Online Energy Efficient Packet Scheduling with Delay Constraints in Wireless Networks

Xiliang Zhong and Cheng-Zhong Xu
Department of Electrical and Computer Engineering
Wayne State University, Detroit, Michigan 48202
{xlzhong, czxu}@wayne.edu

***Abstract*— We study online energy-efficient packet scheduling in a multi-user AWGN channel. The objective is to minimize the overall energy consumption in both transmission and circuit by adapting packet transmission rates with individual packet delay constraints. The scheduling is online in the sense that the transmission rates are determined without any assumption about future packet arrivals. We consider a transmitter with a general input and the transmitter communicates with multiple receivers with different distances, which means packets may have different power characteristics and delay constraints. We propose to conduct online rate assignment based on backlogged packets and derive the optimal algorithm in transmitting the packets. We compare the proposed scheduler with existing online algorithms via simulation in terms of energy consumption and algorithms efficiency. Results demonstrate the effectiveness of the online algorithms in striking a better energy-delay trade-off.**

I. Introduction

Power efficiency is a key design requirement in battery powered systems. Reliable packet delivery over a wireless channel is a major source of energy expenditure. It is essential to reduce power consumption of packet transmission without performance degradation. The objective of this study is to improve the overall energy efficiency while guarantee individual packet delay constraints.

It is known that power consumption between two points over a wireless channel is exponentially related to the information transmission rate [1]. A linear increase of transmission time can achieve super-linear energy savings. On the other hand, applications are usually delay-sensitive. The transmission time cannot be arbitrarily long. Researchers have proposed different approaches to deal with the energy-delay tradeoff. In [1], [5], [16], [18], average queueing delay was considered as a constraint in energy minimization. In [10], [11], [20], [24], a single deadline for all packets was set so that all arrivals in time $[0, T)$ have to be transmitted before time T. Both average delay and single deadline constraints provide a delay guarantee for a group of packets; transmission delay of individual packet can still be large.

A recent research focus is on delay sensitive applications with individual packet delay constraints, under which each packet transmission must finish before its deadline. The objective is to determine packet transmission rates so that the transmission energy consumption is minimized and delay constraint of each packet is satisfied. Extensive studies were conducted along the line for offline packet scheduling, in which all packet timing information is available offline in order to determine packet transmission rates, or service times similarly. Researchers have studied the problem with different focuses, such as on a special application model (periodic streams) [19], offline generalization [23], distinguished characteristics of different packets [15], properties of offline schedulers [2] [3].

Practically, packet arrivals turn to be more dynamic and we cannot always have full knowledge about future arrivals at design time. There exist packet schedulers based on limited assumption about future packet arrivals. The input is assumed to be independent and identically distributed (i.i.d.) over different time slots in [18]. Input time-correlation is required in [26]. A poisson input with known packet statistics is studied in [23]. Knowledge about future packets releases in a small timing window is assumed in [15]. The knowledge about input correlation and/or distribution is not always available; accurate prediction of future packet arrivals even in a small window is hard, if possible.

A more realistic and general approach is to scheduling packets without *a priori* knowledge about packet arrivals. Decisions are made based on history packet arrivals or backlogged packets. As only limited information is available, it is challenging to derive an online energy-efficient scheduler. Furthermore, by making decisions online, the scheduler needs to be invoked frequently at the time of packet arrivals or departures. The decision must be made efficiently with a low time complexity. There are only limited results for online packet scheduling [13] [3]. Packets are assumed to have identical delay constraints in [3]. Both work assume packets have identical power characteristics and they only minimize the transmission power consumption.

The main contribution of this work is that we consider a general case in which a transmitter communicates with multiple receivers. This implies that packets may have different communication distances (power characteristics) and delay constraints, which make the assumptions of previous work no longer valid. In addition, we also consider the overall energy consumption, including both transmission and circuit energy. Transmission power consumption dominates the overall circuit power at long communication radius. At short distance, however, circuit power becomes a non-negligible factor in determining packets transmission rates. These characteristics play a key role in determining energy efficient packet transmission rates. We first consider a static version of the problem in transmitting all backlogged packets and derive the optimal rate assignment for each packet. By dynamically updating the packets in the backlogged queue and invoking the static assignment algorithm, we are able to adapt the algorithm for dynamic packet arrivals.

A special case of our work is when all receivers are at the same distance and the circuit power is small enough to be ignored. In such cases, our problem formulation is similar to those in [13] [3]. Our proposed algorithm can be reduced to an algorithm with a linear time complexity in the number of backlogged arrivals, in contrast to the algorithms in [13] [3] with a square time complexity.

The rest of the paper is organized as follows. Section II presents system model of our online packet scheduler. Optimal online packet scheduling algorithms are presented in Section III. Section IV verifies the analytical results through simulation. In Section V, we review related studies. Section VI concludes the article.

II. System Model

We consider a wireless environment where a single node transmits packets to multiple receivers. The transmitter communicates with one receiver at a time. Data are generated in the transmitter node and put into the transmitter buffer before they are forwarded to different receivers. Each packet has a delay constraint and should be transmitted no later than the deadline.

We assume the channel between the transmitter and the receivers is an Additive White Gaussian Noise (AWGN) channel and the interference to the receiver is negligible, as in [2], [3], [13], [15], [23]. Using adaptive modulation scaling or channel coding can reduce the energy consumption in wireless communication. The maximum channel capacity R under optimal channel coding is:

$$R = \frac{B}{2} log_2(1 + \frac{P^r}{NB}) \text{ bits/s},$$

where N is the noise power, P^r is the received signal power, and B denotes the channel bandwidth. We can express the received power as:

$$P^r = NB(2^{\frac{2R}{B}} - 1). \quad (1)$$

Considering the effect of path attenuation, we have

$$P^t = \frac{P^r}{A} = \frac{NB}{A}(2^{\frac{2R}{B}} - 1), \quad (2)$$

where A is the attenuation factor and P^t is the transmission power. The transmission power function is monotonically increasing and strictly convex with respect to R. The same characteristics of power function apply even if sub-optimal channel coding is deployed [20]. This means even a small reduction in the transmission rate or increase in transmission delay can lead to a large energy saving. Let l be the distance between the transmitter and the receiver. It is typically observed $A \propto l^{-2}$ [6]. As receivers may have different distances to the transmitter, the power consumed in transmitting packets to each receiver is generally different.

Power consumption of a wireless transmitter, however, also includes that of the electronic circuitry for Digital-to-Analog converter, mixer, and frequency synthesizer, etc. The circuit power, denoted by P^e, can be expressed as [4]

$$P^e = C \cdot B, \quad (3)$$

where C is a constant depending on the circuit.

Combining (2) and (3), we can express the total energy consumption in transmitting a bit as

$$E = (P^t + P^e)\frac{1}{R} = (\frac{N}{A}(2^{\frac{2R}{B}} - 1) + C)\frac{B}{R}. \quad (4)$$

Considering the effect of circuit power, the total energy consumption does not strictly increase with transmission rate. Similar to the studies on Quadrature Amplitude Modulation (QAM) by [19] [22], we define the rate that minimizes the total energy E as the energy-efficient rate, denoted by R^{min}. Below this rate, both the transmission energy and time increase, which is not desirable. The best strategy is to transmit packets at a rate no lower than the minimum rate and put the circuit into sleep state afterward. We ignore sleep power in our model, which can be easily extended by incorporating the power into circuit power consumption.

III. Energy Efficient Online Scheduling

As we consider online packet scheduling without assumption about future packet arrivals, it is not possible to derive the offline optimal policy with all packet information known before their arrivals. Instead, we try to minimize energy consumption of all backlogged packets in the transmitter, in a similar way to the reward maximization problem in [8] [7] [14]. In the following, we will first study a higher-level rate assignment problem considering backlogged packets; then we will show it is readily applicable to dynamic packet arrivals by being invoked at each packet arrival.

A. Problem Formulation Based on Backlogged Packets

Consider n packet arrivals ready for transmission at a time, starting from time 0 for brevity. We refer to the n arrivals as message streams (streams for short) and order them according to their increasing order of deadlines, $d_1 \leq ... \leq d_n$. Stream i has X_i bits to transmit. It takes $\frac{X_i}{R_i}$ for stream i to complete at a rate R_i, We denote the attenuation factor of stream i as A_i. The power function at a rate R_i for stream i is

$$P_i(R_i) = \frac{P_i^r(R_i)}{A_i} + CB = (\frac{N}{A_i}(2^{\frac{2R_i}{B}} - 1) + C)B.$$

The energy consumption of stream i is then $P_i(R_i)\frac{X_i}{R_i}$. We represent its minimum energy-efficient rate as R_i^{min}. Given a set of n streams, we need to find an assignment of rates R_i ($1 \leq i \leq n$) so that the overall energy consumption is minimized while all streams finish before their deadlines. The problem can be formulated as

$$\text{minimize} \sum_{i=1}^{n} P_i(R_i)\frac{X_i}{R_i} \quad (5)$$

$$\text{subject to} \sum_{i=1}^{j} \frac{X_i}{R_i} \le d_j, 1 \le j \le n \quad (6)$$

$$R_i^{min} \le R_i \le R_i^{max},\ 1 \le i \le n. \quad (7)$$

The set of constraints (6) ensures that all streams finish before their deadlines. We assume there is an admission control mechanism to ensure the packets can finish before their deadlines if the transmitter operates at the maximum rates (denoted as R_i^{max} in general). Otherwise, we should not use transmission slowdown for energy savings.

The problem is a nonlinear optimization with $3n$ constraints. In general, with a large number of constraints, it is time-consuming to solve the problem and the solution may not converge. We will show that an efficient solution is possible by exploiting properties of the optimal solution. In the next section, we will solve a simplified version of the problem by considering only the delay constraints.

B. Rate Assignment with Delay-constraints (RAD)

We ignore the boundary condition (7) and refer the minimization problem with only delay constraints (6) as Rate Assignment with Delay constraints (RAD). It can be solved by forming the Lagrangian as

$$L(R, \lambda) = \sum_{i=1}^{n} P_i(R_i)\frac{X_i}{R_i} + \sum_{j=1}^{n} \lambda_j (\sum_{i=1}^{j} (\frac{X_i}{R_i} - d_j),$$

where λ_j, $1 \le j \le n$, are the non-negative Lagrange multipliers. Any optimal solution to RAD satisfies the Kuhn-Tucker conditions including (6) and

$$\frac{\partial P_i(R_i)\frac{X_i}{R_i}}{\partial R_i} - \sum_{j=i}^{n} \lambda_j \frac{X_i}{R_i^2}$$

$$= \frac{P_i'(R_i)R_i - P_i(R_i)}{R_i^2} X_i - \sum_{j=i}^{n} \lambda_j \frac{X_i}{R_i^2} = 0, 1 \le i \le n \quad (8)$$

$$\lambda_j (\sum_{i=1}^{j} \frac{X_i}{R_i} - d_j) = 0, \lambda_j \ge 0, 1 \le j \le n. \quad (9)$$

Because of the strict convexity of the objective function in (5), the conditions are necessary and sufficient for the optimal solution [9].

1) Properties of the optimal solution: Condition (8) can be simplified as

$$P_i'(R_i)R_i - P_i(R_i) = \sum_{j=i}^{n} \lambda_j, 1 \le i \le n. \quad (10)$$

The left hand side function represents the rate of power change at a transmission rate R_i. Geometrically, if we draw a tangent line of the power function $P_i(R_i)$ at point R_i, it is the value of the intercept on the power axis, as illustrated in Figure 1. It

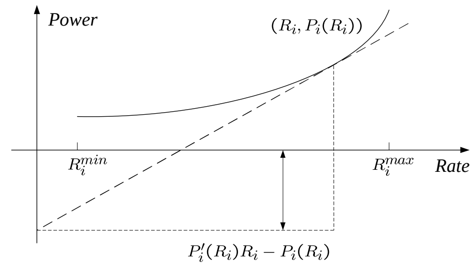


Fig. 1. Illustration of the concept of marginal power.

shares some similarity with the control point concept in [24]. We call this value the *marginal power* of a stream defined as $G_i(R_i) = P_i'(R_i)R_i - P_i(R_i)$ and will omit the parameter R_i for brevity. The completion time of stream j, $\sum_{i=1}^{j} \frac{X_i}{R_i}$ in (9), is no larger than its deadline d_j (stream j finishes at or before its deadline). If a stream (i for example) finishes before its deadline, from (9), $\lambda_i = 0$, and from (10),

$$G_i = G_{i+1} = \sum_{k=i+1}^{n} \lambda_k.$$

If stream i finishes at its deadline. It follows $\lambda_i \ge 0$ and

$$G_i = \sum_{k=i}^{n} \lambda_k \ge G_{i+1} = \sum_{k=i+1}^{n} \lambda_k.$$

Considering the two cases, we have the following lemmas, similar to the analysis in [27].

Lemma 3.1: The marginal power of stream i, G_i, is non-increasing with i. That is, for any i_1 and i_2, $1 \le i_1 < i_2 \le n$, we have $G_{i_1} \ge G_{i_2}$.

Proof: It follows from (10) that $G_{i_2} = \sum_{j=i_2}^{n} \lambda_j$ and $G_{i_1} = \sum_{j=i_1}^{n} \lambda_j = \sum_{j=i_1}^{i_2-1} \lambda_j + G_{i_2}$. As λ_j is non-negative, we get $G_{i_1} \ge G_{i_2}$. □

Lemma 3.2: If all streams between i_1 and i_2, $1 \le i_1 < i_2 < n$, finish before their deadlines, then they have the same marginal power. That is $G_i = G_{i_2+1}$, $i_1 \le i \le i_2$.

Proof: For any stream i, $i_1 \le i \le i_2$, if it finishes before its deadline d_i, i.e., $\sum_{j=1}^{i} \frac{X_j}{R_j} < d_i$, we get $\lambda_i = 0$ according to (9). Combining it with (10) leads to $G_i = \sum_{j=i}^{n} \lambda_j = \sum_{j=i_2+1}^{n} \lambda_j = G_{i_2+1}$. □

We determine the transmission rate of each stream by first solving sub-problems with fewer streams. Define optimization sub-problems for the first i streams as $\mathcal{J}_i$, where $1 \le i \le n$. It is trivial to solve $\mathcal{J}_1$ since there is only one stream to be scheduled. The targeted problem RAD is equivalent to $\mathcal{J}_n$. Let $R_j^{(i)}$ denote the rate of stream j for problem $\mathcal{J}_i$, where $1 \le j \le i$. The optimal solution equals $R_j^{(n)}$. Before we show how to solve $\mathcal{J}_{i+1}$ with a solution to $\mathcal{J}_i$, we first present the following property.

Lemma 3.3: Let $\mathcal{L}$ be the set of streams that have their rates changed by considering a new stream. All streams in $\mathcal{L}$ have their rates increased and lead to the same marginal power.

Proof: We prove the lemma in three steps. First, we note that there must be at least one stream in $\mathcal{L}$ with an increase of rate by considering the new stream, stream i for example. This is because $\sum_{j=1}^{i} \frac{X_j}{R_j} = d_i$ in $\mathcal{J}_i$. Suppose to the contrary, some of the rates are reduced while others remain the same. Then the total transmission time exceeds d_i, which is not allowed.

Let l be the first stream in $\mathcal{L}$ with an increase of its rate after the inclusion of stream $i+1$, i.e., $R_l^{(i+1)} > R_l^{(i)}$. In the second step, we prove no rate decrease is possible for all streams that finish before l by contradiction. Suppose there is such a stream. The completion time of stream $l-1$ in $\mathcal{J}_{i+1}$ is then larger than that in $\mathcal{J}_i$. That is, $\sum_{j=1}^{l-1} \frac{X_j}{R_j^{(i+1)}} > \sum_{j=1}^{l-1} \frac{X_j}{R_j^{(i)}}$. Since the schedule after adding stream $i+1$ must be feasible, we have $\sum_{j=1}^{l-1} \frac{X_j}{R_j^{(i+1)}} \leq d_{l-1}$. It follows $\sum_{j=1}^{l-1} \frac{X_j}{R_j^{(i)}} < d_{l-1}$. This means stream $l-1$ finishes early than its deadline d_{l-1} before we add stream $i+1$. According to Lemma 3.2, we have $G_{l-1}^{(i)} = G_l^{(i)}$, where we denote $G_l(R_l^{(i)})$ as $G_l^{(i)}$ for brevity. After adding stream $i+1$, since stream l has a rate increase, we get $G_l^{(i+1)} > G_l^{(i)}$. The rate of stream $l-1$ can be either reduced or unchanged depending whether stream $l-1$ belongs to $\mathcal{L}$. That is, $G_{l-1}^{(i)} \geq G_{l-1}^{(i+1)}$. Consequently, we have $G_{l-1}^{(i+1)} < G_l^{(i+1)}$, which means stream $l-1$ has a smaller marginal power than that of stream l. This is in contradiction to Lemma 3.1. Therefore, no rates decrease is possible for streams finished before l.

Finally, we prove that all streams starting from l have their rates increased. Because stream l is the first with a rate change (increase), it is completed ahead of the scheduled time according to the solution to $\mathcal{J}_i$. It means stream l finishes before its deadline after the inclusion of stream $i+1$. As a result, $G_l^{(i+1)} = G_{l+1}^{(i+1)}$ according to Lemma 3.2. We then get $G_{l+1}^{(i+1)} > G_l^{(i)}$. Recall $G_l^{(i)} \geq G_{l+1}^{(i)}$ by Lemma 3.1. It follows that $G_{l+1}^{(i+1)} > G_{l+1}^{(i)}$ and the rate of stream $l+1$ increases, i.e., $R_{l+1}^{(i+1)} > R_{l+1}^{(i)}$. Similar analysis can be applied to any stream from $l+2$ to i. All streams between l and i have their rates increased and finish before their deadlines. They have the same marginal power according to Lemma 3.2. □

Let $G_j^{-1}(x)$ denote the reverse function of $G_j(x)$. The inverse function exists because there is a one-to-one mapping between $G_i(R_i)$ and R_i. Recall that the marginal power of a stream is defined as $G_i(R_i) = P_i'(R_i)R_i - P_i(R_i)$. The derivative of the marginal power is $G_i'(R_i) = P_i''(R_i)R_i > 0$, which means the marginal power is monotonically increasing with the transmission rate. The inequality applies because $P_i(R_i)$ is a strictly convex function. We can get the solution to $\mathcal{J}_{i+1}$ according to that of $\mathcal{J}_i$ by Theorem 3.1.

Theorem 3.1: Let l be the first stream with a rate increase as a result of adding stream $i+1$. Given a rate assignment of the first i streams, we get the optimal solution to the first $i+1$ streams according to:

$$\sum_{j=l}^{i+1} \frac{X_j}{G_j^{-1}(\lambda^{(i+1)})} = \sum_{j=l}^{i} \frac{X_j}{R_j^{(i)}} + (d_{i+1} - d_i), \tag{11}$$

where $\lambda^{(i+1)}$ equals $\lambda_{i+1}^{(i+1)}$ and is the value of the Lagrange multipliers for all streams in $\mathcal{L}$.

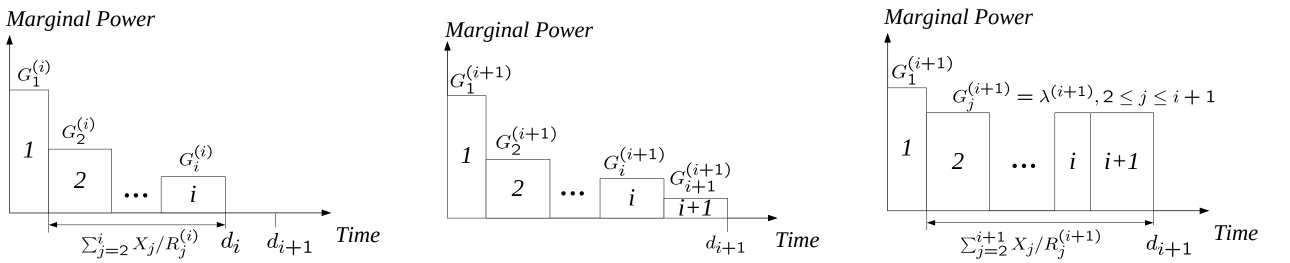
Proof: The overall transmission time of streams from l to i before adding stream $i+1$ is $\sum_{j=l}^{i} X_j/R_j^{(i)}$. The time added by the interval $d_{i+1} - d_i$ is equal to the summed time of streams from l to $i+1$ in $\mathcal{L}_{i+1}$. That is:

$$\sum_{j=l}^{i+1} \frac{X_j}{R_j^{(i+1)}} = \sum_{j=l}^{i} \frac{X_j}{R_j^{(i)}} + (d_{i+1} - d_i). \tag{12}$$

From Lemmas 3.2 and 3.3, we know that all streams from l to i have their rates increased after adding stream $i+1$ and have the same marginal power. That is, there exists a $\lambda_{i+1}^{(i+1)}$ such that $G_j(R_j^{(i+1)}) = \lambda_{i+1}^{(i+1)}$ for any j, $l \leq j \leq i+1$. Substituting $R_j^{(i+1)}$ as $G_j^{-1}(\lambda_{i+1}^{(i+1)})$ in (12) completes the proof. □

2) An iterative rate assignment algorithm: We develop an iterative algorithm to find the optimal rate assignment. According to Theorem 3.1, the optimal solution is to obtain l and $\lambda^{(i+1)}$ such that (11) holds. A straight-forward approach is to try every stream from 1 to i and solve for $\lambda^{(i+1)}$; but this is not guaranteed to have a solution. Take the streams in Figure 2(a) for example. We have i streams and we will get the rate assignment considering the $(i+1)th$ stream. Starting from the interval $[d_i, d_{i+1}]$, we first check if the stream can be finished in the interval with a marginal power no larger than that of stream i. If it is, we do not need to consider other streams and there are no rate changes of existing streams, as shown in Figure 2(b); if the stream cannot finish in the interval, we raise the rate of stream i to determine if the stream can be finished with a marginal power no larger than that of stream $i-1$. This process continues until we find the first stream with a rate change so as to finish stream $i+1$. Figure 2(c) is an example in which stream 2 needs a rate change ($l = 2$). This can be interpreted as a water-filling process [6] of marginal power to different streams, with $G_j^{(i)}$ as the residual waters and $G_j^{(i+1)}$ as the water-level after rate assignment of stream $i+1$.

Instead of starting from the last stream and iterating backward, we can use a binary search due to the non-decreasing nature the marginal power. We present an iterative procedure in Algorithm 1, which determines rate settings of n streams with only delay constraints. In the ith iteration, we have i marginal power in a non-increasing order and add stream $i+1$. For notational convenience, we assume all rates initialized as 0, $G(0) = 0$, and $1/0 = \infty$. We find the minimum index l using a binary search such that rates of streams from l to i increase. The value of $\lambda^{(i+1)}$ is then computed according to Theorem 3.1 and used to determine the rate assignment. In the algorithm, computing sums in the binary search takes $O(n)$ time so that each iteration takes $O(nlogn)$. The running time of the algorithm is then $O(n^2logn)$. The time complexity can be improved by noting that the sums inside the binary search only need to be done once. We can move the sums out of the search loop and maintain an array of sums, which reduce the complexity of the binary search loop to $O(logn)$.



(a) Schedule of the first i streams (b) No rate change after adding stream $i+1$ (c) Rates (2 to i) changes after adding stream $i+1$

Fig. 2. Rate assignment for the first $i+1$ streams based on the schedule of the first i streams

The computation of the sums and other statement during each iteration can be finished in $O(n)$ time. The overall algorithm takes $O(n^2)$.

Algorithm 1 Online rate assignment with delay-constraints (RAD) based on backlogged packets.

$R_1^{(1)} = X_1/d_1$
for $i = 1$ to $n-1$ **do** ▷ adding stream $i+1$ at the ith iteration
  $l = 1$ ▷ set l to 1 if binary search does not return a value
  Using a binary search, find $l \in [2, i+1]$ such that $\sum_{j=l-1}^{i+1} \frac{X_j}{G_j^{-1}(G_{l-1}(R_{l-1}^{(i)}))} - \sum_{j=l-1}^{i} \frac{X_j}{R_j^{(i)}} \le d_{i+1} - d_i < \sum_{j=l}^{i+1} \frac{X_j}{G_j^{-1}(G_l(R_l^{(i)}))} - \sum_{j=l}^{i} \frac{X_j}{R_j^{(i)}}$
  Solve for $\lambda^{(i+1)}$ in $\sum_{j=l}^{i+1} \frac{X_j}{G_j^{-1}(\lambda^{(i+1)})} = \sum_{j=l}^{i} \frac{X_j}{R_j^{(i)}} + (d_{i+1} - d_i)$
  $R_j^{(i+1)} = G_j^{-1}(\lambda^{(i+1)}), l+1 \le j \le i+1$
  $R_j^{(i+1)} = R_j^{(i)}, 1 \le j \le l$
end for

3) A linear rate assignment algorithm with identical power functions: A special case of RAD is all packets have identical power functions, referred to as RAD-IP. This reduces to a similar problem to the online packet scheduling by [13] [3], in which the authors proposed solutions with a time complexity of $O(n^2)$. By contrast, we will present a linear algorithm in the following.

When all streams have the same power functions, characterized by the same transmission distance and circuit power, the same marginal power always leads to the same transmission rate. The water-filling over marginal power of different streams in Figure 2 reduces to water-filling over transmission rates, in a similar way to [13]. Streams with the same transmission rates can be grouped together and treated equally. In such case, we obtain the optimal solution to RAD-IP in Algorithm 2. We group streams with the same rate into one state and maintain a set of states. Each state is represented by (S_i, X_{S_i}, R_{S_i}), where S_i contains indexes of streams in a subset having the same transmission rates R_{S_i} with a request size X_{S_i} summed over all streams in S_i.

Initially, we have a list L containing only the first stream. In each iteration, stream i is added with a rate either smaller than or identical to the rate of the last stream in L. If added with a smaller rate, we continue to the next stream; otherwise, we keep checking the next rate in L and merge streams with the same rate to one state until we reach a rate at which stream i can be completed. After n iterations, rates of all streams are stored in list L. The maximum number of states in L is n, which occurs when the **else** part of the while loop is executed up to its maximum number, $n-1$. On the other extreme, there is only one state in L, meaning the maximum number of state merging is taking, $n-1$. Any other case is a combination of at most $n-1$ executions of the **if** and **else** parts. Therefore, the algorithm has a linear time in determining rate settings of n streams.

Algorithm 2 Rate assignment with delay-constraints for packets with identical power functions (RAD-IP).

$S_1 = \{1\}$, $X_{S_1} = X_1$, $R_{S_1} = X_1/d_1$
Initialize L with state (S_1, X_{S_1}, R_{S_1}), denoted by S_1
for $i = 2$ to n **do** ▷ add stream i at the ith iteration
  $S_i = \{i\}$, $X_{S_i} = 0$, $R_{S_i} = 0$, $\tau = d_i - d_{i-1}$
  append state S_i to the end of L
  while $X_i > 0$ **do**
    let the last state in L as S', the second to the last as S''; S'' as $(0, 0, \infty)$ if there is only one state in L
    if $X_i > (R_{S''} - R_{S'})\tau$ **then** ▷ stream i has a larger rate than $R_{S''}$; need to check next state in L
      $X_i = X_i - (R_{S''} - R_{S'})\tau$
      merge S' and S'' into one state $\bar{S}$, $X_{\bar{S}} = R_{S''}\tau + X_{S''}$, $R_{\bar{S}} = R_{S''}$
      $\tau = \tau + X_{S''}/R_{S''}$
    else ▷ stream i has a smaller rate than $R_{S''}$
      $X_{S'} = X_{S'} + X_i$, $R_{S'} = R_{S'} + X_i/\tau$, $X_i = 0$
    end if
  end while
end for
for each state $S_i \in L$ and each $j \in S_i$ **do** ▷ size of $S_1 \cup S_2 ... \cup S_{|L|}$ is n
  return rate R_{S_i}
end for

C. Rate Assignment with Both Delay and Boundary Conditions (RADB)

The solution to RAD only considers the delay constraints (6). The optimal solution should also satisfy the boundary condition (7) as well. If the solution to RAD satisfies the conditions, it is optimal; otherwise, we need to find the optimal solution. Consider the lower bounds first. We start the water-filling process from rates no lower than the minimum rate R_i^{min}. As no rate decrease is possible in all iterations according to Lemma 3.3, the lower bound of stream i is guaranteed to be satisfied. This is not the case for upper bounds because the increased rates in adding streams after i may exceed the upper bound of stream i. We impose an upper bound to the marginal power of each stream; whenever the marginal power reaches the bound, it stops increasing and we distribute the new stream to others with marginal power below their upper bounds. We can obtain the solution following a similar procedure to Algorithm 1. The difference is, however, that the marginal power is not necessarily non-increasing by starting from the minimum rates. A binary search among all streams is not applicable to find the first stream with a rate increase. We can design a more complicated search to find the range of the appropriate water-level that accommodates the new stream. Alternative, we propose to use a bisection to find water-level in Algorithm 3. The procedure GET_RATES returns the set of rates for the first $i+1$ streams with a given marginal power. The main procedure returns the rate settings given a set of streams.

We first determine the lower and upper bounds (G_{min}^{min} and G_{max}^{max}) of the marginal power and start from the middle (G_{mid}). Under the marginal power, if the new stream can be finished before its deadline, we have a rate higher than necessary. We then decrease the rates by restricting the search range to $[G_{min}^{min}, G_{mid}]$; otherwise, we increase the search range to $[G_{mid}, G_{max}^{max}]$. The algorithm terminates when the search interval is smaller than the given threshold ϵ in the **while** loop. In practice, with a sufficiently small ϵ the computed set of rates can be used effectively as the optimal rate settings. The maximum number of steps needed to converge occurs for the iteration of stream n, with an search interval from $\min_{1 \le j \le n} G_j^{min}$ to $\max_{1 \le j \le n} G_j^{max}$, where G_j^{min} and G_j^{max} are the marginal power of stream j under the minimum transmission rate R_j^{min} and the maximum rate R_j^{max}, respectively. As the values of the two bounds are constant in a transmitter with known transmission rate bounds, it is a constant factor in deriving the time complexity of Algorithm 3. As each bisection step involves $O(n)$ summations and we have n streams, the running time of the algorithm is $O(n^2)$.

D. Dynamic Packet Arrivals

We have studied static rate assignment based on backlogged packet arrivals. This is readily applicable to dynamic packet arrivals and departures, in a similar way to [13] [3]. Each time there are new packet arrivals, we invoke the static rate assignment algorithms (e.g. RADB) to determine transmission rates of all packets. The packets will be transmitted using the rates if there are no new packet arrivals before all backlogged packets are transmitted; otherwise, we need to update their transmission rates by considering new arrivals based on the backlogged packets and their residual delay constraints.

After we determine the transmission rates of all backlogged packets, we still need to choose a low-level scheduling policy to determine the order of packet transmission. Any policy can be used as long as packets transmission can complete before their deadlines. Example service disciplines include earliest deadline first [23], first-come-first-served [2], [3], [15], and weighted fair queuing [13]. By separating rate assignment and packet scheduling, we make algorithm design simple and enable the use of independent policies at either level.

IV. EVALUATION

We conducted simulations to verify the analytical results. In the case of RADB, we use an error threshold ϵ as 10^{-4} to terminate the bisection. We assumed a packet length of 8KBits, a maximum bits per transmission as 16, and a channel bandwidth $B = 10^6$Hz. The power function is based on (1) (2). We set the noise power to 1. The circuit constant C was set to 10^{-8} according to [22]. Assuming it takes 10pJ/bit/m^2 to reliably transmit one bit [22], we have the transmission power function at a rate R and a distance l as $10^6 l^2 (2^{2*10^{-6}*R} - 1)$.

We first consider energy per bit according to (4) for receivers with different communication distances. We present the results of l = 5m, 30m, and 50m in Figure 3. We can observe that in all cases the energy consumption reduces significantly with lower transmission rates. There exists, however, a minimum rate for each receiver, below which energy per bit increases. It means both the transmission time and energy

Algorithm 3 Rate assignment with delay and boundary conditions (RADB) using bisection.

$R_1^{(1)} = \max(X_1/d_1, R_1^{min})$
$R_1^{(j)} = R_j^{min}$, $2 \le j \le n$
for $i = 1$ to $n-1$ **do** ▷ add stream $i+1$ at the ith iteration
  $G_{min}^{min} = \min_{1 \le j \le i+1} G_j^{min}$, $G_{max}^{max} = \max_{1 \le j \le i+1} G_j^{max}$
  while $G_{max}^{max} - G_{min}^{min} > \epsilon$ or $\sum_{j=1}^{i+1} \frac{X_j}{R_j^{(i+1)}} > d_{i+1}$ **do**
    $G_{mid} = \frac{1}{2}(G_{max}^{max} + G_{min}^{min})$
    $[R_1^{(i+1)}, ..., R_{i+1}^{(i+1)}] = \text{GET\_RATES}(G_{mid}, i)$
    if $\sum_{j=1}^{i+1} \frac{X_j}{R_j^{(i+1)}} < d_{i+1}$ **then**
      $G_{max}^{max} = G_{mid}$ ▷ to reduce rates
    else if $\sum_{j=1}^{i+1} \frac{X_j}{R_j^{(i+1)}} > d_{i+1}$ **then**
      $G_{min}^{min} = G_{mid}$ ▷ to increase rates
    else
      break
    end if
  end while
  $R_j^{(1)} = R_j^{(i+1)}, 1 \le j \le i+1$ ▷ to start from updated rates
end for

procedure GET_RATES(G_{mid}, i)
  for $j = 1$ to i **do**
    if $G_j^{max} \le G_{mid}$ **then**
      $R_j = R_j^{max}$ ▷ cannot exceed the upper bound
    else if $G_{mid} > G_j(R_j^{(1)})$ **then**
      $R_j = G_j^{(-1)}(G_{mid})$ ▷ with an increased rate
    else
      $R_j = R_j^{(1)}$ ▷ keep the original setting
    end if
  end for
  return $[R_1, ..., R_i]$
end procedure

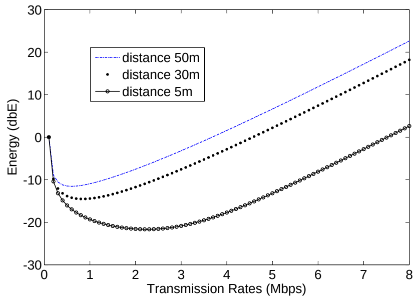


Fig. 3. Energy per bit for receivers with difference communication distances.

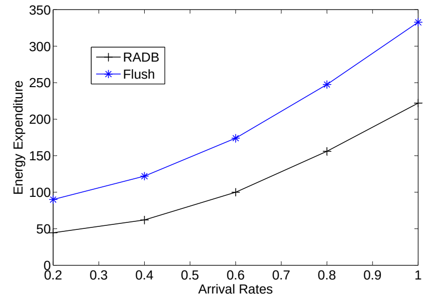


Fig. 4. Energy consumed in communicating to two receivers with identical arrival rates.

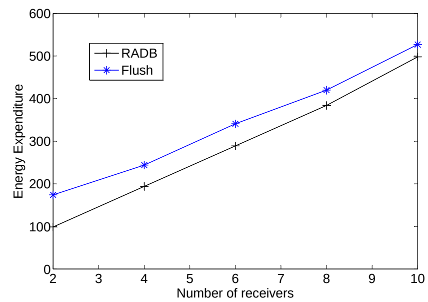


Fig. 5. Energy consumed with varied number of receivers.

increase with lower rates. For receiver at a distance of 50m, the minimum rate is 0.59Mbps; for the receiver at a distance of 5m, the minimum rate increases to 2.11Mbps. Short range communication has an increased minimum rate because less energy is spent in packet transmission and circuit power consumption becomes a more important factor, in consistent with the observations using QAM in [19] [22].

We compared the proposed algorithms with the following two algorithms:

- *Adaptive*, the time-variant algorithm by Khojastepour and Sabharwal [13] which was proved to be optimal for transmission energy minimization among online packet schedulers for streams with identical power characteristics.
- *Flush*, an online algorithm by Chen *et al.* [3] (extended from [23] for static packet arrivals). It is for packet arrivals with identical power characteristics and delay constraints.

We note that although the original Flush only supports packets with identical delay constraints. The solution can be readily extended for different delays. We generalized Flush for packets with different deadlines. For both Flush and the proposed RADB, we use the earliest deadline first as the lower-level packet scheduler due to its effectiveness in satisfying deadlines.

An interesting result is that the energy consumption according to Adaptive and Flush are exactly the same. This is because Flush is an extension of the buffer-flushing algorithm in [23], which is a generalization of Adaptive [13]. In the following, we only present the results of Flush for improved readability.

We generated packet arrivals according to Poisson distributions. The average number of packet arrivals at a time interval dt (1ms) is the arrival rate multiplied by the interval, $\lambda \cdot dt$. We first assumed the transmitter communicates with two receivers, one at a distance of 5m, the other at 30m. Packets deadlines were uniformly drawn from the range [10ms, 20ms]. In the first test, we set half of the packets for short-range communication and half for long-range communication. Figure 4 shows the normalized energy consumption over increasing arrival rates. With a small arrival rate, less number of packets is ready at the time of rate assignment. More packets get transmitted at their lowest possible rates subject to their deadlines. This makes RADB more effective by operating at higher rates than the energy-efficient rate of a stream. Under Flush, however, we may transmit a packet at a low rate resulting in more power consumption and transmission time. In such cases, RADB can save nearly 50% energy compared with Flush. The relative performance gap decreases with larger arrival rates. As RADB considers the difference of the power characteristics, it can still achieve energy-efficiency as much as 30%.

Similar observation can be made with other simulation scenarios. For example, we varied the number of receivers. Distances of the receivers to the transmitters were uniformly drawn from [5m, 50m]. The performance gain of RADB with more receivers becomes less impressive because the power functions are uniformly distributed and less distinguished from each other. Their overall energy consumption is closer to the case of treating them equally, as shown in Figure 5.

Although RADB is more efficient in energy savings, this comes at the cost of higher complexity. We estimated running times of the algorithms on a Pentium(R) 4 machine at a 2GHz clock rate. Figure 6 is the total algorithm running time in rate assignments for a simulated 20 seconds packet transmission with varied number of receivers. We used lookup tables to reduce the time in computing $G_j^{(-1)}$. Recall that RADB has a square time complexity in the number of backlogged streams. From the figure, we can observe that the running time increases at a speed much lower than a square relationship. This is

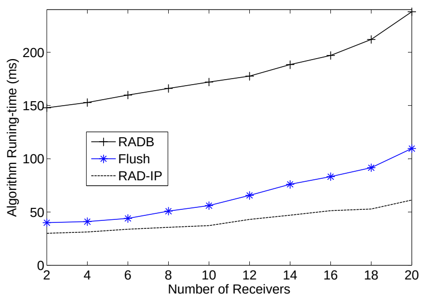


Fig. 6. Algorithm running time with different number of receivers.

because the number of backlogged streams increases at a lower speed than the number of receivers. The algorithm running time of RADB with 20 receivers is 238ms, which is about 1.2% of the simulation duration. Flush also has a square time complexity, but it has a smaller constant factor than RADB. Given the efficiency in energy savings of RADB, we conclude that it achieves a better energy-delay trade-off. We finally experimented with the linear algorithm RAD-IP assuming identical power functions. RAD-IP derives the same rate assignment as Flush, yet with a reduced running time, as shown in Figure 6.

V. Related Work

By transmitting packets over a longer delay, the transmission energy consumption can be significantly reduced. Researchers have studied the energy-delay trade-off from different prospectives. A recent focus is to use explicit deadline for each packet [2], [3], [13], [15], [18], [19], [23], [26]. For example, Schurgers *et al.* analyzed energy-efficient packet scheduling in a time-invariant channel [19]. Their approach is limited only to a periodic task model with constant packet inter-arrival times. More generalized offline packet scheduling with individual deadlines were presented by [23] [15] [2] [3]. Zafer and Modiano presented an offline generalization by a calculus approach and obtained an optimal solution on the earliest deadline first (EDF) basis [23]. Miao and Cassandras proposed a different offline generalization with a focus on packets with different delay constraints and number of bits based on first-come-first-served (FCFS) [15]. Chen *et al.* provided an alternative technique in getting the offline optimal solution for packets with identical deadlines using FCFS [2] and more analytical results for i.i.d. input in [3].

The knowledge about all packet arrivals may not be available offline and an online algorithm is more appropriate in dealing with the dynamic nature of input arrivals. There are a number of online schedulers [23] [15] [18] [26] [13] [3]. The authors of [23] [15] also proposed online schedulers extended from their offline counterparts by requiring partial knowledge about future packet arrivals. In [23], Zafer and Modiano presented an online packet scheduler with information of future packet arrivals for a Poisson input. Miao and Cassandras proposed an online scheduler with required knowledge about future packets arrival information within a small given window [15]. By contrast, the online scheduler we consider in this paper is for general input without any assumption about future packets arrivals.

The online schedulers in [18] [26] and the online time-invariant scheduler in [13] all require a priori knowledge input statistics before scheduling. Rajan *et al.* [18] used a deadline to each packet for an i.i.d. input. They formalized the optimization problem and solved it using a value iteration algorithm. The computational complexity of their algorithm grows exponentially with the delay constraint and the complexity becomes prohibitive when delay gets larger than 3 time units. This limits its usage in environments with limited power and large deadlines.

Zhong and Xu studied online packet scheduling by assuming known input time correlation with a focus on providing statistical quality of service (QoS) control [26]. The QoS control was based on the observation that the use of packet transmission rates slowdown may lead to unexpected delay violations and packet losses. They established bounds on transmission capacity and buffer size for the purpose of performance guarantee. By contrast, this paper focuses on energy-efficient packet scheduling.

Khojastepour and Sabharwal considered a discrete delay constraint for packets with the same size [13]. They established the connection between maximum delay scheduling and a linear filter. Two optimal weighted fair queueing schedulers were proposed. One is a time-invariant (static) policy for i.i.d. input, which determines the transmission rates of new packets by their sizes divided by deadlines. This policy, in essence, leads to the same energy consumption as the Average Rate algorithm for aperiodic tasks in [21] for aperiodic tasks and the online algorithm for sporadic tasks in [17] under earliest deadline first scheduling, both for CPU tasks scheduling using Dynamic Voltage and Frequency Scaling (DVS). The other is a time-variant (adaptive) policy, which makes scheduling decisions according to new packet arrivals and uncompleted packets in the queue backlog. Each time packets arrive, all arrivals in the backlog must be iterated in order for the algorithm to derive new transmission rates for the new arrival and updated rates for those in the backlog. At each invocation, the policy needs $O(n^2)$ to complete, where n is the backlogged arrivals. This policy leads to the same energy consumption as the Optimal Available algorithm for aperiodic tasks in [21], the algorithm for sporadic tasks in [12], and the *linear* algorithm for aperiodic tasks in [25]. A proof of the equality is out of the scope of this paper.

Most recently, Chen *et al.* proposed two online schedulers (referred as Flush and IMET) without knowledge about future packet arrivals [3]. They assumed identical power characteristics and delay constraints for all packets. (Though packets have the same relative deadlines at the arrival time, at any particular time packets can have different remaining timing constraints). They showed the online scheduler, Flush, was more energy-efficient; it achieved a comparable energy consumption compared with their offline approach. The algorithm is a generalization of the time-variant policy in [13] and the static policy in [23]. The static version of all three algorithms lead to the same energy consumption. In this work, we consider a generalized case in which packets can be transmitted to different receivers at different distances, their power characteristics and delay constraints may vary significantly.

Another limitation of all the above solutions is that they only consider the transmission energy. The total power consumption is not necessarily dominated by the transmission power, especially at a short communication range. In this paper, we consider the impact of different receivers and the existence of circuit power. In addition, we separate the problem of rate assignment from the actual packet transmission ordering (scheduling) so as to simplify the rate control problem and make it possible for the choice of service disciplines independent of rate control.

VI. Conclusion

We consider energy minimization over an AWGN channel with a delay constraint for each packet. The energy consumption includes both the transmission energy and static circuit energy. We showed that the existence of circuit power put a limit on the minimum energy-efficient rate. Operating at a rate lower than the limit results in increases of both transmission time and energy. The best strategy is to transmit packets at a rate no lower than the limit and then put the circuit into low power state with negligible power consumption. In deriving the transmission rates, we consider a general case in which a transmitter communicates with multiple receivers at different distances. Power characteristics of packets can vary significantly. The existence of energy-efficient rates and difference in power functions play a key role in determining transmission rates of a packet. Existing online packet scheduling policies are special cases of our algorithms by assuming zero circuit power and the same power function of all packets. For this special case, we propose a linear algorithm in getting the optimal solution, in contrast to the square complexity of existing policies [13] [23] [3].

We have restricted our focus to communication power in an AWGN channel. An interesting future work direction is to consider the trade-off between both communication and computation slowdown, or the impact of more realistic wireless fading channels.

Acknowledgements

This research was supported in part by U.S. NSF grants ACI-0203592, CCF-0611750, DMS-0624849, CNS-0702488, CRI-0708232, and NASA grant 03-OBPR-01-0049.

References

[1] R. A. Berry and R. G. Gallager. Communication over fading channels with delay constraints. *IEEE Trans. on Information Theory*, 48(5):1135–1149, 2002.

[2] W. Chen and U. Mitra. Energy efficient scheduling with individual packet delay constraints. In *Proc. of the IEEE Infocom*, 2006.

[3] W. Chen, M. Neely, and U. Mitra. Energy efficient scheduling with individual packet delay constraints: Offline and online results. In *Proc. of the IEEE Infocom*, 2007.

[4] K.-H. Cho and H. Samueli. A 8.75-mbaud single-chip digital qam modulator with frequency-agility and beamforming diversity. In *Proc. of the IEEE 2000 Custom Integrated Circuits Conference*, pages 27–30, 2000.

[5] B. Collins and R. L. Cruz. Transmission policies for time varying channels with average delay constraints. In *Proc. of Allerton Conference on Communnication, Control, and Computing*, 1999.

[6] T. Cover and J. Thomas. *Elements of Information Theory (pages 250-253)*. John Wiley & Sons, Inc., 1991.

[7] J. K. Dey, J. F. Kurose, and D. F. Towsley. On-line scheduling policies for a class of iris (increasing reward with increasing service) real-time tasks. *IEEE Trans. Computers*, 45(7):802–813, 1996.

[8] J. K. Dey, J. F. Kurose, D. F. Towsley, C. M. Krishna, and M. Girkar. Efficient on-line processor scheduling for a class of iris (increasing reward with increasing service.) real-time tasks. In *Proc. of the ACM SIGMETRICS Conference*, pages 217–228, 1993.

[9] L. R. Foulds. *Optimization Techniques: An Introduction*. Springer-Verlag, 1981.

[10] A. Fu, E. Modiano, and J. N. Tsitsiklis. Optimal transmission scheduling over a fading channel with energy and deadline constraints. *IEEE Trans. on Wireless Communications*, 5(3):630–641, 2006.

[11] A. E. Gamal, C. Nair, B. Prabhakar, E. Uysal-Biyikoglu, and S. Zahedi. Energy-efficient scheduling of packet transmissions over wireless networks. In *Proc. of the IEEE Infocom*, 2002.

[12] I. Hong, M. Potkonjak, and M. B. Srivastava. On-line scheduling of hard real-time tasks on variable voltage voltage processor. In *Proc. of the Int'l Conf. on Computer-Aided Design (ICCAD)*, pages 653–656, 1998.

[13] M. A. Khojastepour and A. Sabharwal. Delay-constrained scheduling: Power efficiency, filter design, and bounds. In *Proc. of the IEEE Infocom*, 2004.

[14] C. Krishna and K. Shin. *Real-Time Systems*. New York: McGraw-Hill, 1997.

[15] L. Miao and C. G. Cassandras. Optimal transmission scheduling for energy-efficient wireless networks. In *Proc. of the IEEE Infocom*, 2006.

[16] M. J. Neely. Optimal energy and delay tradeoffs for multi-user wireless downlinks. In *Proc. of the IEEE Infocom*, 2006.

[17] A. Qadi, S. Goddard, and S. Farritor. A dynamic voltage scaling algorithm for sporadic tasks. In *Proc. of the IEEE Real-Time Systems Symp. (RTSS)*, pages 52–62, 2003.

[18] D. Rajan, A. Sabharwal, and B. Aazhang. Delay-bounded packet scheduling of bursty traffic over wireless channels. *IEEE Trans. on Information Theory*, 50(1):125–144, 2004.

[19] C. Schurgers, V. Raghunathan, and M. B. Srivastava. Power management for energy-aware communication systems. *ACM Trans. Embedded Comput. Syst.*, 2(3):431–447, 2003.

[20] E. Uysal-Biyikoglu, B. Prabhakar, and A. E. Gamal. Energy-eficient packet transmission over a wireless link. *IEEE/ACM Trans. on Networking*, 10(4):487–499, 2002.

[21] F. Yao, A. Demers, and A. Shenker. A scheduling model for reduced cpu energy. In *IEEE Foundations of Computer Science (FOCS)*, pages 374–382, 1995.

[22] Y. Yu, B. Krishnamachari, and V. K. Prasanna. Energy-latency tradeoffs for data gathering in wireless sensor networks. In *Proc. of the IEEE Infocom*, 2004.

[23] M. Zafer and E. Modiano. A calculus approach to minimum energy transmission policies with quality of service guarantees. In *Proc. of the IEEE Infocom*, 2005.

[24] F. Zhang and S. T. Chanson. Throughput and value maximization in wireless packet scheduling under energy and time constraints. In *Proc. of the IEEE Real-Time Systems Symp. (RTSS)*, pages 324–334, 2003.

[25] X. Zhong and C.-Z. Xu. Energy-aware modeling and scheduling for dynamic voltage scaling with statistical real-time guarantee. *IEEE Trans. on Computers*, 56(3):358–372, 2007.

[26] X. Zhong and C.-Z. Xu. Energy-efficient wireless packet scheduling with quality of service control. *IEEE Trans.on Mobile Computing*, 6(10):1158–1170, 2007.

[27] X. Zhong and C.-Z. Xu. Frequency-aware energy optimization for real-time periodic and aperiodic tasks. In *Proc. of the ACM SIGPLAN/SIGBED Conf. on Languages, Compilers, and Tools for Embedded System (LCTES)*, pages 21–30, 2007.