Joint Wireless and Service Allocation for Mobile Computation Offloading with Job Completion Time and Cost Constraints

Hong Chen*, Terence D. Todd*, Dongmei Zhao* and George Karakostas†
*Department of Electrical and Computer Engineering
†Department of Computing and Software
McMaster University
Hamilton, Ontario, CANADA
Email: {chenh151,todd,dzhao,karakos}@mcmaster.ca

***Abstract*—This paper proposes a method of joint wireless network and job service allocation for use with mobile computation offloading where task completion times have deadline constraints. In this design, mobile devices (MDs) may execute a computational task locally or offload the task through a wireless network for execution on an edge server (ES). The network owner offers to lease wireless communication channels at a given set of base stations along with edge server capacity that is used for job execution. The objective is to obtain a wireless and service capacity allocation that minimizes the total energy consumption of the mobile devices, subject to a cost budget constraint and constraints on the delay incurred by offloaded task execution. The design is first formulated as a mixed integer nonlinear programming problem. An approximate solution is then obtained by decomposing it into a collection of convex subproblems that can be efficiently solved. Results are presented that demonstrate that the proposed solution achieves near optimum performance over a wide range of system parameters.**

I. Introduction

Mobile computation offloading (MCO) can be used to improve performance by having the mobile device (MD) offload local task execution to a remote cloud server, as opposed to running the task locally. Wireless communications is typically used by the MD to upload the task/data so that remote execution is possible. There is a large literature that has studied various issues dealing with MCO [1] [2] [3].

MCO incurs added latency that would not otherwise exist due to the time needed to exchange application data with the server. This latency may be compensated for by task execution at a cloud edge server, which is typically more rich in computational resources than the mobile device. An edge server, located close to the network base stations is typically used since it can provide very low latency between the BSs and the server [4]. The basic tradeoffs involving these attributes and how they relate to the decision to offload task execution have been studied extensively, for example cf. [5] [6], and the references therein. The decision to offload job execution is more complicated when the MD interacts with the server over stochastic transmission channels that may also change randomly during the computation offload. This is the environment that is considered in this paper.

In our paper we consider the case where a network leaseholder (NL) leases both wireless channel and edge server (ES) execution services from the network owner, subject to a cost budget constraint. The NL then uses the leased resources to provide MCO to a set of mobile devices. The objective is to find an allocation that minimizes the average MD energy consumption subject to the budget constraint and constraints on the probability that job execution deadlines are violated. This latter constraint significantly increases the difficulty of the problem compared to the unconstrained case. Note that this problem is different than that of network slice creation [7]. In our case, the NL has no interest in operating its own network. Instead, it purchases services from the network owner, who prices the cost of unit channel/computational resources in accordance with the associated performance it offers to the NL. Inside the owner's network this may be accomplished using conventional resource provisioning, and is therefore not a concern of the network leaseholder. Due to the edge server placement, we consider the case where the dominant latencies are that of wireless access and application server execution [4].

More specifically, we introduce a method of combined wireless network and job service allocation for use with MCO where task completion times have deadline constraints. When each MD task is generated, there is an associated deadline, which gives the time by which task execution should be completed with a high degree of certainty [8]. The NL leases wireless channels from the wireless network owner (NO) at a given set of base stations (BSs), along with ES computational capacity so that accumulated tasks can be executed. The objective is for the NL to minimize the total average MD energy consumption subject to a cost budget constraint and constraints on the probability that task execution deadlines are violated. The design is initially formulated as a mixed integer nonlinear programming problem (MINLP). An approximate solution is then obtained by decomposing it into a collection of convex subproblems, which can be solved efficiently, and picking the best of these solutions. A variety of results are presented that characterize the tradeoffs between task deadline violation, MD energy consumption and the rental cost budget.

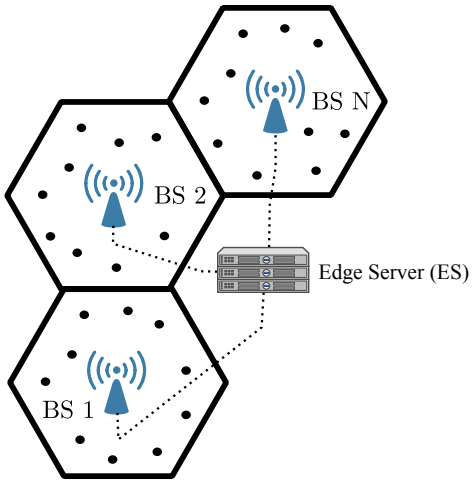

Fig. 1. System Model

Our results also show the quality of the proposed solution, which can achieve close-to-optimum performance for a wide range of system parameters.

II. System Model and Problem Formulation

As shown in Fig. 1, we consider a network that consists of N BSs that are owned and operated by a network owner (NO). The set of BSs is denoted by $\mathcal{N} = \{1, 2, \ldots, N\}$ and indexed by $n \in \mathcal{N}$. The network also contains an edge server (ES). Tasks generated by an MD can be offloaded through the wireless network and executed on the ES.

The NO permits a NL to rent wireless communication and ES computational capacity that the NL can use for mobile computation offloading for its MDs. When this is done, for each BS n, there are up to K_n available channels that can be selected by the NL. The unit cost for including a channel from BS n is specified by the NO as α_n. When a channel is included in the agreement, the NO agrees to provision its network so that sufficient resources are available that allows the traffic generated on the channel to be carried to the ES with an acceptable delay with a high degree of certainty. Since the ES is located at the edge of the network, we focus on the dominant sources of delay, i.e., wireless access at the BSs and task execution at the ES [4].

In order to use the computing resources at the ES, the NL must also lease CPU resources at the ES. The cost (based on the number of CPU cycles per second) for leasing on the CPU resource is denoted by β. The maximum available CPU speed for rental is f^{C} CPU cycles per second.

When an agreement is made between the NO and NL, x_n is defined as the number of channels from BS n that are included, and $y \in [0, 1]$ is defined as the fraction of maximum CPU speed at the ES that is included, i.e., the CPU speed available for the NL will be yf^{C}. It is assumed that the NL has a cost budget, denoted by $B^{\max}$. Accordingly, the total rent must satisfy the following constraint: $\sum_{n=1}^{N} \alpha_n x_n + \beta y f^{\mathrm{C}} \leq B^{\max}$.

There are J classes of tasks generated by the MDs, which may need to be offloaded to the ES. Let $\mathcal{J} = \{1, 2, \ldots, J\}$ be the set of task classes. The class j of a task is defined by parameters s_j, q_j, and d_j, where s_j is the input data size in bits, q_j is the computation load in number of CPU cycles, and d_j is the deadline of the task in seconds. The probability of a task generated by an MD belonging to class j is denoted by P_j^{C}; we assume that this probability is known, e.g., by observing the past history of offloading requests.

Our objective is to create a NO/NL contract for MCO. In MCO, jobs generated by an MD can be executed either locally (at the MD itself) or offloaded through the network and executed on the ES. The goal is to accomplish this so that the mean mobile energy consumption is minimized, subject to the cost budget constraint and so that the probability of a task execution deadline violation is bounded.

We model the wireless channels between the MDs and the BSs as discrete-time Markov processes. It is assumed that there are I_n channel models for BS n, which are a function of the propagation environment that the MDs experience at that BS. $\mathcal{I}_n = \{1, 2, \ldots, I_n\}$ is the set of all wireless channel models in BS n. For each of the channel models, the Markovian transition probabilities are defined in the usual way, i.e., given the channel state in the current time slot, there is a probability associated to its transition to another state in the next time slot. The time slot duration is defined to be τ seconds.

To obtain the design, the decision to offload the execution of a task is made using a *local execute on blocking* (LEB) mechanism as follows. When an MD in BS n generates a class j task, the MD offloads the task if at least one of the x_n channels is available for immediate use. Otherwise, the MD executes the task locally. When a channel is available, the MD begins the offload by uploading the s_j task bits needed for execution on the ES. The LEB mechanism is useful in that either local execution or remote offloading is initiated immediately at task release time, which may be advantageous when task deadlines are tight.

We consider that tasks arrive at BS n according to a stationary process with average arrival rate λ_n tasks per second. When using the LEB mechanism, a new task is blocked from BS channel access if all the x_n channels are busy for uploading other tasks. We denote the blocking probability at BS n as $P_{\mathrm{B}n}$ and the power needed in the MD to process tasks as P^{L}. When a class j task is blocked from offloading and executed locally, the local execution time is given as $L_j = q_j/f$, where f is the MD's execution speed in number of CPU cycles per second. Hence, the average MD energy consumption per unit of time[1] in BS n to execute the tasks that are blocked for offloading due to channel access can be written as $E_n^{\mathrm{L}} = P_{\mathrm{B}n}\lambda_n P^{\mathrm{L}} \bar{L}$, where $\bar{L}$ is the average local execution time of tasks, which can be calculated as $\bar{L} = \sum_{j=1}^{J} P_j^{\mathrm{C}} L_j$. Note that the task blocking is caused by channel access, which is the same for all task classes.

The wireless upload transmission time $t_{n,j,k}^{\mathrm{W}}$ of the jth class of tasks in BS n with wireless channel model k is modelled in number of time slots. The mean wireless transmission

[1] In what follows we just use the term 'energy consumption'

time $\bar{t}_n^{\mathrm{W}}$ for uploading tasks in BS n can be calculated from $t_{n,j,k}^{\mathrm{W}}$ if the distributions of tasks in different classes and the wireless channel models are given. The average per unit time energy consumption of MDs in BS n for uploading tasks can be written as $E_n^{\mathrm{T}} = (1 - P_{\mathrm{B}n})\,\lambda_n P^{\mathrm{T}} \bar{t}_n^{\mathrm{W}}$, where P^{T} is the transmission power used by the MD for uploading the task bits. Therefore, the average energy consumption for tasks arriving at BS n can be expressed as $E_n = E_n^{\mathrm{L}} + E_n^{\mathrm{T}}$.

Under the stated assumptions, the aggregate mean task arrival rate $\tilde{\lambda}$ at the ES is given by

$$\tilde{\lambda} = \textstyle\sum_{n=1}^{N} (1 - P_{\mathrm{B}n})\,\lambda_n. \tag{1}$$

As is normally the case for stability in a single server queueing system, the constraint $\tilde{\lambda} < \mu^{\mathrm{C}}$ must always be satisfied, where μ^{C} denotes the mean service rate at the ES, i.e, $\mu^{\mathrm{C}} = yf^{\mathrm{C}} / \sum_{j=1}^{J} P_j^{\mathrm{C}} q_j$. We can relax this constraint to $\tilde{\lambda} \leq yf^{\mathrm{C}} / \sum_{j=1}^{J} P_j^{\mathrm{C}} q_j$ without affecting our solution.

We consider the distribution of total delay for an offloaded task, which consists of the sum of the delay incurred during the *data upload* and *task execution at the ES*. For BS n, task class j, and channel model k, the former random variable is denoted as $t_{n,j,k}^{\mathrm{W}}$, and the latter as $t_{n,j,k}^{\mathrm{C}}$. As mentioned earlier, the data transmission delay from the BS to the ES is negligible. In addition, in this paper we consider the case of a very small amount of data returned once the execution is completed. Therefore, we only consider the data uploading delay from the MD to the BS. For the jth class of tasks in BS n with wireless channel model k, the delay constraint must satisfy

$$\Pr\left[t_{n,j,k}^{\mathrm{W}} + t_{n,j,k}^{\mathrm{C}} \leq d_j\right] \geq 1 - \varepsilon_j, \tag{2}$$

where ε_j is the probability that the completion time of a task of class j exceeds the required deadline. This commonly used constraint places a limit on the probability that task deadline targets are violated [8]. Note that $t_{n,j,k}^{\mathrm{W}}$ takes discrete values in number of time slots, $t_{n,j,k}^{\mathrm{C}}$ takes discrete values that are multiple of CPU cycle periods, while d_j is continuous in seconds.

Our objective is to create an allocation that minimizes the average MD energy consumption under the deadline and cost budget constraints. This can now be written as follows.

$$\min_{x_n, y} \textstyle\sum_{n=1}^{N} \left[P_{\mathrm{B}n} \lambda_n P^{\mathrm{L}} \bar{L} + (1 - P_{\mathrm{B}n})\,\lambda_n P^{\mathrm{T}} \bar{t}_n^{\mathrm{W}}\right] \text{s.t.} \tag{3}$$

$$x_n \leq K_n,\ \forall n \in \mathcal{N} \tag{4}$$

$$\textstyle\sum_{n=1}^{N} \alpha_n x_n + \beta y f^{\mathrm{C}} \leq B^{\max}, \tag{5}$$

$$\Pr\left[t_{n,j,k}^{\mathrm{W}} + t_{n,j,k}^{\mathrm{C}} \leq d_j\right] \geq 1 - \varepsilon_j, \forall n \in \mathcal{N}, j \in \mathcal{J}, k \in \mathcal{I}_n \tag{6}$$

$$\tilde{\lambda} \leq yf^{\mathrm{C}} / \textstyle\sum_{j=1}^{J} P_j^{\mathrm{C}} q_j, \tag{7}$$

$$x_n \in \mathbb{N},\ \forall n \in \mathcal{N} \tag{8}$$

$$y \in [0, 1]\,. \tag{9}$$

In this formulation, constraint (4) ensures that the number of channels assigned does not exceed the maximum number available in each BS; constraint (5) makes sure that the cost of the allocation does not exceed the cost budget; constraint (6) requires that the probability that tasks are completed before their deadline is bounded from below, and constraint (7) ensures that the queue at ES is stable. The optimization problem formulated in (3)-(9) is a mixed integer nonlinear programming (MINLP) problem. In general, MINLP problems are NP-hard and thus no efficient solutions exist.

III. Approximate solution

In this section, we propose an approximate solution for the optimization problem (3)-(9) by decomposing it into several convex subproblems that can be solved efficiently. More specifically, we discretize variable $y \in [0, 1]$ by breaking $[0, 1]$ into Y equal segments, so that y takes values $y_a = a/Y$, for $a = 0, 1, \ldots, Y$. With y fixed, we show that the relaxation of (3)-(9) can be approximated by a convex optimization problem, which can be solved in polynomial time. The resulting (fractional) x_n's are then rounded to integer values (and this is another source of suboptimality for our solution method). After solving the resulting $Y + 1$ problems, we output the minimum solution x^*, y^*. Obviously, the quality of the approximation depends on the discretization parameter Y.

We consider the relaxed version of problem (3)-(9), i.e., constraint (8) has been replaced by $x_n \geq 0, \forall n$. With y fixed, we show that the non-convex problem (3)-(9) can be transformed into an equivalent convex optimization problem with the $P_{\mathrm{B}n}$'s as the decision variables. First, we concentrate on constraints (6), (7). Note that the distribution of $\Pr[t_{n,j,k}^{\mathrm{W}} + t_{n,j,k}^{\mathrm{C}} \leq d_j]$ is a monotonically decreasing function of the aggregate mean task arrival rate $\tilde{\lambda}$. Hence, by binary search in the range $[0, yf^{\mathrm{C}} / \sum_{j=1}^{J} P_j^{\mathrm{C}} q_j]$, we can approximate within any desired accuracy the maximum possible value of $\tilde{\lambda}$ that satisfies constraints (6) for all n, j, k. Let λ^* be this maximum value (note that $\lambda^* < \mu^{\mathrm{C}}$, so stability is ensured). Using (1), constraints (6), (7) can be replaced by constraint $\sum_{n=1}^{N} (1 - P_{\mathrm{B}n})\,\lambda_n \leq \lambda^*$.

Next, we note that the blocking probability $P_{\mathrm{B}n}$ is monotonically decreasing to x_n; let $P_{\mathrm{B}n}^{\min}$ be the blocking probability when $x_n = K_n$. Then constraints (4) can be replaced by the equivalent constraints $P_{\mathrm{B}n}^{\min} \leq P_{\mathrm{B}n} \leq 1,\ \forall n$.

Constraint (5) is the only remaining constraint with an explicit dependence on the x_n's. Since $P_{\mathrm{B}n}$ is a function of x_n, one could potentially use its inverse to replace x_n with a function of $P_{\mathrm{B}n}$. However, such an inversion function may not exist explicitly (and even if it does, it may be non-convex). In its stead, we can use a convex upper bound approximation F of the inversion of blocking probability, so that $x_n \leq F(P_{\mathrm{B}n}),\ \forall n$.

Hence, the new convex optimization problem that approximates the original one when y is fixed, is the following:

$$\min_{P_{\mathrm{B}n}} \textstyle\sum_{n=1}^{N} \left[P_{\mathrm{B}n} \lambda_n P^{\mathrm{L}} \bar{L} + (1 - P_{\mathrm{B}n})\,\lambda_n P^{\mathrm{T}} \bar{t}_n^{\mathrm{W}}\right] \text{s.t.} \tag{10}$$

$$\textstyle\sum_{n=1}^{N} \alpha_n F(P_{\mathrm{B}n}) + \beta y f^{\mathrm{C}} \leq B^{\max} \tag{11}$$

$$\textstyle\sum_{n=1}^{N} (1 - P_{\mathrm{B}n})\,\lambda_n \leq \lambda^* \tag{12}$$

$$P_{\mathrm{B}n}^{\min} \leq P_{\mathrm{B}n} \leq 1,\ \forall i \in \mathcal{N}. \tag{13}$$

After solving (10)-(13) and obtaining the $P_{\mathrm{B}n}$'s, we can compute the largest integral x_n^* which achieves a blocking probability equal or bigger than $P_{\mathrm{B}n}$, for all $n \in \mathcal{N}$. Algorithm GCA (cf. Algorithm 1) codifies the solution method described above.

Algorithm 1 General Case Approximation (GCA)

Require: $\lambda_n, P^{\mathrm{T}}, P^{\mathrm{L}}, \alpha_n, K_n, \beta, f^{\mathrm{C}}, Y, s_j, d_j, L_j P_j^{\mathrm{C}}$, PDFs of $t^{\mathrm{W}}, t^{\mathrm{C}}$

1: $cost^* = \infty$
2: **for all** $a = 0, \ldots, Y$ **do**
3: $y = a/Y$
4: Obtain λ^*, the upper bound of $\tilde{\lambda}$, by binary search in $[0, \mu^C]$
5: $P_{\mathrm{B}}, cost =$ solution cost of (10)-(13)
6: $x^{int} =$ max integral x with blocking probabilities $\geq P_{\mathrm{B}}$
7: **if** $cost < cost^*$ **then**
8: $x^* = x^{int}; y^* = y; cost^* =$ cost of x^*, y^*
9: **end if**
10: **end for**
11: **return** x^*, y^*

In the remainder of this paper we make the common assumption that tasks arrive from the MDs at BS n according to a Poisson process with mean arrival rate λ_n. In this case, we can invoke the *insensitivity property* of the Erlang B formula, to compute the probability of blocking at each BS [9]. Note that typically, the Erlang B result is derived using the $M/M/N/N$ Markovian queue, which assumes exponentially distributed channel upload (i.e., service) times [10]. Due to insensitivity, the result holds for any service time distribution with the same mean. Therefore, the blocking probability for a task arriving at BS n can be written as

$$P_{\mathrm{B}n} = \left(\frac{\lambda_n}{\mu_n^{\mathrm{W}}}\right)^{x_n} \frac{1}{x_n!} \left[\sum_{r=0}^{x_n} \left(\frac{\lambda_n}{\mu_n^{\mathrm{W}}}\right)^r \frac{1}{r!}\right]^{-1} \tag{14}$$

where μ_n^{W} denotes the mean service rate, i.e., $\mu_n^{\mathrm{W}} = 1/\bar{t}_n^{\mathrm{W}}$. Function (14) is convex in x_n [11].

Note that due to the Poisson process job arrival assumption, the channel state sampled by arriving jobs is given by the steady-state equilibrium probability distribution of the Markovian channel at that MD. This follows from the PASTA rule [12].

We assume that the aggregate task arrival process at ES is Poisson [13], and, therefore, arriving tasks sample the asymptotic equilibrium state distribution of ES. This approximation is justified due to the mixing of arrivals at ES from BSs operating independently, and has been verified in our simulation experiments. In this case, the ES can be modeled as an M/G/1 queue, whose waiting time is given by the random variable w^{C}. Given $\tilde{\lambda}$ and knowledge of the data upload distribution, the distribution of w^{C} can be obtained by numerical inversion of the probability generating function of system waiting time for M/G/1 [14]. In this case, $t_{n,j,k}^{\mathrm{C}} = w^{\mathrm{C}} + s_j/yf^{\mathrm{C}}$ and $\Pr[t_{n,j,k}^{\mathrm{W}} + t_{n,j,k}^{\mathrm{C}} \leq d_j]$ can be easily obtained.

In order to apply algorithm GCA (Algorithm 1), the upper bound F used in problem (10)-(13) is the following [15]: $x_n \leq \lambda_n(1 - P_{\mathrm{B}n})/\mu_n^{\mathrm{W}} + 1/P_{\mathrm{B}n}, \ \forall n$. Then problem (10)-(13) becomes:

$$\min_{P_{\mathrm{B}n}} \sum_{n=1}^{N} \left[P_{\mathrm{B}n}\lambda_n P^{\mathrm{L}} \bar{L} + (1 - P_{\mathrm{B}n})\, \lambda_n P^{\mathrm{T}} \bar{t}_n^{\mathrm{W}}\right] \text{s.t.} \tag{15}$$

$$\sum_{n=1}^{N} \alpha_n \left(\frac{\lambda_n}{\mu_n^{\mathrm{W}}}(1 - P_{\mathrm{B}n}) + \frac{1}{P_{\mathrm{B}n}}\right) + \beta y f^{\mathrm{C}} \leq B^{\max} \tag{16}$$

$$\sum_{n=1}^{N} (1 - P_{\mathrm{B}n})\lambda_n \leq \lambda^* \tag{17}$$

$$P_{\mathrm{B}n}^{\min} \leq P_{\mathrm{B}n} \leq 1, \ \forall i \in \mathcal{N}. \tag{18}$$

Problem (15)-(18) is convex, and can be solved in time $O(\mathcal{L})$, for a polynomial $\mathcal{L}$. Hence Algorithm 1 has a running time of $O(Y(\mathcal{L} + \log \frac{\mu^{\mathrm{C}}}{\epsilon}))$, where Y is the granularity of y, and $O(\log \frac{\mu^{\mathrm{C}}}{\epsilon})$ is the binary search cost of line 4 of the algorithm, in order to get a λ^* within ϵ of the optimal.

IV. Simulation Results

In this section, we present simulation results to demonstrate the performance of algorithm GCA (Algorithm 1) for different system parameter values. We will assume that the jobs generated at the MDs have a fixed bit-size s and a fixed computation load q, i.e., $s_j = s$ and $q_j = q$ for all j, and that each channel model is a two-state Gilbert-Elliot channel [16], i.e., a Markov chain with two states, "Good" (G) and "Bad" (B). This model is commonly used to characterize the effects of burst noise in wireless channels, where the channel can abruptly transition between good and bad conditions [17]. Thus, the transition probability matrix for wireless model k in BS n is given by $P_{n,k}^{\mathrm{GG}}, P_{n,k}^{\mathrm{GB}}, P_{n,k}^{\mathrm{BG}}$, and $P_{n,k}^{\mathrm{BB}}$. Since there is only one class of the tasks, subscript j can be dropped from the notations.

Denote $\pi_{n,k}^{\mathrm{G}}$ and $\pi_{n,k}^{\mathrm{B}}$, respectively, as the stationary probabilities of channels with G and B states in BS n with propagation model k. In each time slot, the channel state Markov chain transitions in accordance with these probabilities. We assume that each task can be uploaded at one time slot when the channel is in the G state, hence, we can obtain the distribution of wireless transmission time $t_{n,k}^{\mathrm{W}}$ in BS n with channel model k, i.e., the probability that one task in BS n with channel model k can be uploaded in l time slots is given as follows

$$\Pr\left[t_{n,k}^{\mathrm{W}} = l\right] = \begin{cases} \pi_{n,k}^{\mathrm{G}}, \ l = 1 \\ \pi_{n,k}^{\mathrm{B}} {P_{n,k}^{\mathrm{BB}}}^{l-2} P_{n,k}^{\mathrm{BG}}, \ l \geq 2 \end{cases} \tag{19}$$

We can then calculate the mean wireless transmission time of tasks in BS n with propagation model k as follows

$$\bar{t}_{n,k}^{\mathrm{W}} = \sum_{l=1}^{\infty} l \Pr\left[t_{n,k}^{\mathrm{W}} = l\right] = 1 + \frac{P_{n,k}^{\mathrm{GB}}}{{P_{n,k}^{\mathrm{BG}}}^2 + P_{n,k}^{\mathrm{GB}} P_{n,k}^{\mathrm{BG}}}. \tag{20}$$

Thus, we can obtain the mean wireless transmission time of tasks in BS n: $\bar{t}_n^{\mathrm{W}} = \sum_{k=1}^{I_n} \eta_{n,k} \bar{t}_{n,k}^{\mathrm{W}}$, where $\eta_{n,k}$ is the probability of tasks in BS n with channel model k. In this case, the ES becomes an M/D/1 queueing system, $t_{n,j,k}^{\mathrm{C}} = t^{\mathrm{C}}$, for all n, j and k, and the distribution of delay is given by

[18]

$$\Pr\left[t^{\mathrm{C}} \le \hat{t}\right] = \left(1 - \frac{\tilde{\lambda}}{\mu^{\mathrm{C}}}\right) \sum_{z=0}^{\lfloor \hat{t}\mu^{\mathrm{C}} \rfloor} \frac{\left[\tilde{\lambda}\left(\frac{z}{\mu^{\mathrm{C}}} - \hat{t}\right)\right]^z}{z!} e^{-\tilde{\lambda}\left(\frac{z}{\mu^{\mathrm{C}}} - \hat{t}\right)} \tag{21}$$

where $\mu^{\mathrm{C}} = yf^{\mathrm{C}}/q$ and $\hat{t}$ is the delay tolerance for ES execution.

For comparison, we also use a discrete event simulation (DES) of the system using the x_n's and y values obtained by the proposed algorithm to validate our model assumptions. In addition, we simulate an optimal scheme (i.e., DES-based OPT) as follows. We first obtain all the possible combinations of x_n's under constraint (4); for each combination, we calculate y from (5), (9), and then check if constraint (7) is satisfied by the current values of x_n's and y. If not, we go to the next combination of x_n's and repeat this procedure. Otherwise, we use this set of x_n's and y to run the DES for the system, and then check if (6) is satisfied. If not, we proceed to the next combination of x_n's and repeat the above procedure. If the constraints are satisfied, we save the obtained energy consumption. After going through all the possible combinations of x_n's, we obtain the minimum energy consumption and the corresponding x_n's and y.

In the simulation, we consider a network consisting of 3 BSs. The tasks arrive at the BSs according to the Poisson process with average arrival rates $\lambda_1 = 11$, $\lambda_2 = 13$ and $\lambda_3 = 15$ tasks per second. There are two propagation models at each BS with transition probabilities $P_{n,1}^{\mathrm{GG}} = 0.9$, $P_{n,2}^{\mathrm{GG}} = 0.7$, $P_{n,1}^{\mathrm{BB}} = 0.1$, and $P_{n,2}^{\mathrm{BB}} = 0.3$ for $n = 1, 2, 3$. The probabilities of the different channel models in BS 1 are $\eta_{1,1} = 0.8$ and $\eta_{1,2} = 0.2$; those in BSs 2 and 3 are $\eta_{2,1} = 0.5$, $\eta_{2,2} = 0.5$, $\eta_{3,1} = 0.2$, and $\eta_{3,2} = 0.8$. The default parameters used in the simulations are summarized in Table I.

TABLE I
DEFAULT SYSTEM PARAMETERS

Parameter	Value
Size of input data of tasks s	2 Mbits
Tolerable delay of tasks d	4s
Task computation load q	3 M CPU cycles
Available ES capacity f^{C}	75 M cycles/s
Available number of channels in BSs K_n	[15 15 20]
Unit price of channel α_n	[1 1 1]$
Unit price of ES CPU speed β	0.5×10^{-6} $
Local processing speed f	1M cycles/s
Local execution power P^{L}	250 mW
Offloading transmission power P^{T}	2.5 mW
Cost budget $B^{\max}$	140 $

Fig. 2 shows the total average energy consumption of MDs versus $B^{\max}$, which is the cost budget of the network customer. When the tolerable violation of latency ε is 1%, the total average energy consumption of MDs for all schemes is a constant, which means that all the tasks are executed locally regardless of the cost budget. This is because the delay constraints cannot be satisfied if the tasks are offloaded due to the tight latency violation condition. When ε is 3% and 5%, some tasks are allowed to be offloaded, and the energy

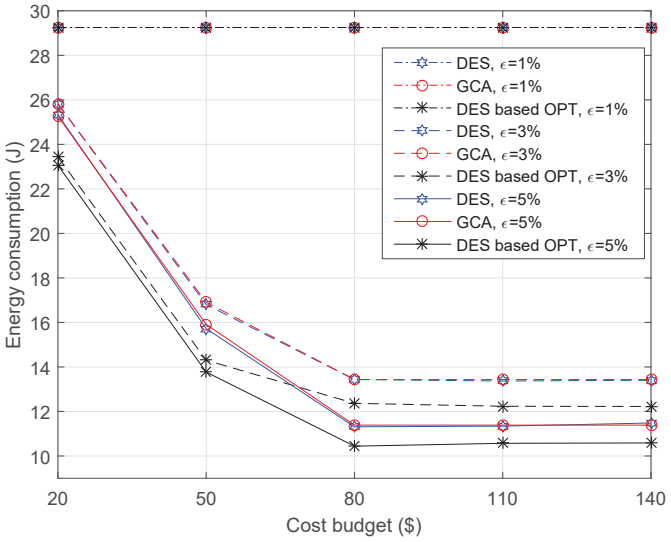

Fig. 2. Average energy consumption versus cost budget

consumption of the MDs decreases as $B^{\max}$ increases for all schemes. This happens because when the cost budget is small, the optimization is constrained by the cost budget, which limits the number of offloaded tasks; but with the increase of $B^{\max}$, more channels or ES capacity can be allocated, leading to more MDs offloading their tasks. When $B^{\max}$ becomes large, the budget constraint is not tight anymore, and the task offloading completion is mainly affected by the changing wireless transmission conditions. It also shows that the average MD energy consumption decreases as ε increases for all schemes, since larger ε makes it easier to meet the latency constraint through offloading, which results in more offloaded tasks and saves energy in the MDs.

By comparing the total average MD energy consumption for $\varepsilon = 3\%$ and $\varepsilon = 5\%$ in Fig. 2, it is seen that the gap is small when the cost budget is small, but then increases as the cost budget increases, and finally becomes constant as the budget constraint becomes non-tight. When the cost budget is small, the amount of channel resources is limited, most tasks have to be executed locally, and the value of ε has little effect on the energy consumption of the MDs. As the cost budget increases, more channel resources are available, and the offloading decisions are determined by both ε and the available channel resources. When the cost budget is sufficiently large, the offloading decisions are less affected by the cost budget. The figure also shows that the average MD energy consumption using GCA is almost the same as DES, which validates the approximations used in our solution. The performance of GCA is also close to DES-based OPT, which further shows good performance of the former.

Fig. 3 shows the MD energy consumption versus λ_n (same for all BSs). The energy consumption increases linearly with λ_n for all schemes, since both local average execution energy and uploading average transmission energy are proportional to the mean task arrival rate. Fig. 4 shows the average MD

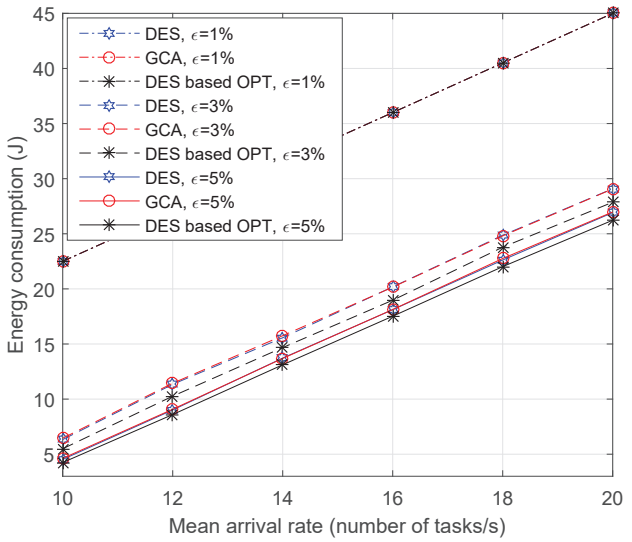

Fig. 3. Average energy consumption versus mean arrival rate

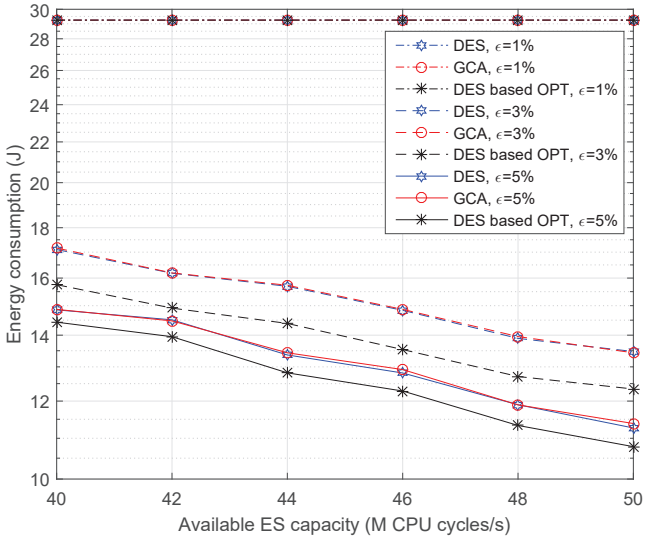

Fig. 4. Average energy consumption versus available ES capacity

energy consumption versus C^{B}, i.e., the available ES capacity. When $\varepsilon = 1\%$, the total average energy consumption of MDs is a constant for all the schemes, since all tasks are executed locally. When $\varepsilon = 3\%$ and 5%, offloading is possible for some tasks, and the number of tasks that can be offloaded increases with the ES capacity, resulting in lower energy consumption of the MDs. Figs. 3 and 4 also show that the performance of our GCA solution is very close to both DES and DES-based OPT, which demonstrates the good performance of GCA and validates the approximations in our solution.

V. Conclusions

We have studied joint wireless network and job service allocation for mobile computation offloading. Our objective is to minimize the total average energy consumption of MDs for completing the arriving tasks, while satisfying the delay constraints of tasks and the cost budget of the network customer. An MINLP problem was formulated and an approximate solution was proposed for the optimization problem by decomposing it into a collection of convex subproblems, which can be solved efficiently. Our results have demonstrated the viability and efficiency of the proposed solution, which can achieve close-to-optimum performance for a wide range of system parameters.

References

[1] H. Ba, W. Heinzelman, C.-A. Janssen, and J. Shi, "Mobile computing - A green computing resource," in *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC)*, July 2013, pp. 4451–4456.

[2] G. Huerta-Canepa and D. Lee, "A virtual cloud computing provider for mobile devices," in *Proceedings of the 1st ACM Workshop on Mobile Cloud Computing Services: Social Networks and Beyond*, June 2010, p. 6.

[3] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "CloneCloud: Elastic Execution between Mobile Device and Cloud," in *Proceedings of the Sixth Conference on Computer Systems*, ser. EuroSys '11. New York, NY, USA: ACM, 2011, pp. 301–314. [Online]. Available: http://doi.acm.org/10.1145/1966445.1966473

[4] Huawei Inc., "5g network architecture - a high-level perspective," https://www.huawei.com/en/technology-insights/industry-insights/outlook/mobile-broadband/insights-reports/5g-network-architecture, 2016.

[5] K. Kumar and Y.-H. Lu, "Cloud Computing for Mobile Users: Can Offloading Computation Save Energy?" *IEEE Computer*, vol. 43, no. 4, pp. 51–56, 2010.

[6] O. Muñoz, A. Pascual-Iserte, and J. Vidal, "Optimization of Radio and Computational Resources for Energy Efficiency in Latency-Constrained Application Offloading," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 10, pp. 4738–4755, October 2015.

[7] L. Cominardi, T. Deiss, M. Filippou, V. Sciancalepore, F. Giust, and D. Sabella, "Mec support for network slicing: Status and limitations from a standardization viewpoint," *IEEE Communications Standards Magazine*, vol. 4, no. 2, pp. 22–30, 2020.

[8] H. Park, Y. Jin, J. Yoon, and Y. Yi, "On the economic effects of user-oriented delayed wi-fi offloading," *IEEE Transactions on Wireless Communications*, vol. 15, no. 4, p. 26842697, 2016.

[9] D. Y. Burman, "Insensitivity in queueing systems," *Advances in Applied Probability*, vol. 13, no. 4, pp. 846–859, 1981.

[10] D. J. Daley and L. D. Servi, "Idle and busy periods in stable m/m/k queues," *Journal of Applied Probability*, vol. 35, no. 4, pp. 950–962, 1998.

[11] E. J. Messerli, "B.s.t.j. brief: Proof of a convexity property of the erlang b formula," *The Bell System Technical Journal*, vol. 51, no. 4, pp. 951–953, 1972.

[12] R. W. Wolff, "Poisson arrivals see time averages," *Operations Research*, vol. 30, no. 2, pp. 223–414, 1982.

[13] D. N. Shanbhag and D. G. Tambouratzis, "Erlang's formula and some results on the departure process for a loss system," *Journal of Applied Probability*, vol. 10, no. 1, pp. 233–240, 1973.

[14] A. Y. Khintchine, "Mathematical theory of a stationary queue," *Matematicheskii Sbornik*, vol. 39, no. 4, pp. 73–84, 1932.

[15] S. A. Berezner, A. E. Krzesinski, and P. G. Taylor, "On the inverse of erlang's function," *Journal of Applied Probability*, vol. 35, no. 1, pp. 246–252, 1998.

[16] E. N. Gilbert, "Capacity of a burst-noise channel," *The Bell System Technical Journal*, vol. 39, no. 5, pp. 1253–1265, 1960.

[17] T. Blazek and C. F. Mecklenbräuker, "Measurement-based burst-error performance modeling for cooperative intelligent transport systems," *IEEE Transactions on Intelligent Transportation Systems*, no. 99, pp. 1–10, 2018.

[18] G. J. Franx, "A simple solution for the m/d/1 waiting time distribution," *Operations Research Letters*, vol. 29, no. 5, pp. 221–229, 2001.