This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2016.2621359, IEEE Journal on Selected Areas in Communications

Energy-efficient Transmission with Data Sharing in Participatory Sensing Systems

Weiwei Wu, *Member, IEEE,* Jianping Wang, *Member, IEEE,* Minming Li, *Senior Member, IEEE,* Kai Liu, *Member, IEEE,* Feng Shan, *Member, IEEE,* Junzhou Luo, *Member, IEEE*

***Abstract*—In a participatory sensing system, data sensed from smartphone users is shared with the general public who requests data through submitting tasks. When multiple tasks request the data from a mobile user, the mobile user can make a transmission schedule to achieve the balance between the amount of data transmitted and energy consumption. Intuitively, reducing the amount of data transmitted by making use of data sharing between the tasks can save the energy consumption. However, due to the convexity of rate-power function for rate-adaptive transmitting devices, a schedule purely minimizing the amount of data transmitted may not always be the optimal one minimizing the energy consumption. Thus there exists a trade-off between the amount of data transmitted and energy consumption. This paper formulates the problem as a bi-objective optimization problem to simultaneously minimize the amount of data transmitted and the energy consumption. Two task models are studied, FIFO (*first-in-first-out*) task model and AD (*arbitrary deadline*) task model, respectively. We first provide optimal algorithms for the offline case. We then study the online case where requests arrive dynamically without prior information. For FIFO tasks, we develop an online algorithm that is $O(\ln L)$-competitive with respect to both the amount of data transmitted and energy consumption, where L is the longest length of the time duration of the tasks. For AD tasks, we devise an online algorithm that is $O(\ln^2 L)$-competitive with respect to both the amount of data transmitted and energy consumption. Our simulation results validate the efficiency of our online algorithms.**

I. Introduction

Participatory sensing with smartphones has become a compelling paradigm with the proliferation of smartphones, in which data collected is shared with the general public [1], [2]. A participatory sensing system consists of a platform as the coordinator, a group of mobile users as data providers,

W. Wu, F. Shan and J. Luo are with School of Computer Science and Engineering, Southeast University, Nanjing, Jiangsu, P. R. China (Emails: {weiweiwu, shanfeng, jluo}@seu.edu.cn).

J. Wang and M. Li are with Department of Computer Science, City University of Hong Kong, Hong Kong (Emails: {jianwang, minming.li}@cityu.edu.hk).

K. Liu is with College of Computer Science, Chongqing University, Chongqing, P.R. China (Email: liukai0807@cqu.edu.cn).

This work is supported in part by National Natural Science Foundation of China under Grants No. 61300024, No. 61632008, No. 61320106007 and No. 61572088, Jiangsu Provincial Natural Science Foundation of China under Grants No. BK20130634 and No. BK20140648, Jiangsu Provincial Key Laboratory of Network and Information Security under Grant No. BM2003201, China Specialized Research Fund for the Doctoral Program of Higher Education under Grant No. 20130092120036, Key Laboratory of Computer Network and Information Integration of Ministry of Education of China under Grants No. 93K-9 and Collaborative Innovation Center of Wireless Communications Technology, Hong Kong Research Grant Council under CRF C7036-15G, NSFC-Guangdong Joint Fund under project U1501254, Research Grants Council of the Hong Kong Special Administrative Region, China [Project No. CityU 117913].

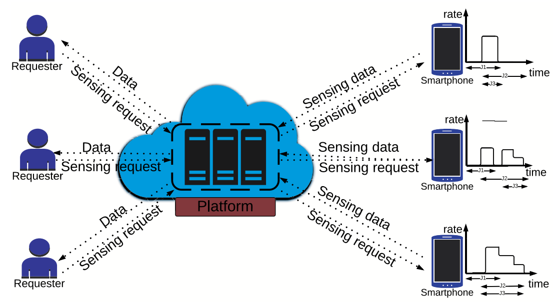


Fig. 1. The architecture of a participatory sensing system.

and the general public as data requesters. Fig. 1 shows the architecture of a participatory sensing system. The general public submits multiple data sensing tasks to the platform, then the platform allocates the tasks to mobile users, and the mobile users transmit the sensed data to the platform. One important feature of the system is data sharing among tasks, where some data from a mobile user can be shared by multiple tasks on the platform, as long as the data meets the time-sensitive QoS constraint.

To reduce the total amount of data transmitted, a mobile user can make a transmission schedule to enable more data sharing among tasks. Such a problem has been studied by [3], [4]. In this paper we further the research by considering the total energy consumption used by transmission, another important concern to mobile users. The relationship between the total data traffic and total energy consumption is not as straightforward as one may assume. Although in general less data traffic through shared data transmission implies less energy consumed, it is not always true because sometimes too much shared data needs a high transmission rate, which may consume more energy. This happens because the rate-power function $p = G(s)$ (which specifies the power p consumed to achieve a desired rate s) is convex in nature for rate-adaptive devices [5], [6], [7].

We now use an example shown in Fig. 2 to illustrate the conflict between data traffic minimization and energy minimization. Fig. 2(a) gives a schedule with the input of one task, J_1, requesting an amount w of data within the interval $[1, 2l]$. In such a case, transmitting data with a constant rate $\frac{w}{2l}$ during the time interval of $[1, 2l]$ will lead to the minimum energy consumption. However, when there are two tasks with the possibility for data sharing, the schedule to minimize the total data traffic will not lead to the minimum energy consumption. Suppose that there is another task J_2, requesting

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2016.2621359, IEEE Journal on Selected Areas in Communications

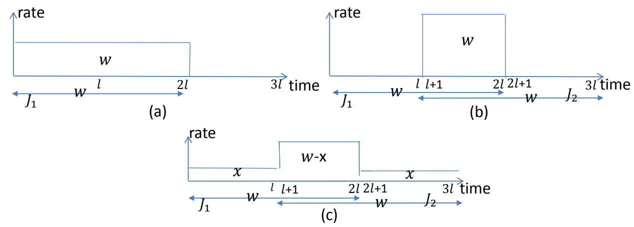


Fig. 2. The trade-off between the data traffic and energy consumption.

an amount w of data within time interval $[l+1, 3l]$. We can see that,

- To minimize the total data traffic, the schedule in (b) transmits all data (with a total amount w) in the overlap of the two time intervals to make it shared by the two tasks. Under such a schedule, the total amount of energy consumption is $l \cdot G(\frac{w}{l})$.
- To minimize the total energy consumption, the schedule in (c) decreases the traffic in the mid area to be $w-x$ and increases the traffic in the low-rate areas to be x where the variant x is a positive real number to be discussed later, resulting in a total amount $w - x + 2x = w + x > w$ of data traffic and an amount $2l \cdot G(\frac{x}{l}) + l \cdot G(\frac{w-x}{l})$ of energy consumption.

In contrast to the intuition that decreasing the data traffic can reduce the energy consumption, we can observe from (b) and (c) in Fig. 2 that, transmitting more data traffic may cause less energy consumption. This leads to a need to design new transmission policies to address the trade-off between the data traffic and energy consumption. Although there may not exist a schedule minimizing the data traffic and energy consumption simultaneously, as shown later, we can always find a schedule of which both data traffic and energy consumption can approach the minimum data traffic and the minimum energy consumption within a good bound, respectively.

To address the trade-off above, this paper formulates the problem as a bi-objective optimization problem to reduce both the data traffic and energy consumption. To model the delay constraints, we consider the following two models, FIFO task model and AD task model. The former typically models the FIFO schedulers with *first-in-first-out* service rule that earlier arrived tasks have earlier deadlines [8], [9]; while the latter generalizes the setting in the former to tasks with *arbitrary deadlines*.

To model the tasks requiring data transmission, we consider both offline setting and online setting. In the offline setting, all information is known in advance, i.e. the arrivals and departures of tasks. In the online setting, we assume arbitrary task arrivals and departures to model the general online setup of rate scheduling, which can provide a worst-case guarantee. We measure the performance of online algorithms in terms of the bi-objectives with the paradigm of competitive analysis [10], which guarantees that the output (both data traffic and energy consumption) of an online algorithm always approximates the optimal offline solution within a bounded factor for all possible inputs.

Developing online scheduling algorithms with proven good performance bounds for both FIFO task model and AD task model is quite challenging. First, the optimal schedule that minimizes the data traffic may not always be the optimal one that minimizes the energy consumption. Second, the optimal solution for minimizing the energy consumption has complex structures and does not admit a combinatorial algorithm due to irregular intersections/sharing of the intervals of the tasks (which can be seen from the example above that the value x can be an irrational number correlated with the rates of other tasks). To the best of our knowledge, no online algorithms with competitiveness for minimizing the energy consumption (even regardless of the data sharing and data traffic) are known in the literature of rate scheduling.

In this paper, we propose optimal algorithms for the offline setting and develop online algorithms with proven worst-case performance bounds with respect to bi-objectives. Since the structure of the optimal solution is too complicated to be characterized, it is almost impossible to directly bound the performance. To this end, we propose two online decomposition methods that are able to partition the tasks into subsets with well-characterized properties (*e.g.* disjoint property, bi-monotonicity) respectively for FIFO tasks and AD tasks, and accordingly devise novel algorithms/analysis to tackle the partitioned tasks, thus indirectly bound the competitiveness of the online algorithm running on the original tasks by merging the performance bounds.

Our contributions are summarized as follows.

- This paper investigates an energy-efficient transmission problem with data sharing in participatory sensing systems in which a trade-off between data traffic and energy consumption is discovered. We theoretically address such a trade-off in terms of bi-objective optimization.
- For the offline case, we provide optimal algorithms with respect to bi-objectives. For common deadline tasks, an iteration-based algorithm is proved optimal to simultaneously minimize the data traffic and the energy consumption. For general tasks with individual deadlines, we provide a method to compute the optimal solutions respectively minimizing the data traffic and the energy consumption
- For online FIFO tasks, we develop an online algorithm that achieves $O(\ln L)$-competitiveness with respect to both of the bi-objectives, i.e., the overall data traffic and the overall energy consumption always approximate the offline optimal solution within a factor of $O(\ln L)$.
- For online AD tasks, we develop an online algorithm that achieves $O(\ln^2 L)$-competitiveness with respect to both of the bi-objectives.

The remains of this paper are organized as follows. Section II reviews the related work. In Section III, we formulate the model and introduce the energy-efficient transmission problem with bi-objectives. In Section IV, we provide the optimal algorithms for the offline case. In Section V, we study FIFO task model and develop an online algorithm with proven $O(\ln L)$-competitiveness. Different from our preliminary work [11], the competitive ratio derived removes its dependency

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2016.2621359, IEEE Journal on Selected Areas in Communications

on the total number T of time slots that is usually large, and moreover, we further investigate the generalized AD task model by developing an $O(\ln^2 L)$-competitive algorithm in Section VI. Section VII performs the simulations for our online algorithms and validates their efficiency. Finally, we conclude the paper in Section VIII.

II. Related work

Extensive research work has been done respectively in the field of participatory sensing and rate scheduling. We only review the most related ones due to space limit.

Participatory sensing: Recently, with its attractive applications, participatory sensing has attracted extensive research attention, both from industry and academia. Various issues in participatory sensing have been addressed in the literature [1], [2], [12], [13]. For example, [14], [15], [16] use social-media-based crowdsourcing to build knowledge base of urban emergency events; [18], [19] consider the privacy preserving problem in participatory sensing; [20], [21] study the incentive mechanism design problem in incentivizing truthfulness and users' participation; [17] considers location-aware collaborative sensing in mobile crowdsourcing. The task allocation issues in crowdsourcing markets are investigated in [22], [23], without concerning about data sharing, however. The data sharing problem within multiple applications is initially formulated in the field of wireless sensing systems, in which Tavakoli *et al.* [3] and Fang *et al.* [4] consider constant-rate schedules and develop algorithms to minimize the communication overhead. Observing the data sharing nature, Zhao *et al.* [24] extend their work to study the sensing task allocation problem in participatory sensing system with the aim of load balancing over multiple participants.

Energy-efficient rate-adaptive scheduling: Much research effort has been made to design rate-adaptive transmission algorithms (without considering data sharing). Various objectives (*e.g.,* throughput, delay, energy consumption) are investigated in prior works which can be referred to in a recent survey [25]. [26], [9], [27] devise the optimal rate schedule to minimize the energy consumption in the offline setting, respectively with the input of common deadline tasks and FIFO tasks. Gatzianas *et al.* [28] investigate the system utility maximization problem from a stochastic aspect. [29], [30], [31], [32] are the first works to theoretically study the algorithms in the general online setup, without relying on any prior information. Among them, Vaze *et al.* [29] develop online algorithms to minimize the completion time; [30], [31], [32] propose online algorithms with competitive ratios to maximize the data throughput. In the literature, no online rate scheduling algorithms with competitive ratios are known for minimizing the energy consumption.

To the best of our knowledge, no prior works considered the rate-adaptive transmission policies with data sharing in participatory sensing systems. This paper addresses such a problem and we notice that there is a surprising conflict between the data traffic and energy consumption in such a rate-adaptive scenario, which is in contrast to the constant-rate scenario [4], [24] where data traffic minimization is consistent with the energy minimization, thus we attempt to balance such a trade-off by developing online algorithms with performance bounds in terms of bi-objectives.

TABLE I
Key notations

Symbol	Semantics
$\mathcal{J}$	input task set
J_i	i^{th} task
r_i	arrival time of task J_i
d_i	deadline of task J_i
w_i	amount of data requested by task J_i
L	the longest length of the time duration among all tasks in $\mathcal{J}$
T	the latest deadline of the tasks in $\mathcal{J}$
$p = G(s)$	rate-power function, the power consumed to achieve the rate s
$s(t)$	data rate specified in time t
$W(A)$	data traffic caused by schedule A
$E(A)$	energy consumption caused by schedule A

III. Preliminaries

In this section, we first introduce the system model, and then present the problem formulation.

A. System model

We model a QoS-constrained request in a participatory sensing system as a task with an arrival time and a deadline, specifying the timeliness of the data request. The time is partitioned into discrete time slots, $1, 2, ..., T$, and we use time interval $[t_1, t_2]$ to refer to time slots $t_1, t_1 + 1, \ldots, t_2$. Thus the length of the interval is $t_2 - t_1 + 1$ time slots. A task J_i, $i = 1, \ldots, n$, can be defined by (r_i, d_i, w_i), where r_i is the arrival time, d_i is the deadline, and w_i is the amount of data requested. To be specific, task J_i can receive data from the beginning of time slot r_i to the end of time slot d_i. For simplification, we just say task J_i arrives at r_i and departures at d_i. The amount of transmitted data in interval $[r_i, d_i]$ should be at least w_i, which is called the *delay/time constraint*. All tasks form the input task set $\mathcal{J} = \{J_1, J_2, ..., J_n\}$. Let $L = \max_{1\le i\le n} d_i - r_i + 1$ be the longest length of the time duration of the tasks.

Without loss of generality, we assume $\min_i r_i = 1$ and $\max_i d_i = T$. For each task $J_i = (r_i, d_i, w_i)$, we say that task J_i remains alive at $t \in [r_i, d_i]$. Also, r_i (d_i) is called *an arrival (deadline) time/point*, and consequently, from time 1 to T, there are $2n$ such points. The time interval between two adjacent points is called a *block/epoch*.

In this paper, we consider two task models, FIFO task model and AD task model. The FIFO models the first-in-first-out service rule where tasks have deadlines in the same order as the arrival times, which is the most general task model studied in the literature of rate scheduling [8], [9]. The AD task model further generalizes FIFO to tasks with Arbitrary Deadlines to model the most general QoS requirement.

We consider a single user point-to-point transmission channel and make the same assumption as previous works that the transmitter can adaptively change its transmission rate s, which is related to transmission power p through a function $p = G(s)$ called *rate-power* function. In many systems with realistic encoding/decoding schemes, the rate-power function

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2016.2621359, IEEE Journal on Selected Areas in Communications

is convex in nature [9], [26], such as the optimal random coding in single-user additive White Gaussian Noise (AWGN) channel, where $s = G^{-1}(p) = \frac{1}{2}\log(1 + \frac{p}{N})$ and N is the thermal noise level and often assumed $N = 1$, and inversely $p = G(s) = e^{2\ln 2 \cdot s} - 1$.

In this paper, instead of investigating a specific rate-power function, we target at the general rate-power function $p = G(s) = e^{\alpha s} - 1$ with constant $\alpha > 0$. To simplify the presentation, we write $p = G(s) = e^{s} - 1$ for short when α does not affect the final performance bound in our deduction. Table I summarizes the key notations for the ease of reference.

B. Problem formulation

We introduce an energy-efficient transmission problem where a transmitter needs to transmit its data, shared by multiple tasks, to the platform with the minimum data traffic and energy consumption. A schedule adapts its rate at the beginning of a time slot (or equivalently at the end of the proceeding time slot). A scheduling algorithm A generates a rate schedule/allocation that specifies the data rate $s(t)$ to transmit at time t. The sensed data can be shared by multiple tasks as long as it fits in the time intervals of the tasks. A *feasible* schedule must fulfill the data requirement of all tasks within their *delay constraints*. That it,

$$\sum_{r_i \le t \le d_i} s(t) \ge w_i, \ \forall J_i. \tag{1}$$

Note that the data transmitted with rate $s(t)$ in time slot t can be shared by (or equivalently used to meet the requirement of) any task J_i alive at time $t \in [r_i, d_i]$.

The *data traffic* incurred by an algorithm A, denoted as $W(A)$, is the total amount of data transmitted,

$$W(A) = \sum_{1 \le t \le T} s(t). \tag{2}$$

The corresponding *energy consumption* is denoted as $E(A)$ where

$$E(A) = \sum_{1 \le t \le T} G(s(t)). \tag{3}$$

In this paper, we consider a bi-objective energy-efficient scheduling problem to minimize both the overall data traffic $W(A)$ and the overall energy consumption $E(A)$, as defined below.

Definition 1. *The bi-objective energy-efficient rate scheduling problem is to determine the rate $s(t)$ over time so as to minimize both the data traffic (2) and energy consumption (3) with the satisfaction of all the delay constraints (1), respectively in the offline setting and online setting.*

Such a definition seems inconsistent to our previous observation because there is a conflict between these two objectives. Interestingly, we will show that it is possible to achieve the simultaneous minimization of both objectives for tasks with a common deadline in the offline problem. Moreover, for the online problem, we can have a schedule that has bounded competitive ratio simultaneously on both objectives.

In the offline setting, full task information is known. In the online setting, task arrives over time without prior information. An online algorithm needs to decide the scheduling strategy on the arrival of the tasks, without relying on any distribution or future information. We adopt the paradigm of competitive analysis, which is widely used to measure the worst-case performance of online algorithms, where an online algorithm ALG is compared with the optimal offline algorithm OPT with full information (as benchmark).

We measure the online algorithm in terms of bi-objectives, the data traffic and the energy consumption. To simplify the presentation, we abuse the notation and use OPT (or s^{opt}) to represent these optimal solutions (or optimal rate functions), without distinguishing between the one minimizing the data traffic and the one minimizing the energy consumption, if no ambiguity arises. An online algorithm is said *γ-data λ-energy competitive* if it always outputs an online schedule respectively within γ times and λ times of the optimal offline solutions with respect to the overall data traffic and energy consumption for any input σ (which is a set of tasks to be served in this paper). That is,

$$\frac{W(ALG(\sigma))}{W(OPT(\sigma))} \le \gamma, \ \forall \sigma \tag{4}$$

and

$$\frac{E(ALG(\sigma))}{E(OPT(\sigma))} \le \lambda, \ \forall \sigma \tag{5}$$

where $W(ALG(\sigma)), W(OPT(\sigma))$ are the total data traffic in the online algorithm and the optimal offline algorithm respectively for a given input σ, and $E(ALG(\sigma)), E(OPT(\sigma))$ are respectively the total energy consumption in the online algorithm and the optimal offline algorithm respectively for a given input σ.

Thus, this paper aims at designing both offline algorithms and online algorithms with proven performance bounds with respect to the bi-objectives to minimize the data traffic and the energy consumption.

IV. Optimal Offline Rate Schedule

In this section, we study the optimal rate scheduling algorithms.

Our major result here is that, due to the sharing nature of data and the convexity of the rate-power function, the energy consumption can be reduced by properly transmitting with more workload, instead of intuitively transmitting with less workload. Furthermore, the optimal solution for minimizing the energy consumption has a surprisingly complex structure. This can be shown by a simple instance with two tasks as follows.

Example 1: Consider $\mathcal{J} = \{J_1, J_2\}$, where $J_1 = (1, 2, 2)$ and $J_2 = (2, 3, 2)$. Obviously, the optimal schedule for minimizing the data traffic is to transmit at rate $s_1 = 0, s_2 = 2, s_3 = 0$ respectively at time $t = 1, 2, 3$ (transmitting with an amount $w = 2$ of data traffic in total and an amount $e^2 - 1$ of power consumption). Since these two tasks have the same average rate $w(J_1) = w(J_2) = 1$ and lower rate consumes less energy, one promising schedule for minimizing the energy consumption is to schedule task J_1 with rate $s_1 = 1, s_2 = 1$ at time $t = 1, 2$, and moreover, let task J_2 share the rate at time 2 and finally finish its remaining workload with rate $s_3 = 1$ at time 3. Compared with the first schedule, the second schedule does reduce the energy consumption (with an amount $3(e-1)$ consumed) by increasing the total data transmission (transmitting with an amount $w - x + 2x = 2 - 1 + 2$ of data traffic in total) in the low-rate areas, instead of intuitively transmitting less data traffic. Surprisingly, further careful calculation will give the result that the rate $s_2 = 1+\ln\sqrt{2}, s_1 = s_3 = 1-\ln\sqrt{2}$ (transmitting with a total amount $w + x = 2 + (1 - \ln\sqrt{2})$ of data traffic) is actually the optimal solution for minimizing the energy consumption (minimizing $e^{2-s_2} + e^{s_2} + e^{2-s_2} - 3$ under the delay constraints of $s_1 + s_2 \geq 2$ and $s_2 + s_3 \geq 2$).

Motivated by the above observation, We first focus on common deadline tasks (with $d_i = T$ for all $1 \leq i \leq n$) and show that an iteration-based algorithm can simultaneously achieve the minimum data traffic and minimum energy consumption. Then, we will provide optimal algorithms respectively to minimize the data traffic and the energy consumption for general tasks with arbitrary deadlines.

A. Optimal schedule for common deadline tasks

In this section, we develop an algorithm that is able to simultaneously minimize the energy consumption and data traffic. Due to the convexity of the rate-power function, an energy-efficient schedule prefers to assign a low rate, instead of a high rate, to complete the workload. This can be seen from the following example. Suppose that a schedule transmits with rates $s(t)$ and $s(t+1)$ in two consecutive time slots t and $t+1$, say $s(t) > s(t+1)$. Then, we can decrease the power consumption by averaging these two rates to be $\frac{s(t)+s(t+1)}{2}$ since $G(s(t)) + G(s(t+1)) \geq 2G(\frac{s(t)+s(t+1)}{2})$ by the convexity. This is called the *equalization* method [27]. However, we should apply the equalization method carefully, because the delay constraints of the tasks may be violated during the equalization. The following lemma can be proved easily by applying the equalization method with the proof moved to Appendix A.

Lemma 1. *The rate function in the optimal solution is a step function that changes its rate either at an arrival time or at a deadline point. If the optimal solution increases its rate at time t, then t is an arrival time. If the optimal solution decreases its rate at time t, then t is a deadline point.*

Let the rate function in the optimal solution for minimizing the energy consumption be s^{opt}. Define the *average rate* of a task J_i to be $w(J_i) = \frac{w_i}{d_i - r_i + 1}$. Assume that J_m is the task that achieves the largest average rate $w(J_m) = \max_{1 \leq i \leq n} \frac{w_i}{T - r_i + 1}$. The following lemma finds the interval with the largest rate in s^{opt}, which is proved by equalizing the rate inside interval $[r_m, T]$ to the rate outside that interval.

Lemma 2. *For common deadline tasks, if J_m is the task that achieves the largest average rate $w(J_m) = \max_{1 \leq i \leq n} \frac{w_i}{T - r_i + 1}$, then the optimal solution minimizing the energy consumption has rate $s^{opt}(t) = w(J_m)$ in interval $[r_m, T]$.*

Proof: Let $r = w(J_m) = \max_{1 \leq i \leq n} \frac{w_i}{T - r_i + 1}$. Since J_m achieves the largest average rate $w(J_m)$ among all possible intervals $[r_i, T]$, allocating by averaging w_m over interval $[r_m, T]$ will accomplish all tasks with $r_i \geq r_m$. This ensures that transmitting at rate r in interval $[r_i, T]$ is feasible for tasks with $r_i \geq r_m$. Moreover, with amount w_m of data to be finished in that interval, transmitting at the constant rate r minimizes the energy consumption by the convexity of the rate-power function. The total amount of data transmitted in that interval clearly cannot be less than w_m.

To show that transmitting at rate r in interval $[r_m, T]$ is optimal, we still have to prove that the total data finished in that interval will not exceed w_m. This can be shown by contradiction. If amount $E + \epsilon$ of data is transmitted in interval $[r_m, T]$ in the optimal solution, then this part of data will be averaged and transmitted in that interval with rate $\frac{w_m + \epsilon}{T - r_m + 1}$ to minimize the energy consumption. We then prove that there exists at least one time slot $t < r_m$ at which the optimal rate $s^{opt}(t) \leq r$. This is true since if on the contrary the rate $s^{opt}(t) > r$ for all $t < r_m$, then this would contradict the fact that $[r_m, T]$ achieves the largest average rate among all possible intervals $[r_i, T]$. Thus, we assume $s^{opt}(t) < r$ at time $t < r_m$. Then, we can apply equalization to time slot r_m and t. This would reduce the energy consumption and leads to a contradiction to the optimality.

Therefore, the rate in interval $[r_m, T]$ is exactly r. ■

Algorithm 1 INTERVAL-DELETE

1: find the task J_m that achieves the largest average rate $w(J_m) = \max_{1 \leq i \leq n} \frac{w_i}{T - r_i + 1}$.
2: transmit with rate $w(J_m)$ in interval $[r_m, T]$.
3: **while** there are some intervals that have not been fixed **do**
4: delete interval $[r_m, T]$ and update the deadline of each task J_i with $r_i < r_m$ to be $r_m - 1$ and the remaining workload to be $w_i - |I_i \cap I_m| \cdot w(J_m)$.
5: find the task J_k among the tasks $\{J_i : r_i < r_m\}$ that achieves the largest average remaining rate $w(J_k) = \max\{\max_{i: r_i < r_m} \frac{w_i - |I_i \cap I_m| \cdot w(J_m)}{r_m - r_i}, 0\}$.
6: transmit with rate $w(J_k)$ in interval $[r_k, r_m - 1]$.
7: set $m = k$.
8: **end while**

Consequently, the optimal solution has rate $w(J_m)$ in interval $I_m = [r_m, T]$. For any task J_i with $r_i < r_m$ that intersects with I_m, it has a remaining workload $\max\{\frac{w_i - |I_i \cap I_m| \cdot w(J_m)}{|(I_i \cup I_m) \setminus (I_i \cap I_m)|}, 0\}$ to be finished in $[r_i, r_m - 1]$. Observing this, we develop an algorithm INTERVAL-DELETE. It finds the interval with the largest average rate in s^{opt}, updates the workload of the task to be the remaining workload, and then iteratively finds all the intervals in s^{opt}. Its optimality with respect to the bi-objectives is stated in the following theorem, of which the detailed proof is presented in Appendix A. Note that the rate function returned in Algorithm INTERVAL-DELETE, and correspondingly in the optimal solution, is non-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2016.2621359, IEEE Journal on Selected Areas in Communications

decreasing for common deadline tasks.

Theorem 1. *For common deadline tasks, Algorithm* INTERVAL-DELETE *computes the optimal rate schedule that simultaneously achieves the minimum energy consumption and the minimum data traffic.*

B. Optimal schedule for tasks with arbitrary deadlines

According to the iteration-based method proposed above, one may naturally expect to extend the method to compute the optimal solution for general tasks. However, observing Example 1, we note that the properties of the optimal solution are quite different with the input of general tasks. First, the solution minimizing the energy consumption may not always minimize the data traffic. Second, the optimal rate function may not be non-decreasing. Third, the calculation process to minimize the energy consumption in that example further reminds us that it does not allow us to compute the optimal solution by a combinatorial or iteration-based method.

Instead, we note that the energy minimization problem can be formulated by a convex programming with a pseudo-polynomial number of constraints as follows.

$$\text{PCP: } \min \sum_{t=1}^{T} G(s(t)) \tag{6}$$

$$\text{subject to} \quad \sum_{r_i \le t \le d_i} s(t) \ge w_i, 1 \le i \le n \tag{7}$$

$$s(t) \ge 0, 1 \le t \le T \tag{8}$$

Note that the properties stated in Lemma 1 hold for the optimal solutions (with respect to both the one minimizing the energy consumption and the one minimizing the data traffic): the rate schedule in the optimal solution is a step function that changes rate either at an arrival point or at a deadline point. To remove the pseudo-polynomial number of constraints introduced by the variant $s(t)$ over time t, we partition the time axis according to these time points as follows. Assume that the arrival points and deadlines of all tasks are resorted with an increasing order and denoted as $t_1 \le t_2 \le ... \le t_{2n}$. The interval between every two adjacent points t_k, t_{k+1} is written as $I(k)$ and called an *epoch*. The length of epoch $I(k)$ is denoted by T_k. According to the property of the step function in Lemma 1, the optimal solution keeps the rate constant in each epoch and we denote the optimal rate in the epoch between t_k and t_{k+1} to be s_k. The total number of the rate variants associated with the epochs is at most $2n$.

We thus obtain the following convex programming formulation with $3n$ constraints, which can be solved in polynomial time and lead to the optimal solution minimizing the energy consumption.

$$\text{CP: } \min \sum_{k=1}^{2n} T_k G(s_k) \tag{9}$$

$$\text{subject to} \quad \sum_{I(k) \subseteq [r_i, d_i]} T_k s_k \ge w_i, 1 \le i \le n \tag{10}$$

$$s_k \ge 0, 1 \le k \le 2n \tag{11}$$

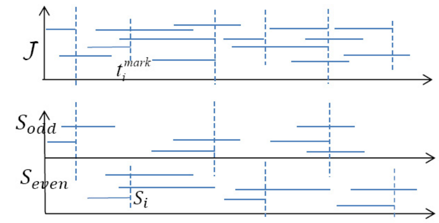


Fig. 3. Online FIFO-decomposition method, where the delay constraint (time interval) of each task is represented as a solid line.

As shown in Example 1, the optimal solution with the minimum energy consumption may not always be the optimal one minimizing the data traffic. We thus separately propose the following linear programming,

$$\text{LPR: } \min \sum_{k=1}^{2n} T_k s_k \tag{12}$$

$$\text{subject to} \quad \sum_{I(k) \subseteq [r_i, d_i]} T_k s_k \ge w_i, 1 \le i \le n \tag{13}$$

$$s_k \ge 0, 1 \le k \le 2n \tag{14}$$

Similar deduction can show that the solution to LPR minimizes the data traffic. We summarize the results above in the following theorem.

Theorem 2. *The solution to convex programming (CP) minimizes the energy consumption. The solution to linear programming (LPR) minimizes the data traffic.*

V. ONLINE RATE SCHEDULE FOR FIFO TASKS

In this section, we investigate FIFO task model and propose an online algorithm to simultaneously minimize the data traffic and energy consumption with bounded competitive ratios.

In fact, to directly bound the performance of an algorithm on all tasks is almost impossible due to the complex structure of the optimal solution. Instead, we propose a decomposition method which partitions the tasks in the optimal solution into several disjoint sets. Such a decomposition method would bring us nice structural properties (*e.g.* disjoint property and bi-monotonicity to be defined later) of the optimal solution for the disjoint sets and help us bound the competitiveness of our algorithm running on the partitioned tasks. By merging performance on the partitioned tasks, we indirectly derive the competitiveness between our online algorithm and the optimal solution with the input of the original tasks.

The arrangement of this section is as follows. First, we present the FIFO-decomposition method and examine the structural property of the partitioned tasks after decomposition. Then, we develop an online algorithm for the partitioned tasks. Next, we prove the performance bound for the algorithm running on the partitioned tasks. Finally, we merge the results and develop a rate schedule that achieves $O(\ln L)$-data $O(\ln L)$-energy competitiveness.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2016.2621359, IEEE Journal on Selected Areas in Communications

A. FIFO-decomposition for FIFO tasks

We first develop an online decomposition method for FIFO tasks, called *FIFO-decomposition*, so that we can focus on the tasks in the partitioned group for which the optimal solution has good structures.

The *FIFO-decomposition* works as follows. We mark the earliest deadline of the tasks to be $t_1^{mark} = d_1$. All tasks alive at time t_1^{mark} will be placed into set $S_1 = \{J_i : t_1^{mark} \in [r_i, d_i]\}$. Then, for the remaining tasks that arrive after t_1^{mark}, we mark the earliest deadline d_i to be $t_2^{mark} = d_i$, and let all tasks alive at time t_2^{mark} form set S_2. Iteratively, we can obtain a set S_q formed by the tasks that arrive after t_{q-1}^{mark} and remain alive at time t_q^{mark}, which is the earliest deadline of tasks that arrive after t_{q-1}^{mark}. Assuming that we obtain Q sets in total after decomposition, we regroup the sets with odd index to be *odd group* $S_{odd} = \cup_k S_{2k-1}$ and the sets with even index to be *even group* $S_{even} = \cup_k S_{2k}$. Obviously, $S_{odd} \cup S_{even} = \mathcal{J}$. Note that, such a decomposition method can be performed in an online manner. Fig. 3 gives an example showing the online decomposition process.

We examine what properties are implied by the FIFO-decomposition proposed above. Given a set S of tasks, let $I(S) = \{[\min_{J_i \in S} r_i, \max_{J_i \in S} d_i]\}$ be the interval of that set.

The FIFO-decomposition leads to the following structural properties.

- (**Disjoint Property**) for every two sets in the same group, say $S_q, S_{q'} \in S_{odd}$ with $q \neq q'$, their intervals are disjoint, i.e., $I(S_q) \cap I(S_{q'}) = \emptyset$.
- (**Pairwise Intersecting Property**) all tasks in the same set, say S_q, are alive at the same marking time, t_q^{mark}.
- (**Bi-group Property**) there are only two groups after decomposition, S_{odd} and S_{even}.

The first property can be verified as follows. Consider two tasks J_u, J_v that are chosen from two sets S_q, S_{q+2}. Assume task J_k in set S_{q+1} arrives after t_q^{mark} and has deadline $d_k = t_{q+1}^{mark}$. Task J_u arrives by time t_q^{mark} with $r_u \leq t_q^{mark} < r_k$, thus $d_u \leq d_k = t_{q+1}^{mark}$ due to the fact that the deadlines of the tasks follow the order as tasks arrive. Combining with the fact $r_v > t_{q+1}^{mark}$, we have $d_u < r_v$ and hence the intervals of J_u, J_v are disjoint. Thus, the intervals of any two sets $S_q, S_{q'}$ with $q \neq q'$ in the same group are disjoint.

The second property and the third one are obviously true according to the decomposition process. We call the tasks in the same set, say S_q, *pairwise intersecting tasks*.

Besides the structural properties above, more significantly, we note that a critical optimal property holds for the partitioned pairwise intersecting tasks, which is called *bi-monotonicity* and crucial to the design of the online algorithm. Given a rate schedule function, we define *peak interval* to be an interval $[t_a, t_b]$ such that the rate function is constant in that interval and satisfies $s(t_a - 1) < s(t_a)$ and $s(t_b) > s(t_b + 1)$. Fig. 4 shows an example for pairwise intersecting tasks and its bi-monotonicity is defined as below.

Lemma 3. *(***Bi-monotonicity of pairwise intersecting tasks***) For pairwise intersecting tasks S, the optimal rate function (both for minimizing the energy consumption and the data traffic) has at most one peak interval, say $[t_a, t_b]$. That is, the rate function is monotonically non-decreasing from the earliest time $\min_{i \in S} r_i$ to t_b and then monotonically non-increasing till $\max_{i \in S} d_i$.*

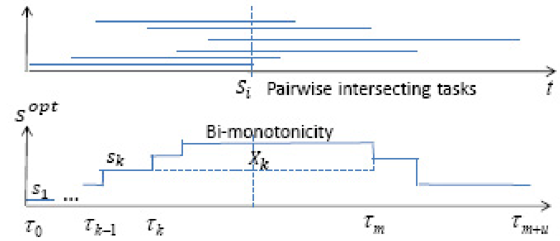


Fig. 4. An example for pairwise intersecting tasks and the bi-monotonicity.

Proof: Suppose on the contrary that there are at least two peak intervals in the optimal solution. We show that this would lead to a contradiction to the property of pairwise intersecting tasks. Assume that $[t_a, t_b], [t_c, t_d]$ are two peak intervals in the optimal solution. Since the optimal solution increases the rate at time t_a, t_c and decreases the rate at t_b, it can be seen that t_a and t_c are arrival times and t_b is a deadline point according to Lemma 1. Note that $t_a < t_b < t_c$. This implies that at least one task has arrival time at t_c which is larger than the other task's deadline at t_b. This contradicts the fact that all tasks are pairwise intersecting. ■

B. Online algorithm for pairwise intersecting tasks

Now we develop an algorithm MAX-REMAIN-ONLINE for pairwise intersecting tasks. The idea of the algorithm is to divide the tasks available at time t into two sets and update the transmission rate according to the comparison of the rates of the tasks in these two sets. The algorithm divides the tasks available at time t into two sets: the tasks that arrive at t, $\{J_i : r_i = t\}$, and the tasks that arrive before, $\{J_i : r_i < t \leq d_i\}$. For any task J_i that arrives at time t, it computes the *average rate* $\mu_i = \frac{w_i}{d_i - r_i + 1}$. The algorithm sets rate function $s(1) = \max_{r_i=1} \mu_i$ at the beginning. For any task that arrives before time t and remains available at time t, it computes the *remaining rate* $\gamma_i(t) = \frac{w_i - \sum_{r_i \leq t' \leq t} s(t')}{d_i - t + 1}$. The algorithm then transmits at rate $s(t) = \max\{\max_{i:r_i=t} \mu_i, \max_{i:r_i<t\leq d_i} \gamma_i(t)\}$ with energy $G(s(t))$ at time t.

According to the strategy of the algorithm, we can see that the algorithm can increase the rate only at a time t, at which a newly arriving task has a larger average rate than the remaining rate of any task that remains alive at that time and arrives before that time. Should this case occur, we have $s(t) = \max_{i:r_i=t} \mu_i > \max_{i:r_i<t\leq d_i} \gamma_i(t)$.

We show an example to illustrate the running process of the algorithm.

Example 2: Set $\mathcal{J} = \{J_1, J_2, J_3\}$ and $J_1 = (1, 3, 6), J_2 = (2, 3, 5), J_3 = (2, 4, 6)$. At time $t = 1$, the algorithm transmits with rate $\mu_1 = 2$ on the arrival of task J_1. At time $t = 2$, tasks J_2 and J_3 arrive, thus the rate is changed to be $\max\{\mu_2, \mu_3, \gamma_1(t)\} = \mu_2 = 2.5$. At time $t = 3$, no task arrives and the algorithm chooses a rate $\max\{\gamma_1(t), \gamma_2(t), \gamma_3(t)\} =$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2016.2621359, IEEE Journal on Selected Areas in Communications

Algorithm 2 MAX-REMAIN-ONLINE

at the beginning time $t = 1$, set initially $\mu_i = \frac{w_i}{d_i - 1 + 1}$ for any task J_i that arrives at time 1 and set the rate $s(1) = \max_{i:r_i=1} \mu_i$
for the arrival of time t **do**
  for each task J_i that arrives at t **do**
    compute the average rate $\mu_i = \frac{w_i}{d_i - r_i + 1}$ for task J_i
  end for
  for each task J_i with $r_i < t \leq d_i$ **do**
    compute the remaining rate $\gamma_i(t) = \frac{w_i - \sum_{r_i \leq t' < t} s(t')}{d_i - t + 1}$ for task J_i
  end for
  transmit at rate $s(t) = \max\{\max_{i:r_i=t} \mu_i, \max_{i:r_i<t\leq d_i} \gamma_i(t)\}$ with energy $G(s(t))$ at time t
end for

$\gamma_2(t) = 2.5$. At time 4, only task J_3 is alive and the algorithm decreases its rate to be $\gamma_3(t) = \frac{6-5}{1} = 1$.

C. Performance of the algorithm running on pairwise intersecting tasks

Based on the bi-monotonicity, we start to study the performance of our online algorithm running on partitioned pairwise intersecting tasks.

We define the following notations for analyzing the algorithm. Let S be the set of partitioned pairwise intersecting tasks, and s^{opt} be the rate function of the optimal solution which follows the bi-monotonicity. Without loss of generality, we assume that $\tau_0 = t_{min} = \min_{i\in S} r_i$ and $\tau_{m+u} = t_{max} = \max_{i\in S} d_i$. Let $[t_a, t_b]$ be the peak interval in s^{opt}, thus s^{opt} is an increasing step function in $[t_{min}, \tau_m = t_b]$ and a decreasing step function in $[t_a, t_{max}]$. Assume that s^{opt} increases the rate at time $\tau_1, \tau_2, ..., \tau_{m-1} = t_a$ and decreases the rate at $\tau_m = t_b, \tau_{m+1}, ..., \tau_{m+u}$. Fig. 4 shows an example of the notations. Write $W(A, [a, b])$ and $E(A, [a, b])$ to be the total amount of data transmitted and energy consumed by schedule A in interval $[a, b]$. Denote by ALG and OPT respectively the schedule in the proposed algorithm and the optimal solution.

The following two lemmas respectively bound the data traffic and energy consumption of our online algorithm with the input of pairwise intersecting tasks.

Lemma 4. *For the input of pairwise intersecting tasks S, in Algorithm* MAX-REMAIN-ONLINE*, the total amount of data transmitted in interval $[\tau_0, \tau_{m+u}]$ is at most $2\ln(\tau_{m+u} - \tau_0 + 1)W(OPT, [\tau_0, \tau_{m+u}])$.*

Proof: Assume that tasks in set S is pairwise intersecting, thus the optimal solution follows bi-monotonicity. Let $s_1, s_2, ..., s_m$ respectively be the increasing rates in s^{opt} in interval $[\tau_0, \tau_m]$. Let $s_{m+1}, s_{m+2}, ..., s_{m+u}$ respectively be the decreasing rates in s^{opt} in interval $[\tau_m, t_{max}]$. Note that for any time t at which s^{opt} increases, which is also called the increasing point, it must be an arrival time of some task (symmetrically, for any time t at which s^{opt} increases, it must be a deadline of some task).

We focus on the interval $[\tau_0, \tau_m]$ first and bound the total amount $W(ALG, [\tau_0, \tau_m])$ of data transmitted in that interval. Note that the algorithm transmits with rate $s(t) = \max\{\max_{i:r_i=t} \mu_i, \max_{i:r_i<t\leq d_i} \gamma_i(t)\}$ at each time point. Therefore, it either increases the rate when a task with large average rate is released and or decreases the rate to be a remaining rate. Accordingly, we say the algorithm increases the rate to be the average rate of task $J_{v(p)}$ or decreases the rate to be the remaining rate of task $J_{v(p)}$. Without loss of generality, we assume that the algorithm sets the rate to be $s(t_{v(p)})$ at time $t_{v(p)}$ to execute $J_{v(p)}$, where $0 \leq p \leq s, \tau_0 \leq t_{v(p)} \leq \tau_m + 1, t_{v(0)} = \tau_0, t_{v(s)} = \tau_m + 1$ and $t_{v(0)} \leq t_{v(1)} \leq ... \leq t_{v(s)}$. Obviously, $s(t_{v(p)}) \leq \frac{w_{v(p)}}{d_{v(p)} - r_{v(p)} + 1}$.

The total amount of data transmitted in interval $[\tau_0, \tau_m]$ is $W(ALG, [\tau_0, \tau_m]) = \sum_{0\leq p\leq s-1} s(t_{v(p)})(t_{v(p+1)} - t_{v(p)}) \leq \sum_{0\leq p\leq s-1} \frac{w_{v(p)}}{d_{v(p)} - r_{v(p)} + 1}(t_{v(p+1)} - t_{v(p)})$ where $s(t_{v(p)})(t_{v(p+1)} - t_{v(p)})$ is the amount of data transmitted in interval $[t_{v(p)}, t_{v(p+1)} - 1]$.

Note that the rate of the algorithm in interval $[t_{v(p)}, t_{v(p+1)} - 1]$ is bounded by $\frac{w_{v(p)}}{d_{v(p)} - r_{v(p)} + 1}$ which is at most $\frac{W(OPT, [r_{v(p)}, d_{v(p)}])}{d_{v(p)} - r_{v(p)} + 1}$ since the optimal solution must complete at least a workload $w_{v(p)}$ in that interval to satisfy the feasibility. We will discuss by considering the following two kinds of tasks, tasks with $d_{v(p)} \leq \tau_m$ and tasks with $d_{v(p)} > \tau_m$.

Consider first the tasks with $d_{v(p)} \leq \tau_m$. By the fact that s^{opt} is non-decreasing in $[r_{v(p)}, \tau_m]$, we have $\frac{W(OPT, [r_{v(p)}, d_{v(p)}])}{d_{v(p)} - r_{v(p)} + 1} \leq \frac{W(OPT, [r_{v(p)}, \tau_m])}{\tau_m - r_{v(p)} + 1}$. As a result,

$$
\begin{aligned}
& s(t_{v(p)})(t_{v(p+1)} - t_{v(p)}) \\
& \leq \frac{w_{v(p)}}{d_{v(p)} - r_{v(p)} + 1}(t_{v(p+1)} - t_{v(p)}) \\
& \leq \frac{W(OPT, [r_{v(p)}, \tau_m])}{\tau_m - r_{v(p)} + 1}(t_{v(p+1)} - t_{v(p)}) \\
& \leq \sum_{t_{v(p)} \leq t \leq r_{t(p+1)} - 1} \frac{(W(OPT, [r_{v(p)}, \tau_m])}{\tau_m - t + 1} \\
& \leq \int_{t_{v(p)}}^{t_{v(p+1)} - 1} (\frac{W(OPT, [r_{v(p)}, \tau_m])}{\tau_m - t + 1})dt \\
& \leq (\ln(\tau_m - t_{v(p)} + 1) - \ln(\tau_m - t_{v(p+1)} + 1))W(OPT, [r_{v(p)}, \tau_m])
\end{aligned}
$$

The third inequality holds since $\frac{W(OPT, [r_{v(p)}, \tau_m])}{\tau_m - r_{v(p)} + 1} \leq \frac{(W(OPT, [r_{v(p)}, \tau_m])}{\tau_m - t + 1}$ with $r_{v(p)} \leq t_{v(p)} \leq t$. The second last one is correct since $\int_{t_{v(p)}}^{t_{v(p+1)} - 1} \frac{1}{\tau_m - t + 1}dt \leq \ln(\tau_m - t_{v(p)} + 1) - \ln(\tau_m - t_{v(p+1)} + 1)$.

Now consider tasks with $d_{v(p)} > \tau_m$. We have $\frac{w_{v(p)}}{d_{v(p)} - r_{v(p)} + 1} \leq \frac{W(OPT, [r_{v(p)}, d_{v(p)}])}{d_{v(p)} - r_{v(p)} + 1} \leq \frac{W(OPT)}{d_{v(p)} - r_{v(p)} + 1}$. Thus,

$$
\begin{aligned}
& s(t_{v(p)})(t_{v(p+1)} - t_{v(p)}) \\
& \leq \frac{w_{v(p)}}{T - r_{v(p)} + 1}(t_{v(p+1)} - t_{v(p)}) \\
& \leq \frac{W(OPT)}{d_{v(p)} - r_{v(p)} + 1}(t_{v(p+1)} - t_{v(p)}) \\
& \leq \sum_{t_{v(p)} \leq t \leq t_{v(p+1)} - 1} \frac{W(OPT)}{d_{v(p)} - t + 1} \\
& \leq (\ln(d_{v(p)} - t_{v(p)} + 1) - \ln(d_{v(p)} - t_{v(p+1)} + 1))W(OPT) \\
& \leq (\ln(\tau_m - t_{v(p)} + 1) - \ln(\tau_m - t_{v(p+1)} + 1))W(OPT).
\end{aligned}
$$

The second last inequality holds because $\ln(d_{v(p)} - t_{v(p)} + 1) - \ln(d_{v(p)} - t_{v(p+1)} + 1) \leq \ln(\tau_m - t_{v(p)} + 1) - \ln(\tau_m - t_{v(p+1)} + 1)$ when $d_{v(p)} > \tau_m$.

Therefore, for both cases, we have obtained that $s(t_{v(p)})(t_{v(p+1)} - t_{v(p)}) \leq (\ln(\tau_m - t_{v(p)} + 1) - \ln(\tau_m - t_{v(p+1)} + 1))W(OPT)$. Summing up all the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2016.2621359, IEEE Journal on Selected Areas in Communications

amount of data transmitted by Algorithm MAX-REMAIN-ONLINE in interval $[\tau_0, \tau_m]$, we have $W(ALG, [\tau_0, \tau_m]) = \sum_{0 \le p \le s-1} s(t_{v(p)})(t_{v(p+1)} - t_{v(p)}) \le \sum_{0 \le p \le s-1} (\ln(\tau_m - t_{v(p)} + 1) - \ln(\tau_m - t_{v(p+1)} + 1))W(OPT) \le (\ln(\tau_m - t_{v(0)} + 1) - \ln(\tau_m - t_{v(s)} + 1))W(OPT) = (\ln(\tau_m - \tau_0 + 1))W(OPT)$.

For the remaining interval $[\tau_m, \tau_{m+u}]$, we can symmetrically derive that $W(ALG, [\tau_{m-1}, \tau_{m+u}]) \le (\ln(\tau_{m+u} - \tau_{m-1} + 1))W(OPT) = (\ln(\tau_{m+u} - \tau_{m-1} + 1))W(OPT, [\tau_0, \tau_{m+u}])$ based on the same fact $s(t_{v(p)}) \le \frac{w_{v(p)}}{d_{v(p)} - r_{v(p)} + 1}$. Combining the results, we have $W(ALG) \le W(ALG, [\tau_0, \tau_{m+u}]) + W(ALG, [\tau_0, \tau_{m+u}]) \le 2(\ln(\tau_{m+u} - \tau_0 + 1)W(OPT, [\tau_0, \tau_{m+u}])$. ■

Lemma 5. *For the input of pairwise intersecting tasks S, in Algorithm* MAX-REMAIN-ONLINE, *the energy consumed in interval $[\tau_0, \tau_{m+u}]$ is at most $2\ln(\tau_{m+u} - \tau_0 + 1)E(OPT, [\tau_0, \tau_{m+u}])$.*

Proof: Comparing with the analysis on the total transmitted data in Lemma 4, we should be more careful to derive the performance bound on the energy consumption, since the lower bound of the optimal solution should be carefully chosen. Let $s_1, s_2, ..., s_m$ respectively be the increasing rates in s^{opt} in interval $[\tau_0, \tau_m]$. Let $s_{m+1}, s_{m+2}, ..., s_{m+u}$ respectively be the decreasing rates in s^{opt} in interval $[\tau_m, t_{max}]$.

We examine interval $[\tau_0, \tau_m]$ and bound the energy consumption $E(ALG, [\tau_0, \tau_m])$ in that interval first.

We first focus on sub-interval $[\tau_{k-1}, \tau_k - 1]$ with $1 \le k \le m$ where the optimal solution has rate s_k. To bound the performance in interval $[\tau_{k-1}, \tau_k - 1]$, we assume that the algorithm changes the rate at time $t_{v(p)}$ to speed $s(t_{v(p)})$ to execute $J_{v(p)}$, where $0 \le p \le s, \tau_{k-1} \le t_{v(p)} \le \tau_k, t_{v(0)} = \tau_{k-1}, t_{v(s)} = \tau_k$ and $t_{v(0)} \le t_{v(1)} \le ... \le t_{v(s)}$. Similar to the proof in Lemma 4, we have $s(t_{v(p)}) \le \frac{w_{v(p)}}{d_{v(p)} - r_{v(p)} + 1}$.

The total amount of energy consumed in interval $[\tau_{k-1}, \tau_k - 1]$ is $E(ALG, [\tau_{k-1}, \tau_k - 1]) = \sum_{0 \le p \le s-1} (e^{s(t_{v(p)})} - 1)(t_{v(p+1)} - t_{v(p)}) \le \sum_{0 \le p \le s-1} (e^{\frac{w_{v(p)}}{d_{v(p)} - r_{v(p)} + 1}} - 1)(t_{v(p+1)} - t_{v(p)})$. Note that $\frac{w_{v(p)}}{d_{v(p)} - r_{v(p)} + 1}$ is at most $\frac{W(OPT, [r_{v(p)}, d_{v(p)}])}{d_{v(p)} - r_{v(p)} + 1}$ since the optimal solution must complete at least a workload $w_{v(p)}$ in that interval to satisfy the feasibility. We will discuss by considering the following two kinds of tasks, tasks with $d_{v(p)} \le \tau_m$ and tasks with $d_{v(p)} > \tau_m$.

Consider first the tasks with $d_{v(p)} \le \tau_m$. By the fact that s^{opt} is non-decreasing in $[r_{v(p)}, \tau_m]$, we have $\frac{W(OPT, [r_{v(p)}, d_{v(p)}])}{d_{v(p)} - r_{v(p)} + 1} \le \frac{W(OPT, [r_{v(p)}, \tau_m])}{\tau_m - r_{v(p)} + 1}$. Let X_k be the amount of data in the area that is covered by s^{opt} but above rate s_k in the optimal solution by time τ_m. We have $s^{opt}(t_{v(p)}) = s_k$ and $\frac{W(OPT, [r_{v(p)}, \tau_m])}{\tau_m - r_{v(p)} + 1} = s_k + \frac{X_k}{\tau_m - r_{v(p)} + 1}$. As a result,

$$
\begin{aligned}
&(e^{s(t_{v(p)})} - 1)(t_{v(p+1)} - t_{v(p)}) \\
&\le G(\tfrac{w_{v(p)}}{d_{v(p)} - r_{v(p)} + 1})(t_{v(p+1)} - t_{v(p)}) \\
&\le G(s_k + \tfrac{X_k}{\tau_m - r_{v(p)} + 1})(t_{v(p+1)} - t_{v(p)}) \\
&\le \textstyle\sum_{t_{v(p)} \le t \le t_{v(p+1)} - 1} (e^{s_k + \frac{X_k}{\tau_m - t + 1}} - 1) \\
&\le \textstyle\sum_{t_{v(p)} \le t \le t_{t(p+1)} - 1} \frac{\tau_m - \tau_k + 1}{\tau_m - t + 1} (e^{s_k + \frac{X_k}{\tau_m - \tau_k + 1}} - 1) \\
&\le (\ln(\tau_m - t_{v(p)} + 1) - \ln(\tau_m - t_{v(p+1)} + 1))(\tau_m - \tau_k + 1)(e^{s_k + \frac{X_k}{\tau_m - \tau_k + 1}} - 1) \\
&\le (\ln(\tau_m - t_{v(p)} + 1) - \ln(\tau_m - t_{v(p+1)} + 1))E(OPT, [\tau_k, \tau_m]) \\
&\le (\ln(\tau_m - t_{v(p)} + 1) - \ln(\tau_m - t_{v(p+1)} + 1))E(OPT).
\end{aligned}
$$

The third inequality holds since $\frac{W(OPT, [r_{v(p)}, \tau_m])}{\tau_m - r_{v(p)} + 1} \le \frac{(W(OPT, [r_{v(p)}, \tau_m])}{\tau_m - t + 1}$ with $r_{v(p)} \le t_{v(p)} \le t$. The forth one is because $e^{a + \frac{b}{c}} - 1 \le c' \cdot e^{a + \frac{b}{c \cdot c'}} - 1$ where $c' = \frac{\tau_m - \tau_k + 1}{\tau_m - t + 1} \le 1$ when $t \le \tau_k$ and we treat s_k to be a constant a. The third last one is correct since $\int_{t_{v(p)}}^{t_{v(p+1)} - 1} \frac{1}{\tau_m - t + 1} dt \le \ln(\tau_m - t_{v(p)} + 1) - \ln(\tau_m - t_{v(p+1)} + 1)$. The second last one is because the optimal solution completes at least an amount $(s_k + \frac{X_k}{\tau_m - \tau_k + 1})(\tau_m - \tau_k + 1)$ of workload in interval $[\tau_k, \tau_m]$ and hence $E(OPT, [\tau_k, \tau_m]) \ge (\tau_m - \tau_k + 1)(e^{s_k + \frac{X_k}{\tau_m - \tau_k + 1}} - 1)$ by the convexity of $G(\cdot)$.

Now consider more complex case that tasks have $d_{v(p)} > \tau_m$. Still, we have $\frac{w_{v(p)}}{d_{v(p)} - r_{v(p)} + 1} \le \frac{W(OPT, [r_{v(p)}, d_{v(p)}])}{d_{v(p)} - r_{v(p)} + 1}$. Let X_k be the amount of data in the area that is covered by s^{opt} but above rate s_k in the optimal solution by time τ_m. If $\frac{W(OPT, [r_{v(p)}, d_{v(p)}])}{d_{v(p)} - r_{v(p)} + 1} \le \frac{W(OPT, [r_{v(p)}, \tau_m])}{\tau_m - r_{v(p)} + 1} = s_k + \frac{X_k}{\tau_m - r_{v(p)} + 1}$, we have the same reduction as above. If $\frac{W(OPT, [r_{v(p)}, d_{v(p)}])}{d_{v(p)} - r_{v(p)} + 1} > s_k + \frac{X_k}{\tau_m - r_{v(p)} + 1}$, it is easy to see that $\frac{W(OPT, [\tau_m + 1, d_{v(p)}])}{d_{v(p)} - \tau_m} > s_k$. The amount $W(OPT, [\tau_m + 1, d_{v(p)}])$ of data transmitted in interval $[\tau_m + 1, d_{v(p)}]$ is at least $s_k \cdot (d_{v(p)} - \tau_m)$ (where $W(OPT, [\tau_m + 1, d_{v(p)}]) - s_k \cdot (d_{v(p)} - \tau_m)$ is always a positive value) and thus $\frac{W(OPT, [r_{v(p)}, d_{v(p)}])}{d_{v(p)} - r_{v(p)} + 1} = s_k + \frac{X_k'}{d_{v(p)} - r_{v(p)} + 1}$ by writing $X_k' = W(OPT, [\tau_k, d_{v(p)}]) - s_k \cdot (d_{v(p)} - \tau_k + 1)$. Moreover, we obtain a lower bound on the energy consumption of the optimal solution that $E(OPT) \ge (d_{v(p)} - \tau_k + 1)G(s_k + \frac{W(OPT, [\tau_k, d_{v(p)}]) - s_k \cdot (d_{v(p)} - \tau_k + 1)}{d_{v(p)} - \tau_k + 1}) = (d_{v(p)} - \tau_k + 1)G(s_k + \frac{X_k'}{d_{v(p)} - \tau_k + 1})$ by the convexity of $G(\cdot)$. Thus,

$$
\begin{aligned}
&(e^{s(t_{v(p)})} - 1)(t_{v(p+1)} - t_{v(p)}) \\
&\le G(\tfrac{W(OPT, [r_{v(p)}, d_{v(p)}])}{d_{v(p)} - r_{v(p)} + 1})(t_{v(p+1)} - t_{v(p)}) \\
&= G(s_k + \tfrac{X_k'}{d_{v(p)} - r_{v(p)} + 1})(t_{v(p+1)} - t_{v(p)}) \\
&\le \textstyle\sum_{t_{v(p)} \le t \le t_{v(p+1)} - 1} (e^{s_k + \frac{X_k'}{d_{v(p)} - t + 1}} - 1) \\
&\le \textstyle\sum_{t_{v(p)} \le t \le t_{t(p+1)} - 1} \frac{d_{v(p)} - \tau_k + 1}{d_{v(p)} - t + 1} (e^{s_k + \frac{X_k'}{d_{v(p)} - \tau_k + 1}} - 1) \\
&\le (\ln(d_{v(p)} - t_{v(p)} + 1) - \ln(d_{v(p)} - t_{v(p+1)} + 1))(d_{v(p)} - \tau_k + 1)(e^{s_k + \frac{X_k'}{d_{v(p)} - \tau_k + 1}} - 1) \\
&\le (\ln(d_{v(p)} - t_{v(p)} + 1) - \ln(d_{v(p)} - t_{v(p+1)} + 1))E(OPT, [\tau_k, d_{v(p)}]) \\
&\le (\ln(d_{v(p)} - t_{v(p)} + 1) - \ln(d_{v(p)} - t_{v(p+1)} + 1))E(OPT) \\
&\le (\ln(\tau_m - t_{v(p)} + 1) - \ln(\tau_m - t_{v(p+1)} + 1))E(OPT).
\end{aligned}
$$

The forth inequality is because $e^{a + \frac{b}{c}} - 1 \le c' \cdot e^{a + \frac{b}{c \cdot c'}} - 1$ where $c' = \frac{d_{v(p)} - \tau_k + 1}{d_{v(p)} - t + 1} \le 1$ when $t \le \tau_k$ and we treat s_k to be a constant a. The third last inequality holds by applying the lower bound for the optimal solution stated above, i.e., $E(OPT, [\tau_k, d_{v(p)}]) \ge (d_{v(p)} - \tau_k + 1)(e^{s_k + \frac{X_k'}{d_{v(p)} - \tau_k + 1}} - 1)$. The last inequality holds because $\ln(d_{v(p)} - t_{v(p)} + 1) - \ln(d_{v(p)} - t_{v(p+1)} + 1) \le \ln(\tau_m - t_{v(p)} + 1) - \ln(\tau_m - t_{v(p+1)} + 1)$ when $d_{v(p)} > \tau_m$.

Therefore, for both cases, we have obtained that $s(t_{v(p)})(t_{v(p+1)} - t_{v(p)}) \le (\ln(\tau_m - t_{v(p)} + 1) - \ln(\tau_m -$

$t_{v(p+1)}+1))E(OPT)$. Summing up all the amount of energy consumed in interval $[\tau_{k-1}, \tau_k - 1]$, we have

$E(ALG, [\tau_{k-1}, \tau_k - 1]) = \sum_{0\le p\le s-1} s(t_{v(p)})(r_{v(p+1)} - r_{v(p)})$
$\le \sum_{0\le p\le s-1}((\ln(\tau_m - t_{v(p)}+1) - \ln(\tau_m - t_{v(p+1)}+1))E(OPT))$
$\le (\ln(\tau_m - \tau_{k-1}+1) - \ln(\tau_m - \tau_k + 1))E(OPT)$.

Finally, summing up all value k, we obtain that
$E(ALG, [\tau_0, \tau_m]) = \sum_{1\le k\le m} E(ALG, [\tau_{k-1}, \tau_k - 1])$
$\le \sum_{1\le k\le m}(\ln(\tau_m - \tau_k + 1) - \ln(\tau_m - \tau_{k-1}+1))E(OPT)$
$\le \ln(\tau_m - \tau_0 + 1)E(OPT)$.

For the remaining interval $[\tau_m, \tau_{m+u}]$, we can symmetrically derive that $E(ALG, [\tau_{m-1}, \tau_{m+u}]) \le \ln(\tau_{m+u} - \tau_{m-1} + 1)E(OPT)$. Combining the results, the energy consumption is at most $E(ALG) \le E(ALG, [\tau_0, \tau_m]) + E(ALG, [\tau_{m-1}, \tau_{m+u}]) \le 2(\ln(\tau_{m+u} - \tau_0 + 1)E(OPT[\tau_0, \tau_{m+u}]))$. This completes the proof. ■

D. Merging the results: rate schedule for FIFO tasks

Now we are ready to combine the results above to develop an online algorithm for FIFO tasks.

The algorithm works as follows. It decomposes the tasks $\mathcal{J}$ into odd group S_{odd} and even group S_{even} in an online manner using Algorithm FIFO-decomposition. Once a task decomposed arrives, it runs MAX-REMAIN-ONLINE, respectively for the tasks in the odd group and even group. Let $s^{odd}(t)$ and $s^{even}(t)$, respectively, be the rate functions returned by Algorithm MAX-REMAIN-ONLINE running on tasks in S_{odd} and S_{even}. Then, the algorithm determines its final transmission rate to be $s(t) = \max\{s^{odd}(t), s^{even}(t)\}$. Algorithm FIFO-SCHEDULE presents the design of the algorithm.

Algorithm 3 FIFO-SCHEDULE

1: run online Algorithm FIFO-decomposition to decompose the tasks into the odd group S_{odd} and even group S_{even}.
2: run online Algorithm MAX-REMAIN-ONLINE over tasks in S_{odd} and determine the rate function $s^{odd}(t)$ at time t.
3: run online Algorithm MAX-REMAIN-ONLINE over tasks in S_{even} to derive the rate function $s^{even}(t)$ at time t.
4: determine the transmission rate at time t to be $s(t) = \max\{s^{odd}(t), s^{even}(t)\}$.

We derive the performance bound of the proposed algorithm. The following theorem proves that the algorithm is $O(\ln L)$-competitive with respect to both data traffic and energy consumption. Its proof crucially relies on the disjoint/bi-monotonicity properties of the decomposition method and the competitiveness of MAX-REMAIN-ONLINE running on partitioned tasks.

Theorem 3. *Algorithm* FIFO-SCHEDULE *is $O(\ln L)$-data $O(\ln L)$-energy competitive for general tasks.*

Proof:
Denote by $W(A, S)$ and $E(A, S)$ respectively the total amount of data transmitted and the overall energy consumption in algorithm A with the input of set S. Assume that ALG stands for algorithm MAX-REMAIN-ONLINE. We have $W(ALG, S_i) \le 2\ln 2L \cdot W(OPT, I(S_i))$ where the last inequality holds by applying the bounds derived for pairwise intersecting tasks in Lemma 4 and the fact that $\tau_{m+u} - \tau_0 + 1 \le 2L$ since the tasks are pairwise intersecting. Similarly $E(ALG, S_i) \le 2\ln 2L \cdot E(OPT, I(S_i))$ by Lemma 5.

Recall that any two sets in the same odd/even group are disjoint, i.e., $I(S_i) \cap I(S_{i+2}) = \emptyset$, and each set is composed of pairwise intersecting tasks. Thus, for tasks in the odd group, it can be seen that $W(ALG, S_{odd}) = \sum_i W(ALG, S_{2i+1}) \le 2\ln 2L \sum_i W(OPT, I(S_{2i+1})) \le 2\ln 2L \cdot W(OPT, [1, T])$ and $W(ALG, S_{even}) = 2\ln 2L \sum_i W(OPT, I(S_{2i})) \le 2\ln 2L \cdot W(OPT, [1, T])$ by the disjoint property. Similarly, $E(ALG, S_{odd}) = \sum_i E(ALG, S_{2i+1}) \le 2\ln 2L \sum_i E(OPT, I(S_{2i+1})) \le 2\ln 2L \cdot E(OPT, [1, T])$ and $E(ALG, S_{even}) = 2\ln 2L \cdot \sum_i E(OPT, I(S_{2i})) \le 2\ln 2L \cdot E(OPT, [1, T])$.

Finally, we combine the results obtained respectively for the odd group and the even group to derive the feasibility and competitiveness of FIFO-SCHEDULE. Note that Algorithm FIFO-SCHEDULE calls MAX-REMAIN-ONLINE twice, respectively for tasks in the odd and even group. In algorithm FIFO-SCHEDULE, setting the rate at time t to be $s(t) = \max\{s^{odd}(t), s^{even}(t)\}$ would be enough to finish the required data of all tasks $\mathcal{J} = S_{odd} \cup S_{even}$ under the delay constraints. This verifies the feasibility of the algorithm. Furthermore, the total amount of data transmitted is $\sum_t s(t) = \sum_t \max\{s^{odd}(t), s^{even}(t)\} \le W(ALG, S_{odd}) + W(ALG, S_{even}) \le 4\ln 2L \cdot W(OPT, [1, T])$. The overall energy consumption is $\sum_t e^{s(t)} - 1 = \sum_t e^{\max\{s^{odd}(t), s^{even}(t)\}} - 1 \le \sum_t (e^{s^{odd}(t)} - 1 + e^{s^{even}(t)} - 1) = E(ALG, S_{odd}) + E(ALG, S_{even}) \le 4\ln 2L \cdot E(OPT, [1, T])$. Therefore, the proposed algorithm is $O(\ln L)$-data $O(\ln L)$-energy competitive. ■

VI. ONLINE RATE SCHEDULE FOR GENERAL TASKS WITH ARBITRARY DEADLINES

In this section, we further consider the generalized AD task model and develop online rate schedule with bounded competitive ratios for tasks with arbitrary deadlines.

In the previous section, the FIFO-decomposition efficiently partitions the tasks into just two (odd/even) groups so that each partitioned set is with good structural properties. For general tasks with arbitrary deadlines, however, such a decomposition method fails to partition the tasks into limited number of well-structured groups, due to irregular intersection/sharing of the tasks. This makes it challenging to design a rate schedule with proven performance bound.

To address this challenge, in this section, we will propose a novel online decomposition method to tackle the tasks with arbitrary deadlines, and design a rate schedule with $O(\ln^2 L)$ competitiveness with respect to both data traffic and energy consumption.

A. AD-decomposition for Tasks with arbitrary deadlines

To tackle tasks with arbitrary deadlines, we propose a new decomposition method, called *AD-decomposition*, which works as follows. On the arrival of a task J_i, we find the value r such that $2^r \le d_i - r_i + 1 < 2^{r+1}$ (we say that J_i belongs

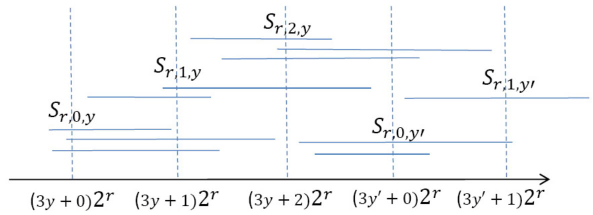


Fig. 5. Online AD-decomposition where tasks in S_r is partitioned into groups $\{S_{r,p}, 0 \leq p \leq 2\}$.

to set S_r). For such a task, we find the earliest time t with the form $t = (3y + p) \cdot 2^r$ such that $t \in [r_i, d_i]$, thus place J_i into set $S_{r,p,y}$. That is, task J_i in set S_r is placed into set $S_{r,y,p}$ if it goes across the earliest time t with the form $t = (3y+p) \cdot 2^r$. It is easy to verify that such a decomposition method can be performed in an online manner. Then, all sets with the same value of r and p are regrouped to be a group $S_{r,p} = \cup_{y \geq 0} S_{r,p,y}$. In such a way, all the tasks in $\mathcal{J}$ are firstly partitioned into $\lceil \log L \rceil$ sets $\{S_r, 0 \leq r \leq \lceil \log L \rceil - 1\}$, and then each set S_r is further partitioned into 3 groups $\{S_{r,0}, S_{r,1}, S_{r,2}\}$, resulting in $3\lceil \log L \rceil$ groups in total. Fig. 5 demonstrates an example on how to partition the set S_r into 3 groups $\{S_{r,p}, 0 \leq p \leq 2\}$ in an online manner. Algorithm 5 implements the online decomposition procedure.

Algorithm 4 *AD-Decomposition*

```
1: t = 0. S = J.
2: for on the arrival of each time t do
3:    for on the arrival of each task J_i ∈ S with t = r_i do
4:       find value r such that 2^r ≤ d_i − r_i + 1 < 2^{r+1}.
5:       if t is with the form t = (3y+p)2^r where y ≥ 0, 0 ≤ p ≤ 2 then
6:          set S_{r,p,y} = S_{r,p,y} ∪ J_i.
7:          S = S\J_i.
8:       end if
9:       t = t + 1.
10:   end for
11: end for
```

Now we examine what properties are implied by the AD-decomposition proposed above.

- (**Disjoint Property**) for every two sets in the same group $S_{r,p}$, say $S_{r,p,y}, S_{r,p,y'} \in S_{r,p}$, their intervals are disjoint, i.e., $I(S_{r,p,y}) \cap I(S_{r,p,y'}) = \emptyset$ for any $y \neq y'$.
- (**Pairwise Intersecting Property**) all tasks in the same set, say $S_{r,p,y}$, are alive at the same time, $(3y + p) \cdot 2^r$.
- (**Logarithmic Groups**) there are at most $3\lceil \log L \rceil$ groups, i.e., $\{S_{r,p}, 0 \leq r \leq \lceil \log L \rceil - 1, 0 \leq p \leq 2\}$.
- (**Bi-monotonicity**) for any set of pairwise intersecting tasks, the optimal solution follows the bi-monotonicity as stated in Lemma 3.

The last three properties can be verified easily by observing the decomposition process. The first property can be shown as follows. For any task in set S_r, say J_i, the length of its interval $d_i - r_i + 1$ is less than $2 \cdot 2^r$. If J_i goes across the earliest time with form $t = (3y + p)2^r$, which implies $(3y + p - 1)2^r < r_i \leq (3y + p)2^r < d_i$, then it will be placed into group $S_{r,p}$. Suppose that two tasks in the same group $S_{r,p}$, say J_u, J_v, respectively belong to two different sets $S_{r,p,y}$ and $S_{r,p,y+1}$. This implies that $d_u \leq r_u + 2 \cdot 2^r \leq (3y + p + 2)2^r$ and $r_v > (3(y + 1) + p - 1)2^r > (3y + p + 2)2^r \geq d_u$, thus the intervals of J_u and J_v are disjoint. This indicates that $I(S_{r,p,y}) \cap I(S_{r,p,y'}) = \emptyset$ for any $y \neq y'$ and hence verifies the disjoint property.

B. Merging the results: rate schedule for AD tasks

Now we design an online algorithm that achieves $O(\ln^2 L)$ competitiveness with respect to both data traffic and energy consumption.

Recall the disjoint property that any two sets in the same group $S_{r,p}$ are disjoint, i.e., $S_{r,p,y} \cap I(S_{r,p,y'}) = \emptyset$ for any $y \neq y'$. Moreover, the tasks in the same set $S_{r,p,y}$ are pairwise intersecting. Thus, when running MAX-REMAIN-ONLINE on $S_{r,p}$, the total amount of data transmitted/energy consumed is bounded within $O(\ln L)$ times of the optimal solution.

Moreover, according to the logarithmic property, there are at most $3\lceil \log L \rceil$ groups in $\{S_{r,p}\}$, which constitutes the tasks $\mathcal{J}$. Let $s_{r,p}(t)$ be the rate function returned by MAX-REMAIN-ONLINE running on set $S_{r,p}$. We observe that the combined rate function, denoted as $s^{comb}(t) = \max_{r,p}\{s_{r,p}(t)\}$, is feasible to finish the workload of all tasks $\mathcal{J}$. As a fact, such a rate function can achieve $O(\ln^2 L)$-competitiveness with respect to the bi-objectives, which will be proved later.

Based on these observations, one alternative idea for deriving $O(\ln^2 L)$-competitive algorithm is to schedule with the rate $s^{comb}(t)$. In order to further improve its performance, we further refine such a strategy by directly running MAX-REMAIN-ONLINE over the input tasks $\mathcal{J}$ to get another rate $s^{ori}(t)$, and then return the better one of $\{s^{comb}(t), s^{ori}(t)\}$. The resulting algorithm will be denoted as AD-SCHEDULE.

Algorithm 5 AD-SCHEDULE

1: Run online Algorithm *AD-Decomposition* to decompose the tasks $\mathcal{J}$ into logarithmic groups.
2: Run online Algorithm MAX-REMAIN-ONLINE over tasks in each decomposed group to get the rate $s^{comb}(t)$.
3: Run online Algorithm MAX-REMAIN-ONLINE over the original input tasks $\mathcal{J}$ directly to get the rate $s^{ori}(t)$.
4: Return the minimum one between schedules $s^{comb}(t)$ and $s^{ori}(t)$

The following theorem proves that AD-SCHEDULE is $O(\ln^2 L)$-data $O(\ln^2 L)$-energy competitive.

Theorem 4. *Algorithm* AD-SCHEDULE *is* $O(\ln^2 L)$*-data* $O(\ln^2 L)$*-energy competitive for generalized AD tasks.*

Proof:
Recall the disjoint property/pairwise intersecting property of the AD-decomposition and the competitiveness of MAX-REMAIN-ONLINE. When running MAX-REMAIN-ONLINE with tasks in group $S_{r,p}$, the total amount of data transmitted is $W(ALG, S_{r,p}) = \sum_{y \geq 0} W(ALG, S_{r,p,y}) \leq 2 \ln 2L \cdot \sum_{y \geq 0} W(OPT, I(S_{r,p,y})) \leq 2 \ln 2L \cdot W(OPT, [1, T])$ where

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2016.2621359, IEEE Journal on Selected Areas in Communications

ALG stands for MAX-REMAIN-ONLINE and $2L$ comes from the fact that $\tau_{m+u} - \tau_0 + 1 \leq 2L$ when applying Lemma 4. Similarly, the total amount of energy consumed is $E(ALG, S_{r,p}) = \sum_{y\geq 0} E(ALG, S_{r,p,y}) \leq 2\ln 2L \cdot \sum_{y\geq 0} E(OPT, I(S_{r,p,y})) \leq 2\ln 2L \cdot E(OPT, [1,T])$.

Let $s_{r,p}(t)$ be the rate function returned by MAX-REMAIN-ONLINE running on set $S_{r,p}$. We examine the combined rate function $s^{comb}(t) = \max_{r,p}\{s_{r,p}(t)(t)\}$. It would be enough to finish the required data of all tasks $\mathcal{J}$ under the delay constraints since $\mathcal{J} = \cup_{r,p} S_{r,p}$. Moreover, the total amount of data transmitted is $\sum_t s^{comb}(t) = \sum_t \max_{r,p}\{s_{r,p}(t)\} \leq \max_{r,p}\{\sum_t s_{r,p}(t)\} \leq \sum_{r,p} W(ALG, S_{r,p}) \leq \sum_{r,p} 2\ln 2L \cdot W(OPT, [1,T]) \leq 6\lceil \log L\rceil \cdot \ln 2L \cdot W(OPT, [1,T])$, where the second last inequality holds by the competitiveness of MAX-REMAIN-ONLINE running on group $S_{r,p}$ and the last one holds by the logarithmic property of the AD-decomposition. Similarly, the overall energy consumption is $\sum_t e^{s^{comb}(t)} - 1 = \sum_t e^{\max_{r,p}\{s_{r,p}(t)\}} - 1 \leq \sum_t \sum_{r,p}(e^{s_{r,p}(t)} - 1) = \sum_{r,p} E(ALG, S_{r,p}) \leq \sum_{r,p} 2\ln 2L \cdot E(OPT, [1,T]) \leq 6\lceil \log L\rceil \cdot \ln 2L \cdot E(OPT, [1,T])$. Thus the combined rate function $s^{comb}(t)$ achieves $O(\ln^2 L)$-data $O(\ln^2 L)$-energy competitiveness.

Therefore, Algorithm AD-SCHEDULE is $O(\ln^2 L)$-data $O(\ln L)$-energy competitive for general tasks with arbitrary deadlines. ■

Discussion. We note that FIFO-SCHEDULE and AD-SCHEDULE respectively apply different partition strategies to deal with the FIFO task model and AD task model, resulting in different worst-case performance bounds, but they call the same sub-procedure MAX-REMAIN-ONLINE in tackling the partitioned pairwise intersecting tasks. FIFO-SCHEDULE solves the scheduling problem for transmitters following first-in-first-out rule in practice. AD-SCHEDULE extends the results and provides solutions to the more general task model. The latter one is with slightly worse but still well-bounded worst-case performance compared to the former one.

VII. SIMULATIONS

Our theoretical analysis has bounded the worst-case performance of the online algorithms with respect to both data traffic and energy consumption. In this section, we perform simulations to further validate their average performances.

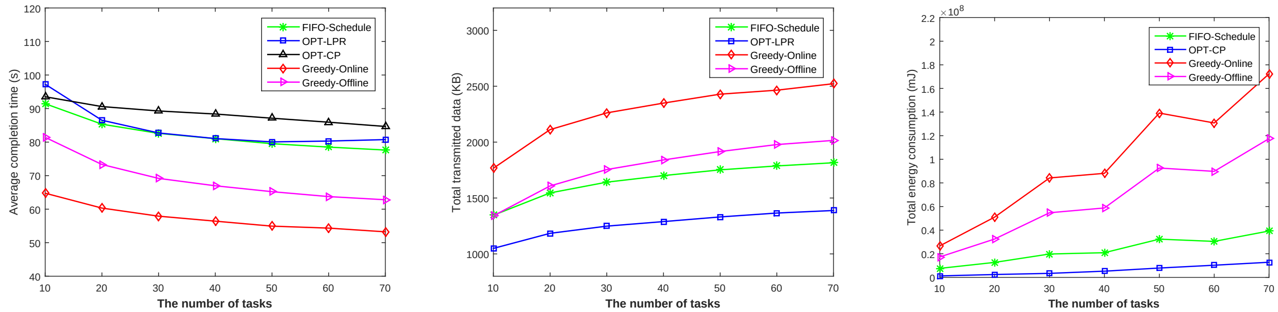
In the simulation, we will compare our rate-adaptive schedule with constant-rate schedules that also consider data sharing. We note that the constant-rate algorithms in [4], [24] also consider data sharing but work only in the offline setting. In this simulation, we will take two algorithms as the baselines to demonstrate the advantage of our rate-adaptive scheduling algorithms, comparing with non-rate-adaptive algorithms: one is a natural online greedy constant-rate algorithm, which transmits with the rate once there is any request waiting in the task queue; the other is the offline greedy constant-rate algorithm in [4]. To ensure the feasibility, the least constant-rate that guarantees the satisfaction of all tasks' time constraints should be $r_{max} = \max_i \frac{w_i}{d_i - r_i + 1}$. Hence, we set the constant rate to be r_{max} in our simulation and consequently each task J_i needs a length $\frac{w_i}{r_{max}}$ of transmission duration. Furthermore, to examine whether our algorithm is close to the best possible solution, we compare our algorithm with the optimal offline solution. In our setting, we not only compare these algorithms with respect to the bi-objectives, the data traffic and the energy consumption, but also with respect to the average delay, which is the average value of the difference between the completion time and the arrival time of the tasks.

We first perform simulations on algorithm FIFO-SCHEDULE for FIFO task model. The rate-power function is set to be $s = \frac{1}{2}\log(1+p)$ (with $\alpha = 2\ln 2$) in AWGN channel where p is in milliwatts and s is in kilobits per second (kbps). Task arrival time is assumed to be a random integer that follows uniform distribution $U(1, 300)$. Task J_i arrives at r_i and the schedule needs to adjust the rate $s(t)$ in every second. Arrival time points are sorted so that $r_1 \leq r_2 \leq \ldots \leq r_n$. For each arrival time r_i, the deadline is generated by randomly selecting an integer in $[r_i, 300], i = 1, 2, \ldots, n$. The amount of requested data of each task is assumed to be a random variable following uniform distribution $U(0kb, 900kb)$. Then all deadlines are sorted so that an earlier arrived task carries an earlier deadline, $d_1 \leq d_2 \leq \ldots \leq d_n$.

Fig. 6 shows the simulation results of algorithm FIFO-SCHEDULE. Each point in these figures is a mean value of 2500 random instances. Although we did not address the delay minimization problem in this paper, we first compare the delay incurred in the five schedules in Figure 6(a), the min-traffic linear programming LPR, min-energy convex programing CP, online greedy constant-rate algorithm Greedy-Online, offline greedy constant-rate algorithm Greedy-Offline, and our online algorithm FIFO-SCHEDULE. As it is shown in the figure, the average delay decreases as the number of tasks increases, which is possibly because of the sharing nature of the tasks; With respect to delay, the online constant-rate greedy algorithm performs good, which is natural since it is actually delay-oriented, and the delay of our algorithm is close to the online greedy algorithm. Then, we examine their performance on the bi-objectives addressed in this paper. Fig. 6(c) shows the simulation results. Here, it is worth noticing that, with respect to minimizing the data traffic, the optimal solution OPT-LPR is computed by the linear programming LPR; while with respect to minimizing the energy consumption, the optimal solution OPT-CP is computed by the convex programming CP. As shown in Fig. 6(b), the data traffic of our online algorithm is less than that of the online/offline greedy constant-rate algorithms, and close to that of the optimal offline solution. Moreover, as demonstrated in the figure, the energy consumption of our online algorithm is much less than the online/offline constant-rate greedy algorithms, and close to that of the optimal solution. These together validate the efficiency of FIFO-SCHEDULE with respect to the bi-objectives.

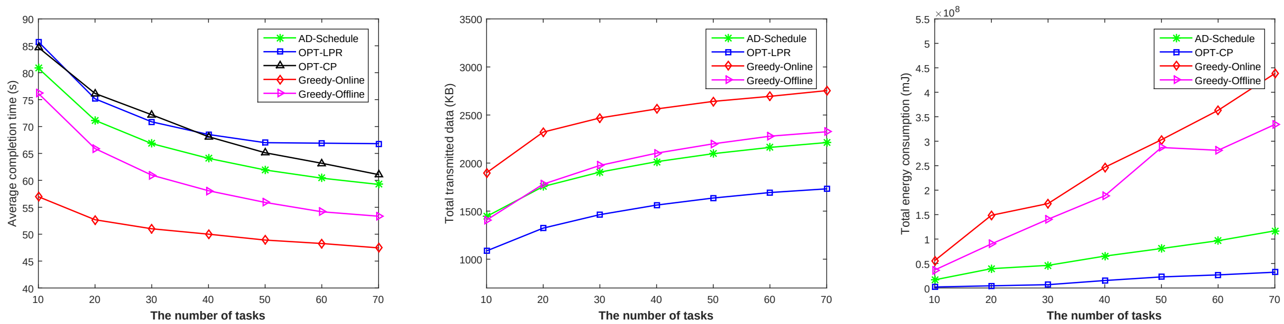
Next, we evaluate our algorithm AD-SCHEDULE for AD task model. The setting is the same as above, except that tasks are allowed to have arbitrary deadlines. Fig. 7 presents the simulation results. We can see from the figure that the trends of different schedules remain unchanged, and both data traffic and the energy consumption are quite close to the optimal solution, which is possibly because AD-SCHEDULE further compares the decomposition-based schedule $s^{comb}(t)$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2016.2621359, IEEE Journal on Selected Areas in Communications



(a) (b) (c)

Fig. 6. Performance of FIFO-SCHEDULE in FIFO task model: (a) average delay; (b) data traffic; (c) energy consumption.



(a) (b) (c)

Fig. 7. Performance of AD-SCHEDULE in AD task model: (a) average delay; (b) data traffic; (c) energy consumption.

with another promising schedule $s^{ori}(t)$ and returns the better one between them to improve the performance.

Considering that our online algorithms do not rely on any distribution or future information, the above simulation results validate the efficiency of the algorithms.

VIII. CONCLUSION

This paper introduces the energy-efficient transmission problem with data sharing and conducts the first theoretical analysis on the trade-off between the data traffic and the energy consumption. We provide optimal algorithms for the offline setting and online algorithms with proven competitive ratios with respect to the bi-objectives of minimizing the data traffic and the energy consumption. By respectively proposing online decomposition methods, called FIFO-decomposition and AD-decomposition, we devise online algorithms that achieve within $O(\ln L)$ times and $O(\ln^2 L)$ times of the optimal solution for FIFO task model and AD task model, with respect to both data traffic and energy consumption. Simulation results further show that the average performances of the online algorithms are close to the optimal solution, thus validate the efficiency.

APPENDIX

A. Proofs of the optimal algorithm for common deadline tasks

Proof of Lemma 1: The first property can be proved by applying the equalization to any two adjacent time slots that belongs to the same epoch, which would not violate the time constraints.

Consider the last two properties. With respect to a specific task J_i, the rate in interval $[r_i, d_i]$ that fulfills the requirement w_i of data may change over time. Consider the case that the optimal solution increases the rate first, say, at time t. Suppose on the contrary that t is not an arrival time. Assume that J_i is the task whose workload is transmitted at time t. Then applying the equalization method to decrease the rate at time t and increase the rate at $t-1$ could reduce the power consumption. This contradicts the optimality of the optimal solution and hence t must be an arrival time. Similar proof can verify that t is a deadline point when the optimal solution decreases at time t.

Proof of Theorem 1: Algorithm INTERVAL-DELETE finds the interval with the largest rate in s^{opt} in the initiation step. The schedule in that interval is fixed by transmitting with the largest average rate $w(J_m)$ among all tasks. Then, it iteratively finds all the intervals in s^{opt}. Write $W(A)$ and $E(A)$ to be the total amount of transmitted data and energy consumption caused by a schedule A.

We first prove its optimality for minimize the energy consumption. By the proof in Lemma 2, the schedule in $[r_m, T]$ achieves the minimum energy consumption. In the second iteration, each task with $r_i < r_m$ has $\max\{\frac{w_i - |I_i \cap I_m| \cdot w(J_m)}{|(I_i \cup I_m) \setminus (I_i \cap I_m)|}, 0\}$ workload to be finished in $[r_i, r_m - 1]$. Thus it can be verified that transmitting with rate $\max\{\max_{i: r_i < r_m} \frac{w_i - |I_i \cap I_m| \cdot w(J_m)}{r_m - r_i}, 0\}$ is optimal for the tasks with updated workload by similarly applying the proof in Lemma 2. Iteratively, in the interval found in each iteration, the algorithm transmits the data with the minimum energy.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2016.2621359, IEEE Journal on Selected Areas in Communications

This verifies that Algorithm INTERVAL-DELETE computes the optimal rate schedule that achieves the energy consumption $E(OPT)$ of the optimal schedule OPT.

We now show that the schedule returned by the algorithm transmits with the minimum amount of data $W(OPT)$. First, in the first interval found by the algorithm, the amount of data transmitted is no more than the requirement of J_m, which is a lower bound for any optimal solution to finish J_m. In the second iteration, assume that the algorithm transmits the remaining workload of J_k in interval $[r_k, r_m - 1]$. Since the remaining workload of J_k is $w_k - |I_k \cap I_m| \cdot w(J_m)$, the total amount of data transmitted in interval $[r_k, T]$ is no more than that of the required workload w_k, which satisfies $w_k \geq w_m$. This is also a lower bound for any optimal solution, i.e. $W(OPT) \geq w_k$. Iteratively, we can see that the total amount of data transmitted by the algorithm in each iteration matches the lower bound of any optimal solution.

Therefore, our proposed algorithm simultaneously achieves the minimum energy consumption $E(OPT)$ and the minimum transmitted data $W(OPT)$.

REFERENCES

[1] K. Ali, D. Al-Yaseen, A. Ejaz, T. Javed, and H. S.Hassanein, Crowdits: crowdsourcing in intelligent transportation systems, in IEEE WCNC, 2012.

[2] R. K. Rana, C. T. Chou, S. S. Kanhere, N. Bulusu, and W. Hu, Earphone: an end-to-end participatory urban noise mapping system, in Proc. ACM IPSN, 2010.

[3] A. Tavakoli, A.Kansal, and S.Nath. On-line sensing task optimization for shared sensors. *Proc. of IEEE IPSN*, 2010.

[4] X. Fang, H. Gao, J. Li, Y. Li, Application-aware data collection in Wireless Sensor Networks, *Proc. of IEEE INFOCOM*, 2013.

[5] R. Berry, R. Gallager, Communication over fading channels with delay constraints, *IEEE Tran. on Information Theory*, vol. 48, no. 5, 2002.

[6] B. Prabhakar, E. Biyikoglu, A. El Gamal, Energy-efficient transmission over a wireless link via lazy packet scheduling, *Proc. of IEEE INFOCOM*, 2001.

[7] Nanda S, Balachandran K, Kumar S. Adaptation techniques in wireless packet data services[J]. *Communications Magazine*, IEEE, 2000, 38(1): 54-64.

[8] W. Chen, M. J. Neely, and U. Mitra, Energy-Efficient Transmissions With Individual Packet Delay Constraints, *IEEE Trans. Information Theory*, vol. 54, no. 5, pp. 2090-2109, 2008.

[9] M. Zafer and E. Modiano. A Calculus Approach to Energy-Efficient Data Transmission With Quality-of-Service Constraints, *IEEE Trans. on Netw.*, vol. 17, no. 3, pp. 898-911, 2009.

[10] A. Borodin and R. El-Yaniv, Online computation and competitive analysis. *Cambridge University Press*, 1998.

[11] W. Wu, J. Wang, M. Li, K. Liu, J. Luo, Energy-efficient Transmission with Data Sharing, *Proc. of IEEE INFOCOM*, 2015.

[12] B. Guo, Z. Wang, Z. Yu, Y. Wang, N. Y. Yen, R. Huang, X. Zhou, Mobile crowd sensing and computing: The review of an emerging human-powered sensing paradigm. *ACM Computing Surveys (CSUR)*, 48(1), 7, 2015.

[13] B.Guo, D. Zhang, Z. Wang, Z. Yu, X. Zhou. Opportunistic IoT: Exploring the harmonious interaction between human and the internet of things. *Journal of Network and Computer Applications*, 36(6), 1531-1539, 2013.

[14] Z. Xu, Y. Liu, N. Yen, L. Mei, X. Luo, X. Wei, C. Hu. Crowdsourcing based description of urban emergency events using social media big data, *IEEE Transactions on Cloud Computing*, 2016.

[15] Z. Xu, H. Zhang, V. Sugumaran, K. K. R. Choo, L. Mei, Y. Zhu. Participatory sensing-based semantic and spatial analysis of urban emergency events using mobile social media. *EURASIP Journal on Wireless Communications and Networking*, 2016(1), 1-9, 2016.

[16] Z. Xu, H. Zhang, C. Hu, L. Mei, J. Xuan, K. K. R.Choo, Y. Zhu. Building knowledge base of urban emergency events based on crowdsourcing of social media. *Concurrency and Computation: Practice and Experience*, 2016.

[17] Z. Feng, Y. Zhu, Q. Zhang, L. M. Ni, and A. V. Vasilakos, Trac:Truthful auction for location-aware collaborative sensing in mobile crowdsourcing, *IEEE Conference on Computer Communica-tions (INFOCOM14)*, pp. 12311239, 2014.

[18] K. Xing, Z. Wan, P. Hu, H. Zhu, Y. Wang, X. Chen, Y. Wang, and L. Huang, Mutual privacy-preserving regression modeling in participatory sensing, in Proc. IEEE INFOCOM, 2013.

[19] K. Vu, R. Zheng, and J. Gao, Efcient algorithms for k-anonymous location privacy in participatory sensing, in Proc. IEEE INFOCOM, 2012.

[20] T. Luo and C.-K. Tham, Fairness and social welfare in incentivizing participatory sensing, in IEEE SECON, 2012.

[21] Y. Zhang and M. v. d. Schaar, Reputation-based incentive protocols in crowdsourcing applications, in Proc. IEEE INFOCOM, 2012.

[22] C.-J. Ho and J. W. Vaughan, Online task assignment in crowdsourcing markets, in AAAI, 2012.

[23] C.-J. Ho, S. Jabbari, and J. W. Vaughan, Adaptive task assignment for crowdsourced classication, in ICML, 2013.

[24] Q. Zhao, Y. Zhu, H. Zhu, J. Cao, G. Xue, B. Li, Fair energy-efficient sensing task allocation in participatory sensing with smartphones. *Proc. of INFOCOM*, 2014.

[25] Ulukus, S., Yener, A., Erkip, E., Simeone, O., Zorzi, M., Grover, P., Huang, K. (2015). Energy Harvesting Wireless Communications: A Review of Recent Advances. *IEEE Journal on Selected Areas in Communications*, 33(3), 360-381.

[26] B. Prabhakar, E. Uysal Biyikoglu, and A. El Gamal, Energy-Efficient Transmission Over a Wireless Link via Lazy packet Scheduling, *Proc. of IEEE INFOCOM*, 2001.

[27] F. Shan, J. Luo, W. Wu, M. Li, X. Shen. Discrete Rate Scheduling for Packets with Individual Deadlines in Energy Harvesting Systems. *Journal on Selected Areas in Communications(JSAC)*, vol. 33, no. 3: 438-451, 2015.

[28] M. Gatzianas, L. Georgiadis, and L. Tassiulas, Control of wireless networks with rechargeable batteries, *IEEE Trans. on Wireless Comm.*, vol. 9, no. 2, pp581-593, 2010.

[29] R. Vaze, Competitive ratio analysis of online algorithms to minimize packet transmission time in energy harvesting communication system, *Proc. of IEEE INFOCOM*, 2013.

[30] R. Vaze, R. Garg, and N. Pathak, Dynamic Power Allocation for Maximizing Throughput in Energy-Harvesting Communication System, *IEEE/ACM Trans. on Netw.*, vol.22, no. 5, pp1621-1630, 2014.

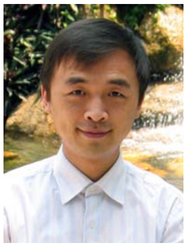
[31] W. Wu, J. Wang, X. Wang, F. Shan, J. Luo, Online Throughput Maximization for Energy Harvesting Communication Systems with Battery Overflow, *Transactions on Mobile Computing*, 2016.

[32] N. Buchbinder, L. Lewin-Eytan, I. Menache, S. Naor and A. Orda, Dynamic power allocation under arbitrary varying channels - an online approach, *Proc. of IEEE INFOCOM*, 2009.

Weiwei Wu is an associate professor in Southeast University, P.R. China. He received his BSc degree in South China University of Technology and the PhD 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. His research interests include optimizations and algorithm analysis, wireless communications, crowdsourcing, cloud computing, reinforcement learning, game theory and network economics.

Jianping Wang is an associate professor in the Department of Computer Science at City University of Hong Kong. She received the B.S. and the M.S. degrees in computer science from Nankai University, Tianjin, China in 1996 and 1999, respectively, and the Ph.D. degree in computer science from the University of Texas at Dallas in 2003. Jianping's research interests include dependable networking, optical networks, cloud computing, service oriented networking and data center networks.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2016.2621359, IEEE Journal on Selected Areas in Communications



Minming Li received the B.Eng., M.Eng., and Ph.D. degrees from Tsinghua University, Beijing, China. He is currently an Associate Professor with the Department of Computer Science, City University of Hong Kong, Hong Kong. His research interests include wireless networks, algorithms design and analysis, combinatorial optimization, scheduling and algorithmic game theory.

Kai Liu received his Ph.D. degree in computer science from City University of Hong Kong in 2011. He is currently an Assistant Professor with the College of Computer Science, Chongqing University, China. From 2010 to 2011, he was a Visiting Scholar with the Department of Computer Science, University of Virginia, Charlottesville, VA, USA. From 2011 to 2014, he was a Postdoctoral Fellow with Nanyang Technological University (Singapore), City University of Hong Kong, and Hong Kong Baptist University (Hong Kong). His research interests include mobile computing, pervasive computing, intelligent transportation systems and internet of vehicles.

Feng Shan received his Ph.D. degree in Computer Science from Southeast University, China in 2015. He is currently an Assistant Professor at School of Computer Science and Engineering, Southeast University. 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 received the BS degree in applied mathematics and the MS and PhD 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. 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 SIGCOMM China. His research interests are next generation network architecture, network security, cloud computing, and wireless LAN.