Optimal Joint Offloading and Wireless Scheduling for Parallel Computing with Deadlines

Xudong Qin* Weijian Xu† Bin Li*
*Dept. of Electrical, Computer and Biomedical Engineering, University of Rhode Island, Rhode Island, USA
†Information Engineering College, Jimei University, Xiamen, China

***Abstract*—In this paper, we consider the problem of joint offloading and wireless scheduling design for parallel computing applications with hard deadlines. This is motivated by the rapid growth of compute-intensive mobile parallel computing applications (e.g., real-time video analysis, language translation) that require to be processed within a hard deadline. While there are many works on joint computing and communication algorithm design, most of them focused on the minimization of average computing time and may not be applicable for mobile applications with hard deadlines. In this work, we explicitly take hard deadlines for computing tasks into account and develop a joint offloading and scheduling algorithm based on the stochastic network optimization framework. The proposed algorithm is shown to achieve average energy consumption arbitrarily close to the optimal one. However, this algorithm involves a strong coupling between offloading and scheduling decisions, which yields significant challenges on its implementation. Towards this end, we first successfully decouple the offloading and scheduling decisions in the case with one time slot deadline by exploring the intrinsic structure of the proposed algorithm. Based on this, we further implement the proposed algorithm in the general setups. Simulations are provided to corroborate our findings.**

I. Introduction

With the rapid growth of Artificial Intelligence (AI) technology, there is a strong need for real-time computation for mobile applications, such as video monitoring, real-time video analysis, and real-time language translation. For example, many users may use real-time language translation services in international events such as World's Fair, where each user expects to experience zero latency and high quality of language translation. Indeed, in these real-time applications, computing tasks become outdated if they are not processed in time. Different from traditional multimedia applications, these real-time mobile applications not only demand for high throughput and low energy consumption but also involve intensive computations.

On one hand, each mobile user has limited computing and battery capacity and thus processes computing tasks slowly, while causing large energy consumption. On the other hand, users can upload computing tasks to edge servers that have powerful computing units and can process computing tasks in a much faster way without any limitation on energy consumption. However, if all users upload their computing tasks to edge servers via wireless works, it will not only cause large transmission delay and thus drop a large amount of computing tasks, but also results in large energy consumption due to significant amount of wireless transmissions. Therefore, in order to meet the desired throughput and energy consumption requirements, each user requires a careful *offloading decision* on the amount of computing traffic that is processed by itself and waits for wireless transmissions to edge servers, respectively, and each AP needs to make a *scheduling decision* that determines which users are allowed for wireless transmissions in each time slot.

While there are many works on joint computation and communication algorithm design, most of them (see [1], [2], [3], [4], [5], [6], [7], [8] and [9] for a thorough survey) focused on the minimization of average computing time and hence are not applicable for mobile applications with hard deadlines. Despite some recent work (e.g., [10], [11]) focused on offloading design for data traffic with hard deadlines, they did not consider scheduling decisions for wireless communications. On the other hand, many efficient wireless scheduling algorithms (e.g., [12], [13], [14], [15]) have already been developed for mobile applications with hard deadlines, however, they did not take offloading decisions into account. Therefore, the proposed solutions in existing work on either joint offloading and communication design or wireless scheduling design with hard deadlines do not apply and new joint offloading and scheduling algorithm designs are required in the presence of mobile compute-intensive applications with hard deadlines.

To that end, we consider the optimal joint offloading and wireless scheduling design for mobile applications with hard deadlines. The main results and contributions of this paper are listed as follows:

- In Section II, we formulate the problem of joint offloading and wireless scheduling design with hard deadline constraints with the goal of minimizing average energy consumption while meeting desired drop rate requirements. We use a simple example to illustrate the need of such a design.
- In Section III, we develop a joint offloading and scheduling (JOS) algorithm based on the stochastic network optimization framework, and show that it yields average energy consumption arbitrarily close to the optimal one. However, the proposed JOS algorithm involves a strong coupling between offloading and scheduling decisions, which yields significant challenges on its implementation.
- In Section IV, we decouple the offloading and scheduling decisions of the proposed JOS algorithm by exploring its

Xudong Qin and Weijian Xu contributed equally to this work, where Weijian completed this work during his visit to the University of Rhode Island.

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

intrinsic structure in the case with one time slot constraint.

• In Section V, based on the insight of implementing the proposed JOS algorithm in the case with one time slot deadline constraint, we successfully decouple the offloading and scheduling decisions of the proposed JOS algorithm in general setups.

II. System Model

We consider a wireless system consisting of N mobile users and one access point (AP), where the AP is directly connected to some powerful servers (referred to as *edge servers*). We consider a time-slotted system. We assume that each user has dynamic and heterogeneous computing demands with strict deadlines of T time slots, where they will be dropped if they are not processed within T time slots. To that end, we group a set of T consecutive time slots into a *frame*.

We note that each mobile user has limited computing capability and thus can only process a portion of computing traffic with a relatively slow speed, while edge servers usually have high-performance computing units and can process a large amount of computing tasks in a much faster way. Due to the wireless interference, only a subset of users is allowed for data transmission in each time slot. Therefore, if all users upload their computing workloads to edge servers via wireless networks, it causes a large network delay and hence significantly increases the total latency of completing computing tasks for each user. As such, a careful *offloading decision* is needed for each user on the amount of computing traffic that is processed by itself and waits for wireless transmissions to edge servers, respectively, and each AP needs to make a *scheduling decision* that determines which users are allowed for wireless transmissions in each time slot.

To facilitate our mathematical modeling and algorithm developments, we unify the units for the processing speed of each user and wireless transmission rate, and ignore the computation time in powerful edge servers. To that end, we use a "*packet*" to denote the minimum amount of computation and wireless communication units. We assume that all computing tasks arrive at the beginning of each time frame. We use $A_n[kT]$ to denote the number of packets arriving at user n at the beginning of time frame k that is independently distributed over users and independently and identically distributed (i.i.d.) over time with mean $\lambda_n > 0$ and $A_n[kT] \leq A_{\max}$ for some $A_{\max} < \infty$. We assume that each user n has a maximum allowable drop rate $\rho_n\lambda_n$, where $\rho_n \in (0,1)$ denotes the maximum fraction of computing demand that can be dropped by user n. For example, $\rho_n = 0.05$ means that user n can drop at most 5% of its computing demand on average.

We assume that each user n can process μ_n packets in each time slot. Due to the wireless channel fading, the transmission rate of each user may change over time. We assume that each user knows its channel rate at the beginning of each time frame that keeps constant over the entire frame. As such, we use $C_n[kT]$ to denote the maximum number of packets that can be transmitted in each time slot within frame k if user n is scheduled for wireless transmission. We assume that $\mathbf{C}[kT] \triangleq (C_n[kT])_{n=1}^N$ are i.i.d. over frames with $C_n[kT] \leq C_{\max}, \forall n, k \geq 0$, for some $C_{\max} < \infty$. Due to the wireless interference, we assume that at most one user is allowed to transmit data in each time slot. Let $S_n[t] = 1$ if user n is scheduled for wireless transmission in time slot t, and $S_n[t] = 0$ otherwise. We use $\mathbf{S}[t] \triangleq (S_n[t])_{n=1}^N$ to denote a *feasible schedule*, where at most one element is equal to one. We use $\mathcal{S}$ to denote the collection of all feasible schedules.

In this paper, we focus on mobile parallel applications with hard deadlines, where each application can be partitioned into two different parts that can be executed simultaneously by users' devices and edge servers (via wireless transmissions). In particular, an *offloading decision* is required for each mobile user that determines the amount of incoming computation that is processed by itself or is uploaded to edge servers via wireless transmission, which is determined by the AP (also referred to as *scheduling decision*). Towards this end, let $A_n^{(L)}[kT]$ and $A_n^{(E)}[kT]$ denote the number of packets that are going to be processed by mobile user n and are waiting for wireless transmissions to edge servers in time frame k, respectively. Let $\mathbf{A}^{(L)}[kT] \triangleq (A_n^{(L)}[kT])_{n=1}^N$ and $\mathbf{A}^{(E)}[kT] \triangleq (A_n^{(E)}[kT])_{n=1}^N$. We use $e_n^{(L)}$ and $e_n^{(E)}$ to denote the energy consumption for local computation and wireless transmission of user n in one time slot, respectively. Let $P_n[k:(k+1)]$ denote the total energy consumption of user n within the frame k, which is equal to $e_n^{(L)} \min\left\{\left\lceil A_n^{(L)}[kT]/\mu_n \right\rceil, T\right\} + e_n^{(E)} \sum_{t=kT}^{(k+1)T-1} S_n[t]$.

In this paper, we are interested in developing an *optimal joint offloading and scheduling algorithm* that minimizes the total energy consumption while meeting the desired drop rate requirements. To illustrate the need for such a design, we consider the power consumption of two simple offloading and scheduling algorithms that meet desired drop rate requirements, Local-First Offloading and Scheduling (LFOS) Algorithm and Edge-First Offloading and Scheduling (EFOS) Algorithm. Under the LFOS Algorithm, at the beginning of each time slot, each user n allocates μ_n packets for local computation if it has sufficient number of packets. Otherwise, each user processes all packets. If there are still packets left, then each user competes for wireless transmission via MaxWeight-type policy (e.g., [14]). Similarly, under the EFOS Algorithm, each user contends for wireless transmission first and then performs local computation in each time slot. Table I illustrates energy consumption in the presence of $N = 2$ users with deadline of $T = 6$ time slots. Each user can process 1 packet in each time slot at the energy expenditure 7 watt, while it can transmit 2 packets if scheduled in each time slot with the energy consumption 4 watt. We assume that both users have 6 packets at time 0. Then, from Table I, we can easily calculate that each user consumes 4.5 watt and 4.25 watt on average under LFOS and EFOS, respectively. However, a better choice will be to let user 1 transmit data in the first three slots and user 2 in the rest of three slots, under which each user only consumes 2 watt on average and saves 55.56% energy compared with LFOS Algorithm.

This simple example indicates the possibility of significant

Policy	User	$t=0$	1	2	3	4	5
LFOS	1	3, 11	0, 11				
	2	5, 7	4, 7	1, 11	0, 7		
EFOS	1	3, 11	0, 11				
	2	5, 7	4, 7	1, 11	0, 4		
Better choice	1	4, 4	2, 4	0, 4			
	2				4, 4	2, 4	0, 4

TABLE I: Average energy consumption under different policies: the first and second number in each cell denote the number of remaining packets and energy expenditure, respectively.

energy saving through a careful joint offloading and scheduling design. To that end, in this paper, we want to determine offloading decisions $(A_n^{(L)}[kT], A_n^{(E)}[kT])_{n=1}^N$ and scheduling decisions $(\mathbf{S}[t])_{t=kT}^{(k+1)T-1}$ within each frame k that solve the following optimization problem:

$$\min \quad \limsup_{k\to\infty} \frac{1}{K} \sum_{k=0}^{K-1} \sum_{n=1}^{N} \mathbb{E}[P_n[k:(k+1)]] \tag{1}$$

$$\text{s.t.} \quad \lambda_n(1-\rho_n) \le \nu_n, \quad \forall n,k, \tag{2}$$

$$A_n^{(L)}[kT] + A_n^{(E)}[kT] = A_n[kT], \quad \forall n,k, \tag{3}$$

where $\nu_n \triangleq \mathbb{E}\Bigg[\min\left\{A_n^{(L)}[kT], T\mu_n\right\} + \min\Bigg\{A_n^{(E)}[kT], C_n[kT]\sum_{t=kT}^{(k+1)T-1} S_n[t]\Bigg\}\Bigg]$ denotes the average total number of packets that can be processed within frame k. Here, the objective (1) is to minimize the average energy consumption and the constraint (2) means that each user should meet its own drop rate requirement.

Next, we develop a joint offloading and scheduling algorithm based on the stochastic network optimization framework (e.g., [16]), which involves a strong coupling between offloading and scheduling decisions.

III. Joint Offloading and Scheduling Design

In this section, we develop a joint offloading and scheduling algorithm that achieves arbitrarily close to the optimal solution to the Problem (1)-(3) (optimal energy consumption) based on the stochastic network optimization framework. To that end, we introduce a virtual queue for each user to keep track of amount of violations of its own drop rate requirement over time. In particular, let $D_n[kT]$ be the number of packets dropped by user n at the end of time frame k, which is expressed as follows:

$$D_n[kT] = \Bigg(A_n[kT] - \min\left\{A_n^{(L)}[kT], \mu_n T\right\} - \min\Bigg\{A_n^{(E)}[kT], C_n[kT]\sum_{t=kT}^{(k+1)T-1} S_n[t]\Bigg\}\Bigg)^+, \tag{4}$$

where $(x)^+ \triangleq \max\{x,0\}$ for any real number x. We generate virtual service $B_n[kT]$ for virtual queue n at the end of time frame k that is independently across users and i.i.d. over time with mean $\rho_n\lambda_n$ and $\mathbb{E}\left[B_n^2[kT]\right] < \infty$. Then, the evolution of virtual queue n can be described as follow:

$$X_n[(k+1)T] = \left(X_n[kT] + D_n[kT] - B_n[kT]\right)^+. \tag{5}$$

We call virtual queue n *rate stable* if $\lim_{k\to\infty} X[kT]/k = 0$. According to [16, Definition 2.2], the average drop rate of user n meets the requirement if its virtual queue is rate stable. By using the stochastic network optimization framework, we develop the following joint offloading and scheduling algorithm.

Joint Offloading and Scheduling (JOS) Algorithm: At the beginning of each frame k, given $\mathbf{X}[kT] \triangleq (X_n[kT])_{n=1}^N$, $\mathbf{A}[kT]$, and $\mathbf{C}[kT]$, find $(\widehat{\mathbf{A}}^{(L)}[kT], \widehat{\mathbf{A}}^{(E)}[kT])$ and $(\widehat{\mathbf{S}}[t])_{t=kT}^{(k+1)T-1}$ that solve the following optimization problem:

$$\max \quad \sum_{n=1}^{N} F_n^{(L)}[kT] + \sum_{n=1}^{N} F_n^{(E)}[kT], \tag{6}$$

where

$$F_n^{(L)}[kT] \triangleq X_n[kT]\min\left\{A_n^{(L)}[kT], T\mu_n\right\} - Me_n^{(L)}\min\left\{\left\lceil\frac{A_n^{(L)}[kT]}{\mu_n}\right\rceil, T\right\}, \tag{7}$$

$$F_n^{(E)}[kT] \triangleq X_n[kT]\min\left\{A_n^{(E)}[kT], C_n[kT]\sum_{t=kT}^{(k+1)T-1} S_n[t]\right\} - Me_n^{(E)}\sum_{t=kT}^{(k+1)T-1} S_n[t], \tag{8}$$

$A_n^{(L)}[kT] + A_n^{(E)}[kT] = A_n[kT], \forall n,k$, and $M > 0$ is some parameter.

In the proposed JOS Algorithm, it requires to solve optimization problem (6) to obtain offloading decisions $(\widehat{\mathbf{A}}^{(L)}[kT], \widehat{\mathbf{A}}^{(E)}[kT])$ and scheduling decisions $(\widehat{\mathbf{S}}[t])_{t=kT}^{(k+1)T-1}$ for the entire frame k. Next, we show that the proposed JOS Algorithm yields average energy consumption arbitrarily close to the optimal one.

Proposition 1: The JOS Algorithm with any $M > 0$ achieves $O(1/M)$ close to the optimal energy consumption at the expense of the mean virtual queue-length growing with $O(M)$, i.e.,

$$\limsup_{K\to\infty} \frac{1}{K} \sum_{k=0}^{K-1} \sum_{n=1}^{N} \mathbb{E}[X_n[kT]] \le \frac{MP_{\max} + H}{\epsilon} \tag{9}$$

$$\limsup_{K\to\infty} \frac{1}{K} \sum_{k=0}^{K-1} \sum_{n=1}^{N} \mathbb{E}\left[\widehat{P}_n[k:(k+1)]\right] \le P_{\min} + \frac{H}{M}, \tag{10}$$

where $H \triangleq \sum_{n=1}^{N} \mathbb{E}[A_n^2[kT] + B_n^2[kT]]$, $\widehat{P}_n[k:(k+1)]$ is the energy consumption in frame k under the JOS Algorithm, $P_{\max}$ is the maximum energy consumption within one time frame which is bounded due to the boundedness of arrivals,

$P_{\min}$ is the optimal solution to Problem (1)-(3), and ϵ is some positive parameter.

Proof: Select the Lyapunov function $V[kT] \triangleq \frac{1}{2}\sum_{n=1}^{N} X_n^2[kT]$ and consider its conditional expected drift:

$$\begin{aligned}
&\Delta V[kT] \triangleq \mathbb{E}\left[V[(k+1)T] - V[kT] | \mathbf{X}[kT]\right] \\
&\overset{(a)}{\leq} \frac{1}{2}\sum_{n=1}^{N} \mathbb{E}\left[(X_n[kT] + \widehat{D}_n[kT] - B_n[kT])^2 - X_n[kT]^2 \Big| \mathbf{X}[kT]\right] \\
&\overset{(b)}{\leq} \sum_{n=1}^{N} \mathbb{E}\left[X_n[kT](A_n[kT] - B_n[kT]) | \mathbf{X}[kT]\right] \\
&- \sum_{n=1}^{N} \mathbb{E}\left[X_n[kT] \min\left\{\widehat{A}_n^{(E)}[kT], C_n[kT] \sum_{t=kT}^{(k+1)T-1} \widehat{S}_n[t]\right\} \Bigg| \mathbf{X}[kT]\right] \\
&- \sum_{n=1}^{N} \mathbb{E}\left[X_n[kT] \min\left\{\widehat{A}_n^{(L)}[kT], T\mu_n\right\} \Big| \mathbf{X}[kT]\right] + H, \qquad (11)
\end{aligned}$$

where the step (a) follows from the fact that $(\max\{x,0\})^2 \leq x^2, \forall x$, and $\widehat{D}_n[kT]$ denote the number of packets dropped at the end of frame k under the JOS Algorithm; (b) is true for $H \triangleq \sum_{n=1}^{N} \mathbb{E}[A_n^2[kT] + B_n^2[kT]] < \infty$ and uses the definition of $\widehat{D}_n[kT]$ (cf. (4)).

Adding the term $M\sum_{n=1}^{N} \mathbb{E}\left[\widehat{P}_n[k:(k+1)] \Big| \mathbf{X}[kT]\right]$ on both sides of (11), we have

$$\begin{aligned}
&\Delta V[kT] + M\sum_{n=1}^{N} \mathbb{E}\left[\widehat{P}_n[k:(k+1)] \Big| \mathbf{X}[kT]\right] \\
&\overset{(a)}{\leq} \sum_{n=1}^{N} \lambda_n(1-\rho_n) X_n[kT] \\
&- \sum_{n=1}^{N} \Bigg[X_n[kT] \min\left\{\widehat{A}_n^{(L)}[kT], T\mu_n\right\} \\
&\qquad - Me_n^{(L)} \min\left\{\left\lceil \frac{\widehat{A}_n^{(L)}[kT]}{\mu_n} \right\rceil, T\right\} \Bigg| \mathbf{X}[kT]\Bigg] \\
&- \sum_{n=1}^{N} \mathbb{E}\left[X_n[kT] \min\left\{\widehat{A}_n^{(E)}[kT], C_n[kT] \sum_{t=kT}^{(k+1)T-1} \widehat{S}_n[t]\right\} \right. \\
&\qquad \left. -Me_n^{(E)} \sum_{t=kT}^{(k+1)T-1} \widehat{S}_n[t] \Bigg| \mathbf{X}[kT]\right] + H \qquad (12) \\
&\overset{(b)}{\leq} \sum_{n=1}^{N} \lambda_n(1-\rho_n) X_n[kT] \\
&- \sum_{n=1}^{N} \mathbb{E}\left[X_n[kT] \min\left\{\widetilde{A}_n^{(L)}[kT], T\mu_n\right\} \right. \\
&-Me_n^{(L)} \min\left\{\left\lceil \frac{\widetilde{A}_n^{(L)}[kT]}{\mu_n} \right\rceil, T\right\} \Bigg| \mathbf{X}[kT]\Bigg] \qquad (13) \\
&- \sum_{n=1}^{N} \mathbb{E}\left[X_n[kT] \min\left\{\widetilde{A}_n^{(E)}[kT], C_n[kT] \sum_{t=kT}^{(k+1)T-1} \widetilde{S}_n[t]\right\} \right. \\
&\left. -Me_n^{(E)} \sum_{t=kT}^{(k+1)T-1} \widetilde{S}_n[t] \Bigg| \mathbf{X}[kT]\right] + H, \qquad (14)
\end{aligned}$$

where step (a) uses the fact that $A_n[kT]$ and $B_n[kT]$ are i.i.d. with mean λ_n and $\rho_n\lambda_n$, respectively, and the definition of $\widehat{P}_n[k:(k+1)]$; (b) follows from the fact that the JOS Algorithm minimizes the drift of (12) and is true for the stationary randomized policy $(\widetilde{\mathbf{A}}[kT], \widetilde{\mathbf{S}}[kT])$ that satisfies

$$\begin{aligned}
&\lambda_n(1-\rho_n) + \epsilon \leq \mathbb{E}\left[\min\left\{\widetilde{A}_n^{(L)}[kT], T\mu_n\right\}\right] \\
&\qquad + \mathbb{E}\left[\min\left\{\widetilde{A}_n^{(E)}[kT], C_n[kT] \sum_{t=kT}^{(k+1)T-1} \widetilde{S}_n[t]\right\}\right], \forall n \\
&\text{and } \sum_{n=1}^{N} \mathbb{E}[\widetilde{P}_n[k:(k+1)]] = P_{av}(\epsilon),
\end{aligned}$$

where $\widetilde{P}_n[k:(k+1)] \triangleq e_n^{(L)} \min\left\{\left\lceil \frac{\widetilde{A}_n^{(L)}[kT]}{\mu_n} \right\rceil, T\right\} + e_n^{(E)} \sum_{t=kT}^{(k+1)T-1} \widetilde{S}_n[t]$ and $\lim_{\epsilon\downarrow 0} P_{av}(\epsilon) = P_{\min}$. The existence of such a randomized policy can be shown by using similar arguments in [17, Theorem 1] and its proof is omitted for brevity. Therefore, we have

$$\begin{aligned}
&\Delta V[kT] + M\sum_{n=1}^{N} \mathbb{E}\left[\widehat{P}_n[k:(k+1)] \Big| \mathbf{X}[kT]\right] \\
&\leq -\epsilon \sum_{n=1}^{N} X_n[kT] + MP_{av}(\epsilon) + H. \qquad (15)
\end{aligned}$$

Summing (15) over $k = 0, 1, 2, \ldots, K-1$ and taking expectation on both sides, we have

$$\begin{aligned}
&\mathbb{E}\left[V[KT] - V[0]\right] + M\sum_{k=0}^{K-1}\sum_{n=1}^{N} \mathbb{E}\left[\widehat{P}_n[k:(k+1)]\right] \\
&\leq -\epsilon \sum_{k=0}^{K-1}\sum_{n=1}^{N} \mathbb{E}[X_n[kT]] + MKP_{av}(\epsilon) + HK \qquad (16)
\end{aligned}$$

Therefore, we have

$$\epsilon \sum_{k=0}^{K-1}\sum_{n=1}^{N} \mathbb{E}[X_n[kT]] \leq MKP_{av}(\epsilon) + HK + \mathbb{E}[V[0]] \qquad (17)$$

$$M\sum_{k=0}^{K-1}\sum_{n=1}^{N} \mathbb{E}\left[\widehat{P}_n[k:(k+1)]\right] \leq MKP_{av}(\epsilon) + HK + \mathbb{E}[V[0]]. \qquad (18)$$

By using the fact that $P_{av}(\epsilon) \leq P_{\max}$, dividing $K\epsilon$ on both sides of (17) and taking the limit, we have (9).

Since (18) holds for any $\epsilon > 0$, we have

$$M\sum_{k=0}^{K-1}\sum_{n=1}^{N} \mathbb{E}\left[\widehat{P}_n[k:(k+1)]\right] \leq MKP_{\min} + HK + \mathbb{E}[V[0]].$$

By dividing MK on both sides of the above inequality and taking the limit, we have (10). ■

Even though the JOS Algorithm can achieve optimal energy consumption asymptotically, it requires to solve the

	$W^{(E)}_{n^*[t]}[t] \geq 0$	$W^{(E)}_{n^*[t]}[t] < 0$
$W^{(L)}_{n^*[t]}[t] \geq 0$	$\overline{A}^{(E)}_{n^*[t]}[t] = \min\left\{A_{n^*[t]}[t], C_{n^*[t]}[t]\right\}$ $\overline{A}^{(L)}_{n^*[t]}[t] = A_{n^*[t]}[t] - A^{(E)}_{n^*[t]}[t]$	$\overline{A}^{(E)}_{n^*[t]}[t] = 0$ $\overline{A}^{(L)}_{n^*[t]}[t] = A_{n^*[t]}[t]$
$W^{(L)}_{n^*[t]}[t] < 0$	$\overline{A}^{(E)}_{n^*[t]}[t] = A_{n^*[t]}[t]$ $\overline{A}^{(L)}_{n^*[t]}[t] = 0$	$\overline{A}^{(E)}_{n^*[t]}[t] = A_{n^*[t]}[t]$ $\overline{A}^{(L)}_{n^*[t]}[t] = 0$

TABLE II: Offloading decisions in one time slot deadline case.

optimization problem (6) at the beginning of each time frame, which involves the strong coupling between offloading decisions $(\widehat{\mathbf{A}}^{(L)}[kT], \widehat{\mathbf{A}}^{(E)}[kT])$ and scheduling decisions $(\widehat{\mathbf{S}}[t])_{t=kT}^{(k+1)T-1}$. Even in the case with one-time slot deadline constraint, it is not clear how to decouple offloading and scheduling decisions at the first glance. To that end, we first consider the decoupled JOS algorithm design in a simplistic case with one-time slot deadline constraint. Then, based on the insights obtained from this simplistic scenario, we develop the decoupled JOS algorithm in general setups.

IV. JOS Algorithm Implementation for the Case with One time slot deadline

In this section, we consider the decoupled algorithm design for the JOS Algorithm in the case with one time slot deadline constraint (i.e., $T = 1$). We explore the intrinsic structure of optimization problem (6) in the presence of one time slot constraint, and develop the following easily implementable algorithm.

Decoupled JOS (DJOS) Algorithm for the case with one time slot deadline: In each time slot t, given $(\mathbf{X}[t], \mathbf{A}[t], \mathbf{C}[t])$,

(1) User $n^*[t]$ is allowed for both local and edge computations, and put $\overline{A}^{(L)}_{n^*[t]}[t]$ and $\overline{A}^{(E)}_{n^*[t]}[t]$ packets for local computation and edge computations via wireless transmission, respectively.
(2) User $n \neq n^*[t]$ is only allowed for local computation and put $\overline{A}^{(L)}_{n}[t]$ packets for computation,

where scheduling decision $n^*[t]$ and offloading decisions $(\overline{A}^{(L)}_{n^*[t]}[t], \overline{A}^{(E)}_{n^*[t]}[t])$ and $(\overline{A}^{(L)}_{n}[t], \forall n \neq n^*[t])$ are determined below:

Wireless Scheduling Decision: Schedule user $n^*[t]$ such that

$$n^*[t] \in \arg\max_{n} \left\{W^{(L,E)}_n[t] - W^{(L)}_n[t]\right\}, \tag{19}$$

where $W^{(L)}_n[t]$ is the maximum weight of user n if it is only allowed for local computation, i.e.,

$$W^{(L)}_n[t] \triangleq \max_{A^{(L)}_n} \left(X_n[t] \min\left\{A^{(L)}_n, \mu_n\right\} - Me^{(L)}_n \mathbb{1}_{\left\{A^{(L)}_n > 0\right\}}\right),$$

and $W^{(L,E)}_n[t]$ is the maximum weight of user n if it is allowed for both local and edge computations, i.e.,

$$W^{(L,E)}_n[t] \triangleq \max_{A^{(L)}_n, A^{(E)}_n} \Bigg(X_n[t] \min\left\{A^{(L)}_n, \mu_n\right\} - Me^{(L)}_n \mathbb{1}_{\left\{A^{(L)}_n > 0\right\}} + \left(X_n[t] \min\left\{A^{(E)}_n, C_n[t]\right\} - Me^{(E)}_n\right)^+\Bigg).$$

Offloading Decision: For all other users $n \neq n^*[t]$, we need to find $\overline{A}^{(L)}_n[t]$ that attains the value of $W^{(L)}_n[t]$. That is, if $X_n[t] \min\{A_n[t], \mu_n\} - Me^{(L)}_n > 0$, then $\overline{A}^{(L)}_n[t] = A_n[t]$. Otherwise, $\overline{A}^{(L)}_n[t] = 0$.

For user $n^*[t]$, we need to find $(\overline{A}^{(L)}_{n^*[t]}[t], \overline{A}^{(E)}_{n^*[t]}[t])$ that yields to the value of $W^{(L,E)}_n[t]$. Let $W^{(E)}_{n^*[t]}[t] = X_{n^*[t]}[t] \min\{A_{n^*[t]}[t], C_{n^*[t]}[t]\} - Me^{(E)}_{n^*[t]}$ be the weight that user n^* performs offloading decision and $W^{(L)}_{n^*[t]}[t] = X_{n^*[t]}[t] \min\{(A_{n^*[t]}[t] - C_{n^*[t]}[t])^+, \mu_{n^*[t]}\} - Me^{(L)}_{n^*[t]}$ be the weight that user n^* performs local computation. Then the offloading decisions are listed in the Table II.

In our proposed DJOS Algorithm for the case with one time slot deadline, the offloading and scheduling decisions are nicely decoupled and it computational complexity is just $O(N)$. Moreover, it yields the optimal solution to the problem (6) in the JOS Algorithm in the case with one time slot constraint, as shown below.

Proposition 2: DJOS Algorithm optimally solves the optimization problem (6) in the JOS Algorithm in the case with one time slot constraint.

Proof: In the case with one time slot deadline (i.e., $T = 1$), the objective function in the optimization problem (6) becomes:

$$\begin{aligned}
&\sum_{n=1}^{N} \left(X_n[t] \min\left\{A^{(L)}_n[t], \mu_n\right\} - Me^{(L)}_n \mathbb{1}_{\left\{A^{(L)}_n[t] > 0\right\}}\right) \\
&\quad + \sum_{n=1}^{N} \left(X_n[t] \min\left\{A^{(E)}_n[t], C_n[t] S_n[t]\right\} - Me^{(E)}_n S_n[t]\right) \\
&= \sum_{n=1}^{N} \left(X_n[t] \min\left\{A^{(L)}_n[t], \mu_n\right\} - Me^{(L)}_n \mathbb{1}_{\left\{A^{(L)}_n[t] > 0\right\}}\right) \\
&\quad + \left(X_m[t] \min\left\{A^{(E)}_m[t], C_m[t]\right\} - Me^{(E)}_m\right)^+
\end{aligned}$$

	$W_n^{(y_n^*)(E)}[kT] \geq 0$	$W_n^{(y_n^*)(E)}[kT] < 0$
$W_n^{(y_n^*)(L)}[kT] \geq 0$	$A_n^{(E)}[kT] = \min\{A_n[kT], y_n^* C_n[kT]\}$ $A_n^{(L)}[kT] = A_n[kT] - A_n^{(E)}[kT]$	$A_n^{(E)}[kT] = 0$ $A_n^{(L)}[kT] = A_n[kT]$
$W_n^{(y_n^*)(L)}[kT] < 0$	$A_n^{(E)}[kT] = A_n[kT]$ $A_n^{(L)}[kT] = 0$	$A_n^{(E)}[kT] = A_n[kT]$ $A_n^{(L)}[kT] = 0$

TABLE III: Offloading decisions in general cases.

$$= \sum_{n\neq m}\left(X_n[t]\min\left\{A_n^{(L)}[t],\mu_n\right\} - Me_n^{(L)}\mathbb{1}_{\left\{A_n^{(L)}[t]>0\right\}}\right) + \left(X_m[t]\min\left\{A_m^{(L)}[t],\mu_m\right\} - Me_m^{(L)}\mathbb{1}_{\left\{A_m^{(L)}[t]>0\right\}}\right) + \left(X_m[t]\min\left\{A_m^{(E)}[t],C_m[t]\right\} - Me_m^{(E)}\right)^+, \quad (20)$$

where the second last step is true for m being the index of user that is scheduled for data transmission in time slot t if $X_m[t]\min\left\{A_m^{(E)}[t],C_m[t]\right\} - Me_m^{(E)} > 0$ and uses the fact that at most one user can be scheduled for data transmission in each time slot.

Therefore, the optimization problem (6) in the case with $T = 1$ is equivalent to

$$\begin{aligned}
&\max_m \sum_{n\neq m}\max_{A_n^{(L)}}\left(X_n[t]\min\left\{A_n^{(L)},\mu_n\right\} - Me_n^{(L)}\mathbb{1}_{\left\{A_n^{(L)}>0\right\}}\right) \\
&+ \max_{A_m^{(L)},A_m^{(E)}}\left(X_m[t]\min\left\{A_m^{(L)},\mu_m\right\} - Me_m^{(L)}\mathbb{1}_{\left\{A_m^{(L)}>0\right\}} + \left(X_m[t]\min\left\{A_m^{(E)},C_m[t]\right\} - Me_m^{(E)}\right)^+\right) \\
\Leftrightarrow &\max_m \max_{A_m^{(L)},A_m^{(E)}}\left(X_m[t]\min\left\{A_m^{(L)},\mu_m\right\} - Me_m^{(L)}\mathbb{1}_{\left\{A_m^{(L)}>0\right\}} + \left(X_m[t]\min\left\{A_m^{(E)},C_m[t]\right\} - Me_m^{(E)}\right)^+\right) \\
&- \max_{A_m^{(L)}}\left(X_m[t]\min\left\{A_m^{(L)},\mu_m\right\} - Me_m^{(L)}\mathbb{1}_{\left\{A_m^{(L)}>0\right\}}\right), \quad (21)
\end{aligned}$$

where we use the fact that the optimal solution does not change if the objective function subtracts a constant $\sum_{n=1}^N \max_{A_n^{(L)}} f(A_n^{(L)})$ and $f(A_n^{(L)}) \triangleq X_n[t]\min\left\{A_n^{(L)},\mu_n\right\} - Me_n^{(L)}\mathbb{1}_{\left\{A_n^{(L)}>0\right\}}$.

Thus, by introducing notations of $W_n^{(L)}[t]$ and $W_n^{(L,E)}[t]$ in the DJOS Algorithm, (21) is equivalent to (19) in the DJOS Algorithm and hence we have the desired result. ∎

The difficulty of the JOS Algorithm implementation lies in the fact that offloading and scheduling decisions are strongly coupled. Our proposed DJOS Algorithm nicely decouples them by first determining the user that is allowed for wireless transmission. Based on this insight, we are ready to develop DJOS Algorithm for the general case with T time slot constraint.

V. JOS Algorithm implementation

In this section, we consider the implementation of our proposed JOS Algorithm (cf. Section III) in general setups. Base on the insight of DJOS Algorithm developed in the case with one time slot deadline, we propose the following Decoupled JOS Algorithm.

Similar to the implementation of JOS Algorithm for the case with one time slot deadline, we use $W_n^{(L)}[kT]$ to denote the weight that user n is allowed to use local computations in time frame k, which can be represented as

$$W_n^{(L)}[kT] \triangleq \max_{z=0,1,2,\ldots,T}\left(X_n[kT]\min\left\{A_n[kT], z\mu_n\right\} - Me_n^{(L)}\min\left\{\left\lceil\frac{A_n[kT]}{\mu_n}\right\rceil, z\right\}\right).$$

Let y_n denote the maximum number of wireless transmissions allowed for user n in frame k. We use $W_n^{(y_n)(L,E)}[kT]$ to the maximum weight of user n when it is allowed for wireless transmissions y_n times in frame k. In particular, $W_n^{(y_n)(L,E)}[kT]$ can be represented as

$$W_n^{(y_n)(L,E)}[kT] \triangleq W_n^{(y_n)(E)}[kT] + W_n^{(y_n)(L)}[kT],$$

where $W_n^{(y_n)(E)}[kT] \triangleq X_n[kT]\min\{A_n[kT], y_n C_n[kT]\} - Me_n^{(E)}\min\left\{\left\lceil\frac{A_n[kT]}{C_n[kT]}\right\rceil, y_n\right\}$ and $W_n^{(y_n)(L)}[kT] \triangleq \max_{z=0,1,\ldots,T} X_n[kT]\min\{(A_n[kT] - y_n C_n[kT])^+, z\mu_n\} - Me_n^{(L)}\min\left\{\left\lceil\frac{(A_n[kT]-y_n C_n[kT])^+}{\mu_n}\right\rceil, z\right\}$.

Next, we propose the Decoupled JOS Algorithm for the case with T time slot deadline. Let $\mathbf{y} \triangleq (y_n)_{n=1}^N$.

Decoupled JOS (DJOS) Algorithm for the case with T time slot deadline: Given $\mathbf{A}[kT]$, $\mathbf{X}[kT]$ and $\mathbf{C}[kT]$ at the beginning of time frame k, perform the following:

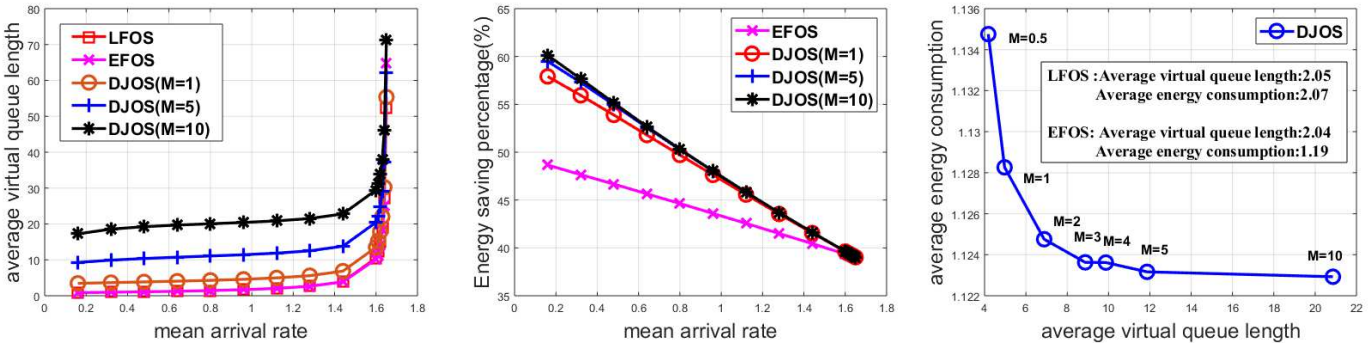
Wireless Scheduling Decision: Wireless scheduling decisions $\mathbf{y}^*$ is obtained by solving the following optimization problem:

$$\mathbf{y}^* \in \arg\max_{\mathbf{y}} \sum_{n=1}^N \left(W_n^{(y_n)(L,E)}[kT] - W_n^{(L)}[kT]\right) \quad (22)$$

$$\text{s.t.} \quad \sum_{n=1}^N y_n = T. \quad (23)$$

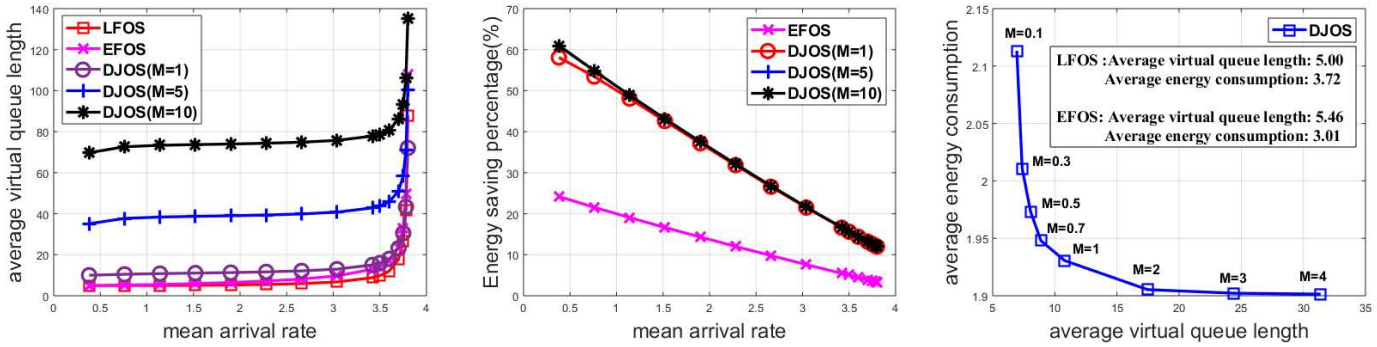
Offloading Decision: Once wireless scheduling decisions $\mathbf{y}^* = (y_n^*)_{n=1}^N$ are made, the offloading decision of each user n depends on whether $W_n^{(y_n^*)(E)}[kT]$ and $W_n^{(y_n^*)(L)}[kT]$ are positive or not, which are listed in the Table III.

In the proposed DJOS Algorithm, (23) is true since only one user is allowed for wireless transmission in each time slot and thus there are T total number of wireless transmissions.



(a) Average virtual queue length (b) Energy saving compared to LFOS (c) Impact of M in the DJOS

Fig. 1: Case with one time-slot deadline.



(a) Average virtual queue length (b) Energy saving compared to LFOS (c) Impact of M in the DJOS

Fig. 2: Case with three time-slot deadline.

The proposed DJOS Algorithm nicely decouples the offloading and wireless scheduling decisions, despite that its computational complexity is $O(N^T)$. Similar to the case with one time slot deadline, we can show that the proposed DJOS Algorithms yields the optimal solution to the problem (6) in the case with general setup.

Proposition 3: DJOS Algorithm optimally solves the optimization problem (6) in the JOS Algorithm in the case with one time slot constraint.

Proof: The proof is similar to that of Proposition 2, and is omitted for brevity. ■

VI. Simulation Results

In this section, we perform simulations to validate the efficiency of proposed low-complexity joint offloading and scheduling algorithm. We consider $N = 5$ users with the maximum allowable dropping rate $\rho = 0.1$. All users suffer from i.i.d. ON-OFF channel fading with probability 0.9 being ON. All users have arriving computing tasks within each frame that follow i.i.d. Bernoulli distribution. The services for all virtual queues follow i.i.d. Bernoulli distribution with probability $\rho\lambda$. We let $e^{(L)} = 7$ watt and $e^{(E)} = 4$ watt. This is motivated by the fact (see [18]) 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 set the local processing rate μ to 1.

The case with one time-slot deadline	The case with three time-slot deadline
Transmission rate 5 when the channel is ON	Transmission rate 4 when the channel is ON
$X = \begin{cases} 5, p = \lambda/5 \\ 0, \text{otherwise} \end{cases}$	$X = \begin{cases} 7, p = \lambda/7 \\ 0, \text{otherwise} \end{cases}$

TABLE IV: Setups, where X denotes number of arriving packets and p denotes the probability of having arrivals.

We consider both cases with one and three time-slot deadline constraints, whose simulation setups are shown in the Table IV.

From Fig. 1a illustrates the simulation results in the case with one time-slot deadline. From Fig. 1a, we can observe that all algorithms yield finite average virtual queue length when the arrival rate is less than 1.62. This implies that all users satisfy maximum allowable drop rate requirements. However, we can see from Fig. 1b that our proposed DJOS Algorithm significantly saves energy compared to the LFOS Algorithm (with the highest average energy consumption) and yields about 60% energy improvement when mean arrival rate is equal to 0.2 and $M = 10$, while EFOS just performs about 48% energy improvement. Fig. 1c studies the impact of design parameter M on the performance of DJOS Algorithm.

From Fig. 1c, we can see that our proposed DJOS Algorithm achieves a better tradeoff between average energy consumption and average virtual queue length than both throughput-optimal EFOS and LFOS Algorithms. In addition, as M increases, the average energy consumption drops at the cost of increasing average virtual queue length, which confirms our result (cf. Proposition 1).

In the case with three time-slot deadline, similar to the one time-slot deadline case, all algorithms can meet the maximum allowable dropping rate requirement, as shown in Fig. 2a. However, form Fig. 2b we can see that DJOS algorithm can save energy at least twice than the EFOS algorithm when $M = 10$. We can also see from Fig. 2c that DJOS Algorithm achieves a better tradeoff between average energy consumption and average virtual queue length than both throughput-optimal EFOS and LFOS Algorithms.

VII. Conclusion

In this paper, we considered the joint offloading and scheduling design for mobile parallel applications with hard deadlines. We first developed a joint offloading and scheduling algorithm by using the stochastic network optimization framework, and showed that it yields average energy consumption arbitrarily close to the optimal one required for meeting the desired drop rate requirements. However, this algorithm involves a strong coupling between offloading and scheduling decisions, which makes its implementation quite challenging. To that end, we first considered a simplistic case with one time slot constraint and developed a decoupled JOS algorithm that successfully decouples offloading and scheduling decisions. Based on this, we further developed a decoupled JOS algorithm in general setup. Simulations were provided to validate the efficiency of our proposed algorithms.

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *arXiv preprint arXiv:1702.05309*, 2017.

[10] G. Gao, M. Xiao, J. Wu, K. Han, L. Huang, and Z. Zhao, "Opportunistic mobile data offloading with deadline constraints," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 12, pp. 3584–3599, 2017.

[11] L. Zhang, D. Fu, J. Liu, E. C.-H. Ngai, and W. Zhu, "On energy-efficient offloading in mobile cloud for real-time video applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 1, pp. 170–181, 2017.

[12] I.-H. Hou, V. Borkar, and P. Kumar, *A theory of QoS for wireless*. IEEE, 2009.

[13] J. J. Jaramillo and R. Srikant, "Optimal scheduling for fair resource allocation in ad hoc networks with elastic and inelastic traffic," in *2010 Proceedings IEEE INFOCOM*. IEEE, 2010, pp. 1–9.

[14] B. Li and A. Eryilmaz, "Optimal distributed scheduling under time-varying conditions: A fast-csma algorithm with applications," *IEEE Transactions on Wireless Communications*, vol. 12, no. 7, pp. 3278–3288, 2013.

[15] I.-H. Hou, "Scheduling heterogeneous real-time traffic over fading wireless channels," *IEEE/ACM Transactions on Networking*, vol. 22, no. 5, pp. 1631–1644, 2014.

[16] M. Neely, *Stochastic network optimization with application to communication and queueing systems*. Morgan & Claypool, 2010.

[17] M. J. Neely, "Energy optimal control for time-varying wireless networks," *IEEE transactions on Information Theory*, vol. 52, no. 7, pp. 2915–2934, 2006.

[18] 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.