Discrete Rate Scheduling for Packets With Individual Deadlines in Energy Harvesting Systems

Feng Shan, *Student Member, IEEE*, Junzhou Luo, *Member, IEEE*, Weiwei Wu, *Member, IEEE*, Minming Li, *Member, IEEE*, and Xiaojun Shen, *Senior Member, IEEE*

***Abstract*—This paper presents an optimal rate scheduling algorithm called *Truncation* for an energy-harvesting enabled wireless transmitter to transmit a set of dynamically arrived packets with minimum transmission energy. Distinct from existing works, we allow packets to have individual delay constraints, which is the most general model ever assumed but is very much desired to guarantee per-application quality-of-service (QoS). Moreover, we restrict the allowable rates to a set of discrete values, which is more practical and required in many real applications. As the first achievement, we obtain an optimal offline algorithm, which assumes the rate is continuously adjustable. Then, we propose a general framework that transforms any algorithm using the continuous-rate model into an algorithm using only discrete-rates, while preserving the optimality as long as the optimality holds for convex rate-power functions. It is possible that the harvested energy is insufficient to guarantee all packets to meet their deadlines. Should this occur, maximizing throughput with the limited available energy becomes the goal to achieve. Our *Truncation* algorithm is able to identify this case and produces a schedule that guarantees maximum throughput, if packets share a common deadline. Furthermore, based on the optimal offline algorithms, an efficient online algorithm is designed which has been shown by simulations to produce near optimal results.**

***Index Terms*—Energy harvesting, packet scheduling, energy-efficient rate scheduling, individual packet deadline, discrete rates, wireless communications.**

I. Introduction

ENERGY harvesting enables wireless devices to receive energy from nature sources to support long life-time operations and is being developed as commercially viable solutions to many applications. Although this technique can provide theoretically sustainable power supply, we must carefully manage the energy usage because the harvested energy is unstable and limited. According to Shannon-Hartley Theorem on wireless channel capacity, a slower transmission rate is preferred to save energy, while a higher rate is preferred to shorten transmission delay. Tremendous research efforts have been made to design energy-efficient rate scheduling algorithms without [1]–[7] or with [8]–[16] energy harvesting.

E. Uysal-Biyikoglu *et al.* [1], [2] are among the first group to formulate the rate scheduling problem which aims at minimizing the energy consumption for delivering a set of packets before a common deadline. They propose a *lazy* schedule to optimally solve the offline problem. Since then, pursuing optimal rate scheduling for delay-constrained packet transmission has received tremendous research interests. W. Chen *et al.* [3], [4] consider a more general problem that allows the packets to have different deadlines, but requires all allowed delays to be equal. Zafer and Modiano [5], [6] further generalize the problem to allow individual packet deadlines provided they follow the same order packets arrive. Most recently, Shan, Luo, and Shen [7] solve the problem that allows arbitrary individual packet deadlines.

Recently, the rate scheduling problems under energy harvesting settings have attracted a lot of research interests [8]–[16]. These works focus on designing offline rate schedules to minimize the transmission completion time for a set of packets [8], [9] with a limited battery capacity [10], [11], or to minimize the transmission time for one data block in a fading channel [12], [13], or to minimize the transmission time for one data block per user in a broadcast communication system [14], [15], or to minimize the transmission time for one queue of packets per user in a broadcast system [16], or to maximize the total throughput for two users in a Gaussian interference channel [17]. However, none of these algorithms can guarantee individual packet delay constraints. Since a communication channel is usually shared by a variety of different applications, and each application may have its own quality-of-service (QoS) constraint, very likely, packets may have individually required delay constraints [3]–[7]. Therefore, the optimal rate scheduling problem that involves both energy-harvesting and individual delay constraints becomes an open problem we wish to solve in this paper.

In addition to the above research results which are directly related to our work, there are also some other relevant research results. Gatzianas *et al.* [18] investigate the system utility

Manuscript received April 1, 2014; revised September 15, 2014; accepted December 16, 2014. Date of publication January 15, 2015; date of current version March 19, 2015. This work was supported by the National High-tech R&D Program under Grant No. 2013AA013503, the National Natural Science Foundation of China under Grants No. 61320106007, No. 61300024, No. 61370208 and No. 61402104, the China Specialized Research Fund for the Doctoral Program of Higher Education under Grants No. 20110092130002 and No. 20130092120036, the Jiangsu Provincial Natural Science Foundation of China under Grants No. BK20130634 and No. BK20140648, the Jiangsu Provincial Key Laboratory of Network and Information Security under Grant No. BM2003201, the Key Laboratory of Computer Network and Information Integration of Ministry of Education of China under Grants No. 93K-9, and the Research Grants Council of the Hong Kong Special Administrative Region, China [Project No. CityU 117913].

F. Shan, J. Luo and W. Wu are with the School of Computer Science and Engineering, Southeast University, Nanjing 210018, China (e-mail: shanfeng@seu.edu.cn; jluo@seu.edu.cn; weiweiwu@seu.edu.cn).

M. Li is with the Department of Computer Science, City University of Hong Kong, Kowloon Tong, Hong Kong (e-mail: minming.li@cityu.edu.hk).

X. Shen is with the School of Computing and Engineering, University of Missouri, Kansas City, MO 64110 USA (e-mail: shenx@umkc.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSAC.2015.2391491

maximization problem from a stochastic aspect. Vaze *et al.* [19], [20] propose an online algorithm with a competitive ratio to maximize the achievable rate in fading channels. Zhang *et al.* [21] consider a packet relay problem that aims at maximizing the throughput. Gurakan *et al.* [22] investigate the throughput maximization problem when energy can be transferred.

Most previous works assume that the transmission rate is continuously adjustable. Although the continuous model can ease mathematical derivations, in many real systems, transmission rate or power is restricted to a set of discrete values. For instance, in CDMA proposal IS-95 [23], the power levels are equally spaced by 0.5 dB, within a dynamic range of 85 dB in the uplink and 12 dB in the downlink. WLAN standard IEEE 802.11b/g/n allows transmission and reception of data at 4/12/32 discrete rates respectively [24]. In academia, this assumption has been used more and more often in many research works [25]–[28]. Most recently, Bacinoglu and Uysal-Biyikoglu [29] consider the discrete rates on an energy harvesting communication link and propose an online schedule policy that maximizes the throughput. Bodin and Gunduz [30] also consider the discrete rate schedule problem. In their settings, an optimal solution for the continuous adjustable rate model can be easily converted to a solution for the model of discrete rates.

This paper also attempts to optimally solve the rate scheduling problem for the discrete rate model. For both models, we allow packets to have individual deadlines, which is the most difficult model ever used in previous research. The combination of individual deadlines and discrete rate model characterizes significant differences between our model and those in the literature for the rate scheduling problem under energy-harvesting settings. As the first step which is also the key step, we need to obtain an optimal scheduling algorithm under the offline setting, where the information regarding packets and harvesting is known in advance and the allowable discrete transmission rates are given as well. This step generalizes and improves on existing algorithms in the literature, but is quite challenging because this rate schedule must (1) consume no more energy than the harvested amount before any time instance, (2) not transmit any data before its arrival, (3) finish each packet before its deadline, (4) transmit only at discrete rate values, (5) consume as little energy as possible. Moreover, an additional consideration must be given to the special case when the harvested energy is insufficient to guarantee deadlines for all packets. Should this situation occur, minimizing energy usage is meaningless, but maximizing throughput with the limited available energy becomes our optimization goal.

Our contributions in this paper are summarized as follows.

- We propose a novel method *Truncation* that produces an optimal rate schedule for the energy minimization problem under the continuous rate model if a feasible schedule exists. This method allows individual deadlines, which is the most general model ever assumed in the literature so far.
- We propose a general framework that transforms any scheduling algorithm computing the continuous rate schedule to an algorithm computing the discrete rate schedule while preserving the optimality. The only requirement is that the optimality of such algorithms remains whenever the rate-power function is convex, which is satisfied by almost all algorithms in the literature, including our *Truncation* method.
- If no feasible schedule exists for the energy minimization problem, our *Truncation* method identifies and reports this situation. Moreover, an optimal schedule is produced that maximizes the throughput if packets share a common deadline. In case individual deadlines are allowed, we provide a numerical solution.
- We design an efficient online heuristic algorithm when individual deadlines are allowed. Simulations demonstrate that, on average, this online algorithm approaches within 93% of the optimal offline solution.

The organization of this paper is as follows. In Section II, we formally define the system model, packet model, and the optimization problems. Sections III and IV solve the energy minimization problem under the continuous rate model. In Section V, we propose a general two-step framework that is capable of deriving the optimal algorithms in discrete rate model. The throughput maximization problem is studied in Section VI. An online algorithm and simulation results are presented in Section VII. Section VIII concludes this paper.

II. Problem Formulation

Let $P = \{P_1, P_2, \ldots, P_n\}$ be a set of n packets to be transmitted. Each packet $P_i = (B_i, a_i, d_i)$ has a size B_i, an arrival time a_i and an individual deadline $d_i (> a_i)$. Packets are sorted so that $a_1 \leq a_2 \leq \ldots \leq a_n$. We assume that packet deadlines follow the order packets arrive, that is $d_1 \leq d_2 \leq \ldots \leq d_n$. This is a reasonable assumption because, in most practical systems, packets wait in queue and are transmitted following FIFO rule so packets completion times are surely in the same order as their arrival times. As a result, the EDF schedule and the FIFO schedule are identical in this paper. Let $T = d_n$ be the last deadline. The transmission of packet P_i can start only after its arrival time a_i and must finish before its deadline d_i. This is called the *causality constraint* [9].

Following the previous works [8]–[16], [19]–[22], let $H = \{H_1, H_2, \ldots, H_m\}$ be a set of m energy harvesting instances (or *harvestings* for short). A harvesting $H_i = (E_i, c_i)$, $1 \leq i \leq m$, means that at time $t = c_i$, the amount E_i of energy is harvested by the transmitter. We assume $0 < c_1 < c_2 < \ldots < c_m \leq T$. The initial energy in the battery of the transmitter is treated as a harvesting H_0 that occurs at time $t = 0$. Since the amount of energy from each harvesting is limited and small compared to the battery size, for simplicity, we assume that the battery is large enough to store all harvested energy. Thus, we assume that, from 0 to any moment t, the amount of consumed energy cannot exceed the total amount of energy harvested. This is called the *energy constraint*.

For each harvesting $H_i = (E_i, c_i)$, $1 \leq i \leq m$, we say that *a harvesting event* occurs at time $t = c_i$ and c_i is called *a harvesting (event) point*. Similarly, arrival time a_i and deadline d_i, $1 \leq i \leq n$, are called *an arrival event/point* and *a deadline event/point*. Therefore, from time 0 to T, there are $(m + 2n)$ event points, $e_i, i = 1, 2, \ldots, m + 2n$ where $e_1 \leq e_2 \leq \ldots \leq$

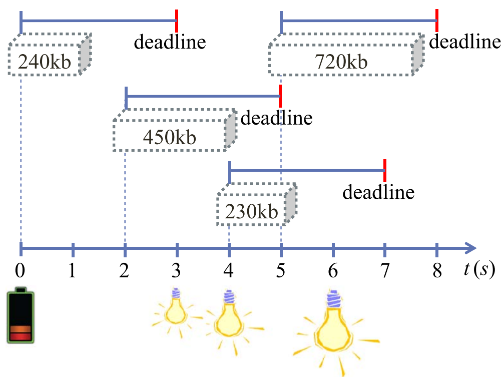


Fig. 1. An example of four packets, P_1 (240 kb, 0 s, 3 s), P_2 (450 kb, 2 s, 5 s), P_3 (230 kb, 4 s, 7 s) and P_4 (720 kb, 5 s, 8 s), and four harvestings, H_0 (2.85 mJ, 0 s), H_1 (1.09 mJ, 3 s), H_2 (3.78 mJ, 4 s) and H_3 (4.80 mJ, 6 s) (initial energy is treated as harvesting H_0 occurring at 0 s).

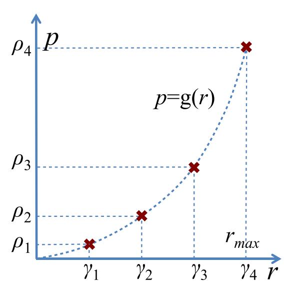


Fig. 2. An illustration of the discrete rate model, where a transmitter can use only one of a set of allowable rates.

$e_{m+2n} = d_n = T$. The time interval between two adjacent event points is called an *epoch*. It is possible that an epoch has a zero length if two event points occur at the same time. Fig. 1 is an example in which there are four packets and four energy harvestings.

We consider a single user point-to-point transmission channel [1]–[6], [8]–[12] and make the same assumption as previous works that the transmitter can adaptively change its transmission rate r, which is related to transmission power p through a function $p = g(r)$. The function $p = g(r)$ is called the *rate-power* function and is convex [1]–[6], [8]–[12]. The convexity of rate-power function is satisfied in many systems with realistic encoding/decoding schemes, such as the optimal random coding in single-user additive White Gaussian Noise (AWGN) channel, where $r = g^{-1}(p) = \frac{1}{2}\log(1+\frac{p}{N})$ and N is the thermal noise level and often assumed $N = 1$ [9].

Previous researchers have assumed that the transmission rate is continuously adjustable. The continuous rate model is not always practical. We thus introduce the *discrete rate model*. Let $R = \{\gamma_0 = 0, \gamma_1, \ldots, \gamma_s = r_{max}\}$ be the set of allowable transmission rates, and $\{\rho_0 = 0, \rho_1, \ldots, \rho_s\}$ be the set of corresponding power levels, such that $\rho_i = g(\gamma_i), i = 1, 2, \ldots, s$. Such a discrete rate model is illustrated in Fig. 2. The discrete rate model is more difficult to deal with than the continuous rate model since it has more restrictions.

Definition 1: The *packet transmission rate scheduler*$_{P_i}(t)$: $[0,T) \to \{\gamma_0, \gamma_1, \ldots, \gamma_s\}$ is defined as the transmission rate for packet P_i at time t, $0 \le t < T, i = 1, 2, \ldots, n$.

Thus, the causality constraints can be expressed as

$$\int_0^T r_{P_i}(t)\mathrm{d}t = \int_{a_i}^{d_i} r_{P_i}(t)\mathrm{d}t = B_i, \quad i = 1, 2, \ldots, n. \tag{1}$$

Definition 2: The *overall rate scheduler*$(t), 0 \le t < T$, is defined as the sum of all packet transmission rates, $r(t) = \sum_{i=1}^{n} r_{P_i}(t), 0 \le t < T$.

An overall rate schedule $r(t)$ uniquely determines the transmission rate at time t for transmitting the packets in FIFO order. Given an overall rate schedule $r(t)$, the corresponding energy consumption from 0 to time t can be calculated by $E(t) = \int_0^t g(r(x))\mathrm{d}x$. So the energy constraint at time t can be expressed as

$$E(t) = \int_0^t g(r(x))\,\mathrm{d}x \le \sum_{i:c_i<t} E_i, \quad t \in [0, T). \tag{2}$$

Definition 3: Given a set of packets P, a set of harvestings H, and a set of allowabel rates R, as described above, a set of n rate schedules $\{r_{P_1}(t), r_{P_2}(t), \ldots, r_{P_n}(t)\}$ is called a *feasible solution* if both the causality constraints (1) and the energy constraints (2) are satisfied. The corresponding overall rate schedule is called the *feasible rate schedule*.

Now, our *min-E* problem is defined as follows.

Definition 4: (*min-E problem*) The *energy minimization problem* is to find a feasible solution whose corresponding overall rate schedule $r(t)$ minimizes the total energy consumption $E(T)$. This overall rate schedule is called an *optimal rate schedule*, denoted $r^{opt}(t)$ (or r^{opt} if no ambiguity arises).

Since *min-E problem* may have no feasible solution if the harvested energy is insufficient to transmit all packets, the causality constraints (1) are thus impossible to be met. If this is the case, we will look at the *max-T* problem which aims to maximize the throughput with the limited available energy. For the *max-T* problem, (1) is not required, but relaxed to the following inequality instead of equality.

$$\int_0^T r_{P_i}(t)\mathrm{d}t = \int_{a_i}^{d_i} r_{P_i}(t)\mathrm{d}t \le B_i, i = 1, 2, \ldots, n \tag{3}$$

We assume that data is transmitted in bits in this case, so that any amount of data can be dropped. The actual amount of data transmitted during $[0, T)$ can be calculated by the following integration,

$$B = \int_0^T r(t)\mathrm{d}t. \tag{4}$$

Now, our *max-T problem* is defined as follows.

Definition 5: (*max-T problem*) The *throughput maximization problem* is to find an overall rate schedule $r(t)$ such that the actual amount of data transmitted determined by (4), is maximized, while the constraints (2) and (3) are satisfied.

Before we present our algorithms in the next section, we need to introduce a closely related known problem which assumes the continuous rate model with no energy harvesting.

Definition 6: ([6], [4]) Under the continuous rate model and a packet model described above, the *delay constrained packet transmission problem* is to find an overall rate schedule $r(t)$

such that the energy consumption, $E(T)$, is minimized subject to causality constraints of (1).

No harvesting is assumed in the delay constrained packet transmission problem defined in Definition 6, where the primary goal is to minimize the total energy consumption with an implicit assumption that the initial energy is infinitely large. We call the optimal overall rate schedule for this problem the *ZM rate schedule* and denoted $r^{zm}(t)$ (or r^{zm} for short). r^{zm} can be computed by the algorithm proposed by Zafer and Modiano in [6] or the algorithm in [4]. A brief description of this algorithm is given in Appendix A.

III. Optimal Rate Scheduling for Packets With a Common Deadline

We start with a simplified *min-E problem*, assuming all packets share a common deadline ($d_1 = d_2 = \ldots = d_n = T$) and adopting continuous rate model ($0 \le r \le r_{max}$). We call this simplified problem *common deadline min-E problem*. Before introducing our *Truncation* method that computes the optimal rate schedule r^{opt}, we first investigate what properties an optimal rate schedule must satisfy.

A. Basic Properties of Optimal Rate Schedule

It has been shown in [9] that, in any epoch $[e_k, e_{k+1}), k = 1, 2, \ldots, m+2n-1$, only one constant transmission rate should be used because of the convexity of the rate-power function. If two rates $r_p \neq r_q$ were used, we can always find a single rate r between these two rates, $r_p < r < r_q$, or $r_p < r < r_q$, to transmit the same amount of data with less energy, or to transmit more data with the same amount of energy. This method is called *equalization*, and accordingly, two rates are said to be *equalized* [9].

Therefore, $r^{opt}(t)$ is a step function which remains constant in each epoch. However, r^{opt} may change from an epoch to the next epoch, which follows the properties below.

Lemma 1: r^{opt} increases only at an arrival point or a harvesting point.

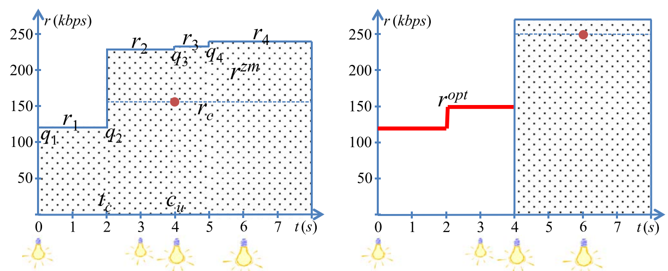
A proof by contradiction can be easily established for Lemma 1 by applying equalization on two adjacent epochs where rate changes occur. We omit the details. Therefore r^{opt} for *common deadline min-E problem* will not decrease until common deadline T.

Lemma 2: If r^{opt} increases at an even point t, then either all packets arrived before point t have been completely transmitted before point t, or all harvested energy before point t has been used up. Accordingly, point t is called a *data critical point* or *energy critical point* of r^{opt}, respectively.

Proof: We prove by contradiction. Suppose r^{opt} increases at t, but both data and energy have non-zero amount available at t, then the epochs immediately before and after t can be equalized, which contradicts the optimality of r^{opt}. Thus, this lemma is correct. □

A similar result for a simpler model is presented in [9].

From Lemma 1 and Lemma 2, it is clear that a data critical point $t < T$ must be an arrival point and an energy critical point $t < T$ must be a harvesting point.



(a) Truncated r^{zm} by r_c. (b) Next iteration.

Fig. 3. An illustration of *Truncation* method. (a) If harvested energy cannot support r^{zm} at all time instances, the higher rate is truncated. The truncation is to find the largest r_c such that the rate $r(t) = \min\{r^{zm}, r_c\}$ can be supported by the harvestings. There must be an energy critical point (the dot). (b) The truncated rate before this dot is optimal (bold line). The same problem repeats in the next iteration.

Note that, the energy in our discussion is used for transmission only. There should be a minimum energy reserved to keep the device operational even at or after an energy critical point. This portion of energy is not counted or considered in this study.

Lemma 3: If t is the first energy critical point of r^{opt}, then any rate schedule that maximizes the data transmission in interval $[0, t)$ must be identical to the optimal rate schedule. In other words, the optimal rate schedule is an unique schedule that can maximize data transmission throughput in $[0, t)$.

Proof: See Appendix B. □

Thus, the key to computing r^{opt} is to find its first energy critical point.

B. Truncation Method

The high level idea of the *Truncation* method is as follows. We first compute the r^{zm} that minimizes the total transmission energy with the assumption that the initial energy is sufficiently large and no harvesting is needed [6]. If, at any time $t \in [0, T)$, harvested energy is sufficient to support r^{zm}, then r^{zm} already minimizes the energy consumption and we have $r^{opt} = r^{zm}$. Otherwise, harvestings are insufficient to support r^{zm}. Therefore, we truncate r^{zm} to $\min\{r^{zm}, r_c\}$, where the horizontal line with rate r_c cuts off r^{zm} as shown in Fig. 3. Clearly, as long as r_c is small enough, the truncated rate can be sufficiently supported by the harvestings. However, if r_c is too small, then less data would be transmitted before the critical time, which results in more data to be transmitted later at higher rate with higher energy consumption. Thus, we want to find the largest rate r_c such that rate $= \min\{r^{zm}, r_c\}$ (called *the rate r^{zm} truncated by r_c*) can be supported by the harvestings. Obviously, in such a rate schedule, there is a point by which all harvested energy is used up but a new harvesting occurs right at this point. This point will be proved to be the first energy critical point, and we will claim that the rate r^{zm} truncated by r_c before this point is optimal. Starting at this new point, the same problem repeats after updating the packet set.

We now discuss in detail the *Truncation* method.

Suppose $r_1, r_2, \ldots, r_s$ are s different rates used by r^{zm} in order and the interval with rate r_i starts at q_i and ends at q_{i+1}. Two known results [2], [6] are: (1) $r_1 < r_2 < \ldots < r_k$; (2) At any q_i, arrived packets must have been delivered. More details are in Appendix A.

Since r^{zm} requires a large initial energy while our model relies upon harvesting, we need a procedure *energy-constraint-check* to check whether r^{zm} is sufficiently supported by the harvestings at all time instances. This can be done by checking all epochs in chronological order as follows. Initially, we set variables $e = h = 0$. For the current epoch, if the beginning event of the epoch is a harvesting, then add the received energy from this harvesting to variable h. The energy consumption during this epoch is added to variable e. If $e < h$, the energy constraints are not violated till the end of this epoch, and the procedure thus goes to the next epoch. If none of the epochs violates the energy constraints, we say r^{zm} passes the *energy-constraint-check*. Otherwise, the procedure returns 'fail'.

If r^{zm} passes the *energy-constraint-check*, then it is already the optimal solution to the *common deadline min-E problem*, because rate schedule r^{zm} minimizes the transmission energy. In case r^{zm} fails, we propose Algorithm TRUNCATION to compute the rate r_c and the first energy critical point c_u. An illustration of the *Truncation* method is shown in Fig. 3.

Algorithm 1 TRUNCATION(P, H, r^{zm}, t)

1: let $r_1, r_2, \ldots, r_s$ be rates of r^{zm} in $[t, T)$
2: let q_i be the start point of the interval with r_i
3: let $r_0 \leftarrow 0, H_{m+1} \leftarrow (0, d_n)$ for loop propose
4: **for** $i \leftarrow 1$ to s**do**
5: let $r(t)$ be the rate r^{zm} truncated by r_i
6: **if** $r(t)$ fails the *energy-constraint-check* **then**
7: break//exit the **for** loop
8: **end if**
9: **end for**
10: $t_c \leftarrow q_i$
11: let $E(t_c)$ be the energy consumed by r^{zm} before t_c
12: $u \leftarrow arg\min_{j: c_j > t_c} \left\{ \frac{\sum_{i=0}^{j-1} E_i - E(t_c)}{c_j - t_c} \right\}$
13: $r_c \leftarrow g^{-1} \left(\frac{\sum_{j=0}^{u-1} E_j - E(t_c)}{c_u - t_c} \right)$
14: let $r(t) \leftarrow \min\{r^{zm}, r_c\}$
15: return $(r(t), c_u)$

The **for** loop determines index i such that $r_{i-1} \le r_c < r_i$. Let $t_c = q_i$ and $E(t_c)$ be the energy consumed by r^{zm} before t_c. For each harvesting event point $c_j > t_c$, let $H(c_j)$ be the total harvested energy in $[0, c_j)$ (not counting c_j), then $E = H(c_j) - E(t_c)$ is the maximum energy we can use in $[t_c, c_j)$. If we use a constant rate in $[t_c, c_j)$, then the largest rate we can use is $r_{c_j} = g^{-1}(E/(c_j - t_c))$. We are looking for a harvesting point such that r_{c_j} is the minimum. We use this rate as the r_c in truncation. This r_c is the largest because any rate that is larger will consume all energy at some point before a harvesting point. Equation (5) directly computes such point c_u and rate r_c.

$$u = \arg \min_{j: c_j > t_c} \left\{ \frac{\sum_{i=0}^{j-1} E_i - E(t_c)}{c_j - t_c} \right\}$$
$$r_c = g^{-1} \left(\frac{\sum_{i=1}^{u-1} E_i - E(t_c)}{c_u - t_c} \right) \quad (5)$$

Theorem 1: Algorithm TRUNCATION correctly computes the first energy critical point c_u of r^{opt} if it exists; the rate r^{zm} truncated by r_c is optimal in $[0, c_u)$.

Proof: We first prove c_u is the first energy critical point of r^{opt} by contradiction. Suppose c_v is the first energy critical point in the optimal schedule, but $v \neq u$. According to (5), $\frac{\sum_{j=1}^{v-1} E_j - E(t_c)}{c_v - t_c} > \frac{\sum_{j=1}^{u-1} E_j - E(t_c)}{c_u - t_c}$, and we have $g^{-1}(\frac{\sum_{j=1}^{v-1} E_j - E(t_c)}{c_v - t_c}) \ge r_c$. If $v < u$, then there must be an epoch p before c_v in which $r^{opt} > r_c$, because all the energy is used up at c_v in rate schedule r^{opt} by the assumption. Consequently, there must be an epoch q in $[c_v, c_u)$ such that $r^{opt} < r_c$. Equalization can thus be applied to epoch p and q to reduce the energy consumption of r^{opt}, which is a contradiction. If $v > u$, there must be an epoch p before c_u in which $r^{opt} < r_c$, because some energy remains at c_u in rate schedule r^{opt}. This leftover energy at c_u by r^{opt} also implies there must be an epoch q in $[c_u, c_v)$ in which $r^{opt} > r_c$. Equalization can thus be applied to epoch p and q, which is a contradiction. Hence, c_u is the first energy critical point in the optimal solution.

Let $r(t)$ be the rate r^{zm} truncated by r_c. According to Appendix A, all packets that arrived in $[0, t_c)$ must have been completely transmitted in $[0, t_c)$ by $r(t)$. Any schedule can not transmit more data in $[0, t_c)$, and the rate $r(t)$ achieves this with the minimum energy consumption [2], [6]. Therefore, rate $r(t)$ has the largest energy to use in $[t_c, c_u)$. Since a constant rate is used by $r(t)$ in $[t_c, c_u)$ which produces the largest throughput with the same amount of energy, $r(t)$ transmits the maximum amount of data in $[0, c_u)$. According to Lemma 3, $r(t)$ is identical to r^{opt} in $[0, c_u)$. □

By repeatedly invoking Algorithm TRUNCATION, we compute the optimal rate schedule by Algorithm COMMON-TRUNCATION. Note that we have assumed that the peak rate is restricted to r_{max}. Thus, in every iteration, we need to make sure that the rate in every epoch does not exceed r_{max} (Line 14). In the last iteration, if r^{zm} passes the *energy-constraint-check*, we still need to check whether $r^{zm} < r_{max}$ in all epochs. If it passes, then we set r^{opt} to be r^{zm} and all data can be delivered; otherwise, we set r^{opt} not to exceed r_{max}, and, of course, the data cannot be transmitted completely no matter how much energy is available.

Alglorithm 2 COMMON-TRUNCATION(P, H, r_{max})

1: let $r^{opt} \leftarrow 0$ in $[0, T)$; $t \leftarrow 0$
2: **while** $t < T$ **do**
3: let r^{zm} be the optimal *ZM rate schedule*
4: if r^{zm} passes the *energy-constraint-check* **then**
5: **if** $r^{zm} < r_{max}$ in all epochs **then**
6: let $r^{opt} \leftarrow r^{zm}$ in $[t, T)$
7: return (r^{opt}, *all-sent*)
8: **else**
9: let $r^{opt} \leftarrow \min\{r^{zm}, r_{max}\}$ in $[t, T)$
10: return (r^{opt}, *partially-sent*)
11: **end if**
12: **else**
13: $(r(t), c_u) \leftarrow$ TRUNCATION (P, H, r^{zm}, t)

```
        r^opt(t) ← min{r(t), r_max}, in [t, c_u)
        schedule transmission according to r^opt until c_u
        update packets set P and harvesting set H
        t ← c_u
    end if
end while
return (r^opt, partially-sent)//handle the case when T = c_u
```

Theorem 2: Algorithm COMMON-TRUNCATION computes the optimal rate r^{opt} for the *common deadline min-E problem*, if *all-sent* is returned.

Proof: Theorem 1 proves the optimality of the partial schedule returned by Algorithm TRUNCATION. The Truncation method guarantees the maximum amount of data to be transmitted before every energy critical point, and thus the leftover data at the energy critical point is minimized. In the last iteration, *all-sent* is returned only if the optimal ZM rate schedule passes the *energy-constraint-check* and all rates are lower than r_{max}. Thus the energy consumption in the last iteration is also minimized. This verifies the optimality of the algorithm. □

IV. OPTIMAL RATE SCHEDULING FOR PACKETS WITH INDIVIDUAL DEADLINES

In this section, we still consider continuous rate model but allow packets to have individual deadlines. We call this new problem *individual deadline min-E problem*. Recall Fig. 1 for an example.

A. Optimality Properties

In the *individual deadline min-E problem*, Lemma 1, 2, and 3 still holds. Since the deadline events may occur at any time in duration $[0,T)$, we introduce the following lemma as a supplement to Lemma 1 and 2.

Lemma 4: r^{opt} decreases only at a deadline point. If r^{opt} decreases at a deadline point d_k, then exactly packets $\{P_1,P_2,\dots,P_k\}$ are delivered before d_k and no transmission has been started for other packets. This point d_k is called a *delay critical point*.

A proof by contradiction can be easily established using equalization argument for above lemmas. We omit the details. Actually, there is a stronger result about *equalization* as stated in Theorem 3.

Theorem 3: A rate schedule is optimal if and only if no two epochs can be equalized.

Proof: See Appendix C. □

We now define a sub-problem and study its optimality properties which will be utilized later.

Definition 7: An *i*-optimal rate schedule $r^{opt(i)}(t)$ (or $r^{opt(i)}$ for short) is an optimal rate schedule to transmit the set of packets $\{P_1,P_2,\dots,P_i\}$, where $1 \le i \le n$.

Obviously, we have $r^{opt(n)} = r^{opt}$. Without loss of generality, we assume $r^{opt(i)} = 0$ in interval $[d_i,T)$ where d_i is the last deadline for the packets $\{P_1,P_2,\dots,P_i\}$.

Theorem 4: The rate in the $(i+1)$-optimal rate schedule is higher than or equal to that of the *i*-optimal rate schedule in any epoch, i.e., $r^{opt(i+1)} \ge r^{opt(i)}$, $i = 1,2,\dots,n-1$.

Proof: See Appendix D. □

Based on the above theorem, we have the following result.

Lemma 5: Before any of its data/energy critical point, rate schedule $r^{opt(i)}$ is identical to $r^{opt(i+1)}$, $i = 1,2,\dots,n-1$.

Proof: If t is an energy critical point, all energy must have been used up by t in rate schedule $r^{opt(i)}$, while $r^{opt(i+1)} \ge r^{opt(i)}$ in all epochs. Thus, the two rates must be identical before time t. Similarly, if t is a data critical point, all data arrived must have been delivered by t in rate schedule $r^{opt(i)}$, while $r^{opt(i+1)} \ge r^{opt(i)}$ in all epochs. Thus, the two rate schedules must be identical before time t. □

There is an immediate theorem that relates the original problem and the sub-problems.

Theorem 5: Before any of its data/energy critical points, rate schedule $r^{opt(i)}$ is identical to r^{opt}, $i = 1,2,\dots,n$.

We now define another similar sub-problem and give its optimality properties.

Definition 8: An *i-ZM optimal rate scheduler*$^{zm(i)}(t)$ (or $r^{zm(i)}$ for short) is an optimal *ZM rate schedule* to transmit the set of packets $\{P_1,P_2,\dots,P_i\}$, where $1 \le i \le n$.

Theorem 6: $r^{zm(i+1)} \ge r^{zm(i)}$, $i = 1,2,\dots,n-1$.

It follows the fact that the *i-ZM optimal rate schedule* problem is a special case of our min-E problem. Obviously, we have $r^{zm(n)} = r^{zm}$.

B. Generalized Truncation Method

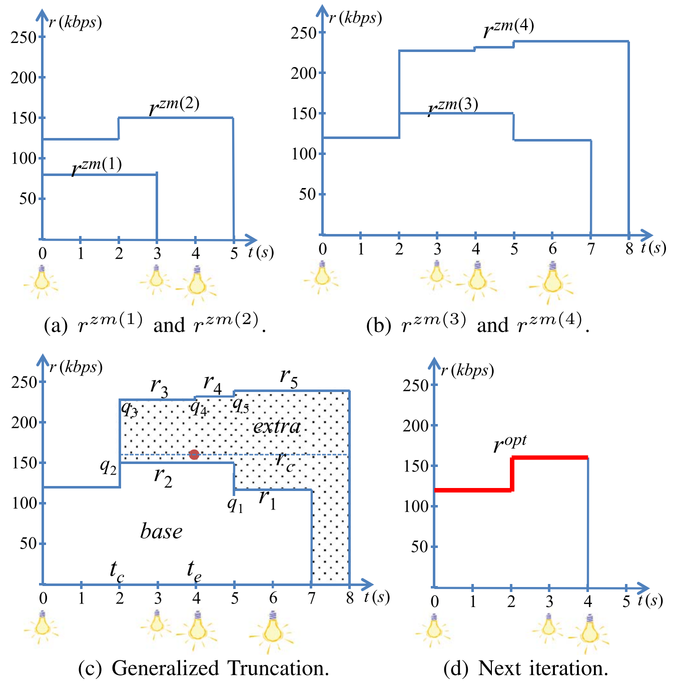
Recall that the idea of Truncation is to compute the largest rate r_c that truncates r^{zm} so that it can be sufficiently supported by harvestings until the first energy critical point. The generalized method follows the similar strategy, but in an indirect way as follows.

Firstly, we perform the *energy-constraint-check* for $r^{zm(i)}$, $i = 1,2,\dots,n$. If all $r^{zm(i)}$ pass, then $r^{opt} = r^{zm(n)}$ and the problem is solved. Otherwise, we will find the smallest index k such that $r^{zm(k)}$ fails. Then, the harvestings would be insufficient to support $r^{zm(k)}$, and truncation becomes necessary. Let procedure SMALLEST-K return this value k.

On one hand, $r^{zm(k)}$ fails the *energy-constraint-check* and it needs to be truncated. On the other hand, $r^{zm(k-1)}$ passes the *energy-constraint-check* and it is the optimal rate schedule to transmit packets $\{P_1,P_2,\dots,P_{k-1}\}$, i.e., $r^{opt(k-1)} = r^{zm(k-1)}$. By Theorem 4, we have $r^{opt(k)} \ge r^{opt(k-1)} = r^{zm(k-1)}$. That is to say, the truncated $r^{zm(k)}$ should not be less than $r^{zm(k-1)}$. Therefore, rate schedule $r^{zm(k-1)}$ is called the *base*. The difference between the *base* and $r^{zm(k)}$ is called the *extra*. The *base* should not be affected by the truncation, and only *extra* should be truncated.

Secondly, we identify the first point t such that $r^{zm(k-1)}(t) < r^{zm(k)}(t)$, which can be easily done by checking their rates in each epoch. We claim that, after point t, (1) The rate of $r^{zm(k)}$ never decreases until d_k, the deadline of P_k; (2) The rate of $r^{zm(k-1)}$ never increases until d_{k-1}, the deadline of P_{k-1}.

Claim (1) can be proved by contradiction. Suppose $r^{zm(k)}$ decreases first time at $t' > t$ and $t' < d_k$. According to Appendix A, t' is a deadline point and all packets with a deadline t' or earlier have been delivered by $r^{zm(k)}$ before t' but no transmission has been started for other packets. However all those transmitted packets must be also delivered by any feasible



(a) $r^{zm(1)}$ and $r^{zm(2)}$. (b) $r^{zm(3)}$ and $r^{zm(4)}$. (c) Generalized Truncation. (d) Next iteration.

Fig. 4. An illustration of Generalized Truncation method. $r^{zm(1)}$ and $r^{zm(2)}$ are shown in (a) and $r^{zm(3)}$ and $r^{zm(4)}$ are shown in (b). Since $k=3$, the shapes of *base* and *extra* are illustrated in (c). The truncation is to find the highest constant rate r_c represented by a straight line in the rate diagram that cuts the *extra* part such that the rate combining the *base* and the truncated *extra* can be sufficiently supported by the harvestings. In this rate schedule, there is a point at which all harvested energy is used up but a new harvesting occurs right at this point (the dot). This point is the first energy critical point. The truncated rate schedule is optimal upto this energy critical point, as the bold line shown in (d).

schedule, including $r^{zm(k-1)}$. This contradicts to the fact that $r^{zm(k-1)}(t) < r^{zm(k)}(t)$.

Claim (2) can also be proved by contradiction. Suppose $r^{zm(k-1)}$ increases first time at $t' > t$ and $t' < d_{k-1}$. According to Appendix A, t' is an arrival time, and all packets with an earlier arrival time have been delivered by $r^{zm(k-1)}$ at time t', meaning that no feasible schedule can transmit more data than $r^{zm(k-1)}$ before time t'. This contradicts to the fact that $r^{zm(k-1)}(t) < r^{zm(k)}(t)$ also.

From these two claims, the shape of *base*, the rate function $r^{zm(k-1)}$ after t looks like a downward staircase, as shown in Fig. 4(c). The rate function $r^{zm(k)}$ after t looks like an upward staircase, as shown in Fig. 4(c) also.

Thirdly, we compute the position of truncation. We will use a straight line with rate r_c to cut off the *extra*, as shown in Fig. 4. Our goal is to find the largest r_c such that this rate can be sufficiently supported by the harvestings at all time instances until d_k. According to the above two claims, we assume $r^{zm(k-1)}(t) = r_d > r_{d-1} > \ldots > r_1$ are the rates used by $r^{zm(k-1)}$ from point t to d_{k-1}. Let $r^{zm(k)}(t) = r_{d+1} < r_{d+2} < \ldots < r_s$ be the rates used by $r^{zm(k)}$ from point t to d_k. Since $r^{zm(k-1)}(t) < r^{zm(k)}(t)$, we have $r_1 < r_2 < \ldots < r_d < r_{d+1} < r_{d+2} < \ldots < r_s$. Let the starting time for each rate r_i be q_i, $1 \le i \le s$. We will decide between which two rates r_c should be located by checking each rate in $\{r_1, r_2, \ldots, r_s\}$ to find the first one r_i that fails the *energy-constraint-check*, which gives range $r_{i-1} < r_c \le r_i$.

If $r_i \le r_d$, then r_c starts from point q_{i-1}, otherwise, $r_d < r_i$ and r_c starts from q_i. We use t_c to denote the starting point of r_c. Then, the exact value of r_c can be determined in the same way as we discussed in Section III. A detailed pseudo code is presented in Algorithm GENERAL-TRUNCATION.

Algorithm 3 GENERAL-TRUNCATION$(P, H, r^{zm(k-1)}, r^{zm(k)})$

```
let t be the first point such that r^{zm(k-1)}(t) < r^{zm(k)}(t);
   r_d > r_{d-1} > ... > r_1 be the rates used by r^{zm(k-1)} from
   point t to d_{k-1}; r_{d+1} < r_{d+2} < ... < r_s be the rates used
   by r^{zm(k)} from point t to d_k
let the starting time for each rate r_i be q_i, 1 <= i <= s
let r_0 <- 0, q_0 <- d_k, H_{m+1} <- (0, d_n) for loop propose
for i <- 1 to s do
   let r(t) be the rate of base plus the extra truncated by r_i
   if r(t) fails the energy-constraint-check then
      break//exit the for loop
   end if
end for
if r_i <= r_d then
   t_c <- q_{i-1}
   E(t_c) <- energy consumed by r^{zm(k-1)} before t_c
else
   t_c <- q_i
   E(t_c) <- energy consumed by r^{zm(k)} before t_c
end if
u <- arg min_{j:c_j>t_c} { (sum_{i=0}^{j-1} E_i - E(t_c)) / (c_j - t_c) }
r_c <- g^{-1}( (sum_{i=0}^{u-1} E_i - E(t_c)) / (c_u - t_c) )
let r(t) be the rate r^{zm} truncated by r_c
return (r(t), c_u)
```

Example We take the example in Fig. 1 to illustrate the generalized truncation method in Fig. 4. The rate-power function is $r = g^{-1}(p) = 10^3 \log(1 + 0.1p)$, where p is the transmission power in milliwatts and r is the transmission rate in kilobits per second (kbps). It can be verified that $r^{zm(3)}$ passes the *energy-constraint-check*, while $r^{zm(4)}$ fails. Thus $k = 3$. The shapes of *base* and *extra* are illustrated in (c). The rates of *extra* are respectively $r_1 = 115$ starting at $q_1 = 5$, $r_2 = 150$ starting at $q_2 = 2$, $r_3 = 225$ starting at $q_3 = 2$, $r_4 = 230$ starting at $q_4 = 4$, and $r_5 = 240$ starting at $q_5 = 5$. The rate that combines the *base* and the *extra* truncated by r_1 is 120 in [0, 2), 150 in [2, 5) and 115 in [5, 8), and it passes the check;the rate truncated by r_2 is 120 in [0, 2), 150 in [2, 8), and it passes the check; the rate truncated by r_3 is 120 in [0, 2), 225 in [2, 8), and it fails the check. Thus, $r_2 \le r_c < r_3$ and $t_c = 2$. The energy consumption before $t_c = 2$ is $E(t_c) = 2g(120) = 1.73$ (mJ). We check every harvesting point: $c_1 : \frac{E_0 - 1.73}{c_1 - 2} = 1.12$, $c_2 : \frac{E_0 + E_1 - 1.73}{c_2 - 2} = 1.10$, $c_3 : \frac{E_0 + E_1 + E_2 - 1.73}{c_3 - 2} = 1.50$, and $c_4 : \frac{E_0 + E_1 + E_2 + E_3 - 22.3}{c_4 - 2} = 1.80$. Note that $H_4 = (0, 8)$ is a virtual harvesting defined before the **for** loop. Obviously, we have $u = 2$, $r_c = g^{-1}(1.10) = 151$ and $c_u = c_2 = 4$. The optimal rate is 120 in [0, 2) and 151 in [2, 4) as shown in Fig. 4(d). After updating the packet set, the same problem repeats by starting at time $c_2 = 4$.

Theorem 7: Algorithm GENERAL-TRUNCATION correctly computes the first energy critical point c_u of r^{opt}, if it exists; the rate determined by the truncation is identical to r^{opt} in $[0, c_u)$.

Proof: The computation of c_u is based on (5). As proved by Theorem 1, c_u is the first energy critical point, which means that all harvested energy will be comsumed by the truncated rate at c_u and a new harvesting occurs at c_u. Moreover, the constant rate r_c is the highest such that the rate determined by the truncation can be sufficiently supported by the harvestings in $[0, c_u)$. Now, we show the rate determined by the truncation is identical to r^{opt} in $[0, c_u)$.

Let $r(t)$ be the rate determined by the truncation which is the rate of *base* plus the *extra* truncated by r_c. By Theorem 3, we only need to show that no two epochs can be equalized in $[0, c_u)$. We now show this is true.

Let $p < q$ be two epochs in $[0, c_u)$ which have rates r_p and r_q respectively. If $r_p = r_q$, then p and q cannot be equalized. If $r_p < r_q$, then let x be the first point after epoch p such that the rate increases at x. Point x must occur at $r^{zm(k)}$ because, from the two claims we discussed above, $r^{zm(k)} \neq r^{zm(k-1)}$ starts at point t and $r^{zm(k-1)}$ will never increase after t. According to Appendix A, data runs out at x, so r_p can not be increased by equalization. If $r_p > r_q$, let x be the first point after epoch p such that the rate decreases at x. Similarly, we can argue that point x must occur at $r^{zm(k-1)}$. According to Appendix A, x must be a delay critical point of $r^{zm(k-1)}$. Thus, r_p can not be decreased by equalization, because otherwise some data would miss a deadline. Therefore, we conclude that $r(t)$ is optimal before c_u. □

After a truncation is done and the energy critical point c_u is computed, packets in the queue are transmitted in FIFO order using this scheduled rate upto this critical point c_u. Then, we use this critical point c_u as a new initial point and compute the remaining packets. Then truncation method repeats until no packets remain. By doing so, the *individual deadline min-E problem* is solved. A pseudo code is given in Algorithm INDIVIDUAL-TRUNCATION.

Algorithm 4 INDIVIDUAL-TRUNCATION(P, H, r_{max})

```
let r^opt ← 0 in [0, T)
t ← 0
while t < T do
    let r^zm be the optimal ZM rate schedule for P
    if r^zm passes the energy-constraint-check then
        if r^zm < r_max in every epoch then
            r^opt ← r^zm in [t, T)
            return r(t)
        else
            return 'non-exist'
        end if
    else
        k ← SMALLEST-K(P, H)
        (r(t), c_u) ← General-Truncation(H, r^zm(k-1), r^zm(k))
        r^opt ← min{r(t), r_max}, in [t, c_u)
        schedule the transmission according to r^opt until c_u
        update packets set P and harvesting set H
        t ← c_u
    end if
end while
return 'non-exist' //handle the case when T = c_u
```

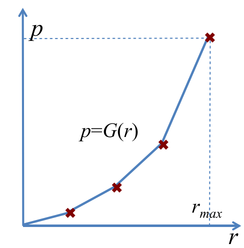


Fig. 5. The curve of new rate-power function $p = G(r)$ that connects discrete points $(\gamma_i, g(\gamma_i)), i = 0, 1, 2, \ldots, s$.

Theorem 8: Algorithm INDIVIDUAL-TRUNCATION computes the optimal rate schedule for the individual min-E problem in $O(mn(n^2+m))$ time.

Proof: The correctness of the algorithm follows by applying Theorem 7 to each truncation in Algorithm INDIVIDUAL-TRUNCATION. The computational complexity is analyzed as follows. The **while** loop repeats at most m times because at least one energy critical point is located in each loop, and there are at most m energy critical points. Within each **while** loop, the most time consuming step is the procedure SMALLEST-K. SMALLEST-K needs at most n times to call procedure *energy-constraint-check* after computing r^{zm}. It is easy to see that the time complexity for computing r^{zm} in [6] is $O(n^2)$. The *energy-constraint-check* checks every epoch and finishes in $O(m+2n)$ steps. Therefore, SMALLEST-K finishes in $O(n(n^2+m))$ steps and the total time complexity is $O(mn(n^2+m))$. □

V. A FRAMEWORK FOR DISCRETE RATE SCHEDULING

In this section, we propose a general framework that transforms a schedule using continuous rate model to a schedule using discrete rate model, while preserving the optimality as long as the rate-power function is convex.

Most rate scheduling algorithms in the literature as well as our algorithm are based on this convexity and the continuous rate model [1]–[6], [8]–[11]. It is the convexity of the rate-power function that yields the fundamental trade-off between transmission delay and energy consumption [1]–[6], or between transmission delay and throughput [8]–[11], which consequently leads to the optimality of these algorithms.

The key idea in our framework is to design a new convex rate-power function $p = G(r)$ such that the discrete rates are embedded in this function. Let $R = \{\gamma_0, \gamma_1, \gamma_2, \ldots, \gamma_s\}$ be the set of allowable rates in discrete rate model such that $0 = \gamma_0 < \gamma_1 < \gamma_2 < \ldots < \gamma_s$, and $\{\rho_0 = 0, \rho_1, \ldots, \rho_s\}$ be the set of corresponding power levels, such that $\rho_i = g(\gamma_i), i = 1, 2, \ldots, s$. As shown in Fig. 5, the allowable rate-power pairs are discrete points $(\gamma_i, g(\gamma_i)), i = 0, 1, 2, \ldots, s$. We connect these discrete points to form a continuous and convex curve, which represents the curve of new rate-power function $p = G(r)$ as shown in

Fig. 5. Obviously, $p = G(r)$ is continuous and convex and can be formulated as follows.

$$p = G(r) = \frac{\gamma_{u+1} - r}{\gamma_{u+1} - \gamma_u} g(\gamma_u) + \frac{r - \gamma_u}{\gamma_{u+1} - \gamma_u} g(\gamma_{u+1}), \quad \text{if } \gamma_u \le r < \gamma_{u+1}. \tag{6}$$

We now describe our two-step framework for above mentioned algorithms in the literature. The *replacing step* is to replace $p = g(r)$ with $p = G(r)$ and run the new algorithm to produce a rate schedule $r(t)$. The *converting step* is to convert $r(t)$ to allowed discrete rates. For each epoch $[e_i, e_{i+1})$, let r be the rate of $r(t)$ in this epoch, where $\gamma_u \le r < \gamma_{u+1}$. We divide it into two intervals, and let rate be γ_u in $[e_i, \frac{r-\gamma_u}{\gamma_{u+1}-\gamma_u} e_i + \frac{\gamma_{u+1}-r}{\gamma_{u+1}-\gamma_u} e_{i+1})$ and let rate be γ_{u+1} in $[\frac{r-\gamma_u}{\gamma_{u+1}-\gamma_u} e_i + \frac{\gamma_{u+1}-r}{\gamma_{u+1}-\gamma_u} e_{i+1}, e_{i+1})$. The combination of the two steps generates a feasible solution for the discrete rate model, while preserving the optimality.

Theorem 9: The framework produces an optimal discrete rate scheduling.

Proof: This theorem is correct because: (1) The rate schedule $r(t)$ produced in the *replacing step* is optimal with continuous function $p = G(r)$. (2) The optimal discrete rate schedule performs no better than the rate schedule $r(t)$. (3) The discrete rate schedule produced in the *converting step* performs the same with $r(t)$.

Claim (1) is correct since $p = G(r)$ is continuous and convex and scheduling algorithms are based on this convexity and the continuous rate model which implies that $r(t)$ is still optimal with $p = G(r)$.

Claim (2) is true because the solution space of the optimal discrete rate schedule is discrete points, while the solution space of $r(t)$ is a continuous piecewise linear function that covers these discrete points. In other word, the solution space of the optimal discrete rate schedule is a subset of that of the $r(t)$. Therefore, this claim is true.

Claim (3) can be proved as follows. In epoch $[e_i, e_{i+1})$, after the *converting step*, the data transmission is $\gamma_u(\frac{r-\gamma_u}{\gamma_{u+1}-\gamma_u} e_i + \frac{\gamma_{u+1}-r}{\gamma_{u+1}-\gamma_u} e_{i+1} - e_i) + \gamma_{u+1}(e_{i+1} - \frac{r-\gamma_u}{\gamma_{u+1}-\gamma_u} e_i - \frac{\gamma_{u+1}-r}{\gamma_{u+1}-\gamma_u} e_{i+1}) = r(e_{i+1} - e_i)$; the energy consumption is $g(\gamma_u)(\frac{r-\gamma_u}{\gamma_{u+1}-\gamma_u} e_i + \frac{\gamma_{u+1}-r}{\gamma_{u+1}-\gamma_u} e_{i+1} - e_i) + g(\gamma_{u+1})(e_{i+1} - \frac{r-\gamma_u}{\gamma_{u+1}-\gamma_u} e_i - \frac{\gamma_{u+1}-r}{\gamma_{u+1}-\gamma_u} e_{i+1}) = (\frac{\gamma_{u+1}-r}{\gamma_{u+1}-\gamma_u} g(\gamma_u) + \frac{r-\gamma_u}{\gamma_{u+1}-\gamma_u} g(\gamma_{u+1}))(e_{i+1} - e_i) = G(r)(e_{i+1} - e_i)$. Both stay the same. □

We take the example in Fig. 1 again, and illustrate how to apply this framework and solve the *min-E problem* in the discrete rate model in Fig. 6.

Example We take the example in Fig. 1 and feature the difference with new function $r = G(p)$. Let $\gamma_1 = 100$, $\gamma_2 = 200$, $\gamma_3 = 300$. So the piecewise linear function $p = G(r)$ is composed of $p = 0.0072r$ if $0 \le r < 100$, $p = 0.0077r - 0.0515$ if $100 \le r < 200$, $p = 0.0082r - 0.1619$ if $200 \le r < 300$. $r^{zm(1)}$ and $r^{zm(2)}$ are given in Fig. 6(a). It can be verified that $r^{zm(1)}$ passes the *energy-constraint-check*, while $r^{zm(2)}$ fails. Thus $k = 1$ with new function G, note that it is $k = 3$ with function g. The rates of the *extra* are $r_1 = 80$ starting at $q_1 = 0$, $r_2 = 120$ starting at $q_2 = 0$, and $r_3 = 150$ starting at $q_3 = 2$.

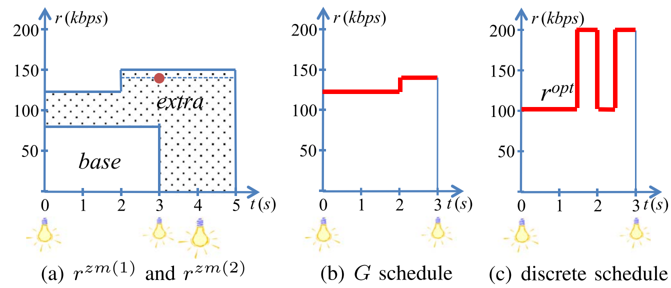


Fig. 6. An illustration of the Truncation method in the discrete rate model. In (a), *extra* (shadow area) is truncated. Bold line in (b) is the rate schedule produced by the *replacing step*. Bold line in (c) is the result of the *converting step*, in which rate is either 1 or 2.

We have $120 < r_c < 150$ and $t_c = \max\{q_2, q_3\} = 2$. $E(t_c) = 2G(120) = 1.74$ (mJ) with new function G, while previously $E(t_c) = 1.73$ (mJ) with function g. According to (5), $c_u = c_1 = 3$ and $r_c = G^{-1}(E_0 - E(t_c)) = 149$, while previously $c_u = 4$. So rate schedule returned by replacing step is 120 in [0, 2) and 149 in [2, 3) as shown in (b). After the converting step, the optimal discrete schedule is 100 (kbps) in [0 s, 1.6 s) and [2 s, 2.51 s), 200 (kbps) in [1.6 s, 2 s) and [2.51 s, 3 s) as shown in (c).

VI. Maximize Transmission Throughput

In case no feasible schedule exists for the *min-E problem*, we study the *max-T problem* which is to find a rate schedule that transmits maximum amount of data for the given set of packets P with the energy harvestings H. We assume the continuous rate model, since an optimal algorithm under continuous rate model can be transformed to an optimal algorithm for the discrete rate model as we discussed in previous section. We first discuss the case where all packets share a common deadline, and then the general *max-T problem*. We assume that the throughput includes the partially transmitted data; when a deadline occurs, the unfinished part is dropped from the queue.

A. Common Deadline Case

The *common deadline max-T problem* assumes all packets sharing a common deadline. In Section III, the *common deadline min-E problem* is solved by Algorithm Common-Truncation. In fact, this algorithm can also identify if no feasible schedules exist for the *min-E problem* and produces the optimal schedule for the *max-T problem* for this case.

Theorem 10: Algorithm Common-Truncation computes the optimal rate schedule for the *common deadline max-T problem* if *partially-sent* is returned at termination.

Proof: This algorithm returns *partially-sent* if the **while** loop ends when $t = T$, or r^{zm} passes the *energy-constraint-check* but the rate in some epoch is higher than r_{max}. The first case means that a truncation occurred in the last iteration before the while loop ends and $t = c_u = T$ is an energy critical point determined by this truncation. From Lemma 3, the data must be maximally transmitted using all harvested energy before T. In the second case, we set $r^{opt} = r_{max}$ if $r^{zm} > r_{max}$, otherwise we set $r^{opt} = r^{zm}$. According to the two properties about r^{zm} in Appendix A, no more data can be transmitted. □

B. Individual Deadline Case

In general *max-T problem*, individual deadlines are allowed. It seems that this problem has more complex structure and does not admit analytical solutions. We thus provide a numerical solution by posing it as a convex program and applying standard convex optimization techniques to solve it within desired error range.

Recall that there are $m+2n$ event points, e_j, $1 \leq j \leq m+2n$, consisting of m harvesting event points, n arrival event points and n deadline event points. They are sorted in the order they occur, $e_1 \leq e_2 \leq \ldots \leq e_{m+2n}$. We use function ξ to map each event point to its rank in this event sequence. For example, if $a_i = e_k$ then $\xi(a_i) = k$. Function ξ is easy to obtain and known before scheduling. Let the length of epoch $[e_j, e_{j+1})$ be l_j, $1 \leq j \leq n+2m-1$.

Since in any epoch only one constant transmission rate should be used, for each packet P_i, we use a constant rate r_{ij}, $1 \leq j \leq n+2m-1$, to denote the rate used in epoch $[e_j, e_{j+1})$ to transmit data of packet P_i. Then, the overall rate used in each epoch can be computed by $r_j = \sum_{i=1}^{n} r_{ij}, j = 1, 2, \ldots, m+2n-1$.

Therefore, the *max-T problem* can be re-defined as follows.

$$\text{max. } B = \sum_{j=1}^{m+2n-1} r_j l_j, \tag{7}$$

$$\text{s.t. } r_j = \sum_{i=1}^{n} r_{ij} \leq r_{max}, j = 1, 2, \ldots, m+2n-1, \tag{8}$$

$$\sum_{j=1}^{m+2n-1} r_{ij} l_j = \sum_{j=\xi(a_i)}^{\xi(d_i)-1} r_{ij} l_j \leq B_i, i = 1, 2, \ldots, n, \tag{9}$$

$$\sum_{j=1}^{\xi(c_k)-1} g(r_j) l_j \leq \sum_{i=1}^{k-1} E_i, k = 2, 3, \ldots, m. \tag{10}$$

Eq. (7) is the actual amount of data transmitted. (9) is the relaxed causality constraints and (10) is the energy constraints. Since $p = g(r)$ is a convex function, this problem is obviously a convex program. Standard convex optimization techniques can be applied to solve it within a desired error. It is worth pointing out however that in the numerical solution, the higher accuracy required, the more iterations (running time) must be involved.

VII. Online Algorithm and Simulations

In this section, based on the truncation method, we develop an online rate scheduling algorithm without any knowledge of distributions of arrival time, deadline, packet size, harvesting time and harvesting amount. We evaluate the performance of our online algorithm by the comparisons with the optimal offline solutions.

A. Online Algorithm

Algorithm 5 Online-Truncation

Step 1: Each remaining packet in FIFO queue is considered as a new packet of which the arrival time is t and the deadline remains the same. If a packet is partially transmitted, its un-transmitted data constitutes its new load.//Note that there is at most one such packet in FIFO queue, while all other data load remains the same.

Step 2: For the new packet set P obtained from step 1, compute optimal rate schedule r^{zm}, assuming the initial energy at t is large enough.//Because packets have a common arrival time t, the rate of r^{zm} is a non-increasing step function according to Appendix A.

Step 3: If initial energy E is not sufficient to support r^{zm}, then apply truncation method to find r_c to truncate r^{zm} such that $r(t) = \min\{r^{zm}, r_c\}$ consumes all E in $[t, d_n)$, where d_n is the last deadline.//As a fact, when continuous rate is available, $r(t)$ maximizes the throughput if no more packets arrive or harvestings occur before d_n according to the discussion in Section IV.

Step 4: Divide each epoch $[e_i, e_{i+1})$ into sub-epochs of a fixed length w except the last sub-epoch, where w is a small adjustable number.//The purpose of the dividing is to avoid transmitting with a constant rate for too long and approach the online adversary of unpredictable packet/harvesting arrival.

Step 5: Convert $r(t)$ in each sub-epoch to an optimal discrete rate schedule using the technique in Section V so that each sub-epoch is then partitioned into two segments using two closest discrete rates, respectively. Let the resulting discrete rate schedule be r^{local}.

Step 6: The online algorithm transmits packets in queue according to r^{local}.

Our online algorithm works in a greedy manner: transmit at the optimal rate schedule computed based on currently known information until a new event occurs (packet arrives or a harvesting occurs). The primary goal of our online algorithm is throughput maximization, however energy minimization is also addressed whenever possible. Let E be the remaining energy at time t. When a new event occurs at time t, the algorithm re-computes a new schedule to be used from time t according to the steps presented in Algorithm Online-Truncation.

Upon a new occurrence of an arrival/harvesting event, our online algorithm repeats the above steps. In order to evaluate the performance of the online algorithm, we have conducted extensive simulations which will be discussed in the following two subsections.

B. Simulation Settings

We compare the performance of our online algorithm with the optimal offline solution, since currently no other existing algorithms have studied the same *min-E problem* or *max-T problem* subject to individual packet deadlines.

Following Yang *et al.* [9], we consider a band-limited additive white Gaussian noise channel, with bandwidth $W = 1$ MHz and the noise power spectral density $N_0 = 10^{-19}$ W/Hz. We assume that the distance between the transmitter and the receiver

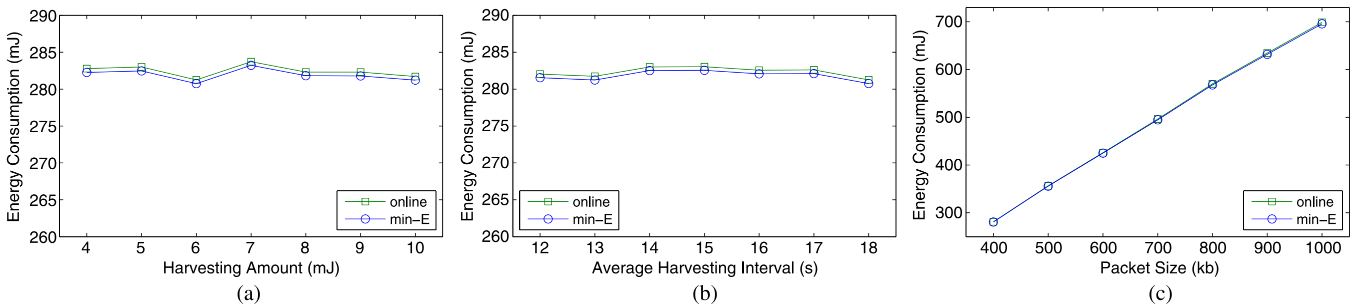


Fig. 7. The energy consumption by the online algorithm and by the offline optimal algorithm for the *min-E problem*. The default setting is the average packet size $z = 400$ kb; the average harvesting inter occurrence time equals 12 s, and the average harvesting amount $h = 8.0$ mJ.

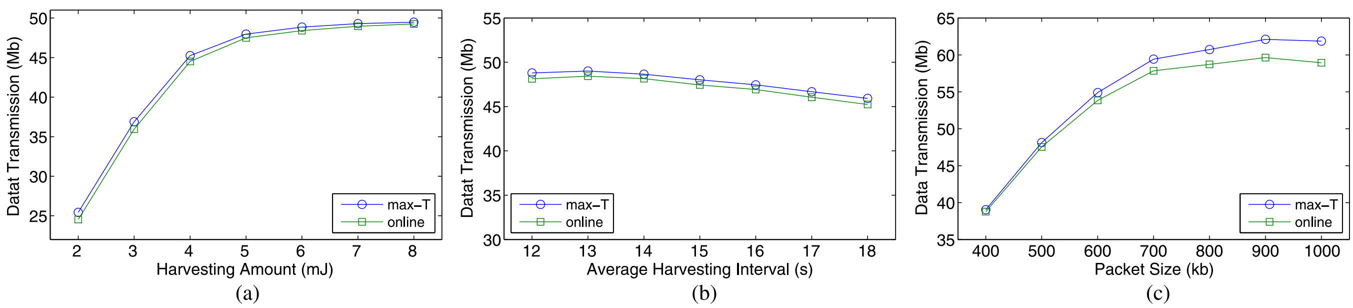


Fig. 8. The amount of data transmitted by the online algorithm and by the offline optimal algorithm for the *max-T problem*. The default setting is the average packet size $z = 500$ kb; the average harvesting inter occurrence time equals 15 s, and the average harvesting amount $h = 5.0$ mJ.

is 1 km, and the path loss h is about 110 dB. Then, we have $r = g^{-1}(p) = W\log_2(1+\frac{ph}{N_0W}) = 10^3\log(1+0.1p)$, where p is in milliwatts, r is in kilobits per second (kbps). The transmission rate is discrete and restricted to $\{0, 50, 100, \ldots, 600\}$. We set algorithm parameter w to be 0.2 s in our simulations.

Following previous works, we also assume the distribution of packet arrival time is a Poisson process, and set the average packet inter arrival time to be 14 s. Packet size is assumed to follow uniform distribution $\mathcal{U}(0.01z, 1.99z)$ where z is the average packet size, and it is set to change from 400 kb to 1000 kb with step 100 kb. Packet delay constraint takes a random number in (0.2 q, 1.8 q), where q is the average delay constraints and set to be 20 s. We also assume that energy harvesting occurs randomly following a Poisson process, and its average inter occurrence time changes from 12 s to 18 s. We assume that the amount of energy harvested follows uniform distribution $\mathcal{U}(0, 2h)$ where h is the average amount which changes from 2 mJ to 8 mJ with step 1 mJ.

Each value shown in figures of this section is the mean value of simulation results from 150 random instances, and in each instance, 100 packets and 100 harvestings are generated according to the above model. Note that, all deadlines are sorted so that an earlier arrived packet carries an earlier deadline.

C. Simulation Results

In Fig. 7, the energy consumption of our online algorithm is compared to the offline optimal solution that minimizes the energy consumptions. To ensure fairness, we only make such comparison on those random instances in which both our online algorithm and the optimal solution completely transmits all packets, where the optimal solution is achieved by solving the *min-E problem* via Truncation method. We set the average packet size $z = 400$ kb. The average harvesting inter occurrence time is 12 s and the average harvesting amount $h = 8.0$ mJ. These three parameters are changed from instance to instance to study their impacts on algorithm performances. We observe that our online algorithm consumes slightly more energy than the optimal truncation method does in all the three figures. In (a) and (b), since the total data of the packets does not change, the energy consumption/curves do not change much. In (c), the two curves increase linearly with the average packet size.

In Fig. 8, we compare the data transmission of our online algorithm with the offline optimal solution that maximizes the throughput. In the default setting, the average packet size is $z = 500$ kb; the average harvesting inter occurrence time is 15 s; the average harvesting amount is $h = 5.0$ mJ. How these parameters make impact on algorithm performance is illustrated respectively in (a), (b), and (c). We observe that in all figures, curves of our online algorithm is closely approaching the curves of the maximum throughput. Fig. 8(a) shows that the throughput increases as the average harvesting amount grows. Fig. 8(b) shows that the throughput slowly decreases as the average harvesting interval enlarges. This is because the larger harvesting interval, the less total energy is harvested to support transmission. Fig. 8(c) shows that the data transmission grows as the average packet size increases.

All figures show that, on average, our online algorithm achieves at least 93% of the throughput that is achievable by the optimal solution. In terms of energy consumption, on average, the online algorithm also achieves ratio of $E^*/E > 93\%$ where E^* is the minimum energy consumption that is achievable by the optimal schedule.

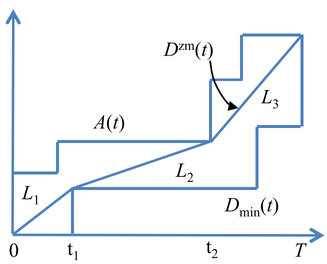


Fig. 9. Example depicting $A(t)$ and $D_{min}(t)$ curves and the constructed $D^{zm}(t)$ curve.

VIII. Conclusion

In this paper, we have studied the energy minimization problem and the throughput maximization problem to deliver a set of packets from a transmitter to a receiver in an energy harvesting system, subject to individual packet deadlines and discrete transmission rates. Based on the fundamental trade-off of the packet delay and energy consumption, our proposed Truncation method computes the optimal solution for the energy minimization problem in the continuous rate model. Then, a general framework is proposed to transform any algorithm designed for continuous rate model (including our Truncation method) to an algorithm in the discrete rate model while preserving the optimality as long as the rate-power function is convex. In case the harvested energy is not sufficient to deliver all packets, we study the throughput maximization problem, which can be solved by the Truncation method also if packets share a common deadline. In this paper, we also have proposed an energy-efficient online algorithm that schedule the transmission for dynamically arrived packets with varying energy harvesting. Simulation results show that, on average, the online algorithm is efficient and well-approximates the optimal solution.

Appendix

A. A Brief Description of ZM Rate Schedule

Let $A(t) = \sum_{i:a_i \le t} B_i$ denote the total amount of bits that have arrived in time interval $[0,t]$; let $D_{min}(t) = \sum_{i:d_i \le t} B_i$ denote the cumulative minimum amount of bits that would satisfy the QoS requirements if it departed by time t. Fig. 9 shows the cumulative data-time diagram. The curve of $A(t)$ is called the arrival curve; the curve of $D_{min}(t)$ is called the minimum departure curve. The key observation is that a feasible schedule can be conveyed by a departure curve that always lies between the arrival curve and minimum departure curve. Now, consider the feasible departure curve as a string. Tie one end of the string at the origin and pass the other end through $D_{min}(T)(= A(T))$. If we now make the string tight, then its trajectory gives the optimal departure curve $D^{zm}(t)$, as depicted in Fig. 9.

The slopes of $D^{zm}(t)$ is the optimal transmission rate r^{zm}. Two important properties about r^{zm} are as follows [2], [6].

- r^{zm} decreases only at deadline points. If r^{zm} decreases at t, then all packets with earlier deadlines (include t) are delivered before t and no transmission has been started for other packets.
- r^{zm} increases only at arrival points. If r^{zm} increases at t, then all packets arrived earlier than t are completely transmitted before point t.

B. Proof of Lemma 3

We prove the lemma by contradiction. Suppose rate schedule $r(t)$ maximizes the data transmission in $[0,t)$, but $r \neq r^{opt}$ in some epochs. Then there exists an epoch p in which $r > r^{opt}$, because otherwise $r(t)$ transmits less data than r^{opt} does. Let r_p and r_p^{opt} be the rate of $r(t)$ and r^{opt} in epoch p respectively. There must also exist another epoch q, with $r < r^{opt}$ because t is an energy critical point of r^{opt} and $r(t)$ cannot consume more energy than r^{opt} does in $[0,t)$. Similarly define r_q and r_q^{opt}. By Lemma 1, we have r^{opt} increases until time T.

If $p < q$, then we must have $r_p^{opt} = r_q^{opt}$, for otherwise, by Lemma 2, there is a data critical point in between, while $r(t)$ transmits more data than r^{opt} before this point, contradicting to the definition of a data critical point. So, $r_p > r_p^{opt} = r_q^{opt} > r_q$. Obviously, r_p and r_q can be equalized to transmit more data. This contradicts the assumption that $r(t)$ maximizes the data transmission. If $q < p$, then we have $r_q < r_q^{opt} \le r_p^{opt} < r_p$. Since $r(t)$ transmits less data than r^{opt} before the end of epoch q, there are non-zero data and energy at the end of epoch q. Thus, r_p and r_q can be equalized to transmit more data in $[0,t)$. It is also a contradiction.

C. Proof of Theorem 3

It is clear that if a rate $r(t)$ is optimal, then no equalization can be performed. We now prove that if no equalization can be performed without violating causality constraints or energy constraints, then the rate schedule is optimal. Suppose on the contrary that such a rate schedule $r(t)$ is not optimal. Let epoch p be the first epoch in which $r(t) \neq r^{opt}$, denote the two rates in epoch p by r_p and r_p^{opt} respectively. There are two cases: $r_p^{opt} > r_p$ or $r_p > r_p^{opt}$. We only prove the first case, while the other proof is similar. Because $r_p^{opt} > r_p$ and both schedules finish all data, there must be some subsequent epoch q in which $r_q^{opt} < r_q$. Let epoch q be the first such epoch.

In duration between epoch p and epoch q, there is no delay critical point of r^{opt}, because otherwise $r(t)$ can not transmit less data than r^{opt} does before this delay critical point. Thus, $r_p^{opt} \le r_q^{opt}$ according to Lemma 4. We have two facts for $r(t)$: (1) $r_p < r_p^{opt} \le r_q^{opt} < r_q$; (2) Both data and energy have a non-zero amount at the end of epoch p. Thus r_p and r_q can be equalized. This is a contradiction. Therefore, $r(t)$ is optimal.

D. Proof of Theorem 4

It is obvious that $r^{opt(i+1)} > r^{opt(i)}$ in at least one epoch in $[0,T)$, since $r^{opt(i+1)}$ finishes one more data packet. Let epoch p be the first such epoch. We first prove $r^{opt(i+1)} = r^{opt(i)}$ before p in Fact 1, and then prove $r^{opt(i+1)} > r^{opt(i)}$ after p in Fact 2.

Fact 1: Prior to epoch p, $r^{opt(i+1)} = r^{opt(i)}$.

Suppose on the contrary, there are some epochs before p in which $r^{opt(i+1)} < r^{opt(i)}$, let epoch q be the last of them. Let t be the end point of epoch q. Thus, at time t, either

$r^{opt(i)}$ decreases or $r^{opt(i+1)}$ increases. If $r^{opt(i)}$ decreases at point t, then t is a delay critical point by Lemma 4. However rate schedule $r^{opt(i+1)}$ transmits even less data than $r^{opt(i)}$ does before this delay critical point, thus some packets must miss their deadlines, which contradicts the feasibility of $r^{opt(i+1)}$. If $r^{opt(i+1)}$ increases at t, then t is either an energy critical point or a data critical point by Lemma 2. If t is an energy (data) critical point, then energy is used up (data runs out) by rate schedule $r^{opt(i+1)}$ at t. This implies that energy (data) is not enough to support the transmission in $r^{opt(i)}$ before t, which is a contradiction.

Fact 2: In subsequent epochs, $r^{opt(i+1)} > r^{opt(i)}$.

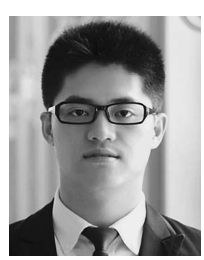
Suppose on the contrary, there is some subsequent epoch q with $r^{opt(i+1)} \leq r^{opt(i)}$, let epoch q be the first of them. Thus, at the beginning of epoch q, either $r^{opt(i+1)}$ decreases or $r^{opt(i)}$ increases. Neither is possible. The proof follows a similar argument as the proof for Fact 1.

ACKNOWLEDGMENT

The authors would like to thank the anonymous referees and guest editors for their constructive comments and valuable suggestions which have helped improve the quality and presentation of the paper.

REFERENCES

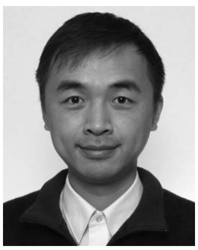
[1] B. Prabhakar, E. Uysal Biyikoglu, and A. El Gamal, "Energy-efficient transmission over a wireless link via lazy packet scheduling," in *Proc. IEEE INFOCOM*, Apr. 2001, vol. 1, pp. 386–394.
[2] E. Uysal-Biyikoglu, B. Prabhakar, and A. El Gamal, "Energy-efficient packet transmission over a wireless link," *IEEE/ACM Trans. Netw.*, vol. 10, no. 4, pp. 478–499, Aug. 2002.
[3] W. Chen and U. Mitra, "Energy efficient scheduling with individual packet delay constraints," in *Proc. 25th IEEE INFOCOM*, Apr. 2006, pp. 1–12.
[4] W. Chen, M. J. Neely, and U. Mitra, "Energy-efficient transmissions with individual packet delay constraints," *IEEE Trans. Inf. Theory*, vol. 54, no. 5, pp. 2090–2109, May 2008.
[5] M. Zafer and E. Modiano, "A Calculus approach to minimum energy transmission policies with quality of service guarantees," in *Proc. IEEE INFOCOM*, Mar. 2005, vol. 1, pp. 548–559.
[6] M. Zafer and E. Modiano, "A Calculus approach to energy- efficient data transmission with quality-of-service constraints," *IEEE/ACM Trans. Netw.*, vol. 17, no. 3, pp. 898–911, Jun. 2009.
[7] F. Shan, J. Luo, and X. Shen, "Optimal energy efficient packet scheduling with arbitrary individual deadline guarantee," *Comput. Netw.*, vol. 75, pp. 351–366, Dec. 2014.
[8] J. Yang and S. Ulukus, "Transmission completion time minimization in an energy harvesting system," in *Proc. 44th Annu. CISS*, Mar. 2010, pp. 1–6.
[9] J. Yang and S. Ulukus, "Optimal packet scheduling in an energy harvesting communication system," *IEEE Trans. Commun.*, vol. 60, no. 1, pp. 220–230, Jan. 2012.
[10] K. Tutuncuoglu and A. Yener, "Short-term throughput maximization for battery limited energy harvesting nodes," in *Proc. IEEE ICC*, Jun. 2011, pp. 1–5.
[11] K. Tutuncuoglu and A. Yener, "Optimum transmission policies for battery limited energy harvesting nodes," *IEEE Trans. Wireless Commun.*, vol. 11, no. 3, pp. 1180–1189, Mar. 2012.
[12] O. Ozel, K. Tutuncuoglu, J. Yang, S. Ulukus, and A. Yener, "Resource management for fading wireless channels with energy harvesting nodes," in *Proc. IEEE INFOCOM*, Apr. 2011, pp. 456–460.
[13] O. Ozel, K. Tutuncuoglu, J. Yang, S. Ulukus, and A. Yener, "Transmission with energy harvesting nodes in fading wireless channels: Optimal policies," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 8, pp. 1732–1743, Sep. 2011.
[14] M. A. Antepli, E. Uysal-Biyikoglu, and H. Erkal, "Optimal packet scheduling on an energy harvesting broadcast link," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 8, pp. 1721–1731, Sep. 2011.
[15] O. Ozel, J. Yang, and S. Ulukus, "Optimal broadcast scheduling for an energy harvesting rechargeable transmitter with a finite capacity battery," *IEEE Trans. Wireless Commun.*, vol. 11, no. 6, pp. 2193–2203, Jun. 2012.
[16] H. Erkal, F. Ozcelik, and E. Uysal-Biyikoglu, "Optimal offline broadcast scheduling with an energy harvesting transmitter," *J. Wireless Commun. Netw.*, vol. 2013, no. 197, pp. 1–14, Jul. 2013.
[17] K. Tutuncuoglu and A. Yener, "Sum-rate optimal power policies for energy harvesting transmitters in an interference channel," *J. Commun. Netw.*, vol. 14, no. 2, pp. 151–161, 2012.
[18] M. Gatzianas, L. Georgiadis, and L. Tassiulas, "Control of wireless networks with rechargeable batteries," *IEEE Trans. Wireless Commun.*, vol. 9, no. 2, pp. 581–593, Feb. 2010.
[19] R. Vaze, "Competitive ratio analysis of online algorithms to minimize packet transmission time in energy harvesting communication system," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 115–1123.
[20] R. Vaze, R. Garg, and N. Pathak, "Dynamic power allocation for maximizing throughput in energy-harvesting communication system," *IEEE/ACM Trans. Netw.*, vol. 22, no. 5, pp. 1621–1630, Oct. 2014.
[21] C. Huang, R. Zhang, and S. Cui, "Throughput maximization for the Gaussian relay channel with energy harvesting constraints," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 8, pp. 1469–1479, Aug. 2013.
[22] B. Gurakan, O. Ozel, J. Yang, and S. Ulukus, "Energy cooperation in energy harvesting two-way communications," in *Proc. IEEE ICC*, Jun. 2013, pp. 3126–3130.
[23] "An overview of the application of Code Division Multiple Access (CDMA) to digital cellular systems and personal cellular networks," San Diego, CA, USA, Doc. EX60-10010, 1992.
[24] *IEEE Wireless Local Area Networks*, IEEE Std. 802.11, 2012.
[25] E. Altman, K. Avrachenkov, G. Miller, and B. Prabhu, "Discrete power control: Cooperative and non-cooperative optimization," in *Proc. 26th IEEE INFOCOM*, May 2007, pp. 37–45.
[26] Y. Xing and R. Chandramouli, "Stochastic learning solution for distributed discrete power control game in wireless data networks," *IEEE/ACM Trans. Netw.*, vol. 16, no. 4, pp. 932–944, Aug. 2008.
[27] T. Kim, H. Lim, and J. C. Hou, "Improving spatial reuse through tuning transmit power, carrier sense threshold, data rate in multihop wireless networks," in *Proc. MobiCom*, 2006, pp. 366–377.
[28] T. Kim, H. Lim, and J. C. Hou, "Understanding and improving the spatial reuse in multihop wireless networks," *IEEE Trans. Mobile Comput.*, vol. 7, no. 10, pp. 1200–1212, Oct. 2008.
[29] B. T. Bacinoglu and E. Uysal-Biyikoglu, "Finite-horizon online transmission scheduling on an energy harvesting communication link with a discrete set of rates," *J. Commun. Netw.*, vol. 16, no. 3, pp. 300–393, Jun. 2014.
[30] A. Bodin and D. Gunduz, "Energy harvesting communication system with a finite set of transmission rates," in *Proc. Eur. Wireless*, May 2014, pp. 1–6.



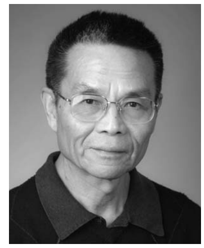
Feng Shan (S'11) received the B.S. degree in computer science from Hohai University, Nanjing, China, in 2008. He is currently pursuing the Ph.D. degree in computer science and engineering at Southeast University, Nanjing, China. He was a Visiting Scholar at the School of Computing and Engineering, University of Missouri-Kansas City, Kansas City, MO, USA, from 2010 to 2012. His research interests are in the areas of energy harvesting, wireless power transfer, algorithm design and analysis.

Junzhou Luo (M'07) received the B.S. degree in applied mathematics and the M.S. and Ph.D. degrees in computer network, all from Southeast University, China, in 1982, 1992, and 2000, respectively. He is a Full Professor in the School of Computer Science and Engineering, Southeast University, Nanjing, China. His research interests are next generation network architecture, network security, cloud computing, and wireless LAN. He is a member of the IEEE Computer Society and co-chair of IEEE SMC Technical Committee on Computer Supported Cooperative Work in Design, and he is a member of the ACM and chair of ACM Nanjing Chapter.

Weiwei Wu (M'14) received the B.Sc. degree from South China University of Technology and the Ph.D. degree from City University of Hong Kong (CityU, Dept. of Computer Science) and University of Science and Technology of China (USTC) in 2011, and went to Nanyang Technological University (NTU, Mathematical Division, Singapore) for post-doctorial research in 2012. He is currently an Associate Professor at Southeast University, Nanjing, China. His research interests include wireless communications, optimizations and algorithm analysis, scheduling, game theory and network economics.



Minming Li (M'13) received the B.E. and Ph.D. degrees from the Department of Computer Science and Technology, Tsinghua University, in 2002 and 2006, respectively. He is currently an Associate Professor in the Department of Computer Science, City University of Hong Kong. His research interests include algorithm design and analysis in wireless networks and embedded systems, combinatorial optimization and algorithmic game theory.



Xiaojun Shen (SM'02) received the B.S. degree in numerical analysis from Tsinghua University, Beijing, China, in 1968, the M.S. degree in computer science from the Nanjing University of Science and Technology, China, in 1982, and the Ph.D. degree in computer science from the University of Illinois at Urbana-Champaign in 1989. He is a Professor in the School of Computing and Engineering, University of Missouri-Kansas City. His current research focuses on fundamental scheduling problems in wired and wireless computer networks.