intermittent. Hence, the drawback of mobile cloud computing is that the performance of cloud services depends strongly on wireless communication networks.

In order to overcome the drawbacks of accessing MCC through cellular and Wi-Fi networks, device-to-device (D2D) communication is exploited for mobile cloud computing while avoiding global network bottlenecks [4]. The increasing density of mobile devices produces an abundance of contact opportunities [5]. When offloading to remote clouds fails in low connectivity scenarios, mobile devices can employ local resources on mobile devices in the vicinity for computing a shared task. By exploiting D2D communication, users could improve the performance of mobile cloud computing in terms of computing speedup and money saving on smartphone data-roaming charges [4, 6, 7].

An important issue in exploring D2D communication for MCC is *how users can access the computing resources on other mobile devices*. Because of user mobility, D2D connection is intermittent. Under such intermittent connectivity, access scheme needs to be carefully developed such that users can utilize computing resources of nearby mobile devices as much as possible while not wasting too much energy on device discovery. In this paper, we propose two access schemes, i.e., *optimal and periodic* access algorithms, in which the initiator optimally or periodically performs node discovery, subtask distribution and retrieval with or without knowledge of other nodes' mobility, respectively.

We study the following MCC performance metrics under

This work is supported by the NSF Award CNS-1018447.

both access schemes: *mobile cloud size*, *serviceable time percentage*, and *task success rate*. More specifically, mobile cloud size is the number of nodes that an initiator discovers and utilizes for computing; serviceable time percentage is the percentage of time that a device computes tasks for an initiator; task success rate is the probability that a task transmitted to a device is executed and successfully retrieved by the initiator. Optimal access scheme provides the optimal performance of MCC based on D2D communication. Using periodic access scheme, performance of mobile cloud is greatly affected by contact and inter-contact time between two nodes. The more frequent a node meets the initiator and the longer their contacts are, the higher the node's serviceable time percentage and the task success rate are.

The remainder of this paper is organized as follows. We present a succinct summary on the existing work in Section II. We give the network model, contact process, and access schemes in Section III. The performance of MCC is analyzed in Section IV. We conclude in Section V.

II. Related Work

With the support of cloud computing of various services for mobile users, mobile cloud computing (MCC) is introduced to facilitate mobile users to take full advantages of cloud computing. Mobile users can access cloud services through wireless networks, including cellular and Wi-Fi networks. However, the increasing mobile data traffic has put a significant strain on cellular network [3]. Overloaded cellular network may incur long latencies and slow data transfers, which make the data uploading and downloading expensive. At the same time, Wi-Fi connection coverage is intermittent. In order to access cloud computing seamlessly, D2D communication is proposed to assist existing wireless communication systems.

Marinelli [4] points out that in many cases, processing mobile data (such as sensor logs and multimedia data) in-place and transferring it directly between smartphones would be more efficient and less susceptible to network limitations than offloading data and processing to remote servers. Therefore, Marinelli develops Hyrax, a platform derived from Hadoop that supports cloud computing on Android smartphones. A central server with access to each mobile device coordinates data and jobs and smartphones communicate with each other on an isolated 802.11g network. Although the performance of Hyrax is poor for CPU-bound tasks, it is shown to tolerate node-departure and offer reasonable performance in data sharing. A distributed multimedia search and sharing application is implemented to qualitatively evaluate Hyrax.

Similarly, Huerta-Canepa and Lee [8] observe that mobile devices can be a virtual cloud computing provider because their pervasiveness means the increasing availability of nearby devices; they are more powerful over the time; they include different network interfaces allowing devices to communicate with each other (with no money cost); moreover they allow us to create communities in which we can execute shared tasks. Huerta-Canepa and Lee propose a virtual cloud computing platform, in which a context manager monitors the location and number of nearby devices. A Korean OCR that reads an image, scans for the Korean characters, and then presents a Romanize version of them was developed for testing purposes.

Paper [6] proposes a framework that uses local resources on mobile devices for computing when offloading to remote clouds fails in low connectivity scenarios. Experiments are conducted in Bluetooth transmission and an initial prototype is also presented. The authors also discuss a preliminary analytical model to determine whether or not a speedup will be possible in offloading. Shi et al. [7] investigates the scenario that a mobile device uses the available, potentially intermittently connected, computation resources of other mobile devices to improve its computational experience, e.g., minimizing local power consumption and/or decreasing computation completion time. The authors propose and implement Serendipity on mobile devices to leverage the frequent contacts between mobile devices in order to speedup computing and conserve energy. A speech-to-text application is implemented to evaluate Serendipity, showing that Serendipity reduces job completion time comparing with executing locally. The authors also implement a preliminary prototype of Serendipity on the Android platforms with two computationally complex applications (i.e., a face detection application and a speech-to-text application).

One fundamental issue in using mobile devices for cloud computing is that an initiator needs to first detect mobile devices in proximity, then perform task distribution and retrieval. Therefore, we propose two access schemes in this paper and study mobile cloud computing performance under them.

III. Models and Definitions

Assume that n mobile devices are moving in a network $\Omega_n = [0, \sqrt{\frac{n}{\lambda}}]$, where λ is the spatial density of mobile users. Each mobile device has a transmission radius r. Denote by $\mathcal{X}_t = \{X_1(t), \ldots, X_n(t)\}$ the positions of users at time t. Nodes are moving according to Mobility Process $\mathcal{M}$. We assume that the mobility process of a node is stationary and ergodic that a node's location $X_i(\cdot)$ has uniform stationary distribution in the network area [9]. Mobility processes of nodes are independent and identically distributed (i.i.d.).

Without loss of generality, we assume that a mobile user initiates to offload computational tasks to nearby mobile devices at time 0. As shown in Fig. 1, the initiator can connect to nodes in its transmission range through direct D2D communication links, forming a *mobile cloud* for computing. We assume that all nodes are willing to support mobile computing because of fast computation of a common task or incentives offered by the initiator.

A. Contact Process Dynamics: Apparently, node mobility affects connectivity of D2D communication. How frequently nodes meet and how long they stay connected affect the size and stability of a mobile cloud, in turn, influence the computing capacity of a mobile cloud. A contact event between a pair of users occurs when two users are close enough to communicate and exchange content with each other. Let $X_u(t)$ and $X_v(t)$ denote the locations of users u and v at time t, we call

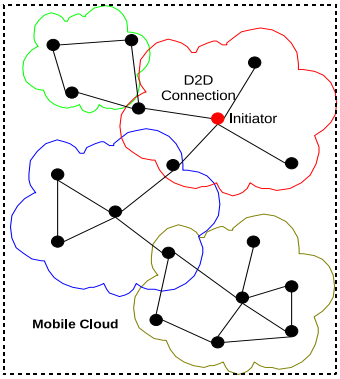

Fig. 1. Devices in the proximity form mobile cloud.

that one *contact event* T_C between users u and v occurs during $[t_0, t_1)$ if $||X_u(t_0^-) - X_v(t_0^-)|| > r$ and $||X_u(t) - X_v(t)|| \leq r$ for all $t \in [t_0, t_1)$, and $||X_u(t_1) - X_v(t_1)|| > r$. The number of contact events between a pair of users within time t is a counting process called the *contact process*. We refer to the time between the end and the start of two consecutive contact events between the same pair of users as the *inter-contact time* T_I.

As task dissemination and retrieval can only be performed when there is a communication link between two nodes, contact and inter-contact time between nodes affect mobile cloud computing performance. Obtaining complete knowledge of contact processe can be extremely difficult. Also mathematically characterizing mobile cloud computing performance is intractable for arbitrary contact process. Thus, we assume that *the contact process of a pair of users is a Poisson process*, which has been shown to be a good approximation and used by other existing studies [10, 11]. In other words, contact and inter-contact time follow exponential distributions with parameters λ_C and λ_I, respectively.

B. Mobile Cloud Access Schemes: In order to enable mobile cloud, an initiator mobile device first needs to discover devices in proximity. Then, the initiator can dispatch/retrieve tasks to/from mobile devices for MCC. Clearly, how an initiator detects other devices and employs them for computing determine mobile cloud performance. Hence, we give two mobile cloud access schemes in the following.

Ideally, if an initiator has perfect knowledge of its future contact with other devices, the initiator can maximally exploit other devices for computing, resulting in the optimal performance of mobile cloud. Therefore, we propose an optimal access scheme to evaluate the optimal performance of MCC based on D2D communications.

Definition 1 (Optimal Access Scheme). Initially, a mobile device u has a task that needs to be computed within time τ. Denote by t_h $(0 \leq t_h \leq \tau)$ the first hitting time when initiator u first meets user v within time period $[0, \tau]$, and t_e $(0 \leq t_e \leq \tau)$ the last exit time that u and v are out of each other's transmission range during $[t_e, \tau]$. User u partitions the task, sends some subtasks to v at time t_h, and retrieves the executed subtasks from v at time t_e. User v can perform task computation during the whole time $[t_h, t_e]$.

In reality, it is difficult for a mobile device to obtain the mobility information of other devices due to privacy issue. A more practical access scheme is the following periodic access scheme, in which an initiator mobile device periodically performs device detection, task distribution and retrieval.

Definition 2 (Periodic Access Scheme). Initiator u periodically scans its neighboring devices. Suppose at t, the set of its neighbors is $N_t = \{v_1, \ldots, v_k\}$. Node u will send a subtask to each of its neighbors. At time $t + \epsilon$, u detects its neighbors $N_{t+\epsilon} = \{v'_1, \ldots, v'_k\}$. If node $v'_k \in N_t$, u will retrieve the subtask sent to v'_k at time t and send a new subtask to v'_k for process; otherwise, u will send a subtask to v'_k.

Note that the node discovery interval ϵ depends on subtask computation time and device's battery level. If battery is sufficient, high node discovery frequency would ensure better utilization of mobile cloud. When $\epsilon \to 0$, an initiator can use a mobile device for computing during all their contact time. If battery level is low, large ϵ can reduce energy consumption with performance compromise. A reasonable setting is letting ϵ equal to the computational time of a subtask, which endures little performance compromise without too much energy consumption. In this paper, we assume ϵ equal to the computational time of a subtask for our analysis. In addition, if an initiator has multiple neighbors in proximity, it can use multiple access scheme (e.g., [12]) to dispatch and retrieve subtasks to all of them at the same time.

C. Mobile Cloud Performance Criteria: We seek to evaluate the D2D computing performance in terms of the size of mobile cloud, and the percentage of a mobile device's serviceable time, and the success rate of a task computation.

- *Mobile cloud size* is the number of mobile devices that an initiator detects within task delay tolerance τ.
- *Serviceable time percentage* is defined as the percentage of time that a mobile device is employed by an initiator to provide computing services.
- *Task success rate* represents the probability that an initiator can successfully send a task to a mobile device for computing and retrieve the task back before it expires.

IV. Mobile Cloud Performance Analysis

We analyze the performance of mobile cloud (i.e., mobile cloud size, serviceable time percentage, task success rate) under optimal and periodic access schemes, respectively. Analysis results under optimal and periodic access schemes will provide the optimal and achievable cloud computing performance based on D2D communication, respectively.

A. Optimal Access Scheme

Under optimal access scheme, an initiator can employ every node that it meets for cloud computing. Thus, mobile cloud size is the number of mobile devices that an initiator meets within task delay tolerance τ. Denote by $\mathcal{N}_\tau^o$ the mobile cloud size, and we have the following theorem on $\mathcal{N}_\tau^o$.

Theorem 1. *The mobile cloud size $\mathcal{N}_\tau^o$ follows Binomial* $\left(n-1, \left[1-\left(1-\frac{\pi r^2}{n/\lambda}\right)e^{-\lambda_I \tau}\right]\right)$.

Proof: Assume at $t=0$, there are N^* nodes in an initiator's transmission range, mobile cloud size increment $\mathcal{N}_\tau^o - N^*$ is the superposition of $n-N^*-1$ number of 0-1 processes $1_{\{\widetilde{T_I}\le\tau\}}$, where $\widetilde{T_I}$ is the residual inter-contact time.

$$\mathcal{N}_\tau^o = N^* + \sum_{i=1}^{n-N^*-1} 1_{\{\widetilde{T_I}\le\tau\}} = n-1-\sum_{i=1}^{n-N^*-1} 1_{\{\widetilde{T_I}>\tau\}}.$$

It is worth noting that N^* is a random variable depending on initial node distribution in the network. Rigorously, $P(\mathcal{N}_\tau^o = k) = E(P(\mathcal{N}_\tau^o = k|N^*))$ and $E(\mathcal{N}_\tau^o) = E(E(\mathcal{N}_\tau^o|N^*))$. Therefore, $\mathcal{N}_\tau^o$ is determined by the initial node distribution and the residual inter-contact time between two nodes. In homogeneous networks, N^* satisfies $P(N^*=m) = \binom{n-1}{m}\left(\frac{\pi r^2}{n/\lambda}\right)^m\left(1-\frac{\pi r^2}{n/\lambda}\right)^{n-1-m}$. Then,

$$\begin{aligned}P(\mathcal{N}_\tau^o = k) &= \sum_{m=1}^{k}\binom{n-1}{m}\left(\frac{\pi r^2}{n/\lambda}\right)^m\left(1-\frac{\pi r^2}{n/\lambda}\right)^{n-1-m}\\ &\times\binom{n-m-1}{k-m}\left(F_{\widetilde{T_I}}(\tau)\right)^{k-m}\left(1-F_{\widetilde{T_I}}(t)\right)^{n-k-1}\\ &=\binom{n-1}{k}\left(\frac{\pi r^2}{n/\lambda}+F_{\widetilde{T_I}}(\tau)\left(1-\frac{\pi r^2}{n/\lambda}\right)\right)^k\\ &\times\left(1-\frac{\pi r^2}{n/\lambda}\right)^{n-1-k}\left(1-F_{\widetilde{T_I}}(\tau)\right)^{n-k-1},\end{aligned}$$

which is a binomial distribution with parameters $n-1$ and $\frac{\pi r^2}{n/\lambda}+F_{\widetilde{T_I}}(\tau)\left(1-\frac{\pi r^2}{n/\lambda}\right)$, where $F_{\widetilde{T_I}}(\tau) = P(\widetilde{T_I}\le\tau)$. Thus,

$$E(\mathcal{N}_\tau^o) = (n-1)\left(\frac{\pi r^2}{n/\lambda}+\left(1-\frac{\pi r^2}{n/\lambda}\right)F_{\widetilde{T_I}}(\tau)\right). \quad (1)$$

The density function of $\widetilde{T_I}$ is $\lambda_I[1-F_{T_I}(x)]$, where $F_{T_I}(x)$ is the CDF of T_I and $\lambda_I^{-1} = \int_0^\infty x * F_{T_I}(dx) = \int_0^\infty (1-F_{T_I}(x))dx$. When the inter-contact time T_I follows exponential distribution with parameter λ_I, $\widetilde{T_I}$ is identically distributed with T_I. Hence, we complete our proof. ■

Under the optimal access scheme, an initiator can schedule to send subtasks to other mobile devices in advance so that they can perform computing even when they are not in contact with the initiator. Thus, the serviceable time for a mobile device is from the first hitting time t_h to the last exit time t_e. In other words, the serviceable time percentage is $(t_e-t_h)/\tau$, for $0\le t_h\le t_e\le\tau$.

Theorem 2. *The serviceable time percentage of a node is shown in Eq. (2), which is approaching* 1 *when τ is large.*

Proof: In a contact process between two nodes, denote by $\xi(t)=1$ when two nodes are in contact at time t, $\xi(t)=0$ otherwise.

(i) When $\xi(0)=\xi(\tau)=1$, $t_h=0$ and $t_e=\tau$. Clearly, its serviceable time $ST_\tau^o=\tau$.

(ii) When $\xi(0)=0$ and $\xi(\tau)=1$, $t_h=\widetilde{T_I}$ and $t_e=\tau$. $ST_\tau^o = \tau-\widetilde{T_I}\cdot 1_{\{\widetilde{T_I}<\tau\}}$.

(iii) When $\xi(0)=1$ and $\xi(\tau)=0$, $t_h=0$ and $t_e=\tau-\widehat{T_I}$, where $\widehat{T_I}$ is the backward recurrence time of T_I. $ST_\tau^o = \tau-\widehat{T_I}\cdot 1_{\{\widehat{T_I}<\tau\}}$.

(iv) When $\xi(0)=0$ and $\xi(\tau)=0$, if the initiator never encounters the node within time τ, the node's serviceable time is 0. Otherwise, $t_h=\widetilde{T_I}$ and $t_e=\tau-\widehat{T_I}$. Thus, $ST_\tau^o = [\tau-(\widetilde{T_I}+\widehat{T_I})\cdot 1_{\{\widetilde{T_I}+\widehat{T_I}<\tau\}}]\cdot 1_{\{\widetilde{T_I}<\tau\}}$.

Denote by $\pi_{ij}(t)$ the equilibrium probability, given that $\xi(0)=i$, that $\xi(t)=j$ $(i,j=0,1)$. Let p_0 and p_1 denote $P(\xi(0)=0)$ and $P(\xi(0)=1)$, respectively. Because T_I and T_C are exponential random variables with parameters λ_I and λ_C, respectively, $\widetilde{T_I}$ and $\widehat{T_I}$ have the same distribution as T_I, and $\widetilde{T_I}+\widehat{T_I}$ follows Erlang-2 distribution Erlang(2,λ_I). Thus,

$$\begin{aligned}E(ST_\tau^o)/\tau &= \tau\lambda_I e^{-\lambda_I\tau}\pi_{00}(\tau)p_0 \quad (2)\\ &+[1+(\pi_{01}(\tau)p_0+\pi_{10}(\tau)p_1+\pi_{00}(\tau)p_0)e^{-\lambda_I\tau}]\\ &-\frac{1}{\lambda_I\tau}(1-e^{-\lambda_I\tau})(\pi_{01}(\tau)p_0+\pi_{10}(\tau)p_1+2\pi_{00}(\tau)p_0),\end{aligned}$$

where $p_1=\frac{\pi r^2}{n/\lambda}$ and $p_0=1-p_1$, and the equilibrium probability $\pi_{ij}(\tau)$ can be derived based on Cox's Renewal Theory (Chapter 7.4) [13]: $\pi_{00}(\tau)=\beta+\gamma e^{-\beta\tau/\lambda_C}$, $\pi_{01}(\tau)=\gamma-\gamma e^{-\beta\tau/\lambda_C}$, $\pi_{10}(\tau)=\beta-\beta e^{-\beta\tau/\lambda_C}$, and $\pi_{11}(\tau)=\gamma+\beta e^{-\beta\tau/\lambda_C}$, where $\beta=\frac{\lambda_C}{\lambda_I+\lambda_C}$ and $\gamma=\frac{\lambda_I}{\lambda_I+\lambda_C}$. When τ is large, $E(ST_\tau^o)/\tau\to 1$, i.e., the mean serviceable time percentage approaches 1 in the long run. ■

Similarly, when an initiator has full knowledge of its contact and inter-contact events with other devices, it can make sure disseminating tasks to a mobile device only if they can be successfully retrieved. Thus, the task success rate of optimal access scheme is 1.

In practice, it is difficult for an initiator to obtain full information of its contact with other users. To make the optimal access scheme practical, we can use the following heuristic access scheme by exploiting the regularity of human mobility. Human mobility shows a very high degree of temporal and spatial regularity [14] and can be predicted with high probability [15]. Benefiting from the predictability of human mobility, an initiator u can estimate its contact time with other users based on their contact history.

Heuristic Access Scheme: Initiator u records the history of its contact with other users over a period of time and uses this information for task computing. For a give period $[t,t+\tau]$ of a day d, initiator u estimates its contact and inter-contact time (e.g., first hitting time t_h and last exit time t_e) with a user v based on mobility history of day $d-c$, where c is a small integer (usually 1 or 2). Then, u applies the optimal access scheme to utilize the computing resource of user v.

Remark 1. When an initiator has pre-knowledge of node mobility or can predict other nodes' mobility based on mobility history, an initiator can utilize computing resources on other devices through optimal or heuristic access schemes. Under the optimal access scheme, the expected number of nodes that an initiator can use within time τ is $(n-$

1) $\left[1-\left(1-\frac{\pi r^2}{n/\lambda}\right)e^{-\lambda_I \tau}\right]$. The long-term serviceable time percentage and task success rate are approaching 1.

B. Periodic Access Scheme

Suppose an initiator periodically detects devices in proximity with frequency $1/\epsilon$. Within time $[0,\tau]$, the initiator performs device discovery at time $\{0,\epsilon,2\epsilon,\ldots,\lfloor\frac{\tau}{\epsilon}\rfloor\epsilon\}$. In other words, a device is detected by the initiator only if it is in contact with the initiator for at least one time point of $\{0,\epsilon,2\epsilon,\ldots,\lfloor\frac{\tau}{\epsilon}\rfloor\epsilon\}$. Applying renewal process theory, we derive the following theorem on the size of mobile cloud.

Theorem 3. *The mobile cloud size $\mathcal{N}_\tau^p$ follows Binomial $\left(n-1,1-\bar{P}\right)$, where $\bar{P}$ can be found in Eq. (5).*

Proof: If a device is not detected by an initiator within time τ, it is in inter-contact with the initiator at all time points $\{0,\epsilon,2\epsilon,\ldots,\lfloor\frac{\tau}{\epsilon}\rfloor\epsilon\}$. Let us consider the contact process between an initiator and a mobile device and let F_{T_C} (f_{T_C}) and F_{T_I} (f_{T_I}) denote the distribution (density) functions of the contact and inter-contact time. Define renewal functions $H_2(t)=\sum_{i=1}^{\infty}F_{T_I}^{(n)}*F_{T_C}^{(n)}(t)$ and renewal density $h_2(t)=dH_2(t)/dt$, where $F_{T_I}^{(n)}$ ($F_{T_C}^{(n)}$) is the n-fold convolution of F_{T_I} (F_{T_C}) itself.

Cox [13] derives the probability that the system is at inter-contact state at time t. Conditioning on the initial state $\xi(0)=0$ (i.e., inter-contact state)

$$IT_0(t)=1-F_{T_I}(t)+(1-F_{T_I})*H_2(t). \tag{3}$$

Further, Baxter [16] derives the joint probability that the system is at inter-contact state at m distinct time points $\{t_1,t_2,\ldots,t_m\}$. Let $m=\lfloor\frac{\tau}{\epsilon}\rfloor$, $t_1=\epsilon,\ldots,t_i=i\epsilon,\ldots,t_m=\lfloor\frac{\tau}{\epsilon}\rfloor\epsilon$,

$$IT_0^{(m)}(t_1,t_2,\ldots,t_m)=R_0(t_m-t_1,t_1)+IT_0(t_1)\sum_{i=1}^{m-1}\int_{t_i-t_1}^{t_{i+1}-t_1}\phi_0(x,t_1)IT_0^{(m-i)}(t_{i+1}-t_1-u,\ldots,t_m-t_1-u)dx, \tag{4}$$

$$R_0(x,t)=1-F_{T_I}(t+x)+\int_0^t h_2(u)(1-F_{T_I}(t+x-u))du,$$

$$\phi_0(x,t)=\frac{1}{IT_0(t)}\left[f_{T_I}(t+x)+\int_0^t h_2(u)f_{T_I}(t+x-u)du\right].$$

$IT_0^{(m)}(t_1,t_2,\ldots,t_m)$ can be computed recursively from Eq. (3). Thus, the probability that a node is not detected by the initiator is

$$\bar{P}=p_0 IT_0^{(m)}(t_1,t_2,\ldots,t_m), \tag{5}$$

where $p_0=1-\frac{\pi r^2}{n/\lambda}$. Hence, the total number of nodes detected by the initiator follows Binomial distribution with parameters $(n-1,1-\bar{P})$. ∎

Remark 2. Clearly, $\bar{P}>\left(1-\frac{\pi r^2}{n/\lambda}\right)e^{-\lambda_I\tau}$. Thus, the mobile cloud size under periodic access scheme is stochastically dominated by that under optimal access scheme.

Whenever an initiator senses a device in proximity, it transmits a subtask to the device for computing, which takes ϵ time to finish. A device computes all subtasks that it receives no matter whether the subtasks are retrieved. Thus, the total serving time of a device is ϵ times the number of task transmissions. We have the following theorem on the serviceable time percentage by studying the number of task transmissions between an initiator and a device.

Theorem 4. *A mobile device's serviceable time percentage is approaching $\frac{\lambda_I}{\lambda_I+\lambda_C}$ when τ is large and ϵ is small.*

Proof: Let us consider a contact event between an initiator and a device during $[t_0,t_1]$. Dividing $[t_0,t_1]$ into time slots $[t_0,t_0+\epsilon],[t_0+\epsilon,t_0+2\epsilon],\ldots$, we have $\lceil(t_1-t_0)/\epsilon\rceil$ number of subintervals with each slot equals to ϵ except the last one. During each time slot, the initiator must perform one and only one node detection. In the best scenario, the initiator performs node detection and transmits the task to a neighbor at the beginning of each time slot, resulting in $\lceil(t_1-t_0)/\epsilon\rceil$ number of task transmissions. In the worst scenario, the initiator performs node detection and transmits the task to a neighbor at the end of each time slot, resulting in $\lfloor(t_1-t_0)/\epsilon\rfloor$ number of task transmissions.

Denote by T_C the contact time random variable and $N(\tau)$ the number of contacts between an initiator and a device within time duration τ. Therefore, the total number of tasks a device computed is upper bounded by $\sum_{i=1}^{N(\tau)}\lceil\frac{T_C^i}{\epsilon}\rceil$ and lower bounded by $\sum_{i=1}^{N(\tau)}\lfloor\frac{T_C^i}{\epsilon}\rfloor$. Hence, the service time satisfies

$$E\left(\sum_{i=1}^{N(\tau)}\left\lfloor\frac{T_C^i}{\epsilon}\right\rfloor\epsilon\right)\le E(ST_\tau^p)\le E\left(\sum_{i=1}^{N(\tau)}\left\lceil\frac{T_C^i}{\epsilon}\right\rceil\epsilon\right).$$

Based on renewal theory, $N(\tau)$ conditioning on the initial state $\xi(0)=0$ (i.e., inter-contact state)

$$E(N^0(\tau))=F_{T_I}(\tau)+\int_0^\tau E(N^0(\tau))dF_{T_I+T_C}(s);$$

and conditioning on $\xi(0)=1$ (i.e., contact state)

$$E(N^1(\tau))=F_{T_C}(\tau)+\int_0^\tau E(N^1(\tau))dF_{T_I+T_C}(s).$$

Taking the Laplace transform of these two equations,

$$L_{E(N^0(\tau))}(s)=L_{T_I}(s)/s+L_{N^0(\tau)}(s)L_{T_I+T_C}(s),$$
$$L_{E(N^1(\tau))}(s)=L_{T_C}(s)/s+L_{N^1(\tau)}(s)L_{T_I+T_C}(s).$$

Because T_C and T_I have exponential distributions with parameters λ_C and λ_I, respectively,

$$L_{E(N^0(\tau))}(s)=\frac{\lambda_I(s+\lambda_C)}{s^2(s+\lambda_I+\lambda_C)},$$
$$L_{E(N^1(\tau))}(s)=\frac{\lambda_C(s+\lambda_I)}{s^2(s+\lambda_I+\lambda_C)}.$$

Taking the inverse Laplace transform, we then have

$$E(N^0(\tau))=\frac{\lambda_I\lambda_C\tau}{\lambda_I+\lambda_C}+\frac{\lambda_I^2}{(\lambda_I+\lambda_C)^2}(1-e^{-(\lambda_I+\lambda_C)\tau}),$$

$$E(N^1(\tau)) = \frac{\lambda_I \lambda_C \tau}{\lambda_I + \lambda_C} + \frac{\lambda_C^2}{(\lambda_I + \lambda_C)^2}(1 - e^{-(\lambda_I + \lambda_C)\tau}).$$

Therefore, the expected number of contact between two nodes within time duration τ satisfies

$$\begin{aligned} E(N(\tau)) &= E(N(\tau)|I_0 = 0)p_0 + E(N(\tau)|I_0 = 1)p_1, \quad (6) \\ &= \frac{\lambda_I \lambda_C \tau}{\lambda_I + \lambda_C} + \frac{p_0 \lambda_I^2 + p_1 \lambda_C^2}{(\lambda_I + \lambda_C)^2}(1 - e^{-(\lambda_I + \lambda_C)\tau}), \end{aligned}$$

where $p_1 = \frac{\pi r^2}{n/\lambda}$ and $p_0 = 1 - p_1$.

Applying this result to the service time, we have

$$E(N(\tau))[E(T_C) - \epsilon] \leq E(ST_\tau^p) \leq E(N(\tau))[E(T_C) + \epsilon].$$

When τ becomes large and ϵ is small,

$$\lim_{\tau \to \infty, \epsilon \to 0} \frac{E(ST_\tau^p)}{\tau} = \frac{\lambda_I}{\lambda_I + \lambda_C}. \quad (7)$$

■

Remark 3. Using the periodic access scheme, an initiator can utilize a mobile device for approximately $\frac{\lambda_I}{\lambda_I + \lambda_C}$ percentage of time, which is determined by $\frac{\lambda_C}{\lambda_I}$. The longer a device and an initiator are in contact and the more frequently they meet, the higher utilization of the device's computing resources is.

After an initiator dispatches a subtask to a mobile device, it can retrieve this subtask if the initiator detects the mobile device again before the task expires. Suppose the initiator detects a mobile device M_τ times during time period τ, the initiator only fails to retrieve the last dispatched subtask. Hence, the task success rate is $1 - 1/M_\tau$. Following this methodology, we have the following theorem on the task success rate.

Theorem 5. *Task success rate satisfies*

$$P_s^p \leq 1 - \frac{1}{E(N(\tau))(\frac{1}{\lambda_C \epsilon} + 1)}. \quad (8)$$

Proof: To derive the task success rate, we need to find out the total number of transmitted subtasks from an initiator to a mobile device. From our proof of serviceable time percentage, we have

$$E\left(\sum_{i=1}^{N(\tau)} \left\lfloor \frac{T_C^i}{\epsilon} \right\rfloor\right) \leq E(M_\tau) \leq E\left(\sum_{i=1}^{N(\tau)} \left\lceil \frac{T_C^i}{\epsilon} \right\rceil\right).$$

As the inverse function is convex on the interval $(0, +\infty)$, the task failure rate satisfies

$$E\left(\frac{1}{M_\tau}\right) \geq 1/E\left(\sum_{i=1}^{N(\tau)} \left\lceil \frac{T_C^i}{\epsilon} \right\rceil\right) \geq \frac{1}{E(N(\tau))(\frac{1}{\lambda_C \epsilon} + 1)},$$

where $E(N(\tau))$ can be found in Eq. (6). Hence, we have the task success rate in Eq. (8). ■

Remark 4. Using periodic access scheme, the task success rate is upper bounded by a function of τ, λ_I, λ_C, and ϵ. The more frequently an initiator meets a device and the longer they stay in contact, the closer this upper bound approaches 1.

V. Conclusion

In this paper, we have studied the performance of mobile cloud computing based on D2D communication, in which an initiator distributes tasks to other nearby devices and retrieves tasks after execution. We propose optimal and periodic access schemes that an initiator detects devices in proximity and distributes tasks to them for computing optimally or periodically. Optimal access scheme results in optimal mobile cloud performance with serviceable time percentage and task success rate approaching 1. Using more practical periodic access scheme, an initiator mobile device can employ a mobile device for computing in $\frac{\lambda_I}{\lambda_I + \lambda_C}$ percentage of time, and the task success rate is upper bounded by a function of λ_I and λ_C. In summary, mobility patterns of users have signicant impact on performance of mobile cloud, which can be measured by $\frac{\lambda_C}{\lambda_I}$. In our future work, we will use mobile applications to evaluate the mobile cloud computing performance.

References

[1] A. K. Gupta, "Challenges of mobile computing," in *2nd National Conference on Challenges & Opportunities in Information Technology (COIT-2008)*, (Mandi Gobindgarh), 2008.

[2] N. Fernando, S. W. Loke, and W. Rahayu, "Mobile cloud computing: A survey," *Future Generation Computer Systems*, vol. 29, no. 1, pp. 84 – 106, 2013.

[3] A. Aijaz, H. Aghvami, and M. Amani, "A survey on mobile data offloading: technical and business perspectives," *IEEE Wireless Communications*, vol. 20, no. 2, pp. 104–112, 2013.

[4] E. E. Marinelli, "Hyrax: Cloud computing on mobile devices using mapreduce," Master's thesis, Carnegie Mellon University, 2009.

[5] S. Liu and A. D. Striegel, "Exploring the potential in practice for opportunistic networks amongst smart mobile devices," in *Proc. of the ACM MobiCom*, 2013.

[6] N. Fernando, S. Loke, and W. Rahayu, "Dynamic mobile cloud computing: Ad hoc and opportunistic job sharing," in *Proc. of IEEE UCC*, 2011.

[7] C. Shi, V. Lakafosis, M. H. Ammar, and E. W. Zegura, "Serendipity: enabling remote computing among intermittently connected mobile devices," in *Proc. of ACM MobiHoc*, 2012.

[8] G. Huerta-Canepa and D. Lee, "A virtual cloud computing provider for mobile devices," in *ACM Workshop on Mobile Cloud Computing & Services (MCS'10)*, 2010.

[9] L. Sun and W. Wang, "On latency distribution and scaling: from finite to large cognitive radio networks under general mobility," in *Proc. of IEEE INFOCOM*, 2012.

[10] Y. Li and W. Wang, "The unheralded power of cloudlet computing in the vicinity of mobile devices," in *Proc. of IEEE GLOBECOM*, 2013.

[11] Y. Li and W. Wang, "Can mobile cloudlets support mobile applications?," in *Proc. of IEEE INFOCOM*, 2014.

[12] R. Mao and H. Li, "A novel multiple access scheme via compressed sensing with random data traffic," *Journal of Communications and Networks*, vol. 12, no. 4, pp. 308–316, 2010.

[13] D. Cox, *Renewal Theory*. Methuen & Co, 1962.

[14] Y. Li, M. Zhao, and W. Wang, "Internode mobility correlation for group detection and analysis in vanets," *Vehicular Technology, IEEE Transactions on*, vol. 62, no. 9, pp. 4590–4601, 2013.

[15] M. C. Gonzalezl, C. A. Hidalgo1, and A.-L. Barabási, "Understanding individual human mobility patterns," *Letters to Nature*, 2008.

[16] L. A. Baxter, "Availability measures for a two-state system," *Journal of Applied Probability*, vol. 18, no. 1, 1981.