Optimal Offloading for Dynamic Compute-Intensive Applications in Wireless Networks

Bin Li
Department of Electrical, Computer and Biomedical Engineering
University of Rhode Island
Kingston, Rhode Island, USA
Email: binli@uri.edu

***Abstract*—With the rapid growth of wireless compute-intensive services (such as image recognition, real-time language translation, or other artificial intelligence applications), efficient wireless algorithm design should not only address when and which users should transmit at each time instance (referred to as *wireless scheduling*) but also determine where the computation should be executed (referred to as *offloading decision*) with the goal of minimizing both computing latency and energy consumption. Despite the presence of a variety of earlier works on the efficient offloading design in wireless networks, to the best of our knowledge, there does not exist a work on the realistic user-level dynamic model, where each incoming user demands a heavy computation and leaves the system once its computing request is completed. To this end, we formulate a problem of an optimal offloading design in the presence of dynamic compute-intensive applications in wireless networks. Then, we show that there exists a fundamental logarithmic energy-workload tradeoff for any feasible offloading algorithm, and develop an optimal threshold-based offloading algorithm that achieves this fundamental logarithmic bound.**

***Index Terms*—Offloading, user-level dynamics, compute-intensive applications, low latency, and energy efficiency.**

I. Introduction

With the recent advances in Artificial Intelligence (AI) techniques, there is a strong need for pushing machine intelligence to mobile devices, such as smartphones, tablets, e-readers, and smart watches. The reason behind this is that mobile devices became aware of users' location, and are capable of providing convenient and seamless connections from anywhere at anytime, and closely accompany humans that helps in learning users' behavioral patterns and monitoring users' health status. For example, users may use mobile devices to recognize an image, translate English to foreign languages (such as Chinese, French, and Spanish) in real-time, or even diagnose their diseases by recording their appearances, behaviors and symptom descriptions. Therefore, intelligence applications are nowadays increasingly developed for every category of mobile services. To support such a technology need and trend, technology giants have recently developed various AI development platforms, such as Facebook Caffe2 and Google TensorFlow Lite, to facilitate and speedup the design of mobile intelligent applications.

Different from traditional wireless data/voice/video services, mobile intelligence services typically require *intensive computations* while demanding low energy consumption and low latency. On one hand, each mobile user has limited computing power and thus can complete computations with a relatively slow speed and large energy consumption, while edge servers usually have high-performance computing units and can process a large amount of computing tasks in a much faster way. On the other hand, if all computing tasks are uploaded to edge servers through resource-restrictive wireless networks, they will not only result in a large transmission delay but also consume a large amount of energy. Therefore, in order to minimize both processing delay of computing tasks and the energy consumption of mobile users, each mobile user needs to decide whether the incoming computing task is processed in his/her local device or is uploaded to edge servers via wireless.

While the offloading design in wireless networks has received great research interest, much of prior work (see [1] for a thorough survey) used simulations and heuristics to reveal its advantage over both local and cloud computing only, and lacks of its fundamental understanding and algorithm design with provable performance guarantees, which is crucial especially with the explosive growth of wireless intelligent applications with high performance needs. Some recent work tried to explore this important direction. For example, [2]–[4] focused on optimal offloading design in single-user cases. [5]–[9] studied the multi-user cases with static users and computational workloads. None of existing work focuses on the realistic case with dynamic users and random computational workloads, which is the typical feature of mobile computing networks and is the main focus of this paper. Although several efficient wireless scheduling algorithms (e.g., [10]–[13]) have been developed in the presence of dynamic users, they did not address offloading designs. Therefore, the proposed solutions in existing work on offloading design (e.g., [5]–[9]) and scheduling design (e.g., [10]–[13])) do not apply and new algorithm designs are required to address the optimal offloading design in the presence of dynamic compute-intensive applications.

To that end, we consider the optimal offloading design in the realistic user-level dynamic model, where each incoming user demands a heavy computation and leaves the system once its computation is completed. Our main contributions are listed as follows:

- In Section II, we formulate a problem of an optimal offloading design in the presence of dynamic users.
- In Section III, we first derive a universal energy-workload

tradeoff for any feasible offloading algorithm, and then develop an optimal algorithm that achieves this fundamental logarithmic bound.

II. System Model

We consider a wireless system with one access point (AP) equipped with powerful servers (referred to as *edge servers*), where each incoming user carries a *compute-intensive* task and leaves the system once its computation is completed. Here, each incoming user should make an *offloading decision* that determines whether its computing task is executed by its portable device with limited computing capability or by powerful edge servers via wireless transmission. Due to wireless interference, at most one user can be scheduled for data transmission at each time instance, which is determined by the AP (referred to as *wireless scheduling*). We assume that the system operates in a *time-slotted* manner, where each mobile user randomly arrives and makes an offloading decision at the beginning of each time slot and the scheduling decision is made by the AP at the end of each time slot.

To facilitate our mathematical modeling, we unify the units for the processing speed of each mobile user and wireless transmission rate, and *ignore the computing time in powerful edge servers*. In particular, we assume that the number of packets[1] that a newly arriving mobile user carries in each time slot follows an arbitrary probability distribution with a finite support, and is independently and identically distributed (i.i.d.) over time and users with the same probability distribution as that of a random variable $\widetilde{F}$.

Let $A_F[t]$ denote the number of arriving users that have a size of F packets in time slot t that is i.i.d. over time with mean $\lambda_F > 0$. Let $F^{\max}$ denote the maximum number of packets a newly user can carry. We also use $\mathcal{A}_F[t]$ to denote the set of arriving users with the size of F packets in time slot t. Let $A[t] \triangleq \sum_F A_F[t]$ denote the number of users arriving in time slot t and thus its mean $\lambda \triangleq \mathbb{E}[A[t]]$ is equal to $\sum_F \lambda_F$. We assume that $A[t] \leq A^{\max}$ for some $A^{\max} > 0$ for all $t \geq 0$, and $q \triangleq \Pr\{A[t] = 0\} > 0$.

Each mobile user j has a limited computing power and thus can only process μ_L packets in each time slot. We use $C_j[t]$ to capture wireless channel fading of user j, which measures the maximum number of packets that can be transmitted in time slot t if user j is scheduled for data transmission. We assume that $\{C_j[t], j \in \mathcal{N}[t]\}$ are independently distributed across users and i.i.d. over time with a finite support[2], where $\mathcal{N}[t]$ denotes the set of mobile users awaiting for wireless transmission in time slot t. Let $C^{\max}$ denote the maximum channel rate. Here, we reasonably assume that the probability that each mobile user achieves the maximum channel rate is strictly positive, i.e., $p_{\max} \triangleq \Pr\{C_j[t] = C^{\max}\} > 0, \forall j \in \mathcal{N}[t], t \geq 0$. We assume that $C^{\max} > \mu_L > 0$. Indeed, in [14], the authors showed that it is faster to transmit the task to edge servers via WiFi than that executed by its mobile device. Fig. 1 shows a snapshot of a wireless system with five mobile users with different computing demands.

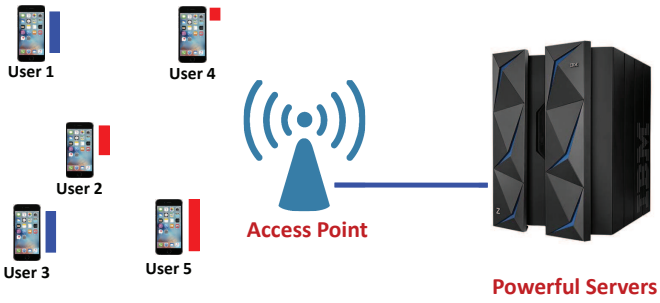


Fig. 1: System model: "blue" and "red" computing tasks denote that they are being processed in local devices and are being uploaded to edge servers, respectively.

To capture the heterogeneous energy consumption due to the computation and wireless communication of each mobile user, we use e_L and e_E to denote the unit power consumption of executing computing tasks and wireless communication, respectively. We assume that $e_L > e_E > 0$. Indeed, it has been reported in [14] that mobile CPU and GPU consume 6.45 watt and 7.89 watt, respectively, while it only takes 4 watt for wireless transmission via WiFi.

We use h_L and h_E to denote the minimum number of time slots required for the user to complete its local computation and transmission to edge servers, respectively, and thus $\mathbf{h} \triangleq (h_L, h_E)$ denotes an offloading decision vector. Since we consider the case that the computation is executed either in a local device or in edge servers via wireless communication, for each incoming user with the size of F, $\mathbf{h}$ is equal to either $(\lceil F/\mu_L \rceil, 0)$ or $(0, \lceil F/C^{\max} \rceil)$. We are interested in minimizing average energy consumption while meeting the desired throughput. This can be achieved by solving the following optimization problem.

$$\min_{\{\alpha(\mathbf{h}|F), \forall \mathbf{h}, F \geq 0\}} \quad \sum_F \lambda_F \sum_{\mathbf{h}} \alpha(\mathbf{h}|F) \left(e_L h_L + e_E h_E\right) \tag{1}$$

$$s.t. \quad \sum_F \lambda_F \sum_{\mathbf{h}} \alpha(\mathbf{h}|F) h_E \leq 1, \tag{2}$$

$$\alpha(\mathbf{h}|F) \geq 0, \forall \mathbf{h}, F \geq 0, \tag{3}$$

$$\sum_{\mathbf{h}} \alpha(\mathbf{h}|F) = 1, \forall F \geq 0, \tag{4}$$

$$\alpha(\mathbf{h}|F) = 0 \quad \text{if } F > h_L \mu_L + h_E C^{\max}, \tag{5}$$

where $\alpha(\mathbf{h}|F)$ denotes the probability that the user with the size of F packets makes an offloading decision vector $\mathbf{h}$. In the above optimization problem, the objective function is to minimize the total average energy consumption of mobile users while guaranteeing all service requests are fulfilled. Inequality (2) is the *network capacity constraint* which means that the average workload awaiting for wireless transmissions should not be greater than 1. Inequality (5) states that the offloading

[1]Here, a packet refers to as a minimum amount of computing and wireless communication units.

[2]Due to the finite number of modulation and coding schemes, each mobile user has a finite number of channel rates.

decision for each mobile user should guarantee sufficient service for completing his/her service request.

Since $C^{\max} > \mu_L$, for each incoming user j, its local workload is always greater than its network workload, i.e., $h_{j,L} > h_{j,E}$. This together with the fact that $e_L > e_E$ implies that it is better to upload all computing tasks in order to minimize energy consumption if the network capacity constraint (2) is satisfied. To avoid this trivial solution, we assume that $\rho \triangleq \lambda \mathbb{E}\left[\left\lceil \widetilde{F}/C^{\max} \right\rceil\right] \geq 1$ and thus the network capacity constraint (2) is violated if all incoming tasks are uploaded to edge servers via wireless communication. However, it is worth mentioning that our developed algorithm (cf. Section III-B) can easily address this case.

In this paper, we focus on the offloading design for each user that optimally solves optimization problem (1)-(5). In the rest of the paper, we consider the following Maximum-Channel-Rate-First scheduling policy (see [10], [11]) within the AP: in each time slot, the AP always serves a user $j^*[t]$ with the maximum channel rate among all its existing users awaiting for wireless transmission, breaking ties uniformly at random, i.e., $j^*[t] \in \operatorname{argmax}_{j\in\mathcal{G}[t]} C_j[t]$, where $\mathcal{G}[t] \triangleq \{j : h_{j,E}[t] > 0\}$ denotes the set of mobile users that are awaiting for uploading their computing tasks to edge servers in time slot t. As we show later, our proposed offloading algorithm together with this specific scheduling policy solves the problem (1)-(5).

We note that our considered user-level dynamic model differs from traditional stochastic optimization framework (e.g., [15]) in the following two aspects: (i) *The dynamics of mobile users is short-term and users will leave the network once they complete their computing tasks*; (ii) *Each mobile user suffers from an independent channel fading and have different service rates in different time slots*. These differences pose significant challenges on our efficient algorithm design and analysis.

III. Optimal Offloading Algorithm Design

In this section, we propose an offloading algorithm that achieves the optimal tradeoff between energy consumption and total network workload. To that end, we first develop a universal energy-workload bound under any policy.

A. Fundamental Energy-Workload Bound

We first characterize the underlying dynamics of mobile users by introducing the following notations. Let $W[t]$ denote the total network workload in time slot t that measures the minimum number of slots required for completing all existing computing requests via wireless communications. Then, the evolution of $W[t]$ can be described as follows:

$$W[t+1] = W[t] + \nu[t] - \gamma[t], \tag{6}$$

where $\nu[t] \triangleq \sum_F \sum_{j\in\mathcal{A}_F[t]} h_{j,E}[t]$ denotes the total network workload increment due to newly arriving users under the offloading decision vector $\mathbf{h}_j[t] \triangleq (h_{j,L}[t], h_{j,E}[t])$ for each user j in time slot t, and $\gamma[t]$ is the amount of network workload decreasing in time slot t.

Next, we prove that there exists a fundamental logarithmic tradeoff between the average energy consumption and the average network workload, i.e., the network workload is at least $\Omega(\log K)$ under any policy that achieves a long-term average energy expenditure within $O(1/K)$ of the optimal solution to the Problem (1)-(5), where $K > 0$ is some parameter.

Proposition 1: If the policy ϕ yields an average energy consumption $P^{(\phi)}$ satisfying

$$P^{(\phi)} \leq P_{\min} + O(1/K), \tag{7}$$

then the average network workload must be at least $\Omega(\log K)$ if $\rho \triangleq \lambda \mathbb{E}\left[\left\lceil \widetilde{F}/C^{\max} \right\rceil\right] \geq 1$, where $P_{\min}$ is the solution to the optimization problem (1)-(5).

Proof: When $\rho \triangleq \lambda \mathbb{E}\left[\left\lceil \widetilde{F}/C^{\max} \right\rceil\right] \geq 1$, it is energy optimal to upload incoming computing tasks with probability $1/\rho$. Therefore, in order to achieve the average energy consumption within $O(1/K)$ of the minimum energy consumption, it should upload incoming computing tasks with probability at least $1/\rho - 1/(K\lambda(e_L h_{L,\max} - e_E h_{E,\max}))$, where $h_{L,\max} \triangleq \mathbb{E}\left[\left\lceil \widetilde{F}/\mu_L \right\rceil\right]$ and $h_{E,\max} \triangleq \mathbb{E}\left[\left\lceil \widetilde{F}/C^{\max} \right\rceil\right]$. Thus, the admitted network workload $\mathbb{E}[\hat{\nu}[t]]$ should satisfy

$$\mathbb{E}[\hat{\nu}[t]] \geq 1 - \delta, \tag{8}$$

where $\delta \triangleq h_{E,\max}/(K(e_L h_{L,\max} - e_E h_{E,\max}))$.

Since we consider the Maximum-Channel-Rate-First scheduling policy in this paper, it is not hard to show that the system is stable. The proof is similar to that in [10], [11] and hence is omitted here. Thus, in steady state, $\Pr\{\gamma[t] = 1\} = \mathbb{E}[\hat{\nu}[t]] \geq 1 - \delta$. The rest of the proof is similar to that of [16, Theorem 2] and we provide here for completeness. Indeed, in steady state,

$$\begin{aligned}
n\delta &\geq \sum_{t=0}^{n-1} \Pr\{\gamma[t] = 0\} \\
&\overset{(a)}{\geq} \Pr\left\{\bigcup_{t=0}^{n-1} \{\gamma[t] = 0\}\right\} \\
&\overset{(b)}{\geq} \Pr\left\{W[0] \leq n-1, \nu[t] = 0, \forall 0 \leq t \leq n-1\right\} \\
&\overset{(c)}{\geq} \Pr\{W[0] \leq n-1\} q^n \\
&= \left(1 - \Pr\{W[0] \geq n\}\right) q^n \\
&\overset{(d)}{\geq} \left(1 - \frac{\mathbb{E}[\widetilde{W}]}{n}\right) q^n \overset{(e)}{\geq} \frac{1}{2} q^n,
\end{aligned} \tag{9}$$

where step (a) uses the union bound; (b) follows from the fact that if the event $\{W[0] \leq n-1, \nu[t] = 0, \forall 0 \leq t \leq n-1\}$ happens, then $\gamma[t] = 0$ for at least one time slot among the first n slots; (c) is true for $q \triangleq \Pr\{\nu[t] = 0\} = \Pr\{A[t] = 0\} \in (0,1)$; (d) is true for $\widetilde{W}$ denoting the steady-state workload and uses Markov Inequality; (e) is true by taking $n = \lceil 2\mathbb{E}[\widetilde{W}] \rceil$.

Therefore, we have

$$\delta \geq \frac{1}{2n} q^n \geq e^{-2n} q^n = e^{-n(2-\log q)}, \tag{10}$$

where the second last step uses the fact that $1/(2x) \geq e^{-x}, \forall x > 0$. Thus, we have

$$n \geq \frac{1}{2 - \log q} \log \frac{1}{\delta}. \tag{11}$$

Since $n = \left\lceil \mathbb{E}[\widetilde{W}] \right\rceil \leq 2\mathbb{E}[\widetilde{W}] + 1$, we have

$$\mathbb{E}[\widetilde{W}] \geq \frac{1}{2} \left(\frac{1}{2 - \log q} \log \frac{1}{\delta} - 1 \right) = \Omega\left(\log K\right), \tag{12}$$

where we recall $\delta \triangleq h_{E,\max}/(K(e_L h_{L,\max} - e_E h_{E,\max}))$. ■

B. Optimal Energy-Workload Tradeoff Algorithm

Having established a fundamental energy-workload logarithmic tradeoff, we develop the following threshold-based policy that achieves this logarithmic tradeoff.

Threshold-Based Offloading (TBO) Algorithm with Parameters $0 < \theta < 1$ and $\overline{W} > 0$: In each time slot t, for each incoming user j, with probability θ/ρ, it attempts to upload all its computing workload to edge servers if $W[t] < \overline{W}$. Otherwise, it keeps all computations in its local device. For those users awaiting for wireless transmissions, the AP deploys the Maximum-Channel-Rate-First Scheduling policy.

Remarks: (1) In the TBO Algorithm, the AP broadcasts its network workload $W[t]$ in each time slot t, and thus each incoming user will upload its computing tasks to edge servers via wireless networks if $W[t] < \overline{W}$, and execute computations locally otherwise.

(2) Even though we assume $\rho \geq 1$, the algorithm can easily adapt to the case with $\rho < 1$ by changing the upload attempting probability θ/ρ to $\min\{1, \theta/\rho\}$.

(3) In our proposed TBO Algorithm, we require the knowledge of arrival intensity ρ, which is usually not available. However, we can use the estimate $\widetilde{\rho}[t] = (1-a)\widetilde{\rho}[t-1] + a\nu[t]$ for some parameter $a \in (0,1)$ to replace ρ.

Next, we will show that the proposed TBO Algorithm can achieve the fundamental logarithmic energy-workload tradeoff. To that end, we first provide an upper bound on the tail probability of network workload under the TBO Algorithm.

Lemma 1: Under the TBO Algorithm, if the admitted throughput θ is equal to $1 - \zeta/2$, then we have

$$\Pr\{W \geq \overline{W}\} \leq M(\zeta) e^{-\overline{W}}, \forall \overline{W} > 0, \tag{13}$$

where

$$M(\zeta) \triangleq e^{\eta(\zeta)(\nu_{\max} + 4H/\zeta)}/(1 - r(\eta(\zeta))), \tag{14}$$

$$\eta(\zeta) \triangleq \frac{1}{4\nu_{\max}} \log\left(1 + \frac{\zeta}{4\nu_{\max}}\right),$$

$$r(\eta(\zeta)) \triangleq e^{\eta(\zeta)\nu_{\max}} - \eta(\zeta)\left(\nu_{\max} + \frac{\zeta}{4}\right) \in (0,1). \tag{15}$$

$H \triangleq \nu_{\max}^2 + \frac{w_{\max}}{-\log(1-p_{\max})} e^{-\log^2(1-p_{\max})}$, $\nu_{\max} \triangleq A^{\max}\lceil F^{\max}/C^{\max}\rceil$, $w_{\max} \triangleq \lceil F^{\max}/C^{\max}\rceil$, and W denotes the steady-state random variable of the workload process under the TBO Algorithm.

Proof: The proof is available in Appendix A. ■

Proposition 2: Under the TBO Algorithm, if θ and $\overline{W}$ satisfy

$$\theta = 1 - \frac{1}{2}\zeta \text{ and } \overline{W} = \log \frac{2M(\zeta)}{\zeta},$$

where $M(\zeta)$ is defined in (14), $\zeta = h_{E,\max}/(K(h_{L,\max}e_L - h_{E,\max}e_E))$, $h_{L,\max} = \mathbb{E}\left[\left\lceil \widetilde{F}/\mu \right\rceil\right]$ and $h_{E,\max} = \mathbb{E}\left[\left\lceil \widetilde{F}/C^{\max} \right\rceil\right]$, then, it yields the average energy consumption within $O(1/K)$ of the minimum energy consumption at the cost of network workload growing with $O(\log K)$.

Proof: By setting $\theta = 1 - \zeta/2$ and $\overline{W}$ to satisfy $\Pr\{W \geq \overline{W}\} \leq \zeta/2$, where $\zeta = h_{E,\max}/(K(h_{L,\max}e_L - h_{E,\max}e_E))$, the network throughput is at least $1 - \zeta$ and thus the average energy expenditure is within $O(1/K)$ of the minimum energy consumption required for network stability. In such a case, the network workload is at most $\overline{W}$, which is equal to $\log(2M(\zeta)/\zeta)$ according to Lemma 1.

Next, we show that $\overline{W} = O(\log K)$. Indeed, we have

$$\begin{aligned} r(\eta(\zeta)) =& e^{\eta(\zeta)\nu_{\max}} - \eta(\zeta)\left(\nu_{\max} + \frac{\zeta}{4}\right) \\ =& \left(1 + \frac{\zeta}{4\nu_{\max}}\right)^{\frac{1}{4}} - \left(1 + \frac{\zeta}{4\nu_{\max}}\right)\log\left(1 + \frac{\zeta}{4\nu_{\max}}\right) \\ =& O\left(1 + \frac{\zeta}{16\nu_{\max}} - \left(1 + \frac{\zeta}{4\nu_{\max}}\right)\frac{\zeta}{4\nu_{\max}}\right) \\ =& 1 - O(\zeta^2). \end{aligned} \tag{16}$$

Therefore, $M(\zeta) = O(1/(1 - r(\eta(\zeta)))) = O(1/\zeta^2)$ and thus $\overline{W} = \log(2M(\zeta)/\zeta) = O(\log(1/\zeta^3))$. This combined with the fact that $\zeta = O(1/K)$ implies that $\overline{W} = O(\log K)$. ■

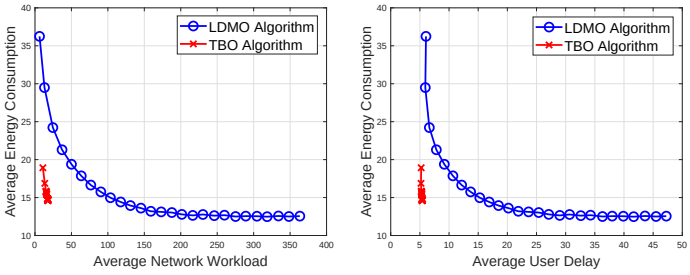
Here, it is worth mentioning that the author in [17] developed an algorithm that achieves optimal energy-delay tradeoff by intelligently dropping packets and controlling transmission power in wireless downlinks. However, the considered setup is quite different from ours, which requires a new technique to perform analysis. In particular, the Kingman bound used to bound the tail probability of workload in [17] does not apply in our case. Instead, we apply [18, Lemma 2.2] to carefully bound such a tail probability and obtain the proper threshold.

IV. Simulation Results

In this section, we study the performance of our proposed TBO Algorithm and compare it with the offloading algorithm derived from the conventional stochastic network optimization framework (e.g. [15]), i.e., in each time slot t, given the current network workload $W[t]$, each arriving user j forwards all its workload to edge servers if $W[t] + K'e_E < K'e_L$, and keeps them for local processing otherwise, where $K' > 0$ is some control parameter. This algorithm is referred to as Lyapunov Drift Minimization based Offloading (LDMO).

In the simulation, we assume that the number of users arriving at the system in each time slot follows a Bernoulli distribution with mean $\lambda = 0.8$. Each user awaiting for wireless transmissions suffers from i.i.d. channel fading with rates $0, 1, 5, 10$ and corresponding probability $0.1, 0.2, 0.5, 0.2$.

The file size $\widetilde{F}$ is equal to 20 with probability $9/19$, and 1 otherwise, and thus the mean file size is equal to 10. We set local processing rate $\mu_L = 1$, $e_L = 7$ and $e_E = 4$. Fig. 2a shows that our developed TBO algorithm achieves a much better energy-workload tradeoff than the LDMO Algorithm derived from the conventional stochastic network optimization framework. Interestingly, we can see from Fig. 2b that the computing latency under the proposed TBO Algorithm is roughly equal to 5, independent from the average energy consumption. Note that if all computing tasks are processed in local devices, it takes 10 time units on average under our setup. Thus, the computation speed under the proposed Threshold-based policy is twice faster than that of local-only approach.



(a) Energy-workload tradeoff (b) Energy-delay tradeoff

Fig. 2: Comparison between TBO and LDMO algorithms

V. Conclusions

In this paper, we first formulated a problem of offloading design with the goal of minimizing average energy consumption while maximizing system throughput in the presence of dynamic users. Then, we proved that no algorithm can beat a logarithmic energy-workload tradeoff and developed an optimal threshold-based algorithm that achieves this fundamental logarithmic bound. Finally, simulation results were provided to demonstrate the efficiency of our proposed algorithm.

Appendix A
Proof of Lemma 1

Under the TB Algorithm with threshold $\overline{W} > 0$, the network workload $W[t]$ evolves as follows:

$$W[t+1] = \min\left\{\overline{W}, W[t] + \widehat{\nu}[t] - \gamma[t]\right\}, \tag{17}$$

where $\mathbb{E}[\widehat{\nu}[t]] = \theta = 1 - \zeta/2$. Let $\mathcal{N}[t]$ be the set of users awaiting for wireless transmission. Define $\{\widehat{W}[t]\}_{t\geq 0}$ with the following evolution

$$\widehat{W}[t+1] = \widehat{W}[t] + \widehat{\nu}[t] - \widehat{\gamma}[t], \tag{18}$$

where $\widehat{\gamma}[t]$ is the amount of workload decreasing in time slot t, and is determined by the Maximum-Channel-Rate-First Scheduling policy as shown in the JOWS Algorithm. Let $\widehat{\mathcal{N}}[t]$ be the set of users awaiting for wireless transmissions in the associated system. If $W[0] = \widehat{W}[0] = 0$, then it is easy to show that $W[t] \leq \widehat{W}[t], \forall t \geq 0$. Indeed, if we couple channel fading of same users and arrival processes in both systems, then under the Maximum-Channel-Rate-First Scheduling policy, we have $\mathcal{N}[t] \subseteq \widehat{\mathcal{N}}[t]$ and $W[t] \leq \widehat{W}[t]$. Therefore, we have

$$\Pr\{W[t] \geq \overline{W}\} \leq \Pr\{\widehat{W}[t] \geq \overline{W}\}, \forall t \geq 0. \tag{19}$$

Next, we focus on upper-bounding $\Pr\{\widehat{W}[t] \geq \overline{W}\}$. We will show that

$$\mathbb{E}\left[\widehat{W}[t+1] - \widehat{W}[t] \middle| \widehat{W}[t]\right] \leq -\frac{1}{4}\zeta, \text{ if } \widehat{W}[t] \geq \frac{4H}{\zeta}, \tag{20}$$

where $H \triangleq \nu_{\max}^2 + \frac{w_{\max}}{-\log(1-p_{\max})} e^{-\log^2(1-p_{\max})}$. This together with the fact that $|\widehat{W}[t+1] - \widehat{W}[t]| \leq \nu_{\max} \triangleq A^{\max}\lceil F^{\max}/C^{\max}\rceil$ implies that, accordingly to [18, Lemma 2.2], there exists a $\eta_1 > 0$ and $r(\eta, \zeta) \triangleq e^{\eta\nu_{\max}} - \eta(\nu_{\max} + \zeta/4) \in (0,1), \forall 0 < \eta < \eta_1$, such that

$$\mathbb{E}\left[e^{\eta(\widehat{W}[t+1] - \widehat{W}[t])}; \widehat{W}[t] \geq \frac{4H}{\zeta} \middle| \widehat{W}[t]\right] \leq r(\eta). \tag{21}$$

Therefore, for $0 < \eta < \eta_1$, we have

$$\begin{aligned}
&\mathbb{E}\left[e^{\eta\widehat{W}[t+1]} \middle| \widehat{W}[t]\right] \\
=&\mathbb{E}\left[e^{\eta\widehat{W}[t+1]}; \widehat{W}[t] \geq \frac{4H}{\zeta} \middle| \widehat{W}[t]\right] \\
&+ \mathbb{E}\left[e^{\eta\widehat{W}[t+1]}; \widehat{W}[t] < \frac{4H}{\zeta} \middle| \widehat{W}[t]\right] \\
\leq& r(\eta)\mathbb{E}\left[e^{\eta\widehat{W}[t]}; \widehat{W}[t] \geq \frac{4H}{\zeta} \middle| \widehat{W}[t]\right] + e^{\eta(\nu_{\max} + 4H/\zeta)} \\
\leq& r(\eta)e^{\eta\widehat{W}[t]} + e^{\eta(\nu_{\max} + 4H/\zeta)},
\end{aligned} \tag{22}$$

where the second last step uses inequality (21) and the fact that $\widehat{W}[t+1] \leq \widehat{W}[t] + \nu_{\max}$. This implies

$$\mathbb{E}\left[e^{\eta\widehat{W}[t+1]}\right] \leq r(\eta)\mathbb{E}\left[e^{\eta\widehat{W}[t]}\right] + e^{\eta(\nu_{\max} + 4H/\zeta)}. \tag{23}$$

By using (23) and iterating over t, we have

$$\mathbb{E}\left[e^{\eta\widehat{W}[t]}\right] \leq (r(\eta))^t \mathbb{E}\left[e^{\eta\widehat{W}[0]}\right] + \frac{1 - (r(\eta))^t}{1 - r(\eta)} e^{\eta(\nu_{\max} + 4H/\zeta)}.$$

Hence, in steady state, we have

$$\mathbb{E}\left[e^{\eta\widehat{W}}\right] \leq \frac{1}{1 - r(\eta)} e^{\eta(\nu_{\max} + 4H/\zeta)}, \tag{24}$$

where we use $\widehat{W}$ to denote the steady-state random variable of $\{\widehat{W}[t]\}_{t\geq 0}$. Let W denote the steady-state random variable of $\{W[t]\}_{t\geq 0}$. Then, according to (19), we have

$$\begin{aligned}
\Pr\{W \geq \overline{W}\} \leq& \Pr\{\widehat{W} \geq \overline{W}\} \\
=& \Pr\{e^{\eta\widehat{W}} \geq e^{\eta\overline{W}}\} \\
\overset{(a)}{\leq}& e^{-\eta\overline{W}}\mathbb{E}\left[e^{\eta\widehat{W}}\right] \leq M(\eta)e^{-\eta\overline{W}},
\end{aligned} \tag{25}$$

where step (a) uses Markov's Inequality; (b) uses (24) and is true for $M(\eta) \triangleq e^{\eta(\nu_{\max} + 4H/\zeta)}/(1 - r(\eta))$.

Next, we show that if $\eta = \frac{1}{2\nu_{\max}} \log(1 + \zeta/(4\nu_{\max}))$, then

$$r(\eta) \triangleq e^{\eta\nu_{\max}} - \eta(\nu_{\max} + \zeta/4) \in (0,1). \tag{26}$$

To see it, let's consider the derivative of $r(\eta)$.

$$r'(\eta) = \nu_{\max} e^{\eta \nu_{\max}} - \left(\nu_{\max} + \frac{\zeta}{4}\right). \tag{27}$$

We note that $r'(0) = -\zeta/4 < 0$. By setting $r'(\overline{\eta})$ equal to 0, we have

$$\overline{\eta} = \frac{1}{\nu_{\max}} \log\left(1 + \frac{\zeta}{4\nu_{\max}}\right). \tag{28}$$

Therefore, $r'(\eta) < 0, \forall 0 < \eta < \overline{\eta}$ and hence $r(\eta)$ decreases in the interval $(0, \overline{\eta})$. We note that

$$\begin{aligned} r(\overline{\eta}) =& e^{\overline{\eta}\nu_{\max}} - \overline{\eta}\left(\nu_{\max} + \frac{\zeta}{4}\right) \\ =& 1 + \frac{\zeta}{4\nu_{\max}} - \left(1 + \frac{\zeta}{4\nu_{\max}}\right)\log\left(1 + \frac{\zeta}{4\nu_{\max}}\right). \end{aligned} \tag{29}$$

Note that $\zeta/(4\nu_{\max}) < 1$ and thus $r(\overline{\eta}) > 0$. In addition, we note that $r(0) = 1$. Hence, by setting $\eta = \overline{\eta}/2$, we have $r(\eta) \in (0, 1)$.

Finally, we prove (20) to complete the proof. To that end, we select Lyapunov function $V_2[t] \triangleq \widehat{W}[t]$. Consider its expected drift $\Delta V_2[t] = \mathbb{E}\left[V_2[t+1] - V_2[t] | W[t]\right]$.

$$\begin{aligned} &\Delta V_2[t] \triangleq \mathbb{E}\left[\widehat{W}[t+1] - \widehat{W}[t] \middle| \widehat{W}[t]\right] \\ =& \mathbb{E}\left[\sqrt{\widehat{W}^2[t+1]} - \sqrt{\widehat{W}^2[t]} \middle| \widehat{W}[t]\right] \\ \overset{(a)}{\leq}& \frac{1}{2\widehat{W}[t]} \mathbb{E}\left[\widehat{W}^2[t+1] - \widehat{W}^2[t] \middle| \widehat{W}[t]\right] \\ \overset{(b)}{=}& \frac{1}{2\widehat{W}[t]} \mathbb{E}\left[(\widehat{W}[t] + \widehat{\nu}[t] - \widehat{\gamma}[t])^2 - \widehat{W}^2[t] \middle| \widehat{W}[t]\right] \\ \overset{(c)}{\leq}& \frac{\mathbb{E}\left[\widehat{W}[t](\widehat{\nu}[t] - \widehat{\gamma}[t]) \middle| \widehat{W}[t]\right] + H_1}{\widehat{W}[t]} \\ \overset{(d)}{\leq}& \frac{\theta\widehat{W}[t] - \mathbb{E}\left[\widehat{W}[t](1 - (1 - p_{\max})^{|\widehat{\mathcal{N}}[t]|}) \middle| \widehat{W}[t]\right] + H_1}{W[t]} \\ \overset{(e)}{\leq}& \frac{\theta\widehat{W}[t] - \mathbb{E}\left[\widehat{W}[t](1 - (1 - p_{\max})^{\widehat{W}[t]/w_{\max}}) \middle| \widehat{W}[t]\right] + H_1}{\widehat{W}[t]} \\ \overset{(f)}{\leq}& \frac{-\frac{1}{2}\zeta\widehat{W}[t] + \widehat{W}[t](1 - p_{\max})^{\widehat{W}[t]/w_{\max}} + H_1}{\widehat{W}[t]} \\ \overset{(d)}{\leq}& -\frac{1}{2}\zeta + \frac{H}{\widehat{W}[t]}, \end{aligned} \tag{30}$$

where step (a) uses the fact that $f(x) = \sqrt{x}$ for $x \geq 0$ so that $f(y) - f(x) \leq f'(x)(y-x) = (y-x)/(2\sqrt{x})$ with $y = \widehat{W}^2[t+1]$ and $x = \widehat{W}^2[t]$; (b) uses the dynamics of $\widehat{W}[t]$, i.e., (18); (c) is true for $H_1 \triangleq v_{\max}^2$; (d) follows from the fact that $\mathbb{E}[\widehat{\nu}[t]] = \theta$ and Maximum-Channel-Rate-First scheduling policy; (e) uses the fact that $|\widehat{\mathcal{N}}[t]| \geq \widehat{W}[t]/w_{\max}$ and we recall that $w_{\max} \triangleq \lceil F^{\max}/C^{\max} \rceil$; (f) uses $\theta = 1 - \zeta/2$; (g) uses the fact that $\lim_{x\to\infty} x(1-p_{\max})^{x/w_{\max}} = 0$ and thus $x(1-p_{\max})^{x/w_{\max}} \leq H_2 \triangleq \frac{w_{\max}}{-\log(1-p_{\max})} e^{-\log^2(1-p_{\max})}$ and is true for $H \triangleq H_1 + H_2$.

Thus, (20) directly follows from (30).

ACKNOWLEDGMENTS

This work has been supported in part by NSF grants CNS-1717108 and CNS-1815563.

REFERENCES

[1] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1628 – 1656, 2017.

[2] M. Molina Pena, O. Muñoz Medina, A. Pascual Iserte, and J. Vidal Manzano, "Joint scheduling of communication and computation resources in multiuser wireless application offloading," in *Proceedings PIMRC 2014*. Institute of Electrical and Electronics Engineers (IEEE), 2014, pp. 1093–1098.

[3] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3590–3605, 2016.

[4] T. Q. Dinh, J. Tang, Q. D. La, and T. Q. Quek, "Offloading in mobile edge computing: Task allocation and computational frequency scaling," *IEEE Transactions on Communications*, vol. 65, no. 8, pp. 3571–3584, 2017.

[5] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Transactions on Networking*, no. 5, pp. 2795–2808, 2016.

[6] H. Cao and J. Cai, "Distributed multiuser computation offloading for cloudlet-based mobile cloud computing: A game-theoretic machine learning approach," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 1, pp. 752–764, 2018.

[7] K. Zhang, Y. Mao, S. Leng, Q. Zhao, L. Li, X. Peng, L. Pan, S. Maharjan, and Y. Zhang, "Energy-efficient offloading for mobile edge computing in 5g heterogeneous networks," *IEEE access*, vol. 4, pp. 5896–5907, 2016.

[8] C. Wang, F. R. Yu, C. Liang, Q. Chen, and L. Tang, "Joint computation offloading and interference management in wireless cellular networks with mobile edge computing," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 8, pp. 7432–7445, 2017.

[9] M.-H. Chen, B. Liang, and M. Dong, "Joint offloading and resource allocation for computation and communication in mobile cloud with computing access point," in *INFOCOM 2017-IEEE Conference on Computer Communications, IEEE*. IEEE, 2017, pp. 1–9.

[10] P. van de Ven, S. Borst, and S. Shneer, "Instability of maxweight scheduling algorithms," in *INFOCOM 2009, IEEE*. IEEE, 2009, pp. 1701–1709.

[11] S. Liu, L. Ying, and R. Srikant, "Throughput-optimal opportunistic scheduling in the presence of flow-level dynamics," *IEEE/ACM Transactions on Networking*, vol. 19, no. 4, pp. 1057–1070, 2011.

[12] B. Sadiq and G. De Veciana, "Throughput optimality of delay-driven maxweight scheduler for a wireless system with flow dynamics," in *Communication, Control, and Computing, 2009. Allerton 2009. 47th Annual Allerton Conference on*. IEEE, 2009, pp. 1097–1102.

[13] B. Li, X. Kong, and L. Wang, "Optimal load-balancing for high-density wireless networks with flow-level dynamics," in *Proceedings of the Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*. ACM, 2018, pp. 316–317.

[14] Y. Kang, J. Hauswald, C. Gao, A. Rovinski, T. Mudge, J. Mars, and L. Tang, "Neurosurgeon: Collaborative intelligence between the cloud and mobile edge," *ACM SIGPLAN Notices*, vol. 52, no. 4, pp. 615–629, 2017.

[15] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," *Synthesis Lectures on Communication Networks*, vol. 3, no. 1, pp. 1–211, 2010.

[16] B. Tan and R. Srikant, "Online advertisement, optimization and stochastic networks," *IEEE Transactions on Automatic Control*, vol. 57, no. 11, pp. 2854–2868, 2012.

[17] M. J. Neely, "Intelligent packet dropping for optimal energy-delay tradeoffs in wireless downlinks," *IEEE Transactions on Automatic Control*, vol. 54, no. 3, pp. 565–579, 2009.

[18] B. Hajek, "Hitting-time and occupation-time bounds implied by drift analysis with applications," *Advances in Applied probability*, vol. 14, no. 3, pp. 502–525, 1982.