# Energy Compensated Cloud Assistance in Mobile Cloud Computing

Jaya Prakash Champati and Ben Liang
Department of Electrical and Computer Engineering, University of Toronto, Canada

***Abstract*—We consider the scenario where a mobile device requires assistance from nearby devices to forward its computational tasks to a cloud server. We incentivize cooperation by allowing helper devices to conserve computational energy by offloading their own tasks to the source device's cloud, as compensation for the communication energy lost during task forwarding. We formulate an optimization problem with the objective of minimizing the cloud cost incurred by the source device due to tasks offloaded from helper devices, subject to no energy loss at the helper devices. We observe that this problem cannot be solved using a standard Lyapunov optimization approach. Instead, we construct an alternate problem that follows the standard form but has the same optimal objective value as the original problem. The resultant Energy Compensated Cloud Assistance (ECCA) algorithm does not require any statistics of the system and can be implemented distributively.**

## I. Introduction

Mobile Cloud Computing (MCC) has been proposed to augment the computation and storage capabilities of a mobile device by offloading the processing of applications or computational tasks/methods from mobile devices to remote cloud resource providers [1][2]. In this work we consider cooperation in MCC. In particular we study a scenario as shown in Figure 1, where a mobile device (source node) having reserved cloud resources requires the help of nearby mobile devices (neighbour nodes) to forward its computing tasks to a base station that is connected to the cloud service provider (e.g., through the Internet). This scenario can arise whenever the source node requires uninterrupted network connectivity for offloading tasks to its cloud or requires higher throughput for its transmissions. Similar to any other cooperative communication scenario, there is a net energy loss for a neighbour node in forwarding the source node's data packets. Hence, suitable incentive is needed to induce the neighbour node to cooperate.

We consider the following incentive for the neighbour nodes. During cooperation a neighbour node may offload its own computational tasks to the cloud of the source node. The advantage of this incentive scheme is two-fold. First, a neighbour node can mitigate the energy loss it incurs during cooperation by saving the required computational energy of its own tasks that are offloaded. Second, this incentive is easy to implement even in a dynamic environment where the neighbour nodes are temporarily cooperating and going out of range due to mobility. This introduces a new paradigm for the usage of cloud resources in wireless networks.

We formulate an optimization problem with the objective of minimizing the cloud cost incurred by the source node due to tasks offloaded from neighbour nodes subject to energy compensation (i.e., no energy loss) at the neighbour nodes and packet queue stability at all the nodes. The energy compensation constraint at a neighbour node involves the transmission energy spent in a time slot, which depends on the state of its packet queue. Therefore, the standard Lyapunov optimization theory [3] cannot be used directly to solve the problem. In particular, the conventional Min-Drift-Plus-Penalty algorithm based on Lyapunov drift analysis is no longer guaranteed to be asymptotically optimal. However, we construct an alternate problem that falls under the standard Lyapunov optimization framework and show that it has the same optimal objective value as the original problem. Furthermore, an optimal control policy for the reformulated problem is also optimal for the original problem. This results in a dynamic Energy Compensated Cloud Assistance (ECCA) algorithm that can be implemented in a distributed fashion. Through simulation of ECCA in a cooperative MCC system, we study the effect of different system parameters on the minimum cloud cost incurred by the source node that ensures energy compensation to the neighbour nodes.

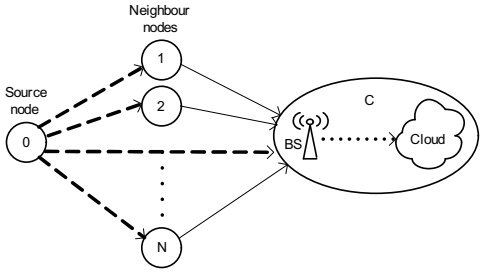


Fig. 1. Cooperation in Mobile Cloud Computing

The rest of the paper is organized as follows. In Section II we present the related work. Section III describes the system model and problem formulation. In Section IV we present an equivalent problem and the ECCA algorithm that solves the problem. Section V presents simulation results, and we conclude the paper in section VI.

## II. Related work

One main purpose of cloud offloading is to save the expenditure of computational energy by mobile devices. Towards this end several research papers have focused on the aspects of offloading mechanisms. Karthik *et al.* in [4] studied the general guideline that applications or tasks with high computational



978-1-4799-3088-3/14/$31.00 ©2014 IEEE

energy requirement and low data load are desirable to be executed remotely. Migration of an entire application for remote execution was studied in [5][6].

In contrast to the above studies, the authors in [7][8][9] considered partitioning an application into fine-grained tasks and then proposed offloading mechanisms. In [10], Cuervo *et al.* proposed MAUI, a system which builds on the above ideas and is shown to provide significant energy gains. One of the salient features of MAUI is that it offloads fine-grained tasks based on network conditions. Clonecloud [11] and Thinkair [12] are other such systems developed for efficient offloading. None of the works mentioned above consider network connectivity loss that may occur when a mobile device needs to offload tasks. In our work, the network connectivity loss is alleviated by obtaining assistance from nearby mobile devices. The practicality of this scenario for the case of smartphones has been studied in a university setting by Liu and Striegel [13]. They have demonstrated that such opportunistic forwarding can be reliable and stable. However, they have not considered energy compensation as incentive for cooperation.

## III. System Model

We consider a discrete time based model. The time slots are indexed by $t \in \{0, 1, \ldots\}$ with slot duration $T$. Let $i \in \{0, 1, \ldots, N\}$ denote the nodes (mobile devices), where node 0 represents the source node and nodes 1 to $N$ represent the neighbouring helper nodes. All nodes access the Internet and the cloud through a Base Station (BS).

We assume that all required cloud service is already reserved by node 0 and the cloud cost is proportional to the computational energy requirement of tasks offloaded. Node 0 and the neighbour nodes enter into the following agreement for cloud assistance and energy compensation. The neighbour nodes may utilize the cloud for their own computing tasks, in exchange for relaying the tasks of node 0. Furthermore, it is guaranteed that the communication energy incurred by each neighbour node is less than the computational energy gain from offloading its own tasks. The detailed model formulation is given in the following subsections.

### A. Task Profile

The mobile devices are equipped with systems such as MAUI [10] or ThinkAir [12] which partition an application into fine-grained tasks. Each task is associated with some data load, which contains a set of operations and input data. We assume that the data load of a task is fragmented into packets of fixed length $l$ bits. The cloud executes a task after reassembling all the packets associated with the task. Since the data load of tasks vary, each task is associated with a random number of packets, which is denoted by $M$ and it takes values from $\mathbb{Z}_{++}$. The computational energy requirement of a task is assumed to be known apriori [10] [11] [12]. We represent the computational energy requirement of a task by random variable $E$ (in joules), and it takes values from $\mathbb{R}_{++}$. The ordered pair $(E, M)$ represents a task profile, and for each task in the system it is chosen independently according to a fixed distribution. At node $i$, let $\bar{M}_i$ represent the average data size per task and $\bar{E}_i$ represent the average computational energy requirement per task. We assume that all the tasks in the system are independently executable and are delay tolerant.

We denote the task arrival process by $\mathbf{A}(t) = (A_0(t), \ldots, A_N(t))$, where $A_i(t)$ represents the number of new tasks that arrive at node $i$ at the beginning of slot $t$. We assume that $\mathbf{A}(t)$ is i.i.d. over time slots. Let us index the tasks that has arrived in slot $t$ at node $i$ by $j \in \mathcal{T}_i(t) = \{1, \ldots, A_i(t)\}$. Let us denote the set of task profiles at node $i$ by $\mathcal{X}_i(t) = \{(E_{ij}(t), M_{ij}(t))\}$ and define $\mathcal{X}(t) = \{\mathcal{X}_0(t), \ldots, \mathcal{X}_N(t)\}$.

### B. Channel Model

The channels are assumed to be block fading with additive white Gaussian noise. We assume that the channel gains remain constant over a slot and their values are normalized by the noise power. In a time slot $t$, $\mathbf{\Gamma}(t) = [(\Gamma_{0i}(t)), (\Gamma_i(t))]$ is the channel state vector, where $\Gamma_{0i}(t), i \neq 0$, represents the normalized channel gain between node 0 and neighbour node $i$, and $\Gamma_i(t)$ represents that between the node $i$ and the BS. Note that, here we allow $i = 0$, meaning that there could be a direct link between node 0 and the BS. We assume that $\mathbf{\Gamma}(t)$ is i.i.d. over time slots.

In time slot $t$, let $P(t) \in [0, P_{max}]$ be the transmission power chosen where, $P_{max}$ is the maximum power that can be used in any time slot and is determined by hardware constraints. We assume that the packets cannot be fragmented further. Let us define $\hat{\mu}(t)$ as the maximum number of packets that can be transmitted using power $P(t)$ in time slot $t$. From the Shannon bound, we have

$$\hat{\mu}(t) = \left\lfloor \frac{WT}{2l} \log_2(1 + P(t)\Gamma(t)) \right\rfloor \tag{1}$$

where $W$ is the bandwidth. We denote by $W_1$ the bandwidth between node 0 and the neighbour nodes and by $W_2$ the bandwidth between the nodes and the BS. Note that, as explained later, even though we have used the rate-power relation (1) for simplicity of illustration, the proposed analysis and ECCA are valid as long as the rate-power relation is concave.

***Boundedness assumption:*** We assume that the task arrivals, data size and energy of a task, and the normalized channel gains are non-negative and satisfy the following condition for all $t$: $\mathbb{E}\{A_i^2(t)\} \leq \sigma^2 \ \forall i, \mathbb{E}\{M_{ij}^2(t)\} \leq \sigma^2$ for any $i, j, \mathbb{E}\{E_{ij}^2(t)\} \leq \sigma^2$ for any $i, j$, and $\mathbb{E}\{\Gamma_{lm}^2(t)\}\} \leq \sigma^2$ for all valid indices $l, m$ where $\sigma < \infty$.

### C. Scheduling and Offloading

In each time slot, node 0 selects either the BS or neighbour node $i$ and chooses a feasible power to transmit packets from its packet queue $Q_0(t)$ to node $i$. The selection of neighbour node $i$ by node 0 is indicated by $I_i(t) = 1$ and $I_i(t) = 0$ otherwise. Similarly, the decision of direct transmission to BS by node 0 is indicated by $I_c(t) = 1$. At most one of the neighbour nodes or the BS can be selected in each time

slot, i.e., the vector $\mathbf{I}(t) = (I_c(t), I_1(t), \ldots, I_N(t)) \in \mathbf{e}$, where $\mathbf{e}$ is a set of $N+1$ dimensional vectors which have at most one non-zero element equal to 1. At node 0, let $P_0(t) \in [0, P_{0,max}]$ be the choice of transmission power and $\hat{\mu}_0(t)$ be the maximum number of packets that can be transmitted. Then, $\hat{\mu}_0(t) = I_c(t)\hat{\mu}_{0c}(t) + \sum_{i=1}^{N} I_i(t)\hat{\mu}_{0i}(t)$, where $\hat{\mu}_{0i}(t), i \neq 0$ is the maximum number of packets that can be transmitted to node $i$ using power $P_0(t)$, given that node $i$ is selected in slot $t$. Similarly, $\hat{\mu}_{0c}(t)$ is the maximum number of packets that can be transmitted to the BS using power $P_0(t)$, given that the BS is selected in slot $t$.

At neighbour node $i$, the packets received from node 0 are enqueued in $Q_i(t)$. Let $\mathcal{B}_i(t) \subseteq \mathcal{T}_i(t)$ be the set of indices of the tasks that node $i$ offloads. Then $M_{\mathcal{B}_i(t)} = \sum_{j \in \mathcal{B}_i(t)} M_{ij}(t)$ represents the total number of packets and $E_{\mathcal{B}_i(t)} = \sum_{j \in \mathcal{B}_i(t)} E_{ij}(t)$ represents the computational energy offloaded. All the $M_{\mathcal{B}_i(t)}$ packets are enqueued in $Q_i(t)$. Also, at node $i$, let $P_i(t) \in [0, P_{i,max}]$ be the choice of transmission power and $\hat{\mu}_i(t)$ is the maximum number of packets that can be transmitted to the BS using power $P_i(t)$ in slot $t$.

The stacked vector $\omega(t) = [\mathbf{\Gamma}(t), \mathcal{X}(t)]$ represents a random network outcome in the system. It can be observed that $\omega(t)$ is i.i.d. over slots. The control actions to be taken in the system is represented by the vector $\alpha(t) = (\mathbf{I}(t); (\mathcal{B}_1(t), \ldots, \mathcal{B}_N(t)); (P_0(t), \ldots, P_N(t)))$. In time slot $t$, we denote the set of all possible control actions by $\mathcal{A}_{\omega(\mathbf{t})}$ which is given by:

$$\mathcal{A}_{\omega(\mathbf{t})} = \{\mathbf{I}(t) \in \mathbf{e}, \{P_i(t) \in [0, P_{i,max}], \mathcal{B}_i(t) \subseteq \mathcal{T}_i(t), \forall i\}\}$$

Let $\mathbf{Q}(t) = (Q_0(t), \ldots, Q_N(t))$ denote the queue backlog vector. The queue update equations are as follows:

$$\begin{aligned} Q_0(t+1) =& \max(Q_0(t) - \hat{\mu}_0(t), 0) + M_0(t) \\ Q_i(t+1) =& \max(Q_i(t) - \hat{\mu}_i(t), 0) + M_{\mathcal{B}_i(t)} \\ &+ I_i(t)\min(Q_0(t), \hat{\mu}_{0i}(t)) \ \forall i \neq 0 \end{aligned} \tag{2}$$

where $M_0(t) = M_{\mathcal{T}_0(t)}$ (all tasks at source node are offloaded) and $I_i(t)\min(Q_0(t), \hat{\mu}_{0i}(t))$ is the number of packets that are transmitted from node 0 to neighbour node $i$ in time slot $t$.

### D. Formulation of Optimization Problem

At the neighbour node we want to maintain no energy loss on average while minimizing the average cost incurred by the source node at it's cloud due to the tasks offloaded by the neighbour nodes. We also want to guarantee the stability of queues at all nodes.

We assume that the cloud cost is proportional to the computational energy of the tasks offloaded. Therefore, we formulate the function $y_0(t) = \sum_{i=1}^{N} E_{\mathcal{B}_i(t)}$, which represents the total computational energy of the tasks offloaded by all neighbour nodes in time slot $t$. Now, the number of packets transmitted by node $i$ in time slot $t$ is given by $\mu_i(t) = \min(Q_i(t), \hat{\mu}_i(t))$. Using $P_i(t)$, the time required to transmit $\mu_i(t)$ is given by $T\frac{\min(Q_i(t), \hat{\mu}_i(t))}{\tilde{\mu}_i(t)}$, where $\tilde{\mu}_i(t) = \frac{W_2 T}{2l}\log_2(1 + P_i(t)\Gamma_i(t))$. Therefore, in time slot $t$ the energy loss at neighbour node $i$ is given by the following function:

$$y_i(t) = P_i(t)T\frac{\min(Q_i(t), \hat{\mu}_i(t))}{\tilde{\mu}_i(t)} - E_{\mathcal{B}_i(t)}$$

We are interested in the long term time average expectation of function $y_0(t)$ defined as $\bar{y}_0 \triangleq \lim_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1} \mathbb{E}\{y_0(\tau)\}$. Let $\bar{y}_i$ represent the corresponding time average expectation of $y_i(t)$. We consider the following optimization problem $\mathcal{P}$:

$$\begin{aligned} \underset{\{\alpha(t)\}}{\text{minimize}} \quad & \bar{y}_0 \\ \text{subject to} \quad & \\ & \bar{y}_i \leq 0, \text{ for } i \in \{1, \ldots, N\} \\ & \text{Queues are mean rate stable} \\ & \alpha(t) \in \mathcal{A}_{\omega(t)} \ \forall t \end{aligned}$$

When the problem is feasible, we use $\bar{y}_{opt}$ to denote its optimum value. Note that $\bar{y}_{opt} \geq 0$.

We note that the function $y_i(t)$ involves the queue length $Q_i(t)$, and hence $\mathcal{P}$ cannot be solved using standard Lyapunov optimization, i.e., the Min-Drift-Plus-Penalty algorithm for solving $\mathcal{P}$ may not guarantee $O(\frac{1}{V})$ asymptotic optimality [3]. However, we formulate a problem $\mathcal{P}'$ below that can be solved using standard Lyapunov optimization. We show that the optimal objective value for $\mathcal{P}'$ is equal to $\bar{y}_{opt}$.

## IV. Energy Compensated Cloud Assistance

In this section we present Energy Compensated Cloud Assistance (ECCA) algorithm for solving $\mathcal{P}$.

### A. Problem Reformulation

Let us define a modified energy loss function $y_i'(t) = P_i(t)T - E_{\mathcal{B}_i(t)}$. Note that in $y_i'(t)$ the accounting of transmission energy assumes that $P_i(t)$ is used for the entire time slot duration $T$. Problem $\mathcal{P}'$ is similar to $\mathcal{P}$ except for the energy compensation constraint. Namely, we use the constraints $\bar{y'}_i \leq 0, \forall i \neq 0$ for $\mathcal{P}'$ instead of $\bar{y}_i \leq 0, \forall i \neq 0$. Let $\bar{y'}_{opt}$ be the optimal value for problem $\mathcal{P}'$.

**Lemma 1.** $\bar{y'}_{opt} = \bar{y}_{opt}$

*Proof.* For any given pair of choices $(P_i(t), E_{\mathcal{B}_i(t)})$ we have $y_i(t) \leq y_i'(t)$. Hence, any control policy that satisfies $\bar{y'}_i \leq 0$ also satisfies $\bar{y}_i \leq 0$. Therefore, $\bar{y'}_{opt} \geq \bar{y}_{opt}$.

Let us consider an optimal control policy $\pi^*$ that achieves $\bar{y}_{opt}$ for problem $\mathcal{P}$. Under this policy let $\alpha^*(t)$ denote the control action in time slot $t$. We design a policy $\pi'$ with control action $\alpha'(t)$ in time slot $t$ as follows: $\mathbf{I}'(t) = \mathbf{I}^*(t)$, $\mathcal{B}_i'(t) = \mathcal{B}_i^*(t), \forall i$, $P_0'(t) = P_0^*(t)$ and,

$$\begin{aligned} P_i'(t) = & \mathbf{1}_{\{Q_i^*(t) \leq \hat{\mu}_i^*(t)\}} \frac{(2^{2lQ_i^*(t)/WT} - 1)}{\Gamma_i(t)} \\ & + \mathbf{1}_{\{Q_i^*(t) \geq \hat{\mu}_i^*(t)\}} \frac{(2^{2l\hat{\mu}_i^*(t)/WT} - 1)}{\Gamma_i(t)}, \forall i \neq 0 \end{aligned}$$

In time slot $t$ the control action $\alpha'(t)$ and $\alpha^*(t)$ only differ in the transmission power choices at the neighbour nodes. Therefore, the objective values achieved under $\pi^*$ and $\pi'$

should be same. Also, at each node $i \neq 0$ the choice of power $P_i'(t)$ at the neighbour node is designed such that the realizations of the queues $Q_i(t), \forall i$ under $\pi^*$ and $\pi'$ are the same. Since $\pi^*$ meets the constraint of mean rate stability of the queues, so does $\pi'$.

Let $y_i(\pi^*, t)$ be $y_i(t)$ induced by policy $\pi^*$ and $y_i'(\pi', t)$ be $y_i'(t)$ induced by policy $\pi'$. We claim that $\bar{y'}_i(\pi') \leq 0, \forall i \neq 0$ is satisfied. To prove this claim we first show that for any given realization of $\omega(t)$ we have $y_i(\pi^*, t) \geq y_i'(\pi', t), \forall t$. Consider the time slots where $Q_i^*(t) \leq \hat{\mu}_i^*(t)$. We have

$$\begin{aligned} y_i(\pi^*, t) &= P_i^*(t) T \frac{Q_i^*(t)}{\tilde{\mu}_i^*(t)} - E_{\mathcal{B}_i^*(t)} \\ &= \frac{(2^{2l\tilde{\mu}_i^*(t)/WT} - 1)}{\Gamma_i(t)} T \frac{Q_i^*(t)}{\tilde{\mu}_i^*(t)} - E_{\mathcal{B}_i^*(t)} \\ &\geq \frac{(2^{2lQ_i^*(t)/WT} - 1)}{\Gamma_i(t)} T - E_{\mathcal{B}_i^*(t)} \\ &= P_i'(t) T - E_{\mathcal{B}_i^*(t)} = y_i'(\pi', t) \end{aligned}$$

where the inequality above is based on the fact that $\frac{2^{ak}-1}{k}$ is an increasing function in $k$ for all $a, k > 0$. Similar argument can be used for the time slots where $Q_i^*(t) \geq \hat{\mu}_i^*(t)$. Noting that the realization of $Q_i(t)$ is the same under $\pi^*$ and $\pi'$, it can be shown that $\bar{y'}_i(\pi') \leq \bar{y}_i(\pi^*) \leq 0, \forall i \neq 0$.

From the above analysis it is clear that we found a control policy $\pi'$ that solves the problem $\mathcal{P}'$ and achieves $\bar{y}_{opt}$ for the objective. Since $\bar{y'}_{opt} \geq \bar{y}_{opt}$, we have $\bar{y'}_{opt} = \bar{y}_{opt}$. □

We note that Lemma 1 holds as long as the rate-power relation is concave and thus the analyses that follow holds for such general rate-power relation. Furthermore, from the proof of Lemma 1, we conclude that an optimal policy for $\mathcal{P}'$ is also feasible and hence optimal for $\mathcal{P}$. Therefore, we focus on solving $\mathcal{P}'$ in the sections that follow.

### B. Lyapunov Optimization on $\mathcal{P}'$

A standard Lyapunov optimization approach can be used to solve $\mathcal{P}'$ [3]. We model the inequality constraint $\bar{y'}_i \leq 0$ as a queue stability problem. Let $\mathbf{Z}(t) = (Z_1(t), \ldots, Z_N(t))$, where $Z_i(t)$ represents a virtual queue. The update equation of $Z_i(t)$ is given by, $Z_i(t+1) = \max(Z_i(t) + y_i'(t), 0)\ \forall\ i \in \{1, \ldots, N\}$. Let $\mathbf{\Theta}(t) = [\mathbf{Q}(t), \mathbf{Z}(t)]$ be the vector of all actual and virtual queues. We define the following weighted quadratic Lyapunov function: $L(\mathbf{\Theta}(\mathbf{t})) \triangleq \frac{w}{2} \sum_{i=0}^{N} Q_i^2(t) + \frac{1}{2} \sum_{i=1}^{N} Z_i^2(t)$. where the weight factor $w$ indicates the relative importance of the virtual queues with respect to the packet queues. The one-slot conditional Lyapunov drift $\Delta(\mathbf{\Theta}(\mathbf{t}))$ is defined as follows: $\Delta(\mathbf{\Theta}(t)) \triangleq \mathbb{E}\{L(\mathbf{\Theta}(t+1)) - L(\mathbf{\Theta}(t)) | \mathbf{\Theta}(t)\}$. Then, the drift-plus-penalty expression is given by $\Delta(\mathbf{\Theta}(t)) + V\mathbb{E}\{y_0(t)|\mathbf{\Theta}(\mathbf{t})\}$. We have the following observation.

**Lemma 2.** *With the assumption that $\omega(t)$ is i.i.d. over slots, under any control algorithm, the drift-plus-penalty expression has the following upper bound for all t, all possible values of $\mathbf{\Theta}(\mathbf{t})$, and all parameters $V \geq 0$:*

$$\begin{aligned} &\Delta(\mathbf{\Theta}(\mathbf{t})) + V\mathbb{E}\{y_0(t)|\mathbf{\Theta}(\mathbf{t})\} \leq B + V\mathbb{E}\{y_0(t)|\mathbf{\Theta}(\mathbf{t})\} \\ &+ \sum_{i=1}^{N} wQ_i(t)\mathbb{E}\{M_{\mathcal{B}_i(t)} - \hat{\mu}_i(t) + I_i(t)\hat{\mu}_{0i}(t)|\mathbf{\Theta}(\mathbf{t})\} \\ &+ \sum_{i=1}^{N} Z_i(t)\mathbb{E}\{P_i(t)T - E_{\mathcal{B}_i(t)}|\mathbf{\Theta}(\mathbf{t})\} \\ &+ wQ_0(t)\mathbb{E}\{M_0(t) - \hat{\mu}_0(t)|\mathbf{\Theta}(\mathbf{t})\} \end{aligned} \tag{3}$$

*where $B$ is a positive constant and its existence is guaranteed by the assumption that $\omega(t)$ is i.i.d. and the boundedness assumption.*

*Proof.* The proof is similar to Lemma 4.6 in [3] and is omitted. □

### C. ECCA

In each slot $t$ we observe the queue backlog vector $\mathbf{\Theta}(t)$ and the network event $\omega(t)$ and choose a control action $\alpha(t) \in \mathcal{A}_{\omega(t)}$ that greedily minimizes the RHS of (3). For convenience of exposition, define

$$\begin{aligned} \varsigma(t) \triangleq &\sum_{i=1}^{N} I_i(t)\hat{\mu}_{0i}(t)[Q_i(t) - Q_0(t)] \\ &- I_c(t)\hat{\mu}_{0c}(t)Q_0(t) \\ \chi_i(t) \triangleq &wQ_i(t)M_{\mathcal{B}_i(t)} - (Z_i(t) - V)E_{\mathcal{B}_i(t)} \\ \varphi_i(t) \triangleq &[Z_i(t)P_i(t)T - wQ_i(t)\hat{\mu}_i(t)] \end{aligned} \tag{4}$$

Then, we aim to minimize $\Psi(t) = w\varsigma(t) + \sum_{i=1}^{N} \chi_i(t) + \sum_{i=1}^{N} \varphi_i(t)$.

### D. Algorithm Details

ECCA can be implemented in a distributed fashion because the decision variables in the expression $\Psi(t)$ are segregated and can be solved independently. To be specific, it is sufficient for node 0 to choose control actions that minimizes $\varsigma(t)$ and for node $i$ to choose control actions that minimizes $\chi_i(t)$ and $\varphi_i(t)$ subject to respective constraints.

*1) Optimization Problem at Source Node:* The problem to be solved at the source node is

$$\underset{(\mathbf{I}(t) \in \mathbf{e}, P_0(t) \in [0, P_{0,max}])}{\text{minimize}} \quad \varsigma(t) \tag{5}$$

Problem (5) is easily solvable and the solution is presented below. Let $\mu_{0i,max}(t) = \left\lfloor \frac{W_1 T}{2l} \log_2(1 + P_{0,max}\Gamma_i(t)) \right\rfloor$. If $Q_0(t) = 0$, then $\mathbf{I}(t) = 0$ and $P_0(t) = 0$. Otherwise, set $P_0(t) = P_{0,max}$, and if $\min_i\{\mu_{0i,max}(t)[Q_i(t) - Q_0(t)]\} \leq -\mu_{0c,max}(t)Q_0(t)$, then $k = \arg\min_i\{\mu_{0i,max}(t)[Q_i(t) - Q_0(t)]\}$ and $I_k(t) = 1$ else $I_c(t) = 1$.

*2) Optimization Problem at Neighbour Node i:* Neighbour node $i$ needs to find $P_i^*(t)$ that minimizes $\varphi_i(t)$ and choose $\mathcal{B}_i^*(t)$ that minimizes $\chi_i(t)$. We solve the former problem first:

$$\underset{P_i(t) \in [0, P_{i,max}]}{\text{minimize}} \quad \varphi_i(t) \tag{6}$$

We note that problem (6) is non-convex. However, for a given $\hat{\mu}_i(t)$, $\varphi_i(t)$ increases with $P_i(t)$. Therefore, to minimize $\varphi_i(t)$ we need to choose $P_i(t)$ such that $\hat{\mu}_i(t) = \frac{W_2 T}{2l}\log_2(1 + P_i(t)\Gamma_i(t))$. Using this property, a naive method to solve the problem is to check the power values corresponding to $\hat{\mu}_i(t) \in \{0, 1, \ldots, \hat{\mu}_{i,max}(t)\}$ where $\hat{\mu}_{i,max}(t) = \left\lfloor \frac{W_2 T}{2l}\log_2(1 + P_{i,max}\Gamma_i(t)) \right\rfloor$. The time complexity of this approach in any slot is $O(\mu_{i,max})$, where $\mu_{i,max} = \left\lfloor \frac{W_2 T}{2l}\log_2(1 + P_{i,max}\Gamma_{max}) \right\rfloor$ represents the maximum number of packets that can be transmitted in any time slot and $\Gamma_{i,max} \geq \Gamma_i(t)\ \forall t$. Note that $\mu_{i,max}$ can be arbitrarily large. Instead, we propose the following approach that solves problem (6) with a time complexity of $O(1)$. The key idea we use is that the problem (6) is convex if we choose $P_i(t) \in [0, P_{i,max}]$ and use $\hat{\mu}_i(t) = \frac{W_2 T}{2l}\log_2(1+P_i(t)\Gamma_i(t))$.

For the degenerate case $Q_i(t) = 0$ or $\Gamma_i(t) = 0$, we have $P_i^*(t) = 0$. Otherwise, if $Q_i(t) > 0$, $\Gamma_i(t) > 0$, and $Z_i(t) = 0$ then find $P_i^*(t)$ such that $\hat{\mu}_{i,max}(t)$ packets are transmitted using the entire time slot. Otherwise, we solve the problem (6) as follows:

**Lemma 3.** *For $Q_i(t) > 0$, $\Gamma_i(t) > 0$, $Z_i(t) > 0$, allowing $P_i(t) \in [0, P_{i,max}]$ and $\hat{\mu}_i(t) = \frac{W_2 T}{2l}\log_2(1+P_i(t)\Gamma_i(t))$, an optimal solution $\hat{P}_i^*(t)$ that minimizes $\varphi_i(t)$ is given by:*

$$\hat{P}_i^*(t) = \min\left\{\left(\frac{W_2 Q_i(t)}{2l Z_i(t)\log 2} - \frac{1}{\Gamma_i(t)}\right)^+, P_{i,max}\right\}$$

*Proof.* Under the given conditions, it can be shown that the function $\varphi_i(t)$ is continuous and strictly convex. Therefore, an optimal solution $\hat{P}_i^*(t)$ is found by solving the first-order condition subject to $\hat{P}_i^*(t) \in [0, P_{i,max}]$. □

Now, an optimal solution to problem (6) can be found by appropriately choosing $P_i^*(t)$ using $\hat{P}_i^*(t)$. Let $P_{i,L}(t)$ and $P_{i,R}(t)$ represent the immediate power values in the left and right neighbourhood of $\hat{P}_i^*(t)$ which provide integer rates $\mu_{i,L}(t)$ and $\mu_{i,R}(t)$ respectively, where $\mu_{i,L}(t) = \left\lfloor \frac{W_2 T}{2l}\log_2(1 + \hat{P}_i^*(t)\Gamma_i(t)) \right\rfloor$ and $\mu_{i,R}(t) = \left\lceil \frac{W_2 T}{2l}\log_2(1 + \hat{P}_i^*(t)\Gamma_i(t)) \right\rceil$. If $P_{i,R}(t) > P_{i,max}$ then set $P_i^*(t) = P_{i,L}(t)$. Otherwise, among the values $P_{i,L}(t)$ and $P_{i,R}(t)$ choose the one which results in the minimum value for $\varphi_i(t)$ and assign it to $P_i^*(t)$.

For finding $\mathcal{B}_i^*(t)$, node $i$ considers the problem

$$\underset{\mathcal{B}_i(t)\subseteq\mathcal{T}_i(t)}{\text{minimize}} \quad \chi_i(t) \tag{7}$$

which can be solved by simply selecting all those tasks for which $Q_i(t)M_{ij}(t) - (Z_i(t) - V)E_{ij}(t)$ is negative. Therefore, $\mathcal{B}_i^*(t) = \{j \in \mathcal{T}_i(t) : Q_i(t)M_{ij}(t) - (Z_i(t) - V)E_{ij}(t) < 0\}$.

In any time slot $t$, the time complexity of ECCA at neighbour node $i$ is $O(A_{i,max})$, where $A_{i,max} > A_i(t)\ \forall t$. At the source node its time complexity is $O(N)$.

### E. Performance Bound

In the following theorem we present the performance bound for ECCA, which is a direct consequence of Theorems 4.5 and 4.8 in [3].

**Theorem 1.** *Assume $\mathbb{E}\{L(\mathbf{\Theta}(0))\} < \infty$. For any $V > 0$, ECCA ensures that the packet queues $\mathbf{Q}(t)$ are mean rate stable, all the required constraints are satisfied, and $\bar{y}_0$ satisfies the following inequality:*

$$\bar{y}_0 \leq \bar{y}_{opt} + \frac{B}{V}.$$

## V. Simulation Results

In this section we present simulation results to study the effect of various system parameters on the performance of ECCA. We choose $T = 1$ sec and $w = 0.005$. The packet size in the system is chosen to be 1280 bytes (minimum datagram size of IPv6). We assume the communication between neighbour nodes and the cloud go through a cellular BS and use parameters from the 3GPP standard. The maximum power of neighbour nodes is set to $P_{i,max} = 500$ mW, which is the maximum output from a UMTS/3G power class 2 mobile phone. The bandwidth is set to $W_2 = 3.84$ MHz. We choose the normalized channel gains from an exponential distribution with mean 4. We assume node 0 communicates with neighbour nodes using Bluetooth and hence choose the following values [14]. The maximum power used by the source node is set to $P_{0,max} = 100$ mW. The bandwidth is set to $W_1 = 1$ MHz. We choose the normalized channel gains from an exponential distribution with mean 80.

Task arrivals at each node $i$ in any time slot is chosen according to a Poisson distribution with parameter $\lambda_i$. The data size of a task $M$ in the system is chosen uniformly from the set $\{\bar{M}_i - 20, \bar{M}_i - 10, \bar{M}_i, \bar{M}_i + 10, \bar{M}_i + 20\}$ for all $i$. Default value for the average data size per task is $\bar{M}_i = 50$ KB. The computational energy requirement of a task $E$ is chosen uniformly from an interval of length 1.6. Default value for the average computational energy requirement per task is $\bar{E}_i = 1$ joule, for all $i$. Our choice for the average data size and average energy requirement of tasks is motivated by the corresponding values of the tasks offloaded for the face recognition application in MAUI [10]. We choose the following set of default parameters to study the affect of various parameters on the objective value achieved by ECCA. $\lambda_i = 2, V = 5000$ and $N = 2$. Each simulation run spans $2\text{x}10^7$ time slots.

Figure 2 shows that, under large $V$ values, as the number of neighbour nodes $N$ increases, a better objective value is achieved. This is as expected, since the overall cost should be non-increasing in $N$ under an optimal control policy. Interestingly, for small $V$ values this is not the case. The reason is that, for small $V$ values, more weight is given to the constraints compared with the objective. From the utility-delay trade-off, it can be seen that the objective value achieved can be quite far from the optimal value. Since the number of constraints increases with $N$, the value of $B$ potentially increases resulting in such a pattern.

Figure 3 shows an interesting phenomenon that the objective value decreases with the average computational energy requirement per task at the neighbour nodes. Even though offloading a task by a neighbour node provides it with an

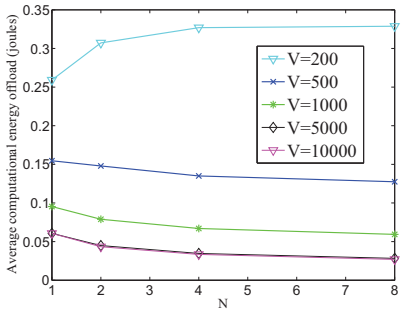


Fig. 2. Average computational energy offload per slot vs. $N$, for different values of $V$

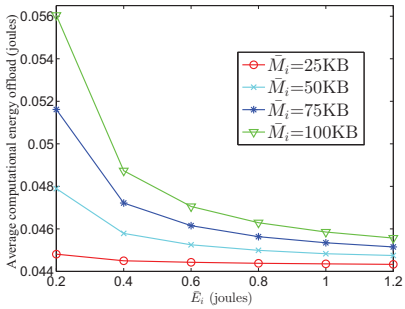


Fig. 3. Average computational energy offload per slot vs. $\bar{E}_i, i \neq 0$, for different values of $\bar{M}_i, i \neq 0$

energy gain equal to the computational energy requirement of the task, it has to compensate for the transmission energy that will be spent for transmitting the data packets of that task. For a fixed average data size per task, the relative transmission energy cost is reduced when the average computational energy per task in the system increases, and hence the required compensation in the form of computational energy offload per slot decreases. Another observation from Figure 3 is that the objective value increases with the average data size per task. In summary, it costs the source node less if the tasks being offloaded by the neighbour nodes have higher computational energy requirement and lower data size.

Figure 4 shows that the objective value increases with the task arrival rate $\lambda_0$ as well the average data size per task $\bar{M}_0$ at the source node. An interesting observation is that the increase is super-linear with respect to $\lambda_0$. We explain this as follows. An increase in data load at the source results in a higher data load to be forwarded by neighbour nodes. This results in higher transmission energy and hence higher computational energy offload per slot. Therefore, the relation between the objective and $\lambda_0$ is similar to that of transmission power and data rate, which is super-linear.

## VI. Conclusion

In this work we have studied one scenario of cooperation in MCC where nearby mobile devices cooperate with a source mobile device to forward its computing tasks to the cloud. Using a discrete time queuing model, we have formulated a stochastic network optimization problem in which we aim to minimize the cloud cost incurred by the source device due to

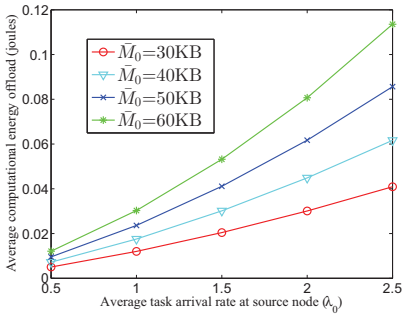


Fig. 4. Average computational energy offload per slot vs. $\lambda_0$, for different values of $\bar{M}_0$

tasks offloaded from neighbour nodes subject to no energy loss at the neighbour nodes and packet queue stability. Noting that no performance guarantee can be obtained by using the standard Min-Drift-Plus-Penalty algorithm, we have formulated an equivalent problem and proposed ECCA algorithm to solve it for a distributed solution.

## References

[1] N. Fernando, S. W. Loke, and W. Rahayu, "Mobile cloud computing: A survey," *Future Gener. Comput. Syst.*, vol. 29, no. 1, pp. 84–106, Jan 2013.

[2] D. Hoang T., L. Chonho, N. Dusit, and W. Ping, "A survey of mobile cloud computing: architecture, applications, and approaches," *Wireless Communications and Mobile Computing*, Oct. 2011.

[3] M. J. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Morgan and Claypool Publishers, 2010.

[4] K. Kumar and Y.-H. Lu, "Cloud computing for mobile users: Can offloading computation save energy?" *Computer*, vol. 43, no. 4, pp. 51–56, 2010.

[5] Y. Wen, W. Zhang, and H. Luo, "Energy-optimal mobile application execution: Taming resource-poor mobile devices with cloud clones." in *Proc. IEEE INFOCOM*, 2012, pp. 2716–2720.

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

[7] J. Flinn, S. Park, and M. Satyanarayanan, "Balancing performance, energy, and quality in pervasive computing," in *Proc. International Conference on Distributed Computing Systems (ICDCS)*, 2002, pp. 217–226.

[8] R. K. Balan, M. Satyanarayanan, S. Y. Park, and T. Okoshi, "Tactics-based remote execution for mobile computing," in *Proc. International Conference on Mobile Systems, Applications and Services (MobiSys)*, 2003, pp. 273–286.

[9] R. Balan, J. Flinn, M. Satyanarayanan, S. Sinnamohideen, and H. i Yang, "The case for cyber foraging," in *Proc. ACM SIGOPS European Workshop*, 2002, pp. 87–92.

[10] E. Cuervo, A. Balasubramanian, D.-k. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, "Maui: making smartphones last longer with code offload," in *Proc. International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2010, pp. 49–62.

[11] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "Clonecloud: elastic execution between mobile device and cloud," in *Proc. Conference on Computer systems (EuroSys)*, 2011, pp. 301–314.

[12] 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," in *Proc. IEEE INFOCOM*, 2012, pp. 945–953.

[13] S. Liu and A. D. Striegel, "Exploring the potential in practice for opportunistic networks amongst smart mobile devices," in *Proc. Annual International Conference on Mobile Computing (MobiCom)*, 2013, pp. 315–326.

[14] "Specification of the bluetooth system v4.0," Bluetooth SIG, Specification, June 2010. [Online]. Available: https://www.bluetooth.org/en-us/specification/adopted-specifications