# Packet Scheduling and Computation Offloading for Energy Harvesting Devices without CSIT

Ibrahim Fawaz[1,2], Mireille Sarkiss[3], and Philippe Ciblat[2]
[1] CEA, LIST, Communicating Systems Laboratory PC 173, 91191 Gif-sur-Yvette, France
[2] Télécom ParisTech, Institut Polytechnique de Paris, 75013 Paris, France
[3] Télécom SudParis, Institut Polytechnique de Paris, 91011 Evry, France
Email: ibrahim.fawaz@cea.fr, mireille.sarkiss@telecom-sudparis.eu, philippe.ciblat@telecom-paristech.fr

***Abstract*—This paper proposes a joint packet scheduling and computation offloading policy for an Energy Harvesting (EH) mobile terminal wirelessly connected to a Base Station (BS) when the channel between the mobile and the BS is unavailable at the mobile side. The mobile terminal has to decide if its packet related to one application is computed either locally or remotely by the BS within a strict delay imposed by this application without knowing the channel in advance. Our objective is to guarantee reliable communication by minimizing the packet loss. This packet loss is due to buffer overflow, strict delay violation and channel mismatch. We formulate the problem using a Markov Decision Process (MDP) and we propose and implement the optimal deterministic offline policy to solve it. This optimal policy decides: (i) the execution location (locally or remotely), (ii) the number of packets to be executed and (iii) the corresponding transmission power. This policy offers a dramatic increase in the number of executed packets and a significant energy saving.**

## I. Introduction

Nowadays, mobile communication systems face unprecedented growth of connected devices which applications require high computation ability. In addition, some applications need to satisfy a strict delay. These constraints lead to an increase demand in high-speed processing and energy. In order to meet these challenges, a novel system combining Energy Harvesting (EH) [1], [2] with Multi-Access Edge Computing (MEC) [3], [4] was recently proposed. Such a system is able to harness energy from the surrounding environment to power its communication and, at the same time, get rid of extensive computations by dispatching them to resourceful nodes deployed within the BS, which are in charge of processing them and sending back the results. This emerging topic brings tremendous potential for enhancing the performance of mobile devices, but also new issues for designing efficient decision-making policies. Recently, the resource management problem of an EH-MEC system was studied in [5]. The authors came up with a dynamic computation offloading policy for mobile terminals. This problem was also addressed for a group of edge servers sharing the same cell of a BS in [6]. An improved policy was proposed based on a combination of offline value iteration algorithm and online reinforcement learning.

This work has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 675891.

In this paper, we address packet scheduling and computation offloading for a single EH mobile user served by a BS when strict delay constraint has to be fulfilled and no channel information is available at the transmitter. Unlike [5], [6], we impose here a strict delay constraint instead of an average delay constraint which prevent to re-use results from [5], [6]. This strict delay constraint was introduced in our previous works devoted to packet scheduling without EH and MEC capabilities [7] and to packet scheduling and computation offloading with perfect CSI at the Transmitter (CSIT) [8]. So the main difference between [8] and this current work lies in the assumption on the channel knowledge. Hereafter, we assume that the current channel state is not available at the terminal before making the decision. Actually, we consider that as *previous* channel realization can be acquired through feedback, the decision relies on it. Compared to [8], we need to redesign the system model partly, to rewrite the corresponding Markov Decision Process (MDP) completely (states, transition probabilities, etc), and more importantly, to refine the ways to loose a packet. Indeed, a new type of packet loss, called channel mismatch, has to be added. Our objective is to minimize the number of discarded packets due to the strict delay violation, the buffer overflow, and the channel mismatch. The problem casts into an MDP and we find the optimal offline stationary policy through the Policy Iteration (PI) algorithm. We compare this policy with three naive ones described later.

The paper is organized as follows. In Section II, we provide the system model. In Section III, the related MDP problem is described and solved. In Section IV, numerical results are given and analyzed. In Section V, a conclusion is drawn.

## II. System Model

We consider a MEC system involving an EH mobile user and its serving cloud-enabled BS. The mobile terminal stores its data packets in a finite buffer and the harvested energy in a limited-capacity battery. The time is slotted into consecutive epochs of equal duration $T_s$. At the beginning of each time slot, the system decides the execution type (locally or remotely), the number of packets to be processed and the transmission power. Hereafter, we provide a full description of the data, energy and channel models, followed by the different execution options and their corresponding energy consumption.

### A. Data model

The data arrival process is modeled as an independent identically distributed (i.i.d.) Poisson distributed process with an average arrival rate $\lambda_d$. All packets are assumed to have the same size of $L$ bits. Before making any decision, these packets are kept in the buffer of the mobile terminal that can store a maximum of $B_d$ packets. A packet is discarded from the buffer if we observe

- **delay violation**, i.e., the packets are stayed in the buffer more than $A_0$ time-slots; and
- **buffer overflow**, i.e., the buffer is already full and there is no room for any additional packet.

The age of each packet in the buffer is required to track the information of the system. Consequently, we denote $a_i(n)$ the age of the $i$-th packet in the buffer at slot $n$. By definition, we have $a_i(n) \in \{-1, \cdots, A_0\}, \forall i, n$ where $a_i(n) = -1$ stands for an empty space in the buffer. Note that packets are ordered in the buffer, i.e. $a_j(n) \leq a_i(n)$, $\forall i \leqslant j$.

### B. Energy model

We model the EH process as a sequence of energy unit (e.u.) arrivals of an i.i.d. Poisson distributed process with an average arrival rate $\lambda_e$. We assume that each e.u. can provide $\mathcal{E}_U$ Joules (J). Before being available for use, this energy is saved in a battery of limited capacity $B_e$, but wasted if the battery is already full. At the beginning of slot $n$, we denote $b_n \in \{0, \cdots, B_e\}$ the battery's energy level after collecting $e_n$ e.u.. By construction, the consumed energy $E_n$ during slot $n$ cannot exceed $b_n$. Moreover, we assume that $b_n$ is causally known at the decision instant in slot $n$.

### C. Channel model

The wireless channel is assumed to be block-fading, i.e. the channel remains constant within each slot and can change state only at the beginning of a slot. The allocated bandwidths are $W_{UL}$ (Hz) in the uplink and $W_{DL}$ (Hz) in the downlink, with an additive white Gaussian noise of power spectral density $N_0$. We consider that the channel process takes values from a finite set $\mathcal{X}$. We define the channel gain by $x = |h|^2$ where $h$ is its complex-valued amplitude. Channel gains are time-correlated following a Markovian model, i.e. the current channel state depends only on the previous channel state. The transition probability from a channel state $j$ at time slot $n$ to a channel state $i$ at time slot $n+1$ is given by

$$p(x_{n+1} = i | x_n = j) = \frac{(1-\rho)^{|i-j|}}{\sum_{k=0}^{|\mathcal{X}|-1} (1-\rho)^{|k-j|}}, \qquad (1)$$

where $\rho \in [0, 1[$ is the correlation coefficient. We assume Outdated CSIT, i.e., only $x_{n-1}$ is known when making decision at time slot $n$. Packets can thus be lost due to a **channel mismatch** since the mobile terminal makes the execution decision based on the previous (and so potentially wrong) channel state.

### D. Execution decisions and related energy consumption

At the beginning of slot $n$, the mobile terminal can choose amongst three possible decisions:

- **Idle:** The mobile terminal does not process any packet and awaits the following slot. In that case, the electronic circuits are considered to be in standby mode leading to zero energy consumption
$$E_I = 0. \qquad (2)$$
- **Local execution:** The mobile terminal executes $u$ packets from its buffer ($u \leqslant q_n$, where $q_n$ is the number of packets in the buffer at time-slot $n$) using its internal processor. The associated energy consumption, expressed as an integer multiple of the e.u., is calculated as
$$E_\ell(u) = \left\lceil u.P_\ell.\frac{T_s}{\mathcal{E}_U} \right\rceil, \qquad (3)$$
where $P_\ell$ is the power consumed to process one packet locally.
- **Remote execution:** The mobile terminal transfers $u$ packets for processing in the BS and then receives the result. The energy is thus consumed to send data, to wait for the remote processing and to receive the result. As the current channel state is unknown, packets are sent at a rate tuned according to the previous channel state. As a consequence, the energy consumption, expressed as an integer multiple of the energy unit, is derived as
$$E_o(x_{n-1}, u) = \left\lceil \frac{u}{\mathcal{E}_U} \left( \frac{L.P_t}{W_{UL}.\log_2\left(1 + \frac{P_t.x_{n-1}}{W_{UL}.N_0}\right)} + T_w.P_w + \frac{L_{DL}.P_r}{W_{DL}.\log_2\left(1 + \frac{P_s.x_{n-1}}{W_{DL}.N_0}\right)} \right) \right\rceil \qquad (4)$$
where $P_t$ is the power consumed by the mobile terminal to send packets. $T_w$ and $P_w$ are the time spent for the BS to execute one packet and the power consumed by the mobile terminal while waiting for the remote packets to be processed, respectively. The result of the computation of size $L_{DL}$ [bits] is sent back by the BS with the power $P_s$ and finally acquired by the mobile terminal consuming in the process a power $P_r$. While the BS is capable of adapting its rate to the current channel (since it can estimate it via the training sequence of the uplink received packets), we force the BS to consider $x_{n-1}$ since the transmitter only knows $x_{n-1}$ to evaluate Eq. (4).
Furthermore, this offloading operation must be carried out within a slot leading to the following constraint
$$u \left( \frac{L}{W_{UL}.\log_2\left(1 + \frac{P_t.x_{n-1}}{W_{UL}.N_0}\right)} + T_w + \frac{L_{DL}}{W_{DL}.\log_2\left(1 + \frac{P_s.x_{n-1}}{W_{DL}.N_0}\right)} \right) \leqslant T_s. \qquad (5)$$
Notice that $W_{DL}, W_{UL}, N_0, L_{DL}, P_s, T_w$ are pre-defined parameters. With respect to $u$ and $x_{n-1}$, $P_t$ can be computed by forcing the equality in Eq. (5).

## III. Problem Formulation and Resolution

Our goal is to devise an optimal policy $\mu^\star$ minimizing the number of discarded packets due to buffer overflow, delay violation and channel mismatch. At the beginning of each slot, the policy $\mu^\star$ specifies the best action to make, namely the processing decision (idle, local or remote processing), the number of packets $u$ to execute and the corresponding transmission power $P_t$. This section describes the states and actions of our system proving that the problem can be formulated as an MDP. The transition matrix and the cost function of this MDP are first defined. Then, the optimization problem is solved through an offline policy iteration algorithm.

### *A. State Space*

The state space $\mathcal{S}$ is the set of $\mathbf{s} = (\mathbf{a}, b, x)$ where

- $\mathbf{a} = [a_1, \cdots, a_{B_d}]$ is the vector of each packet's age,
- $b$ is the current battery level, and
- $x$ is the previous channel gain.

The state space is finite, and the total number of possible states $|\mathcal{S}|$ is upper-bounded by $(A_0+2)^{B_d}.|B_e+1|.|\mathcal{X}|$. However, the state space is considerably reduced, since the packets are ordered in the buffer according to their age. For instance, with $B_d = 6$, $K_0 = 3$, $B_e = 4$ and $|\mathcal{X}| = 5$, our system has only 5250 states out of the 390625 possible combinations.

### *B. Action Space*

The action space $\mathcal{V}$ is the set of possible decisions that the mobile device can make at the beginning of each slot. In particular, the selected action $\nu_n$ in slot $n$ includes:

- the processing decision (idle, local or remote execution),
- the number of packets $u_{\nu_n}$ to be executed, and
- the corresponding transmission power $P_{\nu_n}$.

On one hand, the mobile terminal can execute locally a maximum of $U_\ell$ packets during a slot, depending on the capacity of its internal processor. On the other hand, according to Eq. (5) with equality and using the maximum transmission power $P_{\max}$ and the best channel gain $x_{\max} = \max_{x\in\mathcal{X}}$, the mobile terminal can execute remotely up to $U_o$ packets during a slot. Moreover, $P_\nu$ can be one of the calculated powers with Eq. (5) or equal to $P_{\max}$ if possible. Therefore, the action space is finite with cardinality $|\mathcal{V}| = U_\ell + U_o \times (|\mathcal{X}|+1) + 1$.

### *C. Markov Decision Process*

If during slot $n$, $m_n$ queued packets have reached the maximum delay ($A_0$), the mobile terminal will execute and/or discard $w_n = \max(u_{\nu_n}, m_n)$ packets. Then, it will increment the age of the remaining $q_n - w_n$ queued packets by 1 and store the new $d_{n+1}$ received packets in the buffer with age 0. Thus, the vector $\mathbf{a}$ at slot $n+1$ will be derived as

**for** $i = 1$ **to** $q_n - w_n$ **do**
  $a_i(n+1) = a_{w_n+i}(n) + 1$
**end for**
**for** $i = q_n - w_n + 1$ **to** $q_n - w_n + d_{n+1}$ **do**
  $a_i(n+1) = 0$
**end for**
**for** $i = q_n - w_n + d_{n+1} + 1$ **to** $B_d$ **do**
  $a_i(n+1) = -1$
**end for**

Simultaneously, the execution of these $u_{\nu_n}$ packets consumes $E_n$ e.u. from the battery according to Eqs. (3) or (4). Then, the mobile terminal harvests and stores $e_{n+1}$ e.u. in its battery. Thus, the battery state at slot $n+1$ will be

$$b_{n+1} = \min\{b_n - E_n + e_{n+1}, B_e\}. \tag{6}$$

We therefore notice that $s_{n+1}$ depends only on $s_n$, $\nu_n$ and the external disturbance $(d_{n+1}, e_{n+1})$, which is in compliance with the MDP's fundamental property.

### *D. Transition Matrix*

The transition matrix of an MDP specifies the probability of moving from a state $\mathbf{s} = (\mathbf{a}, b, x)$ to a state $\mathbf{s}' = (\mathbf{a}', b', x')$ after performing an action $\nu$. This transition probability can be expressed as the product of the transition probabilities of the buffer, battery and channel states, if and only if the latter are independent of each other, i.e.

$$p(\mathbf{s}'|\mathbf{s},\nu) = p(\mathbf{a}'|\mathbf{a},b,\nu).p(b'|b,x,\nu).p(x'|x), \tag{7}$$

We first identify for each state $\mathbf{s}$, the set of unfeasible actions $\mathcal{A}(\mathbf{s}) = \mathcal{A}_0(\mathbf{s}) \cap \mathcal{A}_1(\mathbf{s}) \cap \mathcal{A}_2(\mathbf{s})$ where each subset $\mathcal{A}_i(\mathbf{s})$ is defined as follows: *i*) the set $\mathcal{A}_0(\mathbf{s})$ is composed by the offloading actions that require a transmit power $P_t > P_{\max}$ according to Eq. (5); *ii*) the set $\mathcal{A}_1(\mathbf{s})$ includes all the actions that comply with at least one of the following criteria

1: $u_\nu > q$ **or** $a'_i > a_i + 1$ **or** $q' < q - w$
2: $a'_i \neq a_{i+u_\nu} + 1$ **and** $a_{i+u_\nu} \neq -1$
3: $a'_i > 0$ **and** $a_{i+u_\nu} = -1$
4: $q = B_d$ **and** $u_\nu \neq 0$ **and** $a'_i > 0, \forall i \in \{q-w+1, ..., B_d\}$

*iii*) the set $\mathcal{A}_2(\mathbf{s})$ consists of all the actions that fulfill at least one following condition

1: $0 > b - E$
2: $b' < b - E$

where $E$ is the consumed energy according to Eqs. (3) or (4).

Finally, when $\nu \in \mathcal{V} \setminus \mathcal{A}(\mathbf{s})$, the transitions are as follows

**if** $q' < B_d$ **then**
  $p(a'_i|a_i,b,\nu) = e^{-\lambda_d}.\frac{(\lambda_d)^{q'-q+w}}{(q'-q+w)!}$
**else**
  $p(a'_i|a_i,b,\nu) = 1 - \mathrm{Q}(B_d - q + w, \lambda_d)$,

and

**if** $b' < B_e$ **then**
  $p(b'|b,x,\nu) = e^{-\lambda_e}.\frac{(\lambda_e)^{b'-b+E}}{(b'-b+E)!}$
**else**
  $p(b'|b,x,\nu) = 1 - \mathrm{Q}(B_e - b + E, \lambda_e)$.

where Q is the regularized Gamma function.

### *E. Cost and optimal policy*

Let $\mathbf{s}_n = (\mathbf{a}_n, b_n, x_{n-1})$ and $\mu(\mathbf{s}_n) = \nu_n$ be the system state and the selected action at a given slot $n \in \{0, \cdots, N\}$, respectively.

We aim at minimizing the average number of discarded packets under policy $\mu$. Hence, the cost function is given by

$$\overline{D}(\mu) = \lim_{N\to\infty} \frac{1}{N}\mathbb{E}^{\mu}\left[\sum_{n=1}^{N}(\varepsilon_d(\mathbf{s}_n,\nu_n)+\varepsilon_o(\mathbf{s}_n,\nu_n)+\varepsilon_c(\mathbf{s}_n,\nu_n))\right], \tag{8}$$

where $\mathbb{E}$ is the mathematical expectation with respect to the policy $\mu$. $\varepsilon_o(\mathbf{s}_n,\nu_n)$, $\varepsilon_d(\mathbf{s}_n,\nu_n)$ and $\varepsilon_c(\mathbf{s}_n,\nu_n)$ are the instantaneous number of discarded packets due to buffer overflow, delay violation, and channel mismatch, respectively.

The buffer overflow occurs when $q_n - w_n + d_{n+1} > B_d$, therefore the number of discarded packets due to buffer overflow is obtained as

$$\begin{aligned}
\varepsilon_o(\mathbf{s}_n,\nu_n) &= \sum_{t=B_d-q_n+w_n+1}^{+\infty}(q_n - w_n + t - B_d).e^{-\lambda_d}.\frac{(\lambda_d)^t}{t!} \\
&= \lambda_d.(1-\mathrm{Q}(B_d - q_n + w_n, \lambda_d)) \\
&+ (q_n - w_n - B_d) \\
&\times (1-\mathrm{Q}(B_d - q_n + w_n + 1, \lambda_d)).
\end{aligned} \tag{9}$$

During slot $n$, the number of discarded packets due to delay violation is given by

$$\varepsilon_d(\mathbf{s}_n,\nu_n) = \begin{cases} 0 & \text{if } m_n = 0 \text{ or } m_n \leqslant u_{\nu_n} \\ m_n - u_{\nu_n} & \text{otherwise.} \end{cases} \tag{10}$$

During slot $n$, the channel mismatch occurs because the current channel state $x_n$ is unknown, and the decisions are made based on the knowledge of the previous channel state $x_{n-1}$. This situation arises when:

- the mobile device decides to offload with a rate $R_{UL}(P_t, x_{n-1}) > R_{opt,UL}$
- the BS uses the rate $R_{DL}(P_s, x_{n-1}) > R_{opt,DL}$

Both conditions are equivalent to $x_{n-1} > x_n$. Thus, the number of discarded packets due to channel mismatch is

$$\varepsilon_c(\mathbf{s}_n,\nu_n) = u_n \times \mathrm{Prob}(x_{n-1} > x_n), \tag{11}$$

Finally, the MDP optimization problem is stated as

$$\mu^{\star} = \arg\min_{\mu} \overline{D}(\mu). \tag{12}$$

This optimization problem can be solved using the PI algorithm [9]. The resulting optimal offline deterministic policy assigns to each state $\mathbf{s} \in \mathcal{S}$ one and only one action $\nu \in \mathcal{V}$.

## IV. Numerical Results

We consider the system defined in Section II with the following parameters: the slot duration is $T_s = 1$ ms. The data buffer can store $B_d = 6$ packets, each of size $L = 5000$ bits and can stay in the buffer $A_0 = 3$ slots before being discarded. The battery of the mobile device can store $B_e = 4$ e.u with $\mathcal{E}_U = 40$ nJ. The allocated bandwidth is $W_{UL} = 500$ kHz in the uplink and $W_{DL} = 5$ MHz in the downlink with a noise power spectral density of $N_0 = -87$ dBm/Hz. The channel state $x$ can take 5 values from the finite set $\mathcal{X} = \{-5.41, -1.59, 0.08, 1.42, 3.18\}$ dB following the correlation model described in Eq. (1). The packets carrying the computation result have the same size $L_{DL} = 500$ bits. During a slot, the mobile device can execute up to $U_\ell = 2$ packets locally or $U_o = 4$ packets remotely. The rest of the parameters are as follows: $P_\ell = 30$ $\mu$W, $P_r = 0.2$ mW, $P_s = 1.6$ kW, $P_w = 0.1$ mW, $T_w = 0.1$ ms, and $P_{\max} = 0.74$ mW.

In Fig. 1 and Fig. 2, we plot the percentage of discarded packets versus the data arrival rate $\lambda_d$ for $\rho = 0.99$ and $\rho = 0.75$, respectively. Two values of energy arrival rate are also considered, $\lambda_e = 1.0$ and $\lambda_e = 2.0$. The performance of the optimal policy is compared to three different policies, namely the *immediate*, *local*, and *offload* policies. The immediate policy processes, locally or remotely, the maximum number of packets using the current energy in the battery. The *local* policy is obtained by the PI algorithm when the actions are restricted to belong to the set of "local execution" decisions. The *offload* policy is obtained by the PI algorithm when the actions are restricted to belong to the set of "remote execution" decisions. We can see that the proposed policy gives better performance than the other policies. As the buffer overflow occurs more frequently when $\lambda_d$ increases, we observe an increase in the number of discarded packets for all the policies. For both values of $\lambda_e$, we can notice that the *local* policy tends to the optimal policy, as $\rho$ decreases since the probability of losing packets during the transmission increases. When $\rho$ is high, the performance of the *local* policy decreases when $\lambda_d$ increases due to the insufficient computing capacity of the mobile terminal's processor. Moreover, the gap between the optimal and the *immediate* policy increases when $\rho$ decreases. Finally, the performance of the *offload* policy decreases when $\rho$ decreases because packets can be lost at each decision due to channel fluctuation.

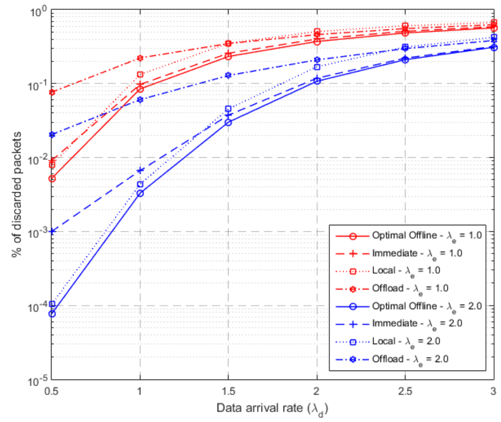


Fig. 1: Percentage of the discarded packets versus $\lambda_d$ for $\lambda_e = \{1, 2\}$ and $\rho = 0.99$.

In Fig. 3 and Fig. 4, the average consumed energy and the average battery state are respectively plotted versus the data arrival rate $\lambda_d$ for $\rho = 0.99$ and energy arrival rates $\lambda_e = \{1, 2\}$. The proposed optimal policy consumes roughly as much energy as the *offload* policy while executing more packets. We can clearly notice that the *local* and *immediate* policies consume the highest amount of energy since local

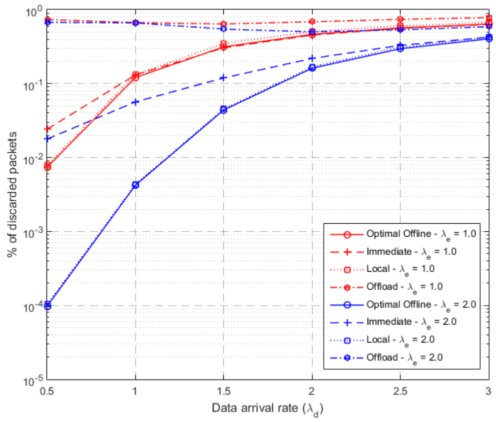


Fig. 2: Percentage of the discarded packets versus $\lambda_d$ for $\lambda_e = \{1, 2\}$ and $\rho = 0.75$.

packet processing is expensive, thereby depleting the battery. In fact, the optimal use of the available energy is reflected in a higher energy level in the battery, ensuring better sustainable communication with fewer discarded packets.

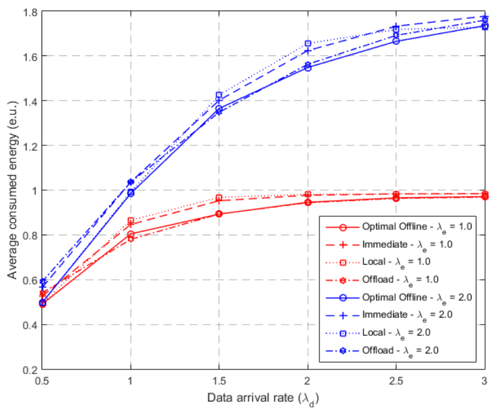


Fig. 3: Average consumed energy versus $\lambda_d$ for $\lambda_e = \{1, 2\}$.

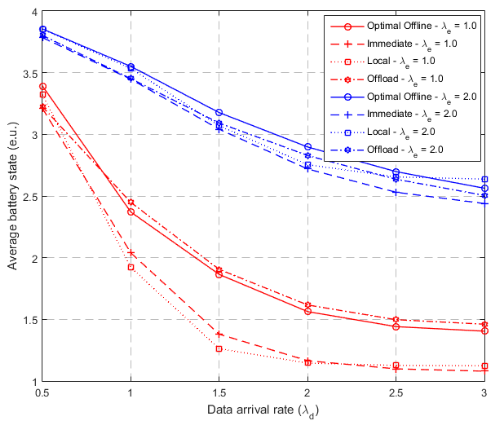


Fig. 4: Average battery state versus $\lambda_d$ for $\lambda_e = \{1, 2\}$.

In Fig. 5, we show the percentage of processing decisions of the optimal policy at $\lambda_d = \{1, 2\}$ for $\rho = 0.99$ in Fig. 5(a)(c), and for $\rho = 0.75$ in Fig. 5(b)(d). When $\rho$ decreases, the system executes more packets locally to minimize the number of discarded packets. When $\lambda_e$ is large, the system reduces the offloading decisions in order to prevent the packet loss due to channel mismatch. However, when $\lambda_e$ is small, the system is forced to use offloading as executing packets locally is costly.

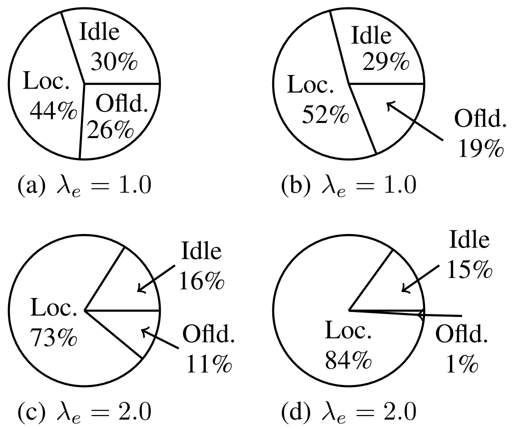


Fig. 5: Percentage of processing decisions for $\rho = 0.99$ (a) and (c), and $\rho = 0.75$ (b) and (d).

## V. Conclusion

Packet scheduling and offloading policy for an EH mobile terminal to its BS was addressed under strict delay constraint and without CSIT. Through MDP framework, an optimal policy was proposed to minimize the packet loss rate.

## References


[1] S. Ulukus, A. Yener, E. Erkip, O. Simeone, M. Zorzi, P. Grover and K. Huang, "Energy Harvesting Wireless Communications: A Review of Recent Advances," IEEE Journal on Selected Areas in Communications, vol. 33, no. 3, pp. 360-381, March 2015.

[2] M. L. Ku, W. Li, Y. Chen and K. J. R. Liu, "Advances in Energy Harvesting Communications: Past, Present, and Future Challenges," IEEE Communications Surveys & Tutorials, vol. 18, no. 2, pp. 1384-1412, February 2016.

[3] K. Kumar, J. Liu, Y-H. Lu and B. Bhargava, "A Survey of Computation Offloading for Mobile Systems," Mobile Networks and Application, vol. 18, no. 1, pp. 129-140, February 2013.

[4] P. Mach and Z. Becvar, "Mobile Edge Computing: A Survey on Architecture and Computation Offloading," IEEE Communications Surveys & Tutorials, vol. 19, no. 3, pp. 1628-1656, Third Quarter 2017.

[5] 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, December 2016.

[6] J. Xu, L. Chen and S. Ren, "Online Learning for Offloading and Autoscaling in Energy Harvesting Mobile Edge Computing," IEEE Transactions on Cognitive Communications and Networking, vol. 3, no. 3, pp. 361-373, September 2017.

[7] I. Fawaz, P. Ciblat and M. Sarkiss, "Energy Minimization based Resource Scheduling for Strict Delay Constrained Wireless Communications," IEEE Global Conference on Signal and Information Processing (GLOBALSIP), December 2017.

[8] I. Fawaz, P. Ciblat and M. Sarkiss, "Joint Resource Scheduling and Computation Offloading for Energy Harvesting Communications," IEEE International Conference on Telecommunications (ICT), June 2018.

[9] D. P. Bertsekas, "Dynamic Programming and Optimal Control," vol.2, Fourth Edition.