Joint Channel and Queue Aware Scheduling for Latency Sensitive Mobile Edge Computing with Power Constraints

Di Han, Wei Chen, *Senior Member, IEEE*, and Yuguang Fang, *Fellow, IEEE*

***Abstract*—Mobile edge computing (MEC) is a promising technique to improve the quality of computation experience for mobile devices by providing computation resources in close proximity. However, the design of scheduling policies for MEC systems inevitably encounters a challenging optimization problem that should take both transmissions and computations into consideration. In particular, how to jointly schedule transmissions and the computations should adapt to the cross-layer system dynamics, i.e., random task arrivals and channel state variations. We formulate this scheduling problem as a joint optimization problem for both transmissions and computations in order to minimize the power consumption of mobile devices, while meeting the latency requirement. With given distributions of the system dynamics, Markov decision process (MDP) is used to model the system operations. Based on this model, the power-optimal scheduling policy can be obtained by converting the joint optimization problem to linear programming (LP) by using variable substitutions and thus the optimal power-latency tradeoff can be achieved. When the distribution information of the system dynamics is unknown, we exploit the Lyapunov optimization to present a low complexity scheduling policy. Our theoretical analysis and extensive simulation studies show that our approach can offer a good tradeoff between power consumption and latency.**

***Index Terms*—Mobile edge computing, Markov decision process, Lyapunov optimization, power-latency tradeoff.**

I. Introduction

In recent years, the demand for mobile devices to execute high computation tasks is increasing rapidly. However, the resources of mobile devices, e.g., battery life and computation capability, are still insufficient due to the limited form factors [2] and [3]. The conflicting design considerations between resource-hungry computation tasks and resource-constrained mobile devices hence pose as a significant challenge to the future mobile computing. To overcome this challenge, mobile edge computing (MEC) is proposed, whose aim is to provide computation resources, i.e., edge server,in close proximity of mobile devices [4].

Nowadays, MEC is recognized as one of the key technologies for the next generation 5G networks. The benefits of MEC consist of high energy efficiency and low latency [5], which are also considered as the key performance indices (KPIs) for the support of low latency sensitive applications under the limited energy resource of mobile devices in computation offloading [6] and [7]. For the computation offloading in MEC systems, the power consumption and latency in both the task transmissions and computations should be jointly taken into consideration [8] and [9]. Hence, a joint optimization from both transmission and computation aspects is required to schedule a task to make the best use of the system resources [10]. Unfortunately, this may result in higher complexity and more difficulty in the design of the task scheduling policy owing to the system dynamics, i.e., random task arrivals and channel state variations. Therefore, how to minimize the power consumption at the mobile device while completing the required computation task on time is highly challenging, but of paramount importance. This paper is to tackle this problem.

Many works have been conducted to study the design of scheduling policies from transmission and computation aspects in order to achieve higher energy efficiency. To meet this goal, various optimization schemes have been proposed to minimize energy consumption at both the network side and the device side. In [8], Zhang et al. designed an energy efficient computation offloading scheme, which jointly optimizes radio resource allocation and offloading to minimize the energy consumption of the offloading under the latency constraint in MEC systems. In [11], Sardellitti, Scutari, and Barbarossa formulated the task offloading as a joint optimization problem for both radio resources and computation resources, and proposed an iterative algorithm to solve the problem. In [12], a computation offloading game among the mobile device users is formulated by Chen et. al, which takes into account both communication and computation aspects of MEC systems. The power-delay tradeoff for multi-user MEC systems was investigated in [13] via the joint management of transmission and computational resources. In [14], Bi et. al considered a multi-user wireless powered MEC system and focus on a weighted sum computation rate maximization problem by optimizing computing mode selection and system transmission time allocation. In [15], Mao et. al proposed a low-complexity online algorithm for an MEC system exploiting energy harvesting, which jointly determines the CPU frequencies and the transmission power for offloading. In addition, in [16], Munoz et. al proposed a joint allocation scheme for transmission and computational resources for femto-cloud computing systems, where each computation task should be completed within

This work was supported in part by the National Natural Science Foundation of China under Grant No. 61971264 and No. 61671269, and the Beijing Natural Science Foundation under Grant No. 4191001. The work of Y. Fang is partially supported by U.S. National Science Foundation under grant CNS-1717736. The preliminary work has been presented at IEEE ICC'2018 [1].

D. Han and W. Chen are with the Department of Electronic Engineering, Tsinghua University, Beijing 100084, China (e-mail: hd15@mails.tsinghua.edu.cn, wchen@tsinghua.edu.cn).

Y. Fang is with the Department of Electrical and Computer Engineering and University of Florida, Gainesville, FL 32611, USA (e-mail: fang@ece.ufl.edu).

required latency tolerance. In [17], Wen et. al proposed an efficient offloading policy by jointly scheduling the clock frequency and the data transmission at a mobile device to minimize the energy consumption. In [18], Yu et al. proposed an offloading policy with caching enhancement for mobile edge cloud networks to minimize the execution delay. In [19], Wang et al. proposed an efficient wireless powered multiuser MEC design with joint energy beamforming, offloading, and computing optimization. In [20], Liu et al. proposed a two-timescale framework for UE-server association, task offloading, and resource allocation by incorporating techniques from Lyapunov stochastic optimization and matching theory.

As mentioned before, it turns out that analyzing and optimizing system performance such as queueing delay and power consumption is a challenging task due to the varying system dynamics. To achieve better system performance, the cross-layer scheduling has been widely investigated in both academia and industry, as it holds the promise to obtain a good scheduling policy by jointly optimizing different protocols with cross-layer in the considered systems, e.g., the random task arrivals in the network layer and random channel state variations in the physical layer. To our best knowledge, Collins and Cruz firstly proposed the idea of the cross-layer scheduling based on the queue and channel states jointly in [21] to minimize the average power consumption under the constraints of average latency and peak transmission power. In [22], Berry and Gallager obtained the optimal power-delay tradeoff with a discrete Markov decision process (MDP). Xu et al. in [23] considered cross-layer optimization for wireless sensor networks powered by heterogeneous energy sources through the Lyapunov optimization approach. In [24] Chen et al., investigated the situation with Bernoulli arrivals and a fixed modulation over a block fading channel, and proposed a probabilistic policy. Based on this, a generalized model for data arrivals, fading channels, as well as the transmission rates was investigated in [25] and [26]. Furthermore, we extended this cross-design method into the design of joint transmission and computation scheduling (JTCS) policy in the computation offloading systems in our previous work [1]. Based on this method, the optimal power-latency tradeoff can be achieved.

In our previous work [1], we focus on the cross-layer scheduling in the computation offloading systems under simple assumptions that the task arrival follows a Bernoulli process and channel state is fixed in different time slots. However, considering the burstiness of traffic arrivals in practical applications and uncertainty in wireless channels, it is necessary to consider the systems with an arbitrarily random task arrival distribution and a time-varying wireless fading channel. Moreover, we only consider the case that the distribution information of task arrivals is already known for the scheduler, which might not be guaranteed in practical systems.

In this paper, we adopt an arbitrarily random task arrival distribution to capture the burstiness of the traffic, i.e., there is no limitation on the number or the distribution of the arriving tasks during one time slot. In addition, multi-state of a wireless link is assumed in this paper instead of a fixed channel state in our previous work. Moreover, the design of the scheduling policy without distribution information is also considered. We formulate a cross-layer scheduling problem as a joint optimization problem for the transmission rate together with the computation rate in order to minimize the power consumption at the mobile device, while meeting the latency requirement of tasks.

With known distribution information on task arrivals and channel state variations, the scheduling problem can be formulated into a Markov decision process (MDP) framework with the proposed probabilistic scheduling policy, where MDP is used to model the system operations. Based on this framework, the average latency and power consumptions in transmissions and computations can be analyzed and the original optimization problem can be converted into linear programming (LP), which can be solved efficiently in polynomial time. Then, the power-optimal scheduling policy can be obtained and thus the optimal power-latency tradeoff can be achieved. For the design of scheduling without distribution information, a Lyapunov drift-plus-penalty approach based on the Lyapunov optimization is implemented to schedule the task offloading. By minimizing the Lyapunov drift-plus-penalty function, an online algorithm is proposed and thus the scheduling decisions can be obtained at each time slot.

The main contribution of this work is to formulate the power-optimal scheduling problem as a joint optimization problem from the transmission and computation aspects, by considering the random task arrivals in the network layer, the queueing behavior in the data link layer, and power adaption and random channel state variations in the physical layer. A simple but applicable system model is used to capture the essence of the real system operations in two important aspects: task arrivals are usually bursty, and the channel state is quantized and fed back to the scheduler with a limited channel state information (CSI) feedback. Based on this model, the tradeoff between the power consumption and the latency is analyzed theoretically. Various optimization approaches are given to address the above optimization problem in two cases that the distribution information is known and unknown, respectively. With the known distribution information, the optimal scheduling policy can be obtained efficiently and thus optimal power-latency tradeoff can be achieved based on the MDP framework. For the case with unknown distribution information, a Lyapunov drift-plus-penalty approach is developed based on Lyapunov optimization to obtain a sub-optimal scheduling policy with low complexity and an analytic bound on the performance can be given from the theoretical analysis when the buffer sizes are sufficiently large.

The rest of this paper is organized as follows. Section II presents the system model. The joint scheduling problem is formulated in Section III. The scheduling policies with distribution information being known and unknown are given in Section IV and V, respectively. Numerical results are presented in Section VI, and Section VII concludes the paper.

II. System Model

In this section, we introduce our model for cross-layer scheduling development, consisting of the random task arrival in the network layer, the queueing behavior in the data link

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2020.2979136, IEEE Transactions on Wireless Communications

layer, and power adaption and random channel state in the physical layer.

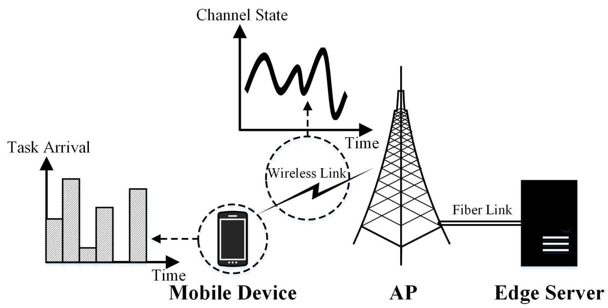


Fig. 1. An illustration of an MEC system.

As shown in Fig. 1, we consider an MEC system consisting of one mobile device and one edge server. The computation tasks arrived at the mobile device can be offloaded to the edge server via a multi-state wireless fading channel and then be computed by the edge server. In this way, a significantly improved computation experience can be achieved. All control operations, i.e., transmissions and computations, are managed by a scheduler. The time is discretized into time slots, each of which is of equal duration T_{s} and is indexed by an integer t. In this paper, we focus on optimizing the average power consumption at the mobile device, while guaranteeing the latency requirement of tasks.

The latency experienced by a task in the offloading scheduling is defined as the time from when the computation task is arrived to when the result of the computation task is obtained at the mobile device. Thus, it incorporates the time to upload the task to the edge server, the time necessary to compute the task at the edge server, and the time to download the computation result of the task back to the mobile device. For the sake of simplicity, we assume that the time necessary for download is a fixed small value, which can be negligible compared with the time necessary for upload and computation. Therefore, the latency for each task can be considered as the sum of the time necessary for transmission during the upload step and for computation at the edge server. Thus, we can characterize the considered system by a tandem queue model, consisting two queues of the mobile device and edge server, as illustrated in Fig. 2.

Let $a[t]$ denote the number of tasks newly arrived at the mobile device at the beginning of the t-th time slot. We suppose that $a[t]$ is a random variable, which follows an independent and identically distribution across the time slots. To capture the burstiness and variability of real-time applications, the distribution of $a[t]$ can be characterized by

$$\Pr\{a[t]=m\}=p_m, \quad m=0,1,2,\ldots, \tag{1}$$

where m is a nonnegative integer and $p_m \in [0,1]$. Moreover, considering traffic shaping and admission control adopted in the system [26], the number of tasks newly arrived at each time slot must be upper-bounded by a large integer M, where $p_m = 0$ for all $m > M$. Thus, the average task arrival rate is obtained by

$$\lambda_1 = \lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}a[t]=\sum_{m=0}^{M}m\cdot p_m. \tag{2}$$

Due to the normalization constraint, we have

$$\sum_{m=0}^{M}p_m = 1. \tag{3}$$

Let Q_1 and Q_2 the buffer sizes of the mobile device and the edge server, which are finite. Each task is set to be offloaded to the edge server via the wireless link. The tasks arrived but not transmitted can be backlogged in the buffer of the mobile device. The queue length of the mobile device, defined as the number of tasks backlogged in the buffer of the mobile device at the end of t-th slot, is denoted by $q_1[t]$ and updated as

$$q_1[t]=\max\left\{0,\min\left\{Q_1,q_1[t-1]+a[t]\right\}-s[t]\right\}, \tag{4}$$

where $s[t]$ is the transmission rate, defined as the number of tasks transmitted in the t-th time slot, and $\min\{Q_1, q_1[t-1]+a[t]\}$ is to guarantee that the tasks backlogged cannot exceed the buffer size Q_1. Then the tasks received by the edge server can be computed in the next time slot. Likewise, the tasks that have not been computed can be backlogged in the buffer of the edge server and the queue length is denoted by $q_2[t]$, which is updated as

$$q_2[t]=\min\left\{Q_2,\max\left\{0,q_2[t-1]-c[t]\right\}+s[t]\right\}, \tag{5}$$

where $c[t]$ is the computation rate, defined as the number of tasks computed in the t-th time slot. Due to the constraints of the mobile device and edge server, it is reasonable to assume that at most S and C tasks can be transmitted and computed in one time slot. Thus, we have $s[t] \in \{0,1,\ldots,S\}$ and $c[t] \in \{0,1,\ldots,C\}$, which are scheduled at the beginning of one time slot. Moreover, to avoid any possible packet dropping, we assume that $S \geq M$ and $C \geq M$.

The wireless channel is assumed to experience N-state block fading, where N is a positive integer. Let $h[t] > 0$ denote the continuous channel power gain in the t-th time slot. Assume that the channel power gain stays constant during one time slot, and follows an independent and identically distribution across the time slots. With the limited CSI feedback, the continuous channel state $h[t]$ is quantized into N discrete states. The discrete channel state in the t-th time slot is denoted by $l[t] \in \{1,2,\ldots,N\}$. Let $h_1 = \infty > h_2 > \ldots > h_N > h_{N+1} = 0$ be the channel power gain levels. Assume that the channel state is at the n-th channel state, i.e., $l[t] = n$, if $h[t]$

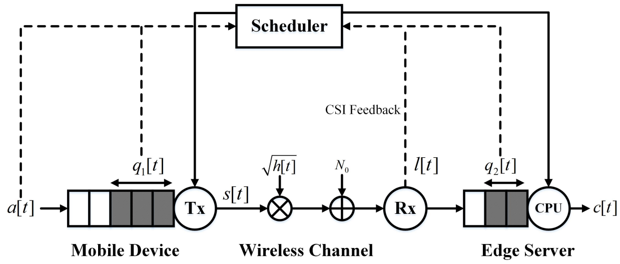


Fig. 2. System model.

ranges in the interval $[h_n, h_{n+1})$. Accordingly, the probability that the channel is at the n-th state is denoted by ϕ_n, which is given by

$$\phi_n = \Pr\{l[t] = n\} = \int_{h_{n+1}}^{h_n} p(h)dh, \tag{6}$$

where $n \in \{1, 2, \ldots, N\}$ and $p(h)$ is the probability density function of the continuous channel power gain $h[t]$. Due to the normalization constraint, we have

$$\sum_{n=0}^{N} \phi_n = 1. \tag{7}$$

Suppose that there exists a feedback channel through which CSI is sent back from the edge server to the mobile device. Then, the channel state $l[t]$ can be obtained by the scheduler at the beginning of one time slot. Therefore, the scheduler can determine the action $s[t]$ and $c[t]$ based on the queue states $q_1[t]$ and $q_2[t]$, the task arrival $a[t]$, and the channel state $l[t]$ at one time slot.

In practical systems, the transmission power consumption should be adapted to the channel state and the transmission rate to meet the bit error rate (BER) requirement. Let $P_{\rm s}(s, n)$ denote the power required to transmit s tasks successfully in one time slot with the n-th channel state. Since more power is required to combat channel fading when the channel condition is poor, for any given transmission rate $s \in \{1, \ldots, S\}$, we have

$$P_{\rm s}(s, n_1) < P_{\rm s}(s, n_2), \quad \forall n_1 < n_2. \tag{8}$$

Moreover, when the transmission rate $s = 0$, we always have $P_{\rm s}(0, n) = 0$ for any channel state.

From the perspective of the physical layer, for one given channel state, to transmit more bits without increasing the BER, a higher constellation should be implemented. Therefore, more power will be consumed for every bit on average. As a result, for any given channel state $n \in \{1, \ldots, N\}$, we have

$$\frac{P_{\rm s}(s_1, n)}{s_1} < \frac{P_{\rm s}(s_2, n)}{s_2}, \quad \forall s_1 < s_2. \tag{9}$$

Moreover, the transmission power consumption in the t-th time slot is denoted by

$$P_{\rm s}[t] = P_{\rm s}(s[t], l[t]). \tag{10}$$

Let $P_{\rm c}(c)$ denote the power needed to compute c tasks in one time slot. According to [27], the power consumption per CPU cycle at the edge server can be expressed as

$$P_{\rm cycle} = \kappa {f_{\rm c}}^2, \tag{11}$$

where $f_{\rm c}$ is the clock frequency and κ is the effective switched capacitance depending on the chip architecture. In each time slot, the total CPU cycles should satisfy

$$f_{\rm c} T_{\rm s} = Lc, \tag{12}$$

where L is the number of required CPU cycles to compute one task. Then $P_{\rm c}(c)$ can be rewritten as

$$P_{\rm c}(c) = P_{\rm cycle} \cdot f_{\rm c} = \frac{\kappa L^3}{{T_{\rm s}}^3} c^3. \tag{13}$$

Based on the above analysis, we have

$$\frac{P_{\rm c}(c_1)}{c_1} < \frac{P_{\rm c}(c_2)}{c_2}, \quad \forall c_1 < c_2. \tag{14}$$

Moreover, the computation power consumption in the t-th time slot is denoted by

$$P_{\rm c}[t] = P_{\rm c}(c[t]). \tag{15}$$

Our system model captures the essence of the real system in two important aspects. Task arrivals are usually bursty in practice, and the channel state is quantized and fed back to the scheduler. Hence, this simple but applicable model is general enough, and allows tractable analysis to provide insights for further study and practical protocol design.

III. Problem Formulation

In this section, we should mathematically formulate the joint transmission and computation scheduling (JTCS) problem in the considered system. Based on this joint optimization problem, we aim to propose the JTCS scheduling policies to determine the transmission rate $s[t]$ and the computation rate $c[t]$ in each time slot.

Intuitively, by recalling Eq. (8), we notice that more transmission power consumption can be saved if the mobile device is willing to wait for better channel condition. However, the waiting time can be undesirably prolonged if the wireless channel stays at poor conditions for a long time. To reduce the latency, the mobile device has to transmit tasks backlogged in the buffer when the wireless channel state is not so good, which inevitably leads to more transmission power consumption. On the other hand, from Eqs. (9) and (14), we see that the latency becomes shorter as the increase of the transmission rate $s[t]$ and the computation rate $c[t]$, but with the cost of more power consumption. Hence, there naturally exists a tradeoff between the power consumption and the latency in the considered system.

We are interested in the average latency from the task generation time to its accomplish time, denoted by $D^{\rm ava}$. Due to the QoS constraint, $D^{\rm ava}$ should be less than or equal to a latency constraint denoted by $D^{\rm th}$. In other words, instead of minimizing the average latency, we only need to guarantee that the average latency of tasks $D^{\rm ava}$ does not exceed $D^{\rm th}$.

According to Little's Law, the average length $L^{\rm ava} = \lim\limits_{T\to\infty} \frac{1}{T} \sum\limits_{t=1}^{T} (q_1[t] + q_2[t])$ is equal to the arrival rate λ_1 multiplied by the average waiting time W that a task spends in the queue. Moreover, the total latency experienced by each task is the sum of waiting time in the queue and the time of the task being computed at the edge server, i.e., one time slot. Thus, we have $D^{\rm ava} = W + 1$. Hence,the average latency of tasks $D^{\rm ava}$ can be calculated by

$$D^{\rm ava} = 1 + \frac{L^{\rm ava}}{\lambda_1} = 1 + \lim_{T\to\infty} \frac{1}{\lambda_1 T} \sum_{t=1}^{T} (q_1[t] + q_2[t]). \tag{16}$$

As mentioned before, the battery life of the mobile device is limited. Hence, it is necessary to minimize the average transmission power consumption at the mobile device, which is give by

$$P_{\rm s}^{\rm ava} = \lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}P_{\rm s}[t]. \tag{17}$$

Furthermore, there also should have a constraint on the average computation power consumption, which is given by

$$P_{\rm c}^{\rm ava} = \lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}P_{\rm c}[t] \le P_{\rm c}^{\rm th}, \tag{18}$$

where $P_{\rm c}^{\rm th}$ is the constraint of the power consumption at the edge server.

Thus, the power-optimal JTCS problem in the considered system can be formulated as

$$\min_{s[t],c[t]} P_{\rm s}^{\rm ava} = \lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}P_{\rm s}[t] \tag{19.a}$$

$$\text{s.t.}\quad D^{\rm ava} = 1 + \lim_{T\to\infty}\frac{1}{\lambda_1 T}\sum_{t=1}^{T}(q_1[t]+q_2[t]) \le D^{\rm th}, \tag{19.b}$$

$$P_{\rm c}^{\rm ava} = \lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}P_{\rm c}[t] \le P_{\rm c}^{\rm th}, \tag{19.c}$$

$$q_1[t] = \max\{0, \min\{Q_1, q_1[t-1]+a[t]\} - s[t]\}, \tag{19.d}$$

$$q_2[t] = \min\{Q_2, \max\{q_2[t-1]-c[t], 0\} + s[t]\}, \tag{19.e}$$

$$s[t] \in \{0,1,\ldots,S\}, \tag{19.f}$$

$$c[t] \in \{0,1,\ldots,C\}, \tag{19.g}$$

where the transmission rate $s[t]$ and the computation rate $c[t]$ in each time slot are decision variables.

Based on this problem, we next study the JTCS policies for the two scenarios in Section IV and V, where the distribution information of the task arrivals and channel state variations is known and unknown, respectively, In these scenarios, an MDP-based approach and a Lyapunov drift-plus-penalty approach are implemented to obtain the scheduling policies. The key features of the two policies are summarized in Table I.

IV. Scheduling with MDP-based Approach

In this section, we focus on the design of power-optimal JTCS policy with known distribution information of task arrivals and channel state variations. A MDP-based approach is implemented to analyze the proposed scheduling policy.

A. The Probabilistic Scheduling Policy

Given the information of distributions of the task arrival $a[t]$ and the channel state $l[t]$, we aim to find an power-optimal JTCS policy based on the problem (19), which can achieve an optimal tradeoff between the average transmission power consumption, the average computation power consumption, and the average latency. To this end, we propose a probabilistic scheduling policy to schedule the transmission rate $s[t]$ and the computation rate $c[t]$ for the mobile device and the edge server in each time slot based on the current queue lengths $(q_1[t], q_2[t])$, channel state $l[t]$, and task arrival $a[t]$.

The transmission rate can be determined by the scheduler after the task arrival in each time slot. Thus, in the sense of the average latency, the scheduling policy is only aware of how many tasks waiting for transmission in the buffer of the mobile device, irrespective of the tasks are newly arrived or have already backlogged before. Hence, a new queue state is defined as

$$\zeta[t] = q_1[t-1] + a[t], \tag{20}$$

where $\zeta[t]$ denotes the number of tasks backlogged in the buffer of the mobile device at the beginning of the t-th time slot. From Eqs. (4) and (20), $\zeta[t]$ is updated as

$$\zeta[t+1] = \min\{Q_1, \max\{0, \zeta[t]-s[t]\} + a[t+1]\}. \tag{21}$$

To avoid any possible packet dropping, i.e., $\zeta[t] > Q_1$, the constraint $q_1[t] \le Q_1 - M$ should always be guaranteed.

Then, the probabilistic scheduling policy is described by the probability $f_{k_1,k_2,n}^{s,c}$ of $s[t] = s$ and $c[t] = c$ given that $(\zeta[t], q_2[t]) = (k_1, k_2)$ and $l[t] = n$, which is given by

$$\Pr\{s[t]=s, c[t]=c | (\zeta[t], q_2[t]) = (k_1,k_2), l[t]=n\}. \tag{22}$$

Furthermore, the normalization constraint of the probabilistic scheduling policy, i.e.,

$$\sum_{c=0}^{C}\sum_{s=0}^{S} f_{k_1,k_2,n}^{s,c} = 1, \tag{23}$$

holds for any $k_1 \in \{0,1,\ldots,Q_1\}$, $k_2 \in \{0,1,\ldots,Q_2\}$ and $n \in \{1,2,\ldots,N\}$.

In addition, since the transmission and computation rate cannot exceed the number of tasks backlogged in buffers, the probabilistic scheduling policy should satisfy the constraints, which are given by, for any given channel state $l[t] \in \{1,2,\ldots,N\}$,

$$f_{k_1,k_2,n}^{s,c} = 0 \qquad \text{if } s > k_1, \tag{24.a}$$

$$f_{k_1,k_2,n}^{s,c} = 0 \qquad \text{if } c > k_2. \tag{24.b}$$

Moreover, to avoid packet dropping caused by the overflow of the buffers, we have the constraints, which are given by

$$f_{k_1,k_2,n}^{s,c} = 0 \qquad \text{if } k_1 - s > Q_1 - M, \tag{25.a}$$

$$f_{k_1,k_2,n}^{s,c} = 0 \qquad \text{if } k_2 + s - c > Q_2, \tag{25.b}$$

This completes the description for our probabilistic scheduling policy in this scenario.

B. The MDP Framework

For obtaining the proposed scheduling policy, we construct an MDP framework to analyze the average latency and power consumption. Based on the MDP framework, an optimization problem which is equivalent to the original problem (19) can be formulated.

According to the description of the system model, the system state can be characterized by the queue lengths $(\zeta[t], q_2[t-1])$, which denote the number of tasks backlogged

TABLE I
DIFFERENCES BETWEEN MDP-BASED AND LYAPUNOV DRIFT-PLUS-PENALTY APPROACHES.

Features	MDP-based approach	Lyapunov drift-plus-penalty approach
Algorithm type	Offline	Online
Distribution information	Necessary	Unnecessary
Power and latency analysis	Accurate formulas in Eqs. (35-38)	Upper-bound in Theorem 4
Power-latency tradeoff	Optimal	Sub-optimal

in the buffers at the beginning of t-th time slot. The system state at the next time slot only depends on the current state, and not on the states at the previous slots. Therefore, the system state can be completely characterized by a two-dimensional Markov chain, whose state space is given by

$$\{(\zeta, q_2)|0 \le \zeta \le Q_1, 0 \le q_2 \le Q_2\}. \tag{26}$$

In addition, the state transition at the next time slot depends on the current action $s[t]$ and $c[t]$, and task arrivals at the beginning of the next time slot $a[t+1]$, given the current state information.

For the ease of understanding, an instance with transition diagram for the state $\zeta[t] = 3$ and $q_2[t] = 3$ is given in Fig. 3. We denote $(a[t+1], s[t], c[t]) = (a, s, c)$ as (a, s, c) for each link . As shown in Fig. 3, the state $\zeta[t] = 3$ and $q_2[t] = 3$ cannot make transition to the states that do not have a link with it, e.g., $\zeta[t] = 2$ and $q_2[t] = 1$. For keep the figure legible, the cases that will result in packet dropping, e.g., $(1, 0, 0)$, are neglected in this figure.

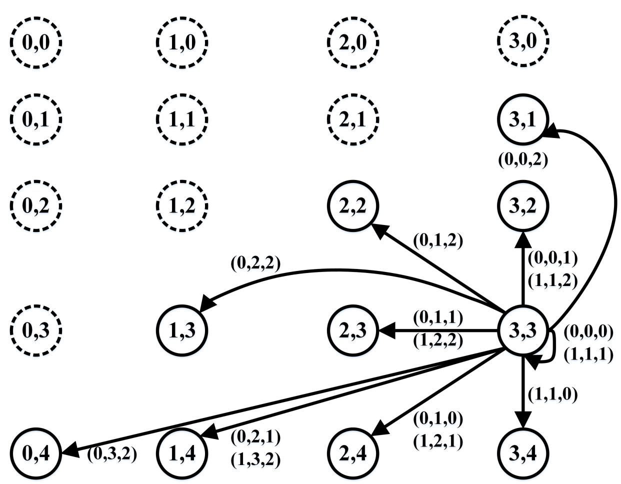


Fig. 3. Transition diagram at $\zeta[t] = 3$ and $q_2[t-1] = 3$ ($Q_1 = 3$, $Q_2 = 4$, $S = 3$, $C = 2$, $A = 1$), where (a, s, c) of one link denotes $a[t+1] = a$, $s[t] = s$, and $c[t] = c$.

Based on this Markov chain, our scheduling policy can be modeled by an MDP as a 5-tuple, defined as

$$(\mathcal{Q}, \mathcal{W}, \mathcal{A}, \boldsymbol{P}_{(.,.)}(.,.), \boldsymbol{F}). \tag{27}$$

Here, $\mathcal{Q} = \{0, 1, 2, \ldots, Q_1\} \times \{0, 1, 2, \ldots, Q_2\}$ denotes the set of all feasible system states, each of which represents the state of the Markov chain defined in Eq. (26). In addition, let $\mathcal{W}$ denote all possible combinations of task arrivals and channel states, indicating the system uncertainty. Moreover, let $\mathcal{A}$ denote the set of all feasible actions of transmission rate and computation rate. In one time slot, one action $v[t] = (s[t], c[t])$ is taken from $\mathcal{A}$ based on the system state $\boldsymbol{q} = (\zeta[t], q_2[t])$ and the system uncertainty $w[t] = (a[t+1], l[t])$. Besides, $P_{v,w}(\boldsymbol{q}_1, \boldsymbol{q}_2) = \Pr\{\boldsymbol{q} = \boldsymbol{q}_2 | \boldsymbol{q} = \boldsymbol{q}_1, v[t] = (s, c), w[t] = (m, n)\}$ is the probability that given the system uncertainty $w[t] = (m, n)$, action $v[t] = (s, c)$ in state $\boldsymbol{q}_1$ leads to state $\boldsymbol{q}_2$ in the next time slot. Finally, $\boldsymbol{F} = (\bar{D}, \bar{P}_{\rm s}, \bar{P}_{\rm c})$ denotes the state-action cost function, where $\bar{D}$ is the average latency, $\bar{P}_{\rm s}$ is the average transmission power, and $\bar{P}_{\rm c}$ is the average computation power, respectively.

Let $\lambda_{k_1,k_2}^{q_1,q_2}$ denote the transition probability from the state (k_1, k_2) to the state (q_1, q_2) in the next time slot in the considered MDP, which is summarized in the following result.

Theorem 1. The transition probability $\lambda_{k_1,k_2}^{q_1,q_2}$ satisfies

$$\lambda_{k_1,k_2}^{q_1,q_2} = \sum_{m=0}^{M} p_m \sum_{n=1}^{N} \phi_n f_{k_1,k_2,n}^{s(m),c(m)} \mathbb{1}_{\{s(m)\in\mathcal{S}, c(m)\in\mathcal{C}\}}, \tag{28}$$

where

$$\begin{cases} s(m) = k_1 - q_1 + m, \\ c(m) = k_1 + k_2 - q_1 - q_2 + m, \end{cases} \tag{29}$$

and the indicator function $\mathbb{1}_{\{s(m)\in\mathcal{S}, c(m)\in\mathcal{C}\}}$ is equal to 1 if $s(m) \in \mathcal{S}$ and $c(m) \in \mathcal{C}$; otherwise it is equal to 0.

Proof: The core idea of this proof is that the transition of the state only relies on the transmission and computation rate in the current time slot, and the task arrivals in the next time slot. From Eqs. (5) and (21), considering constraints (24) and (25), we have

$$\begin{cases} \zeta[t+1] - \zeta[t] = a[t+1] - s[t], \\ q_2[t+1] - q_2[t] = s[t] - c[t]. \end{cases} \tag{30}$$

Then, given that $a[t+1] = m$ and $l[t] = n$, whose probability is $f_{k_1,k_2,n}^{s(m),c(m)}$, the corresponding transmission rate and computation rate at the current time slot can be calculated as

$$\begin{cases} s[t] = \zeta[t] - \zeta[t+1] + a[t+1] = k_1 - q_1 + m = s(m), \\ c[t] = q_2[t] - q_2[t+1] + s[t] = k_2 - q_2 + s(m) = c(m), \end{cases} \tag{31}$$

where $s(m)$ and $c(m)$ are already given in Eq. (29). Therefore, for any $a[t+1] = m$ and $l[t] = n$, the transition probability from the state (k_1, k_2) to the state (q_1, q_2) in the next time slot is $p_m \phi_n f_{k_1,k_2,n}^{s(m),c(m)}$. Then, by summing up transition probabilities of all feasible combinations of $a[t+1]$ and $l[t]$, $\lambda_{k_1,k_2}^{q_1,q_2}$ in Eq. (28) can be obtained and hence this theorem is proved. ■

In this MDP framework, we denote π_{k_1,k_2} as the steady-state distribution probability of the state (k_1, k_2). The steady-state distribution of this Markov chain is denoted by

$$\boldsymbol{\pi}_{(Q_1+1)\times(Q_2+1)} = \begin{bmatrix} \pi_{0,0} & \cdots & \pi_{0,q_2} & \cdots & \pi_{0,Q_2} \\ \pi_{1,0} & \cdots & \pi_{1,q_2} & \cdots & \pi_{1,Q_2} \\ \cdots & & \cdots & & \cdots \\ \pi_{1,0} & \cdots & \pi_{q_1,q_2} & \cdots & \pi_{q_1,Q_2} \\ \cdots & & \cdots & & \cdots \\ \pi_{Q_1,0} & \cdots & \pi_{Q_1,q_2} & \cdots & \pi_{Q_1,Q_2} \end{bmatrix}. \quad (32)$$

For convenience, we convert the matrix $\boldsymbol{\pi}_{(Q_1+1)\times(Q_2+1)}$ into a column vector, i.e., the vectorization of a matrix. Then we have the column vector $\boldsymbol{\pi}_{(Q_1+1)(Q_2+1)\times 1} = \mathbf{vec}(\boldsymbol{\pi}_{(Q_1+1)\times(Q_2+1)})$. In addition, define $\mathbf{1} = [1, \ldots, 1]$ and $\mathbf{0} = [0, \ldots, 0]$, and let $\boldsymbol{I}$ denote the identity matrix. We do not specify their sizes if there is no ambiguity. In the rest of this paper, we refer to $\boldsymbol{\pi}_{(Q_1+1)(Q_2+1)\times 1}$ as $\boldsymbol{\pi}$, where satisfies

$$\boldsymbol{H}\boldsymbol{\pi} = \boldsymbol{\pi}, \quad (33.a)$$

$$\mathbf{1}\boldsymbol{\pi} = 1, \quad (33.b)$$

where $\boldsymbol{H}$ is the transition matrix of the Markov chain and the elements of which are defined in Eq. (28), e.g., the first row of $\boldsymbol{H}$ is given by $[\lambda_{0,0}^{0,0}, \lambda_{0,1}^{0,0}, \ldots, \lambda_{0,Q_2}^{0,0}, \lambda_{1,0}^{0,0}, \ldots, \lambda_{Q_1,Q_2}^{0,0}]$. Hence, we have

$$\boldsymbol{\pi} = \boldsymbol{G}^{-1}\boldsymbol{c}, \quad (34)$$

where $\boldsymbol{G} = \begin{bmatrix} \mathbf{1} \\ \boldsymbol{H} - \boldsymbol{I} \end{bmatrix}$ and $\boldsymbol{c} = [1, \mathbf{0}]^T$. In other words, the transition matrix $\boldsymbol{H}$ determines the steady-distribution $\boldsymbol{\pi}$.

C. Average Latency and Power Consumption

According to the probabilistic scheduling policy defined in Eq. (22), when $\zeta[t] = k_1$, $q_2[t] = k_2$ and $l[t] = n$, the power consumption for the mobile device and the edge server in the t-th time slot is $P_{\rm s}(s, n)$ and $P_{\rm c}(c)$ with the probability $f_{k_1,k_2,n}^{s,c}$, $\forall s \in \mathcal{S}$ and $\forall c \in \mathcal{C}$. The average transmission power consumption at the mobile device is given by

$$P_{\rm s}^{\rm ava} = \sum_{k_2=0}^{Q_2}\sum_{k_1=0}^{Q_1}\sum_{n=1}^{N}\sum_{c=0}^{C}\sum_{s=0}^{S} \pi_{k_1,k_2}\phi_n f_{k_1,k_2,n}^{s,c} P_{\rm s}(s, n). \quad (35)$$

In addition, the average computation power consumption at the edge server is given by

$$P_{\rm c}^{\rm ava} = \sum_{k_2=0}^{Q_2}\sum_{k_1=0}^{Q_1}\sum_{n=1}^{N}\sum_{c=0}^{C}\sum_{s=0}^{S} \pi_{k_1,k_2}\phi_n f_{k_1,k_2,n}^{s,c} P_{\rm c}(c). \quad (36)$$

Moreover, the average queue length $L^{\rm ava}$ is given by

$$\begin{aligned} L^{\rm ava} &= \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T}(q_1[t] + q_2[t]) \\ &= \lim_{t\to\infty} \mathbb{E}\{q_1[t] + q_2[t]\} \\ &= \lim_{t\to\infty} \mathbb{E}\{\zeta[t+1] + q_2[t]\} - \lim_{t\to\infty} \mathbb{E}\{a[t+1]\} \\ &= \sum_{k_1=0}^{Q_1}\sum_{k_2=0}^{Q_2}(k_1 + k_2)\pi_{k_1,k_2} - \lambda_1. \end{aligned} \quad (37)$$

Then, by substituting Eq. (37) into Eq. (16), the average latency can be obtained as

$$\begin{aligned} D^{\rm ava} &= 1 + \frac{\sum_{k_1=0}^{Q_1}\sum_{k_2=0}^{Q_2}(k_1 + k_2)\pi_{k_1,k_2} - \lambda_1}{\lambda_1} \\ &= \frac{1}{\lambda_1}\sum_{k_1=0}^{Q_1}\sum_{k_2=0}^{Q_2}(k_1 + k_2)\pi_{k_1,k_2}. \end{aligned} \quad (38)$$

Based on above analysis of the average transmission power consumptions, the average computation power consumption, and the average latency, we can convert the original optimization problem (19) into an equivalent problem based on the MDP framework.

D. Power-optimal Scheduling based on MDP

Based on the anlaysis of average latency and power consumption in the MDP framework, we have the following optimization problem with $f_{k_1,k_2,n}^{s,c}$ being optimization variables.

$$\min_{f_{k_1,k_2,n}^{s,c}} \sum_{k_2=0}^{Q_2}\sum_{k_1=0}^{Q_1}\sum_{n=1}^{N}\sum_{c=0}^{C}\sum_{s=0}^{S} \pi_{k_1,k_2}\phi_n f_{k_1,k_2,n}^{s,c} P_{\rm s}(s, n) \quad (39.a)$$

$$\text{s.t.} \ \frac{1}{\lambda_1}\sum_{k_1=0}^{Q_1}\sum_{k_2=0}^{Q_2}(k_1 + k_2)\pi_{k_1,k_2} \leq D^{\rm th}, \quad (39.b)$$

$$\sum_{k_2=0}^{Q_2}\sum_{k_1=0}^{Q_1}\sum_{n=1}^{N}\sum_{c=0}^{C}\sum_{s=0}^{S} \pi_{k_1,k_2}\phi_n f_{k_1,k_2,n}^{s,c} P_{\rm c}(c) \leq P_{\rm c}^{\rm th}, \quad (39.c)$$

$$\mathbf{1}\boldsymbol{\pi} = 1, \quad (39.d)$$

$$\boldsymbol{H}\boldsymbol{\pi} = \pi, \quad (39.e)$$

$$\sum_{c=0}^{C}\sum_{s=0}^{S} f_{k_1,k_2,n}^{s,c} = 1 \quad \forall k_1, k_2, n, \quad (39.f)$$

$$f_{k_1,k_2,n}^{s,c} \geq 0 \quad \forall k_1, k_2, n, s, c, \quad (39.g)$$

$$\pi_{k_1,k_2} \geq 0 \quad \forall k_1, k_2, \quad (39.h)$$

where (39.b) and (39.c) denote the constraints of the average latency and the computation capability at the edge server, respectively.

In order to solve this problem, it is necessary to realize that the objective function and the constraints in the problem (39) are linear combinations of $\{\pi_{k_1,k_2}\phi_n f_{k_1,k_2,n}^{s,c}\}$, $\{f_{k_1,k_2,n}^{s,c}\}$, or $\{\pi_{k_1,k_2}\}$. Thus, we define a new variable as

$$\begin{aligned} \mathbf{y} = \{y_{k_1,k_2,n}^{s,c} | k_1 \in \{0, 1, \ldots, Q_1\}, k_2 \in \{0, 1, \ldots, Q_2\}, \\ n \in \{1, \ldots, N\}, s \in \mathcal{S}, c \in \mathcal{C}\}, \end{aligned} \quad (40)$$

where

$$y_{k_1,k_2,n}^{s,c} = \pi_{k_1,k_2}\phi_n f_{k_1,k_2,n}^{s,c}, \quad (41)$$

with the normalization constraint:

$$\sum_{k_2=0}^{Q_2}\sum_{k_1=0}^{Q_1}\sum_{n=1}^{N}\sum_{c=0}^{C}\sum_{s=0}^{S} y_{k_1,k_2,n}^{s,c} = \sum_{k_2=0}^{Q_2}\sum_{k_1=0}^{Q_1}\pi_{k_1,k_2}\sum_{n=1}^{N}\phi_n = 1. \quad (42)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2020.2979136, IEEE Transactions on Wireless Communications

The size of $\mathbf{y}$ is $(Q_1+1)(Q_2+1)N(S+1)(C+1)$. For the ease of presentation, we denote $y^{s,c}_{k_1,k_2,n}$ as the entry at position $\{(Q_2+1)(S+1)(C+1)N\cdot k_1+(S+1)(C+1)N\cdot k_2+(S+1)(C+1)\cdot(n-1)+(C+1)\cdot s+c+1\}$ in $\mathbf{y}$, e.g., $y^{0,0}_{0,0,1}$ and $y^{S,C}_{Q_1,Q_2,N}$ are the first and the last elements, respectively.

By recalling the normalization constraints in Eqs. (7) and (23), π_{k_1,k_2} can be expressed as

$$\pi_{k_1,k_2}=\sum_{n=1}^{N}\sum_{c=0}^{C}\sum_{s=0}^{S}\pi_{k_1,k_2}\phi_n f^{s,c}_{k_1,k_2,n}=\sum_{n=1}^{N}\sum_{c=0}^{C}\sum_{s=0}^{S}y^{s,c}_{k_1,k_2,n}. \tag{43}$$

Moreover, constraint (39.f) can be rewritten as:

$$\sum_{k_2=0}^{Q_2}\sum_{k_1=0}^{Q_1}\sum_{c=0}^{C}\sum_{s=0}^{S}y^{s,c}_{k_1,k_2,n}=\sum_{k_2=0}^{Q_2}\sum_{k_1=0}^{Q_1}\pi_{k_1,k_2}\phi_n=\phi_n, \tag{44}$$

where $\sum\limits_{n=0}^{N}\phi_n=1$.

By replacing optimization variables $f^{s,c}_{k_1,k_2,n}$ with $y^{s,c}_{k_1,k_2,n}$, constraints (39.d) and (39.e) can be expressed as a matrix equation $\mathbf{Q}\mathbf{y}=0$, where the constraint (39.e) can be rewritten as:

$$\begin{aligned}
&\pi_{q_1,q_2}\\
=&\sum_{k_2=0}^{Q_2}\sum_{k_1=0}^{Q_1}\pi_{k_1,k_2}\lambda^{q_1,q_2}_{k_1,k_2}\\
=&\sum_{k_2=0}^{Q_2}\sum_{k_1=0}^{Q_1}\sum_{n=1}^{N}\sum_{m=0}^{M}\pi_{k_1,k_2}p_m\phi_n f^{s(m),c(m)}_{k_1,k_2,n}\mathbb{1}_{\{s(m)\in\mathcal{S},c(m)\in\mathcal{C}\}}\\
=&\sum_{k_2=0}^{Q_2}\sum_{k_1=0}^{Q_1}\sum_{n=1}^{N}\sum_{m=0}^{M}p_m y^{s(m),c(m)}_{k_1,k_2,n}\mathbb{1}_{\{s(m)\in\mathcal{S},c(m)\in\mathcal{C}\}}\\
=&\sum_{n=1}^{N}\sum_{c=0}^{C}\sum_{s=0}^{S}y^{s,c}_{q_1,q_2,n},
\end{aligned} \tag{45}$$

where the last line is obtained from Eq. (43). Then, based on the last two parts of Eq. (45), the matrix $\mathbf{Q}$ can be obtained. In this way, the problem (39) is converted into linear programming which is summarized in the following theorem.

Theorem 2. *The problem (39) is equivalent to the following linear programming.*

$$\min_{y^{s,c}_{k_1,k_2,n}}\sum_{k_2=0}^{Q_2}\sum_{k_1=0}^{Q_1}\sum_{n=1}^{N}\sum_{c=0}^{C}\sum_{s=0}^{S}P_{\rm s}(s,n)y^{s,c}_{k_1,k_2,n} \tag{46.a}$$

$$\text{s.t. }\frac{1}{\lambda_1}\sum_{k_1=0}^{Q_1}\sum_{k_2=0}^{Q_2}\sum_{n=1}^{N}\sum_{c=0}^{C}\sum_{s=0}^{S}(k_1+k_2)y^{s,c}_{k_1,k_2,n}\le D^{\rm th}, \tag{46.b}$$

$$\sum_{k_2=0}^{Q_2}\sum_{k_1=0}^{Q_1}\sum_{n=1}^{N}\sum_{c=0}^{C}\sum_{s=0}^{S}P_{\rm c}(c)y^{s,c}_{k_1,k_2,n}\le P^{\rm th}_{\rm c}, \tag{46.c}$$

$$\sum_{k_2=0}^{Q_2}\sum_{k_1=0}^{Q_1}\sum_{c=0}^{C}\sum_{s=0}^{S}y^{s,c}_{k_1,k_2,n}=\phi_n, \tag{46.d}$$

$$\mathbf{Q}\mathbf{y}=0, \tag{46.e}$$

$$y^{s,c}_{k_1,k_2,n}\ge 0\quad\forall k_1,k_2,n,s,c, \tag{46.f}$$

where $\mathbf{y}$ is a column vector with $y^{s,c}_{k_1,k_2,n}$ as components.

Proof: Firstly, we proceed to prove each component in those two problems can be converted equivalently. Define $y^{s,c}_{k_1,k_2,n}=\{\pi_{k_1,k_2}\phi_n f^{s,c}_{k_1,k_2,n}\}$ and π_{k_1,k_2} can be expressed as $\pi_{k_1,k_2}=\sum_{n=1}^{N}\sum_{c=0}^{C}\sum_{s=0}^{S}y^{s,c}_{k_1,k_2,n}$. Therefore, each component in the problem (39) can be converted into the corresponding form in the problem (46). Then we need to prove that all the feasible solution of those two problems are bijective. For each feasible solution π_{k_1,k_2} and $f^{s,c}_{k_1,k_2,n}$ in the problem (39), $y^{s,c}_{k_1,k_2,n}=\{\pi_{k_1,k_2}\phi_n f^{s,c}_{k_1,k_2,n}\}$ is still feasible to the problem (46). For each feasible solution $y^{s,c}_{k_1,k_2,n}$ of problem (46), the corresponding solution of the problem (39) can be obtained by $\pi_{k_1,k_2}=\sum_{n=1}^{N}\sum_{c=0}^{C}\sum_{s=0}^{S}y^{s,c}_{k_1,k_2,n}$. Moreover, another part of feasible solution of the problem (39), i.e., $f^{s,c}_{k_1,k_2,n}$, is considered in the following Eqs. (48) and (49).

Therefore, all the feasible solutions of those two problems can be converted equivalently, i.e., bijective, and thus they can be converted equivalently. ■

The converted problem (46) can be solved efficiently in polynomial time using interior-point method [28]. After the optimal solution ${y^{s,c}_{k_1,k_2,n}}^*$ of the linear programming (46) is obtained, the corresponding steady-state distribution can be represented as

$$\pi^*_{k_1,k_2}=\sum_{n=1}^{N}\sum_{c=0}^{C}\sum_{s=0}^{S}{y^{s,c}_{k_1,k_2,n}}^*. \tag{47}$$

To obtain the power-optimal scheduling JTCS policy, we can derive ${f^{s,c}_{k_1,k_2,n}}^*$ from ${y^{s,c}_{k_1,k_2,n}}^*$, which is summarized below.

Case 1. *When $\pi^*_{k_1,k_2}\neq 0$, the optimal policy is given by*

$${f^{s,c}_{k_1,k_2,n}}^*=\frac{{y^{s,c}_{k_1,k_2,n}}^*}{\phi_n\pi^*_{k_1,k_2}}. \tag{48}$$

Case 2. *When $\pi^*_{k_1,k_2}=0$, which means that the state (k_1,k_2) is a transient state. Denote $\mathcal{A}$ as the feasible set of action (s,c) where $s\in\mathcal{S}$ and $c\in\mathcal{C}$, satisfying constraints (24) and (25). Then, a simple policy can be used, i.e.,*

$${f^{s,c}_{k_1,k_2,n}}^*=\frac{1}{|\mathcal{A}|},\quad\forall(s,c)\in\mathcal{A}, \tag{49}$$

where $|\mathcal{A}|$ is the cardinality of $\mathcal{A}$.

Since the time complexity of deriving ${f^{s,c}_{k_1,k_2,n}}^*$ from ${y^{s,c}_{k_1,k_2,n}}^*$ is also polynomial, the power-optimal JTCS policy can be obtained through the MDP-based approach in polynomial time. It is worth emphasizing that this MDP-based approach is an offline algorithm which only needs to be optimized for once before the offloading rather not in each time slot. Then, based on this policy, the optimal power-latency tradeoff can be achieved.

V. Scheduling with Lyapunov Drift-plus-penalty Approach

In this section, we focus the design of JTCS policy with unknown distribution information of the task arrivals and the channel state variations. To solve this problem, an online algorithm is proposed through a Lyapunov drift-plus-penalty approach since it can operate without requiring distribution information.

A. Lyapunov Drift-plus-penalty Approach

Intuitively, more computation power consumed by the edge server will provide higher computation rates, which will not affect the transmission of the mobile device while reducing the average delay. Thus, the inequality constraint (19.c) can be regarded as an equality constraint $P_{\rm c}^{\rm ava} = P_{\rm c}^{\rm th}$. Similar to [29], in order to satisfy this constraint, we define a virtual queue of computation power consumption, denoted by $q_3[t]$, whose initial value $q_3[0] = 0$ and updating equation is given by

$$q_3[t] = q_3[t-1] - P_{\rm c}^{\rm th} + P_{\rm c}[t]. \tag{50}$$

If virtual queue $q_3[t]$ is stable, its input rate $\lim\limits_{T\to\infty} \frac{1}{T}\sum\limits_{t=1}^{T} P_{\rm c}[t]$ will be less than its output rate $P_{\rm c}^{\rm th}$ [30]. In this way, the constraint of the computation capability in (19.c) holds, which is more effective than by optimizing the weighted sum of $P_{\rm s}^{\rm ava}$ and $P_{\rm c}^{\rm ava}$ in [13].

Based on this virtual queue, we define the Lyapunov drift function of the problem (19) as

$$\Delta L[t] = \frac{1}{2}\mathbb{E}\left\{\sum_{i=1}^{3}(q_i[t]^2 - q_i[t-1]^2)|\mathbf{Q}[t-1]\right\}, \tag{51}$$

where $\mathbf{Q}[t-1] = \{q_1[t-1], q_2[t-1], q_3[t-1]\}$. The expectation depends on the scheduling policy and is with respect to the random channel states and the control actions made in reaction to these channel states. The quadratic function is implemented since it has important dominant cross terms that include an inner product of the current queue lengths and rates of transmission and computation. The Lyapunov drift-plus-penalty function can be defined as $\Delta L[t] + V\mathbb{E}\{P_{\rm s}(s[t], l[t])|\mathbf{Q}[t-1]\}$, where the penalty parameter $V \geq 0$ is set to balance the trade-off between the average latency and the average transmission power consumption. Intuitively, from Lyapunov drift analysis, with the increase of V, the objective of the optimization can be arbitrarily close to optimal average transmission power consumption, while with the increasing cost on the average latency. When V is very large, the queues might be non-convergence, resulting in the average delay is infinite.

From Eqs. (4) and (5), we have the Lyapnuov drifts of $q_1[t]$ and $q_2[t]$ satisfying

$$\frac{q_1[t]^2 - q_1[t-1]^2}{2} \leq \frac{M^2 + S^2}{2} + q_1[t-1](a[t] - s[t]), \tag{52}$$

and

$$\frac{q_2[t]^2 - q_2[t-1]^2}{2} \leq \frac{S^2 + C^2}{2} + q_2[t-1](s[t] - c[t]), \tag{53}$$

since $(\max\{Z - Y, 0\} + X)^2 \leq Z^2 + X^2 + Y^2 + 2Z(X - Y)$ holds for any X, Y, and $Z \geq 0$. In addition, from Eq. (50), we also have the Lyapnuov drift of the virtual queue $q_3[t]$ satisfying

$$\frac{q_3[t]^2 - q_3[t-1]^2}{2} \leq \frac{{P_{\rm c}^{\rm th}}^2 + {P_{\rm c}^{\rm m}}^2}{2} + q_3[t-1](P_{\rm c}[t] - P_{\rm c}^{\rm th}), \tag{54}$$

where $P_{\rm c}^{\rm m} = P_{\rm c}(C) \geq P_{\rm c}[t]$. Then the Lyapunov drift-plus-penalty function is bounded by

$$\begin{aligned} &\Delta L[t] + V\mathbb{E}\{P_{\rm s}(s[t], l[t])|\mathbf{Q}[t-1]\} \\ \leq& B_{\max} + \mathbb{E}\{q_1[t-1](a[t] - s[t]) + q_2[t-1](s[t] - c[t]) \\ &+ q_3[t-1](P_{\rm c}[t] - P_{\rm c}^{\rm th}) + VP_{\rm s}(s[t], l[t])|\mathbf{Q}[t-1]\}, \end{aligned} \tag{55}$$

where

$$B_{\max} = \frac{M^2 + C^2 + {P_{\rm c}^{\rm th}}^2 + {P_{\rm c}^{\rm m}}^2}{2} + S^2. \tag{56}$$

From the above bound in Eq. (55), we can develop an online algorithm as summarized in Algorithm 1 to greedily minimize the upper bound of the Lyapunov drift-plus-penalty function given the queue lengths $\mathbf{Q}[t-1]$ and the current channel state $l[t]$. Note that the algorithm only uses the current system states $a[t]$, $l[t]$, and $\mathbf{Q}[t-1]$, while does not require any distribution information of task arrival and channel state. This algorithm is based on a heuristic approach when infinite buffers are considered, while it can approach to the original drift-plus-penalty method in [30] when the buffer sizes are large sufficiently.

Algorithm 1 A heuristic scheduling policy

1: Input current system state: V, $a[t]$, $l[t]$, $q_1[t-1]$, $q_2[t-1]$ and $q_3[t-1]$.
2: Define available transmission decision set $\mathcal{S}_{\rm a} = \{x | 0 \leq q_1[t-1] + a[t] - x \leq Q_1 - M, x \in \mathcal{S}\}$.
3: Define available computation decision set $\mathcal{C}_{\rm a} = \{y | 0 \leq q_2[t-1] + s[t] - y \leq Q_2, y \in \mathcal{C}\}$.
4: Choose the control decisions $s[t] \in \mathcal{S}_{\rm a}$ and $c[t] \in \mathcal{C}_{\rm a}$ which can minimize the upper bound of Lyapunov drift-plus-penalty function defined in the right hand side of Eq. (55).
5: Update queue lengths for next time slot, i.e., $q_1[t]$, $q_2[t]$, and $q_3[t]$.
6: Output $s[t]$ and $c[t]$.

Notice that, the objective of the problem (19) is to optimize the average transmission power consumption, while the average latency constraint must be guaranteed before, rather not just keep the queues stable [13]. To ensure the average latency constraint holds, the penalty parameter V should be adjusted over time, regarded as $V[t]$, since it plays the role to balance the average transmission power and latency. Hence, by sampling the average latency over a period of time, we can update $V[t]$ according to the sampling results. For instance, $V[t]$ can be updated through a dichotomy method, i.e., $V_{\min} \leftarrow V[t]$ and $V[t] \leftarrow \frac{V[t]+V_{\max}}{2}$ if the sampled average latency is less than $D^{\rm th}$; otherwise $V_{\max} \leftarrow V[t]$ and $V[t] \leftarrow \frac{V[t]+V_{\min}}{2}$. After several updates, the penalty parameter $V[t]$ can converge to a constant, and then the scheduling policy can be obtained according to Algorithm 1. With the scheduling policy, the average delay and transmission power consumption can be counted based on Eqs. (16) and (17).

B. Performance Analysis

For analyzing the upper bound of the optimization performance, we consider an ideal scenario that the buffer size Q_1 and Q_2 are sufficiently large such that the buffer overflow can be negligible. For convenience, we assume that the buffer sizes are infinite, i.e., $Q_1 = Q_2 = \infty$. Therefore, in Algorithm 1, steps 3 and 4 can be relaxed to $\mathcal{S}_{\mathrm{a}} = \mathcal{S}$ and $\mathcal{C}_{\mathrm{a}} = \mathcal{C}$. Then, according to [30] (Definition 2.3 and Theorem 2.4), we have the following theorem and definition.

Theorem 3. (Rate Stability Theorem) The queues $q_1[t]$ and $q_2[t]$ are rate stability, i.e.,

$$\lim_{T\to\infty} \frac{q_1[T] + q_2[T]}{T} = 0, \tag{57}$$

if and only if $\lambda_1 \le \min\{\mu_1, \mu_2\}$, where $\mu_1 = \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} s[t]$ and $\mu_2 = \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} c[t]$.

Definition 1. Define $\lambda_{\max} = \min\{\mu_1, \mu_2\}$ as the feasible region of the arrival rate λ_1 defined in Eq. (2), where there always exists a scheduling policy that can keep queues $q_1[t]$ and $q_2[t]$ rate stable if $\lambda_1 \le \lambda_{\max}$.

Since we do not consider the transmission power consumption in the above definition, the transmission rate $s[t]$ can be scheduled at the highest rate S in each time slot and thus $\mu_1 = S \ge M \ge \lambda_1$ holds. Then, the feasible region of the arrival rate can be rewritten as $\lambda_{\max} = \mu_2$, which is the optimal solution of the following problem, i.e.,

$$\begin{aligned}
\mu_2 = \max_{g(c)} \quad & \sum_{c\in\mathcal{C}} c \cdot g(c) && (58.a)\\
\text{s.t.} \quad & \sum_{c\in\mathcal{C}} P_{\mathrm{c}}(c)\cdot g(c) \le P_{\mathrm{c}}^{\mathrm{th}} && (58.b)\\
& \sum_{c\in\mathcal{C}} g(c) = 1 && (58.c)\\
& g(c) \ge 0, \forall c \in \mathcal{C}, && (58.d)
\end{aligned}$$

where the optimization variable $g(c)$ is the probability that computation rate is chosen as c at one time slot. This optimization problem is linear programming and can also be solved to obtain the feasible region $\lambda_{\max} = \mu_2$, which is the maximal arrival rate that can be supported by the edge server to keep the queue $q_2[t]$ stable under the power constraint in (58.b).

Furthermore, denote $\Psi(\lambda_1)$ as the minimal average transmission power consumption that can be achieved without the average latency constraint, while guaranteeing the queue $q_1[t]$ stable by any feasible control policy. It is the optimal value of the objective function in the following optimization problem, i.e.,

$$\begin{aligned}
\Psi(\lambda_1) = \min_{g_{c,n}} \quad & \sum_{n=1}^{N}\sum_{s\in\mathcal{S}} \phi_n P_{\mathrm{s}}(s,n)\cdot f(s,n) && (59.a)\\
\text{s.t.} \quad & \sum_{n=1}^{N}\sum_{s\in\mathcal{S}} \phi_n s\cdot f(s,n) > \lambda_1 && (59.b)\\
& \sum_{s\in\mathcal{S}} f(s,n) = 1, \forall n \in \{1,\cdots,N\} && (59.c)\\
& f(s,n) \ge 0, \forall s\in\mathcal{S}, \forall n \in \{1,\cdots,N\}, && (59.d)
\end{aligned}$$

where the optimization variable $f(s,n)$ is the probability that the transmission rate is chosen as $s[t] = s$ when the channel state $l[t] = n$ at one time slot. Likewise, this problem is also linear programming and can be solved efficiently. Based on the above analysis and results, we have the following theorem.

Theorem 4. Assume that the task arrival rate is within the feasible region, i.e., $\lambda_1 \le \lambda_{\max}$, and the online scheduling decision algorithm in Algorithm 1 is applied in each time slot. For any control parameter $V > 0$, the average transmission power consumption $P_{\mathrm{s}}^{\mathrm{ava}}$ and average latency D^{ava} can be bounded as

$$P_{\mathrm{s}}^{\mathrm{ava}} \le \Psi(\lambda_1) + \frac{B_{\max}}{V}, \tag{60}$$

and

$$\begin{aligned}
D^{\mathrm{ava}} &\le \frac{B_{\max} + V(\Psi(\lambda_{\max}) - \Psi(\lambda_1))}{\lambda_1(\lambda_{\max} - \lambda_1)} + 1\\
&\le \frac{B_{\max}}{\lambda_1(\lambda_{\max} - \lambda_1)} + \frac{V P_{\mathrm{s}}(S,N)}{\lambda_1 S} + 1.
\end{aligned} \tag{61}$$

Proof: Since $\lambda_1 < \lambda_{\max}$, there exists one scheduling policy algorithm $\{s^*[t], c^*[t]\}$ depends on the current channel state $l[t]$ can keep all queues rate stable, satisfying that:

$$\mu_1 = \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} s^*[t] \ge \lambda_1 + \varepsilon, \tag{62}$$

and

$$\mu_2 = \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} c^*[t] \ge \lambda_1 + \varepsilon, \tag{63}$$

and

$$P_{\mathrm{s}}^{\mathrm{ava}} = \mathbb{E}\{P_{\mathrm{s}}(s^*[t], l[t])\} = \Psi(\lambda_1 + \varepsilon), \tag{64}$$

where ε is a non-negative constant [30].

Since the proposed online scheduling algorithm in Algorithm 1 always attempts to greedily minimize the right-hand-side of Eq. (55) for each time slot over all possible feasible decision, we have the following inequality, which is given by

$$\begin{aligned}
&\Delta L[t] + V\mathbb{E}\{P_{\mathrm{s}}(s[t], l[t])|\mathbf{Q}[t-1]\}\\
\le& B_{\max} + V\mathbb{E}\{P_{\mathrm{s}}(s^*[t])\} + q_1[t-1](\lambda_1 - \mathbb{E}\{s^*[t]\})\\
&+ q_2[t-1](\lambda_1 - \mathbb{E}\{c^*[t]\}) + q_3[t-1](\mathbb{E}\{P_{\mathrm{c}}(c^*[t])\} - P_{\mathrm{c}}^{\mathrm{th}}),
\end{aligned} \tag{65}$$

where the left-hand-side of this inequality is the Lyapunov drift-plus-penalty function obtained by the scheduling decisions based on Algorithm 1. By plugging Eqs. (62), (63), and

(64), and the constraint $\mathbb{E}\{P_c(c^*[t])\} \le P_c^{\text{th}}$ into the right-hand-side of the above inequality, we have

$$\begin{aligned}&\Delta L[t] + V\mathbb{E}\{P_s(s[t], l[t])|\mathbf{Q}[t-1]\}\\ \le& B_{\max} + V\Psi(\lambda_1+\varepsilon) - \varepsilon(q_1[t-1]+q_2[t-1]).\end{aligned} \tag{66}$$

Taking expectations of the above inequality and using the law of iterated expectations, we have

$$\begin{aligned}&L[t] - L[t-1] + V\mathbb{E}\{P_s(s[t], l[t])\}\\ \le& B_{\max} + V\Psi(\lambda_1+\varepsilon) - \varepsilon\mathbb{E}\{q_1[t-1]+q_2[t-1]\},\end{aligned} \tag{67}$$

where $L[t] = \frac{1}{2}\mathbb{E}\left\{\sum_{i=1}^3 q_i[t]^2\right\}$. By summing the above inequality over T time slots, we have

$$\begin{aligned}&L[T] - L[0] + V\sum_{t=1}^{T}\mathbb{E}\{P_s(s[t], l[t])\}\\ \le& B_{\max}T + VT\Psi(\lambda_1+\varepsilon) - \varepsilon\sum_{t=1}^{T}\mathbb{E}\{q_1[t-1]+q_2[t-1]\}.\end{aligned} \tag{68}$$

Thus, the following two inequalities can be obtained by

$$\Psi(\lambda_1) \le \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\{P_s(s[t], l[t])\} \le \Psi(\lambda_1+\varepsilon) + \frac{B_{\max}}{V} + \frac{L[0]}{VT}, \tag{69}$$

and

$$\begin{aligned}&\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\{q_1[t-1]+q_2[t-1]\}\\ \le& \frac{B_{\max} + V(\Psi(\lambda_1+\varepsilon) - \Psi(\lambda_1))}{\varepsilon} + \frac{L[0]}{\varepsilon T}.\end{aligned} \tag{70}$$

Then, by taking $T \to \infty$, we have

$$P_s^{\text{ava}} = \lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\{(P_s(s[t], l[t])\} \le \Psi(\lambda_1) + \frac{B_{\max}}{V}, \tag{71}$$

where ε is set as 0, and

$$\begin{aligned}L^{\text{ava}} =& \frac{1}{T}\lim_{T\to\infty}\sum_{t=1}^{T}(q_1[t-1]+q_2[t-1])\\ \le& \frac{B_{\max}}{\lambda_{\max}-\lambda_1} + \frac{V(\Psi(\lambda_{\max}) - \Psi(\lambda_1))}{\lambda_{\max}-\lambda_1},\end{aligned} \tag{72}$$

where ε is set as $\lambda_{\max} - \lambda_1$ to satisfy $\lambda_1 + \varepsilon \le \lambda_{\max}$. Furthermore, we have

$$\Psi(\lambda_{\max}) - \Psi(\lambda_1) \le (\lambda_{\max} - \lambda_1)\frac{P_s(S,N)}{S}, \tag{73}$$

where the inequality holds because it requires at most $\frac{P_s(S,N)}{S}$ units of power to transmit each task according to Eq. (9). By substituting Eq. (73) into Eq. (72), we have

$$\begin{aligned}D^{\text{ava}} =& \frac{L^{\text{ava}}}{\lambda_1} + 1\\ \le& \frac{B_{\max}}{\lambda_1(\lambda_{\max}-\lambda_1)} + \frac{V(\lambda_{\max}-\lambda_1)\frac{P_s(S,N)}{S}}{\lambda_1(\lambda_{\max}-\lambda_1)} + 1\\ =& \frac{B_{\max}}{\lambda_1(\lambda_{\max}-\lambda_1)} + \frac{VP_s(S,N)}{\lambda_1 S} + 1.\end{aligned} \tag{74}$$

Then the theorem is proved. ■

VI. Numerical Results

In this section, we will validate our theoretical results by the simulation studies. Throughout this section, we set $S = 2$, $C = 2$, $M = 2$, and $N = 3$. Power consumption functions defined in Eqs. (8), (9), and (14) are assumed as $P_s(s,n) = \frac{n(2^s-1)}{10}$ and $P_c(c) = \frac{c^3}{5}$. The distributions of task arrivals and channel state variations are summarized in Table II. Based on this, the average arrival rate can be calculated as $\lambda_1 = \sum_{m=1}^{M} p_m \cdot m = 0.75$. Based on the optimization problem (46) and Algorithm 1, the scheduling policy with and without distributions of task arrivals and channel state variations can be obtained. Each simulation result runs 10^7 time slots.

TABLE II
Simulation Parameter Settings.

m	0	1	2	n	1	2	3
p_m	0.5	0.25	0.25	ϕ_n	0.25	0.25	0.5

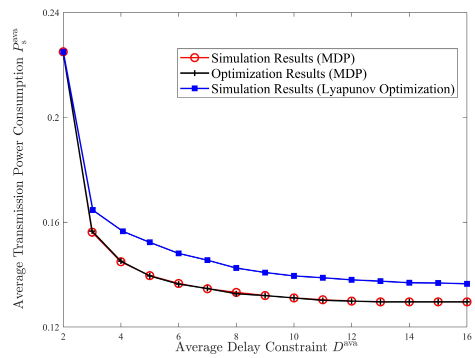


Fig. 4. Comparison of Optimization and Simulation Results.

The average transmission power consumption achieved by the MDP-based approach and the Lyapunov drift-plus-penalty approach are compared in Fig. 4, where the constraint of the computation capability at the edge server $P_c^{\text{th}} = 1.25$ and buffer sizes $Q_1 = Q_2 = 20$. For the MDP-based approach, the optimization results are obtained by Eqs. (35) and (38) via linear programming in (46), while the simulation results are given by Eqs. (16) and (17) via Monte-Carlo simulation, where they match perfectly well. For the Lyapunov drift-plus-penalty approach, the penalty parameter V should be adjusted to meet each latency constraint $D^{\text{ava}} \le D^{\text{th}}$ first. Then, with the obtained stationary value of V, the simulation results of the Lyapunov drift-plus-penalty approach are also obtained via Monte-Carlo simulation. With the increase of the average latency constraint D^{th}, the average transmission power consumptions achieved by two approaches decrease significantly. Moreover, by making full use of the distribution information of task arrivals and channel state variations, the MDP-based approach can achieve lower average transmission power consumption compared to the Lyapunov drift-plus-penalty approach.

The optimal tradeoff among the average latency, transmission power consumption, and computation power consumption

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2020.2979136, IEEE Transactions on Wireless Communications

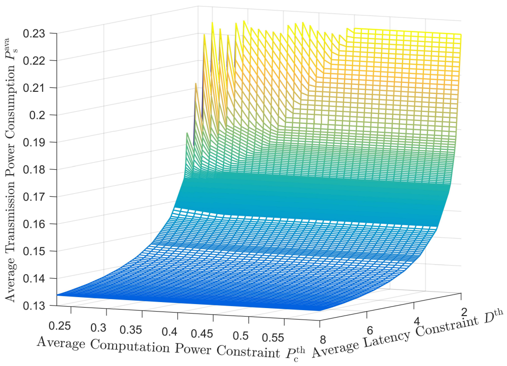


Fig. 5. Optimal Power-Latency Tradeoff.

with buffer sizes $Q_1 = Q_2 = 20$ achieved by MDP-based approach is shown in Fig. 5. As it is expected, the average transmission power consumption decreases when both the average latency constraint and the computation capability constraint increase, which is also matched with the results in Fig. 4. When the latency constraint D^{th} are small enough, the mobile device always transmits all arriving tasks under any channel state without waiting and, hence, the average transmission power consumption approaches to the maximal value, i.e., $\sum_{m=0}^{M} \sum_{n=1}^{N} \phi_n P_{\text{s}}(m, n) = 0.225$. Moreover, when $P_{\text{s}}^{\text{ava}}$ and P_{c}^{th} are large enough, the tasks can be transmitted as soon as each task arrives, i.e., the tasks can be transmitted as soon as each task arrivals, i.e., the transmission rate is $\min\{\zeta[t], S\}$, and then be computed at the rate $\min\{q_2[t], C\}$ in each time slot. As a result, the average latency is two time slots.

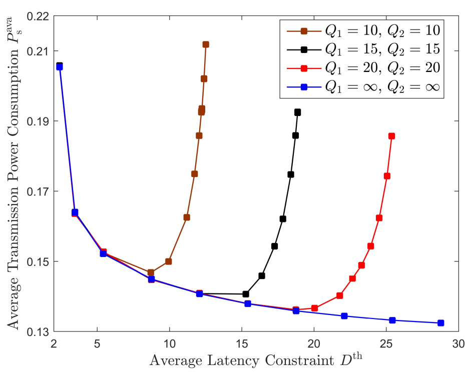


Fig. 6. Results of the Lyapunov Drift-plus-penalty Approach with Different Buffer Sizes.

The simulation results for the Lyapunov drift-plus-penalty approach with different buffer sizes $Q_1 = Q_2 = 10$, 15, 20, and ∞, respectively, are shown in Fig. 6, where the constraint of the computation capability at the edge server $P_{\text{c}}^{\text{th}} = 1.25$. With the increase of buffer sizes, the Lyapunov drift-plus-penalty approach can achieve a better performance, which matches with the theoretical results. With the increasing of penalty parameter V, the mobile device will prefer to transmit tasks at a lower transmission rate at the time slots with good channel state to reduce power consumption. When V is large enough, the average transmission rate at the early time will be less than the average arrival rate λ_1, which will lead to a full buffer. In this way, the mobile device with full buffer state has to transmit all arriving tasks at any channel states to avoid overflow and then the average transmission power will rise up with the increase of V, which is also matched with the simulation results in Fig. 6.

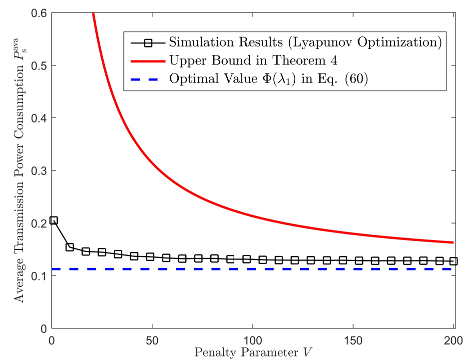


Fig. 7. Average Transmission Power Consumption $P_{\text{s}}^{\text{ava}}$ versus Penalty Parameter V.

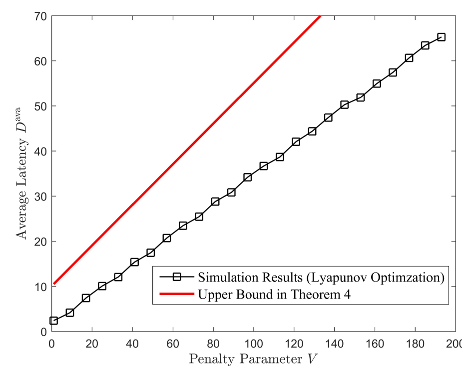


Fig. 8. Average Latency D^{ava} versus Penalty Parameter V.

The simulation results and theoretical upper bounds of the average transmission power consumption $P_{\text{s}}^{\text{ava}}$ and average latency D^{ava} through the Lyapunov drift-plus-penalty approach are shown in Fig. 7 and Fig. 8, respectively. The constraint of the computation capability at the edge server is given by $P_{\text{c}}^{\text{th}} = 1.25$. In Fig. 7, with the increase of the penalty parameter V, both simulation results and upper bound of $P_{\text{s}}^{\text{ava}}$ approach to the optimal value $\Phi(\lambda_1)$ in Eq. (60). In Fig. 8, both of the simulation results and upper bound of D^{ava} increases with the increasing of V. Thus, both the average transmission power consumption $P_{\text{s}}^{\text{ava}}$ and average latency D^{ava} can be bounded well with different V and thus the theoretical results in Theorem 4 can be validated.

VII. Conclusion

In this paper, we have investigated the power-optimal scheduling policies to achieve the optimal power-latency trade-off in MEC systems by considering the system dynamics in different layers. In particular, we have formulated a scheduling problem to minimize the average power consumption at the mobile device under the constraints of the average latency of tasks and the computation capability at the edge server. With known distribution information of the system dynamics, the scheduling problem can be formulated into an MDP framework to analyze the average latency and power consumptions. By solving this problem, the power-optimal scheduling policy is obtained and thus the optimal power-latency tradeoff can be achieved. When distribution information of the system dynamics is unknown, we have also presented an online algorithm based on a Lyapunov drift-plus-penalty approach to optimize the scheduling decisions in each time slot. Through theoretical analysis, the power-latency tradeoff based on the Lyapunov drift-plus-penalty approach can also be found. Furthermore, an analytic bound on the performance can be obtained when the buffer sizes are sufficiently large. Important future topics include considering the worst-case latency as the latency constraint rather than the average latency.

References

[1] D. Han, W. Chen, and Y. Fang, "Power-optimal scheduling for delay constrained mobile computation offloading," in *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–6.

[2] R. Kemp, N. Palmer, T. Kielmann, and H. Bal, "Cuckoo: a computation offloading framework for smartphones," in *International Conference on Mobile Computing, Applications, and Services*. Springer, 2010, pp. 59–79.

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

[4] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing: A key technology towards 5G," *ETSI white paper*, vol. 11, no. 11, pp. 1–16, 2015.

[5] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, and W. Wang, "A survey on mobile edge networks: Convergence of computing, caching and communications," *IEEE Access*, vol. 5, pp. 6757–6779, 2017.

[6] K. Kumar, J. Liu, Y.-H. Lu, and B. Bhargava, "A survey of computation offloading for mobile systems," *Mobile Networks and Applications*, vol. 18, no. 1, pp. 129–140, 2013.

[7] Z. Li, C. Wang, and R. Xu, "Computation offloading to save energy on handheld devices: a partition scheme," in *Proceedings of the 2001 international conference on Compilers, architecture, and synthesis for embedded systems*. ACM, 2001, pp. 238–246.

[8] 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.

[9] L. Lei, Z. Zhong, K. Zheng, J. Chen, and H. Meng, "Challenges on wireless heterogeneous networks for mobile cloud computing," *IEEE Wireless Communications*, vol. 20, no. 3, pp. 34–44, 2013.

[10] D. Han, W. Chen, B. Bai, and Y. Fang, "Offloading optimization and bottleneck analysis for mobile cloud computing," *IEEE Transactions on Communications*, 2019.

[11] S. Sardellitti, G. Scutari, and S. Barbarossa, "Joint optimization of radio and computational resources for multicell mobile-edge computing," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 1, no. 2, pp. 89–103, 2015.

[12] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795–2808, 2015.

[13] Y. Mao, J. Zhang, S. Song, and K. B. Letaief, "Stochastic joint radio and computational resource management for multi-user mobile-edge computing systems," *IEEE Transactions on Wireless Communications*, vol. 16, no. 9, pp. 5994–6009, 2017.

[14] S. Bi and Y. J. Zhang, "Computation rate maximization for wireless powered mobile-edge computing with binary computation offloading," *IEEE Transactions on Wireless Communications*, vol. 17, no. 6, pp. 4177–4190, 2018.

[15] 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.

[16] O. Munoz, 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, 2014.

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

[18] S. Yu, R. Langar, X. Fu, L. Wang, and Z. Han, "Computation offloading with data caching enhancement for mobile edge computing," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, pp. 11 098–11 112, 2018.

[19] F. Wang, J. Xu, X. Wang, and S. Cui, "Joint offloading and computing optimization in wireless powered mobile-edge computing systems," *IEEE Transactions on Wireless Communications*, vol. 17, no. 3, pp. 1784–1797, 2017.

[20] C.-F. Liu, M. Bennis, M. Debbah, and H. V. Poor, "Dynamic task offloading and resource allocation for ultra-reliable low-latency edge computing," *IEEE Transactions on Communications*, vol. 67, no. 6, pp. 4132–4150, 2019.

[21] B. Collins and R. L. Cruz, "Transmission policies for time varying channels with average delay constraints," in *Proceedings of the Annual Allerton Conference on Communication Control and Computing*, vol. 37. The University; 1998, 1999, pp. 709–717.

[22] R. A. Berry and R. G. Gallager, "Communication over fading channels with delay constraints," *IEEE Transactions on Information theory*, vol. 48, no. 5, pp. 1135–1149, 2002.

[23] W. Xu, Y. Zhang, Q. Shi, and X. Wang, "Energy management and cross layer optimization for wireless sensor network powered by heterogeneous energy sources," *IEEE Transactions on Wireless Communications*, vol. 14, no. 5, pp. 2814–2826, 2015.

[24] W. Chen, Z. Cao, and K. B. Letaief, "Optimal delay-power tradeoff in wireless transmission with fixed modulation," in *2007 International Workshop on Cross Layer Design*. IEEE, 2007, pp. 60–64.

[25] X. Chen, W. Chen, J. Lee, and N. B. Shroff, "Delay-optimal buffer-aware scheduling with adaptive transmission," *IEEE Transactions on Communications*, vol. 65, no. 7, pp. 2917–2930, 2017.

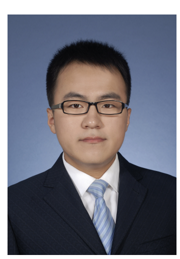
[26] M. Wang, J. Liu, W. Chen, and A. Ephremides, "Joint queue-aware and channel-aware delay optimal scheduling of arbitrarily bursty traffic over multi-state time-varying channels," *IEEE Transactions on Communications*, vol. 67, no. 1, pp. 503–517, 2018.

[27] T. D. Burd and R. W. Brodersen, "Processor design for portable systems," *Journal of VLSI signal processing systems for signal, image and video technology*, vol. 13, no. 2-3, pp. 203–221, 1996.

[28] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

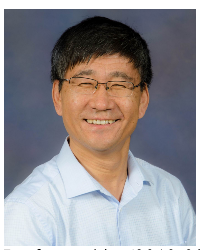
[29] Y. Long, H. Li, H. Yue, M. Pan, and Y. Fang, "Sum: Spectrum utilization maximization in energy-constrained cooperative cognitive radio networks," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 11, pp. 2105–2116, 2014.

[30] 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.



Di Han (S'16) received the B.S. degree in information engineering from Shanghai Jiao Tong University, Shanghai, China, in 2015, where he is currently pursuing the Ph.D. degree with the Department of Electronic Engineering in Tsinghua University, Beijing, China. His research interests are in wireless communications, with an emphasis on mobile computation offloading technologies. He has received the Samsung Scholarship in 2013 and National Scholarship in 2014.

Wei Chen (S'05-M'07-SM'13) received the B.S. and Ph.D. degrees (Hons.) from Tsinghua University in 2002 and 2007, respectively. Since 2007, he has been a Faculty Member with Tsinghua University, where he is currently a Tenured Full Professor, the Director of the Degree Office, and a University Council Member. During 2014-2016, he was the Deputy Head of the Department of Electronic Engineering in Tsinghua University. From 2005 to 2007, he was a Visiting Ph.D. Student with the Hong Kong University of Science and Technology. He visited the University of Southampton in 2010, Telecom ParisTech in 2014, and Princeton University, Princeton, NJ, USA, in 2016. His research interests are in the areas of communication theory, stochastic optimization, and statistical learning. He is a Cheung Kong Young Scholar and a member of the National Program for Special Support of Eminent Professionals, also known as 10,000 talent program. He is also a member of All-China Youth Federation. He received the IEEE Marconi Prize Paper Award in 2009 and the IEEE Comsoc Asia Pacific Board Best Young Researcher Award in 2011. He was a recipient of the National May 1st Labor Medal and the China Youth May 4th Medal. He has also been supported by the National 973 Youth Project, the NSFC Excellent Young Investigator Project, the New Century Talent Program of the Ministry of Education, and the Beijing Nova Program. He serves as an Editor for the IEEE TRANSACTIONS ON COMMUNICATIONS. He has served as a TPC Co-Chair for IEEE VTC-Spring in 2011 and a Symposium Co-Chair for IEEE ICC and Globecom.



Yuguang Fang (S'92, M'97, SM'99, F'08) received an MS degree from Qufu Normal University, Shandong, China in 1987, a PhD degree from Case Western Reserve University in 1994, and a PhD degree from Boston University in 1997. He joined the Department of Electrical and Computer Engineering at University of Florida in 2000 as an assistant professor, then was promoted to an associate professor in 2003, a full professor in 2005, and a distinguished professor in 2019. He holds a University of Florida Foundation Preeminence Term Professorship (2019-2022), a University of Florida Foundation Professorship (2017-2020, 2006-2009), and a University of Florida Term Professorship (2017-2021).

Dr. Fang received the US NSF Career Award in 2001, the US ONR Young Investigator Award in 2002, the 2018 IEEE Vehicular Technology Outstanding Service Award, the 2019 IEEE Communications Society AHSN Technical Achievement Award, the 2015 IEEE Communications Society CISTC Technical Recognition Award, the 2014 IEEE Communications Society WTC Recognition Award, and the Best Paper Award from IEEE ICNP (2006). He has also received a 2010-2011 UF Doctoral Dissertation Advisor/Mentoring Award, and the 2009 UF College of Engineering Faculty Mentoring Award. He served as the Editor-in-Chief of IEEE Transactions on Vehicular Technology (2013-2017), the Editor-in-Chief of IEEE Wireless Communications (2009-2012), and serves/served on several editorial boards of journals including Proceedings of the IEEE (2018-present), ACM Computing Surveys (2017-present), IEEE Transactions on Mobile Computing (2003-2008, 2011-2016), IEEE Transactions on Communications (2000-2011), and IEEE Transactions on Wireless Communications (2002-2009). He has been actively participating in conference organizations such as serving as the Technical Program Co-Chair for IEEE INFOCOM?2014 and the Technical Program Vice-Chair for IEEE INFOCOM'2005. He is a fellow of the IEEE and the American Association for the Advancement of Science (AAAS).