Chang CW, Zhang TF, Yang CY *et al.* Data transmission with the battery utilization maximization. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 26(3): 392–404 May 2011. DOI 10.1007/s11390-011-1142-7

Data Transmission with the Battery Utilization Maximization*

Che-Wei Chang[1] (张哲维), Tie-Fei Zhang[3] (章铁飞), Chuan-Yue Yang[1] (杨川岳), Ying-Jheng Chen[1] (陈应正)
Shih-Hao Hung[1,2] (洪士灏), *Member*, *IEEE*, Tei-Wei Kuo[1,2] (郭大维), *Fellow*, *IEEE*
and Tian-Zhou Chen[3] (陈天洲), *Member*, *CCF*, *ACM*, *IEEE*

[1] *Department of Computer Science and Information Engineering, "National Taiwan University", Taipei*

[2] *Graduate Institute of Networking and Multimedia, "National Taiwan University", Taipei*

[3] *College of Computer Science, Zhejiang University, Hangzhou 230027, China*

E-mail: {f95093, r92032, r97154, hungsh, ktw}@csie.ntu.edu.tw; {tfzhang, tzchen}@zju.edu.cn

Received January 25, 2011; revised March 20, 2011.

Abstract With the growing popularity of 3G-powered devices, there are growing demands on energy-efficient data transmission strategies for various embedded systems. Different from the past work in energy-efficient real-time task scheduling, we explore strategies to maximize the amount of data transmitted by a 3G module under a given battery capacity. In particular, we present algorithms under different workload configurations with and without timing constraint considerations. Experiments were then conducted to verify the validity of the strategies and develop insights in energy-efficient data transmission.

Keywords data transmission, battery utilization, 3G devices, bandwidth scheduling

1 Introduction

In the past decades, there is strong demand for mobile devices with networking and/or telecommunication capabilities, such as mobile Internet devices (MIDs). An example market growth is 300% plus for MIDs from 2008 to 2010. Moreover, it is believed that all smartphones will be equipped with mobile networking functionalities, such as 3G, in 2011, compared to 60% of them in 2009. The market trends reveal the popularity of various applications with mobile networking needs in the coming years. One of such popular applications is portable media player, that has its predicted market share 15.1% among related products in 2012[2].

One critical problem in mobile device designs is on the energy efficiency consideration because of the trade-off between the system performance and the battery capacity. In this direction, a number of excellent research results, e.g., [3-8], have been proposed by researchers based on dynamic power management (DPM) and/or dynamic voltage scaling (DVS) methodologies. Here DPM considers the switching among operating modes of subsystems/hardware to reduce the energy consumption, and DVS considers the switching of operating frequencies or supply voltages of processing elements to trade the performance with the energy consumption. In recent years, the energy consumption of peripheral devices and memory has received a lot of attention, as the energy consumption ratio of microprocessors, like the IBM P7 power processor, keeps dropping in computer systems. It is worth noticing that the networking function, such as 3G, of mobile devices now contributes a very significant portion of the energy consumption to the entire system, compared with those of the microprocessor and LCD screen. With such an observation, researchers explored energy-efficient resource management over Wi-Fi with various considerations, such as activities[9-10] and handoff between Wi-Fi and 3G[11]. There are also strong demands on energy-efficient designs of 3G resource usages because of the growing popularity of 3G-powered devices, such as smartphones, and their serious energy consumption problem. In particular, researchers have explored power management in communication-parameter control with respect to traffic patterns[12]. Issues on Quality of Services and multimedia workloads were also

Regular Paper

This work was supported by the Excellent Research Projects of "National Taiwan University" under Grant No. 99R80304. The author Tei-Fei Zhang was supported by the Mediatek.

*This paper is extended from a conference paper[1] appeared in ASP-DAC 2011.

explored[13-14]. The research on energy-efficient designs highly depends on the characteristics of batteries and their usages. In this respect, researchers have proposed excellent results on battery models (e.g., [15-16]), recover behaviors (e.g., [10]), fuel batteries (e.g., [17]), multi-battery usages (e.g., [18]), etc.

Different from past work on energy-efficient real-time task scheduling, e.g., [19-23], we are interested in the maximization of the amount of data transmitted by a 3G module under various conditions. The characteristics of a 3G module are considered with the best utilization of the battery capacity. The study starts with the maximization of the amount of data transmission and then with a joint consideration of a receiving task. Algorithms are proposed to derive optimal bandwidth schedules, that consist of a sequence of pairs of an operating mode and its corresponding time interval. We later extend the work with the consideration of multiple tasks with their amount of to-be-transmitted data and deadlines. The capabilities of the proposed algorithms are evaluated with a series of experiments to provide insights to the study.

The rest of the paper is organized as follows. Section 2 presents a battery model, a model of a 3G module, and the task models for data sending and receiving. We propose algorithms to derive optimal bandwidth schedules with and without a receiving task in Subsection 3.2. We later extend the work to consider tasks with deadlines in data transmissions in Subsection 3.3. Finally, Subsection 3.4 extends the previous strategies with the consideration of multiple tasks. The performance study of the proposed algorithms is presented in Section 4. Section 5 concludes this paper.

2 System Model

In the paper, we are interested in the methodology to maximize the usage of a battery for 3G-equipped mobile devices, such as a smartphone. Before we proceed with further discussion, we shall define the battery model and the task model in the transmission and receiving of data over the 3G module. The battery model considered in this paper is based on a Lithium-Ion battery model from [15], where the parameters of the model are summarized in Table 1.

$$V_{(t)} = V_0 - \gamma I_j - \phi \ln \frac{\alpha_n + I_j(t-t_j) + \sum_{k=1}^{j-1} I_k d_k}{\alpha_p - I_j(t-t_j) - \sum_{k=1}^{j-1} I_k d_k}. \quad (1)$$

The supply voltage of a battery for some specific usage pattern and the corresponding time interval is denoted as $V_{(t)}$. When the battery voltage drops to a certain level, referred to as the *cut-off voltage* and denoted as V_{cut_off}, battery-powered devices will stop functioning and be turned off. Thus, the *lifespan* of a battery for some specific usage pattern and the current voltage can be derived from (1), from the current voltage $V_{(t)}$ to V_{cut_off}. As indicated by the battery model, a large current leads to a short lifespan of a battery, and γI_j is referred to as the instant voltage drop (IVD). We shall reduce the IVD, whenever possible, so as to extend the lifespan of a battery, especially when the battery voltage approaches to V_{cut_off}.

Table 1. Battery Model Parameters

Symbol	Descriptions	Unit
γ	Ohmic resistance	Ohm
V_0	Reference voltage	Volt
ϕ	Voltage curve flatness	Volt
α_n, α_p	Initial battery capacity	mAh
I_j	The latest current	Coulomb
I_k	The previous current	Coulomb
d_j	The duration of I_j	Second
d_k	The duration of I_k	Second
t_j	The time switch to I_j	Second
t	The current time point	Second

The operating of a 3G module consumes different amounts of energy for different bandwidths in data transmission, provided that an effective bandwidth, due to interference, is not a focus in the operating mode definition. The operating modes of a 3G module could be modeled by a set $\boldsymbol{M} = \{M_1, M_2, \ldots, M_N\}$, where each mode $M_i = (B_i, I_i)$ is associated with an ideal bandwidth B_i (with a unit: Kbps) and the required corresponding current I_i (with a unit: milliampere or mA), and let the operating modes in the set $\boldsymbol{M}$ be sorted in a non-increasing order of their bandwidths. Note that a high bandwidth in data transmission implies a short lifespan for the battery because it usually requires a large current, i.e., $B_i > B_j$ implies $I_i > I_j$ for $1 \leqslant i \neq j \leqslant N$. On the other hand, the promoting of a low current and a longer lifespan for the battery might miss the deadline in data transmission. When data receiving is considered, a server-based mechanism draws a fixed current I_r from the battery in each receiving interval. The effective receiving bandwidth might not be controllable by any individual mobile device because of competition in the radio resource. We shall consider the following task model in the usage of a 3G module.

From the viewpoint of an operating system, tasks might send or receive data over a 3G module in an on-demand fashion without any priori resource declaration. In order to properly manage the system resource and to guarantee the system performance, we assume the existence of a periodic task $\tau_r = (P_r, I_r, T_r)$, referred to as the *receiving task* and it receives whatever data that could be received on its residing device, and demands the current I_r for a time duration T_r from the battery in every period P_r. The operating system then dispatches the received data to the proper receiving applications. There could be one or more on-demand tasks $\tau_i = (C_i, R_i, D_i)$ in the system, referred to as a *sending task*, that arrives at time R_i (due to a request from some application) and requests to send a C_i amount of data with the deadline D_i.

3 Energy-Efficient Bandwidth Switching

3.1 Overview

In this section, we explore methodologies in the best utilization of the battery capacity to transmit data over a 3G module. Different from the past work on task-centric energy-efficient scheduling, we focus on how to transmit the maximum amount of data with a given battery capacity and different workload considerations. In Subsection 3.2, an optimal algorithm is presented to transmit the maximum amount of data with a given initial battery voltage. The algorithm is then extended to the consideration of a periodic task that is responsible for data receiving, and the optimality of the algorithm is proven to remain. Finally, multiple sending tasks with different amounts of to-be-transmitted data and deadlines are exploited in Subsection 3.4. Two on-line algorithms are proposed with the deadline satisfaction or the maximization of the battery utilization as the highest priority in scheduling is explored.

3.2 Battery Utilization Maximization

3.2.1 Basic Approach

Given a set of possible operating modes, we are interested in how to maximize the amount of data that could be sent over a 3G module with a given initial voltage. In other words, we shall derive a *bandwidth schedule* $\boldsymbol{S} = \{S_1, S_2, \ldots, S_k\}$ to maximize the amount of to-be-transmitted data, where $S_i = (M_i, T_i)$ denotes the executing of a mode M_i for the time duration T_i.

Problem Definition 1 (Battery Utilization Maximization (BUM)). *Given a beginning voltage V_0 and a set $\boldsymbol{M}$ of possible operating modes, the objective is to derive a bandwidth schedule $\boldsymbol{S}$ such that $\sum_{S_i \in \boldsymbol{S}} B_i T_i$ is maximized within the corresponding battery lifespan.*

The rationale behind the to-be-proposed algorithm could be explained by the relationship between a bandwidth and the corresponding current. We shall start with a 3G module of two modes $M_h = (B_h, I_h)$ and $M_l = (B_l, I_l)$ to illustrate the idea, where $B_h > B_l$ and $I_h > I_l$. Let the amount of data sent in a time interval be linearly propositional to the adopted bandwidth. In other words, if a 3G module sends data at bandwidth B_i for time t_i, then the amount of transmitted data is the multiplication of B_i and t_i. When the 3G module adopts various bandwidths in a time interval, the total amount of transmitted data is the sum of the multiplications of each adopted bandwidth and its corresponding time duration, that is, when t_h and t_l are the time intervals that a 3G module operates under B_h and B_l, respectively, then the total amount of transmitted data is $B_h t_h + B_l t_l$.

As pointed out by Rakhmatov and Vrudhula[15], the best way, from the battery perspective, is to schedule the workloads in nondecreasing order of their currents. In other words, given two possible modes M_h and M_l, the maximum amount of battery utilization can be explored by transmitting data with M_h first and switching to M_l at a proper time point. For the maximum amount of the transmitted data, referred to as *maximal sending data* (MSD), the bandwidth of different modes should also be considered for the device mode scheduling. We could define the lifespans T_h and T_l of the battery with the adoption of modes M_h and M_l, respectively:

$$T_h = \frac{A_h\alpha_p - \alpha_n}{(A_h+1)I_h}, \quad A_h \text{ is } e^{\frac{V_0-\gamma I_h - V_{cut\_off}}{\phi}}, \tag{2}$$

$$T_l = \frac{A_l\alpha_p - \alpha_n}{(A_l+1)I_l}, \quad A_l \text{ is } e^{\frac{V_0-\gamma I_l - V_{cut\_off}}{\phi}}. \tag{3}$$

We shall show the following properties in the adoption of modes of different bandwidths/currents.

Lemma 1. *Given two modes M_h and M_l, if $B_h > B_l$, then $I_l T_l > I_h T_h$.*

Proof. We first define the function $f(x) = \frac{x\alpha_p - \alpha_n}{(x+1)}$. For every $x > 0$, $\frac{d}{dx}f(x) = \frac{\alpha_p+\alpha_n}{(x+1)^2} > 0$, and therefore $f(x)$ is a monotonically increasing function. Since $A_l > A_h$, $I_l T_I = \frac{A_l\alpha_p-\alpha_n}{(A_l+1)} > \frac{A_h\alpha_p-\alpha_n}{(A_h+1)} = I_h T_h$. □

Theorem 1. *Given two modes M_h and M_l, if $\frac{B_l}{I_l} \geqslant \frac{B_h}{I_h}$, then a 3G module should always operate at the mode M_l to maximize the amount of transmitted data (within its corresponding lifespan).*

Proof. Since $B_l T_l = \frac{B_l}{I_l} I_l T_l$ and $B_h T_h = \frac{B_h}{I_h} I_h T_h$, $\frac{B_l}{I_l} I_l T_l > \frac{B_h}{I_h} I_h T_h$ is true according to the assumption $\frac{B_l}{I_l} \geqslant \frac{B_h}{I_h}$ and Lemma 1. Therefore, $B_l T_l > B_h T_h$ is always true because the IVD of M_l is also lower than M_h. In other words, the 3G module should always operate at B_l to achieve MSD. □

Similar to Lemma 1, if $\frac{B_h}{I_h} > \frac{B_l}{I_l}$, then a 3G module should operate at mode M_h first. The decision to switch from B_h to B_l depends on whether $B_h T_h < B_h t_h + B_l t_l$. Here t_l is the remaining time for a 3G module to operate under B_l after the module had worked for the duration t_h at bandwidth B_h. Given $t_h = T_h - \Delta$, the technical problem is to determine the existence of a positive number Δ such that the switching is justified. Since $t_l = \frac{A_l\alpha_p - \alpha_n - (A_l+1)I_h t_h}{(A_l+1)I_l}$, we know that

$(B_h I_l - B_l I_h)\Delta < B_l(I_l T_l - I_h T_h)$. Because of Lemma 1, we have $I_l T_l > I_h T_h$. Moreover, based on the assumption of this case discussion, we have $\frac{B_h}{I_h} > \frac{B_l}{I_l}$, and therefore $\frac{B_l(I_l T_l - I_h T_h)}{B_h I_l - B_l I_h} > 0$ is true. As a result, $B_h T_h < B_h t_h + B_l t_l$ if $0 < \Delta < \frac{B_l(I_l T_l - I_h T_h)}{B_h I_l - B_l I_h}$. We shall show the property of Δ with the following theorem.

Theorem 2. *Given two modes M_h and M_l, if $\frac{B_h}{I_h} > \frac{B_l}{I_l}$, then a 3G module needs to switch at $\Delta \to 0$ to maximize the amount of transmitted data.*

Proof. Let $F(\Delta)$ denote $B_h t_h + B_l t_l - B_h T_h$. $F(\Delta)$ could be rephrased as:

$$F(\Delta) = \frac{B_l}{I_l}\frac{(A_l - A_h)\alpha_n + (A_l - A_h)\alpha_p}{(A_l + 1)(A_h + 1)} + \Big(\frac{I_h}{I_l}B_l - B_h\Big)\Delta. \tag{4}$$

We can maximize the amount of transmitted data when $F(\Delta)$ is maximized. Since $\frac{B_h}{I_h} > \frac{B_l}{I_l}$, $\frac{I_h}{I_l}B_l - B_h < 0$. In other words, $F(\Delta)$ is maximized when $\Delta \to 0$. □

Algorithm 1. BUM Algorithm

Input: $\boldsymbol{S}_0$, V_0, $\boldsymbol{M}$;
Output: a bandwidth schedule $\boldsymbol{S}$;

```
1   S ← S_0;
2   k ← |S| + 1;
3   while true do
4   |   Let M_j ∈ M be the mode with the maximal B/I
    |   that is affordable by the current battery;
5   |   if M_j is the mode with the minimum current
    |   among all of the modes then
6   |   |   T_k ← T_interval(V_0, S, M_j);
7   |   |   S ← S ∪ {(M_j, T_k)};
8   |   |   return S;
9   |   else
10  |   |   T_k ← (T_interval(V_0, S, M_j) − ε_k);
11  |   |   S ← S ∪ {(M_j, T_k)};
12  |   |   k ← k + 1;
13  |   end
14  end
```

Theorem 2 implies that we should reduce the IVD right before its voltage approaches to V_{cut_off} so as to maximize the amount of transmitted data. Furthermore, Theorem 1 and Theorem 2 suggest to adopt a mode with a high value of $\frac{B}{I}$. Based on the observations, the algorithm BUM is proposed to derive a bandwidth schedule $\boldsymbol{S}$ to maximize the amount of data sent by a 3G module: the initial bandwidth schedule is set as $\boldsymbol{S}_0$, e.g., $\varnothing$ in this case, and the index k is reset to track the progress in the construction of the derived schedule, i.e., 1 in this case (Step 2). M_k is said being *affordable* if $V_0 - \phi \ln \frac{\alpha_n + \sum_{i=1}^{k-1} I_i T_i}{\alpha_p - \sum_{i=1}^{k-1} I_i T_i} > IVD(M_j)$, where $IVD(M_j)$ is the *IVD* of M_j, and it is assumed that M_j is the k-th mode in the schedule $\boldsymbol{S}$ (Step 4). The time interval of mode M_j is defined as $T_{interval}(V_0, \boldsymbol{S}, M_j) = \frac{(A\alpha_p - \alpha_n) - (1+A)\sum_{i=1}^{k-1} I_i T_i}{(1+A)I_k}$, where A is $e^{V_0 - \gamma I_k - V_{cut\_off}}$ (Step 6). In Step 10, ϵ_k denotes the minimal time interval allowed for a switch. ϵ_k should not only satisfy $B_k T_k < B_k t_k + B_{k+1} t_{k+1}$ but also be the minimum possible value allowed by a target system.

Theorem 3. *For a given set of operating modes, any bandwidth schedule derived by Algorithm 1 could maximize the amount of transmitted data over a 3G module.*

Proof. This theorem can be proved by a contradiction. Let $\boldsymbol{S}$ be a schedule derived by the algorithm, and $\boldsymbol{S}'$ be an arbitrary bandwidth schedule such that i, if it exists, is the smallest index value and the i-th mode-interval pair (denoted as (M_i, T_i)) of $\boldsymbol{S}$ is different from that (denoted as (M'_i, T'_i)) of $\boldsymbol{S}'$. According to Algorithm BUM, we have $\frac{B_i}{I_i} > \frac{B'_i}{I'_i}$. If $I_i T_i \geqslant I'_i T'_i$, let the 3G module operate at M_i for t_i, that satisfies $I_i t_i = I'_i T'_i$, which would have the same effect on the battery, compared to that at M'_i for T'_i. Since $B_i t_i = \frac{B_i}{I_i} I_i t_i = \frac{B_i}{I_i} I'_i T'_i > \frac{B'_i}{I'_i} I'_i T'_i = B'_i T'_i$, we can send more data for $\boldsymbol{S}'$ by replacing (M'_i, T'_i) with $(M_i, \frac{I'_i T'_i}{I_i})$. If $I_i T_i < I'_i T'_i$, $\boldsymbol{S}'$ should first schedule (M_i, T_i) at the i-th step to explore more data sending with the same battery consumption $I_i T_i$. In other words, S is a bandwidth schedule that could send more data than any other schedule does. □

3.2.2 *Consideration of Receiving Activities*

The purpose of this subsection is to consider not only the maximization of the amount of transmitted data but also the energy consumption of a periodic task for the receiving of incoming data.

Problem Definition 2 (Battery Utilization Maximization with a High-Priority Periodic Task (BUMP)). *Given a beginning voltage V_0, a set $\boldsymbol{M}$ of possible operating modes, and a periodic task $\tau_r = (P_r, I_r, T_r)$ with the highest priority, the objective is to derive a bandwidth schedule $\boldsymbol{S}$ such that $\sum_{S_i \in \boldsymbol{S}} B_i T_i$ is maximized within the corresponding battery lifespan.*

Since the receiving task has a higher priority than all of the sending tasks do, the 3G module would be occupied for every P_r time units for the time interval T_r. We propose to invoke Algorithm BUM with the initial schedule as the current schedule $\boldsymbol{S}$ to derive a sub-schedule between the finishing of τ_r and its next arrival (Step 6). The algorithm, referred to as Algorithm BUMP, then reconstructs the derived schedule

Algorithm 2. BUMP Algorithm

Input: V_0, $\boldsymbol{M}$, τ_r;

Output: a bandwidth schedule $\boldsymbol{S}$;

1 $\boldsymbol{S} \leftarrow \varnothing$;
2 **while** true **do**
3   **if** $T_{interval}(V_0, \boldsymbol{S}, M_r) > 0$ **then**
4     $\boldsymbol{S} \leftarrow \boldsymbol{S} \cup \{(M_r, \min\{T_{interval}(V_0, \boldsymbol{S}, M_r), T_r\})\}$;
5   **if** $T_{interval}(V_0, \boldsymbol{S}, M_r) > T_r$ **then**
6   Invoke $BUM(V_0, \boldsymbol{S}, \boldsymbol{M})$ with $\boldsymbol{S}'$ as the derived schedule;
7       $\Phi \leftarrow P_r - T_r$;
8   **for** $i \leftarrow |\boldsymbol{S}| + 1$ to $|\boldsymbol{S}'|$ **do**
9   **if** $T'_i \geqslant \Phi$ **then**
10           $\boldsymbol{S} \leftarrow \boldsymbol{S} \cup \{(M'_i, \Phi)\}$;
11           $\Phi \leftarrow 0$;
12   **break**;
13   **else**
14           $\boldsymbol{S} \leftarrow \boldsymbol{S} \cup \{(M'_i, T'_i)\}$;
15           $\Phi \leftarrow \Phi - T'_i$;
16   **end**
17   **end**
18   **end**
19   **else**
20   **return** $\boldsymbol{S}$;
21   **end**
22 **end**

based on the schedule derived by Algorithm BUM (Steps 7~17). The algorithm terminates (Step 20) when the battery voltage reaches the cut-off voltage. The optimality of the algorithm is shown as follows.

Theorem 4. *For a given set of operating modes and a highest-priority receiving task τ_r, any bandwidth schedule derived by Algorithm* 2 *could maximize the amount of transmitted data over a* 3*G module.*

Proof. The correctness of this theorem follows directly from the optimality of Algorithm BUM and the fact that the receiving task adopts a constant current I_r for every given time interval. □

3.3 Deadline Satisfaction with the Maximization of the Battery Utilization

3.3.1 Basic Approach

The purpose of this subsection is to consider the timing constraint of a sending task with the maximization of the battery utilization. We shall then extend the result to the consideration of a receiving task in the next section.

Problem Definition 3 (Deadline Satisfaction for a Single Task with the Battery Utilization Maximization (DSSBU)). *Given a beginning voltage V_0, a set $\boldsymbol{M}$ of possible operating modes, a sending task $\tau = (0, C, D)$, the objective is to derive a schedule $\boldsymbol{S}$ which not only completes the task τ in time but also maximizes the total amount of data that can be processed by the device before the voltage of the battery falls under V_{cut_off}.*

Algorithm 3. DSSBU Algorithm

Input: $V_0, \boldsymbol{M}, \tau$;

Output: a bandwidth schedule $\boldsymbol{S}$ or unschedulable;

1 Invoke $BUM(V_0, \varnothing, \boldsymbol{M})$ with $\boldsymbol{S}$ as the derived schedule;
2 **while** true **do**
3   **if** the amount of transmitted data in $\boldsymbol{S}$ before the deadline D is no less than C **then**
4   **return** $\boldsymbol{S}$;
5   **end**
6   $S' \leftarrow \{S_j | S_j \in \boldsymbol{S}$ and $\sum_{i=1}^{j} T_i > D\}$;
7   **if** $\boldsymbol{S}' = \varnothing$ **then**
8   **return** unschedulable;
9   **end**
10   Let S'_k be the executing of the mode with the maximum bandwidth B'_k in $\boldsymbol{S}'$ and $\check{S}_k$ be its previous executing of a mode in $\boldsymbol{S}$ if it exists;
11   $\boldsymbol{M}' \leftarrow \{M_j | M_j \in \boldsymbol{M}$, $B_j > B'_k$, and $B_j < \check{B}_k$ if $\check{S}_k$ exists$\}$;
12   **if** $\boldsymbol{M}' = \varnothing$ **then**
13   **return** unschedulable;
14   **else**
15   Insert (M_l, T_l) before S'_k, where M_l has the maximum $\frac{B_l}{I_l}$ in $\boldsymbol{M}'$, and T_l is derived by $T_{interval}()$;
16   Invoke Algorithm 1 for the remaining BUM problem instance;
17   **end**
18 **end**

A schedule is feasible if the task τ does not miss its deadline D, and the battery constraint is not violated. Algorithm 3 proposes a methodology that tries to meet the deadline of the task τ first by giving it the highest priority among all sending tasks implicitly and takes the energy efficiency in terms of the total amount of data transmitted by this device as the secondary consideration. First, we apply Algorithm BUM to derive an initial bandwidth schedule, $\boldsymbol{S}$, with the maximum amount of transmitted data (Step 1). If the amount of data that can be transmitted before the deadline D, when we follow the schedule $\boldsymbol{S}$, is no less than the to-be-transmitted data, i.e., C, we achieve the objective and return the schedule $\boldsymbol{S}$ as the solution (Steps 3~5). Otherwise, the algorithm should sacrifice a certain amount of data that can be transmitted after the deadline D for a larger amount of data that can be transmitted before the deadline D, where each $S_i \in \boldsymbol{S}$ is a pair of its mode M_i (with bandwidth B_i) and time interval T_i ordered by its execution sequence (Steps 6∼17). We find the mode M'_k which is adopted to transmit data right after the deadline D and its previous mode $\check{M}_k$ in the schedule $\boldsymbol{S}$ if M'_k is not the first mode adopted in the schedule $\boldsymbol{S}$. Let M_l be the mode with the maximum $\frac{B_l}{I_l}$ value among all modes in $\boldsymbol{M}$ whose bandwidths are larger than the bandwidth B'_k of the mode M'_k, and B_l be smaller than the bandwidth $\check{B}_k$ of the mode $\check{M}_k$ if $\check{M}_k$ exists. In order to increase the amount of data transmitted before the deadline D, we then shorten some part of the executing of the mode M'_k to insert an executing of the mode M_l, i.e., (M_l, T_l), before the executing of the mode M'_k (Step 15). Algorithm 1 is then invoked to handle the remaining BUM problem instance. Based on the definition of Algorithm 1, it will provide a shorter interval for the mode M'_k and keep the following schedule unchanged. If the mode M'_k or M_l does not exist, Algorithm 3 returns "unschedulable" to indicate it fails in finishing the task τ in time because there is no sufficient amount of data that is transmitted after the deadline D which can be traded for that before the deadline D (Step 8) or there is no mode with a higher bandwidth that can be adopted to increase the amount of data transmitted by the deadline D (Step 13).

3.3.2 Consideration of Deadline and Receiving Activities

Problem Definition 4 (Deadline Satisfaction for a Single Task with the Battery Utilization Maximization and a Periodic Task with Highest Priority (DSSBUP)). *Given a beginning voltage V_0, a set $\boldsymbol{M}$ of possible operating modes, a sending task $\tau = (0, C, D)$, and a periodic task $\tau_r = (P_r, I_r, T_r)$ with the highest priority, the objective is to derive a schedule $\boldsymbol{S}$ which not only satisfies the requirements of tasks τ and τ_r but also maximizes the total amount of data that can be processed by the device before the voltage of the battery falls under V_{cut_off}.*

Algorithm 4 shows our proposed algorithm for the DSSBUP problem. In Algorithm 4, the periodic task τ_r preempts the sending task at a period of P_r constantly because of its higher priority over sending tasks. After the completion of an task instance of τ_r, we try to finish the task τ in time by invoking Algorithm DSSBU with $V_{remaining}(V_0, \boldsymbol{S})$, $\boldsymbol{M}$, and τ as its input data, where $V_{remaining}(V_0, \boldsymbol{S})$ stands for the remaining voltage of the battery if we use the battery according to the bandwidth schedule $\boldsymbol{S}$ from the initial voltage V_0. Then, we follow the derived schedule $\boldsymbol{S}'$ before the next arrival of τ_r. Note that the amount of to-be-sent data C of the task τ is updated so that we can know whether the task τ is complete or not. When the battery voltage reaches the cut-off voltage (Steps 22∼28), if the task τ is complete, i.e., $C = 0$, the derived schedule $\boldsymbol{S}$ is returned. Otherwise, Algorithm 4 returns "unschedulable" to indicate it fails in finishing the task τ in time.

Algorithm 4. DSSBUP Algorithm

Input: V_0, $\boldsymbol{M}$, τ_r, τ
Output: a bandwidth schedule $\boldsymbol{S}$ or unschedulable;

1 $\boldsymbol{S} \leftarrow \varnothing$;
2 **while** true **do**
3   **if** $T_{interval}(V_0, \boldsymbol{S}, M_r) > 0$ **then**
4         $\boldsymbol{S} \leftarrow \boldsymbol{S} \cup \{(M_r, \min\{T_{interval}(V_0, \boldsymbol{S}, M_r), T_r\})\}$;
5   **if** $T_{interval}(V_0, \boldsymbol{S}, M_r) > T_r$ **then**
6   **if** $DSSBU(V_{remaining}(V_0, \boldsymbol{S}), \boldsymbol{M}, \tau)$ returns $\boldsymbol{S}'$ as its derived schedule **then**
7                 $\Phi \leftarrow P_r - T_r$;
8   **for** $i \leftarrow 1$ to $|\boldsymbol{S}'|$ **do**
9   **if** $T'_i \geqslant \Phi$ **then**
10                         $\boldsymbol{S} \leftarrow \boldsymbol{S} \cup \{(M'_i, \Phi)\}$;
11                         $C \leftarrow \max\{C - B'_i\Phi, 0\}$;
12                         $\Phi \leftarrow 0$;
13   **break**;
14   **else**
15                         $\boldsymbol{S} \leftarrow \boldsymbol{S} \cup \{(M'_i, T'_i)\}$;
16                         $C \leftarrow \max\{C - B'_iT'_i, 0\}$;
17                         $\Phi \leftarrow \Phi - T'_i$;
18   **end**
19   **end**
20   **end**
21   **end**
22   **else**
23   **if** $C = 0$ **then**
24   **return** $\boldsymbol{S}$;
25   **else**
26   **return** unschedulable;
27   **end**
28   **end**
29 **end**

3.4 Battery Utilization Maximization with Multiple Tasks

In this subsection, we extend previous results to deal with multiple tasks which may arrive during runtime with different deadlines. As we can see from Algorithm 3, in order to meet the deadline of a task, we might have to adopt operating modes in $\boldsymbol{M}'$ to serve the task. Since operating modes in $\boldsymbol{M}'$ are not so efficient in terms of the ratio $\frac{B_l}{I_l}$ compared to the operating modes adopted by Algorithm 1 automatically, there would be a trade-off between deadline satisfaction and the maximization of battery utilization when multiple tasks with different deadlines are considered. Thus, we proposes two approaches according to different objectives in this subsection.

3.4.1 Maximization of Transmitted Data

Problem Definition 5 (Battery Utilization Maximization with Multiple Tasks (BUMM)). *Given a beginning voltage V_0 and a set $\boldsymbol{M}$ of possible operating modes, the objective is to derive a bandwidth schedule $\boldsymbol{S}$ which maximizes the amount of transmitted data for a set of dynamically arrived tasks with different deadlines.*

Algorithm 5. DUMM-Arrival Algorithm

Input: V_0, $\boldsymbol{M}$, $\boldsymbol{S}'$, STK, WTL, t_{idle}, t_{cur}, τ_i

1 **if** $STK = \varnothing$ **then**
2 $\quad$ $\boldsymbol{S} \leftarrow \boldsymbol{S} \cup \{(M_N, (t_{cur} - t_{idle}))\}$ and $t_{idle} \leftarrow t_{cur}$;
3 $\quad$ Invoke $BUM(V_0, \boldsymbol{S}, \boldsymbol{M})$ and let $\boldsymbol{S}'$ be the derived bandwidth schedule;
4 **else**
5 $\quad$ Let τ_h be the task with the earliest deadline in STK;
6 $\quad$ **if** $D_h < D_i$ **then**
7 $\quad\quad$ Insert τ_i into WTL and **goto** Step 16;
8 $\quad$ **end**
9 **end**
10 **if** τ_i and tasks in STK can be completed in time by following $\boldsymbol{S}'$ with an earliest-deadline-first order **then**
11 $\quad$ Push τ_i onto STK;
12 **else**
13 $\quad$ Discard τ_i;
14 **end**
15 **if** $STK \neq \varnothing$ **then**
16 $\quad$ Execute tasks in STK in an earliest-deadline-first order by following $\boldsymbol{S}'$ and update $\boldsymbol{S}$ accordingly;
17 **end**

Algorithm BUMM-Arrival and Algorithm BUMM-Completion are two online algorithms to address the above problem. Algorithm BUMM-Arrival is invoked whenever a new sending task arrives, while Algorithm BUMM-Completion is invoked when a sending task completes its execution. Let WTL and STK be a pending queue and a stack to track task preemption, respectively. When a sending task arrives but could not be scheduled, it is inserted into WTL. When a task is going to start its execution, it is pushed onto the top of STK. During the system operation, a global system variable $\boldsymbol{S}$ is maintained to record the history of the bandwidth schedule which is required when we invoke Algorithm BUM. In addition, $\boldsymbol{S}'$ is maintained as a reference bandwidth schedule. When we execute sending tasks, we refer to $\boldsymbol{S}'$ to determine when to switch the operating mode and which operating mode should be adopted. The system starts at the operating mode M_N, i.e., the operating mode which has the lowest required current I_N in all modes in $\boldsymbol{M}$, with a beginning voltage V_0, current time $t_{cur} = 0$, the last idle time $t_{idle} = 0$, and empty WTL and STK.

When a new task τ_i arrives, Algorithm 5 first examines whether there is an active task, i.e., STK is not empty, or not. If there is no active task, $\boldsymbol{S}$ is revised by appending the executing of the lowest bandwidth mode M_N, i.e., $(M_N, t_{cur} - t_{idle})$, and the reference bandwidth schedule $\boldsymbol{S}'$ is then updated by invoking Algorithm BUM with the latest historical bandwidth schedule $\boldsymbol{S}$ (Steps 1∼3). Otherwise, we check whether the current arrived task τ_i has a deadline D_i earlier than that of the current active task τ_h in STK. If τ_h has an earlier deadline D_h, we just insert τ_i into WTL (Step 7)

Algorithm 6. BUMM-Completion Algorithm

Input: $\boldsymbol{M}$, $\boldsymbol{S}'$, STK, WTL, t_{idle}, t_{cur}

1 Pop the completed task from STK and remove tasks which miss their deadlines from WTL;
2 **while** $WTL \neq \varnothing$ **do**
3 $\quad$ Let τ_i be the task with the earliest deadline in WTL;
4 $\quad$ **if** $STK \neq \varnothing$ **then**
5 $\quad\quad$ Let τ_h be the task with the earliest deadline in STK;
6 $\quad\quad$ **if** $D_h < D_i$ **then**
7 $\quad\quad\quad$ **Goto** Step 18;
8 $\quad\quad$ **end**
9 $\quad$ **end**
10 $\quad$ Remove τ_i from WTL;
11 $\quad$ **if** τ_i and tasks in STK can be completed in time by following $\boldsymbol{S}'$ with an earliest-deadline-first order **then**
12 $\quad\quad$ Push τ_i onto STK and **goto** Step 18;
13 $\quad$ **else**
14 $\quad\quad$ Discard τ_i;
15 $\quad$ **end**
16 **end**
17 **if** $STK \neq \varnothing$ **then**
18 $\quad$ Execute tasks in STK in an earliest-deadline-first order by following $\boldsymbol{S}'$ and update $\boldsymbol{S}$ accordingly;
19 **else**
20 $\quad$ Let $t_{idle} \leftarrow t_{cur}$ and switch to operating mode M_N;
21 **end**

and continue to execute τ_h by following the reference bandwidth schedule $\boldsymbol{S}'$ (Step 16). When τ_i has an earlier deadline, or there is no active task, we check up whether τ_i and all tasks in STK can be completed in time if we follow the reference bandwidth schedule $\boldsymbol{S}'$ to switch the operating modes and execute tasks in an earliest-deadline-first order[1] . If the condition holds, we push τ_i into STK (Step 11) and execute tasks (Step 16). If the condition is not satisfied, we discard τ_i and keep the system idle at the lowest bandwidth mode M_N (Step 13).

When a task is completed, Algorithm 6 removes it from STK and all other tasks whose deadlines are missed from WTL (Step 1). Algorithm 6 then keeps getting the task τ_i with the earliest deadline from WTL when WTL is not empty. For each time a task τ_i is gotten from WTL, τ_i is treated as a newly arrived task in Algorithm 5, except that it is already in WTL. Since task τ_i is already in WTL, there is no need to insert τ_i into WTL before we continue to execute τ_h in STK when τ_h has an earlier deadline (Step 7), but we have to remove τ_i from WTL (Step 10) before we can push it into STK (Step 12) or discard it (Step 14). Another major difference between Algorithms 5 and 6 is that the reference bandwidth schedule $\boldsymbol{S}'$ is updated in the previous invocation of Algorithm BUMM-Arrival. Therefore, Algorithm 6 does not need to update $\boldsymbol{S}'$. However, when both STK and WTL become empty, i.e., the system becomes idle, the operating mode should be switched to the lowest bandwidth mode M_N for power saving, and the last idle time t_{idle} should be updated to the current time t_{cur} (Step 20).

3.4.2 Maximization of Deadline Satisfied Tasks

Problem Definition 6 (Deadline Satisfaction for Multiple Tasks with the Battery Utilization Consideration (DSMBU)). *Given a beginning voltage V_0 and a set $\boldsymbol{M}$ of possible operating modes, the objective is to derive a bandwidth schedule which maximizes the number of tasks which complete by their deadlines, where the tasks may arrive dynamically and have different deadlines.*

We proposed Algorithms DSMBU-Arrival and DSMBU-Completion to deal with the DSMBU problem. Similar to Algorithms BUMM-Arrival and BUMM-Completion, Algorithms DSMBU-Arrival and DSMBU-Completion are invoked when a new task arrives and a task completes, respectively. The fundamental idea behind Algorithms DSMBU-Arrival and DSMBU-Completion is also very close to that of Algorithms BUMM-Arrival and BUMM-Completion. The most significant difference between Algorithms DSMBU-Arrival and DSMBU-Completion and Algorithms BUMM-Arrival and BUMM-Completion is that Algorithms DSMBU-Arrival and DSMBU-Completion take the deadline satisfaction as the first priority. Therefore, instead of invoking Algorithm BUM to derive the reference bandwidth schedule for checking whether a set of tasks can be completed in time or not, Algorithms 7 and 8 adopt a single operating mode to maximize the amount of data that could be processed before deadlines of tasks (Step 9 in Algorithm 7 and Step 11 in Algorithm 8). As a result, the reference bandwidth schedule $\boldsymbol{S}'$ in Algorithms 5 and 6 is replaced with a reference bandwidth M' in Algorithms 7 and 8. Although Algorithms 7 and 8 do not apply Algorithm BUM directly to derive a reference bandwidth schedule, they adopt the mode with the largest ratio of a bandwidth to its required current as the reference bandwidth M' (Step 11 in Algorithm 7 and Step 13 in Algorithm 8) if more than one mode satisfies the requirement of task τ_i and tasks in STK.

Algorithm 7. DSMBU-Arrival Algorithm

Input: V_0, $\boldsymbol{M}$, $\boldsymbol{M}'$, STK, WTL, t_{idle}, t_{cur}, τ_i

1 **if** $STK = \varnothing$ **then**
2 $\quad$ $\boldsymbol{S} \leftarrow \boldsymbol{S} \cup \{(M_N, (t_{cur} - t_{idle}))\}$ and $t_{idle} \leftarrow t_{cur}$;
3 **else**
4 $\quad$ Let τ_h be the task with the earliest deadline in STK;
5 $\quad$ **if** $D_h < D_i$ **then**
6 $\quad\quad$ Insert τ_i into WTL and **goto** Step 17;
7 $\quad$ **end**
8 **end**
9 $\boldsymbol{M}' \leftarrow \{M_j | M_j \in \boldsymbol{M}$ and $T_{interval}(V_0, \boldsymbol{S}, M_j) \cdot B_j$ is no less than the total amount of remaining data of τ_i and all tasks in $STK\}$;
10 **if** $\boldsymbol{M}' \neq \varnothing$ **then**
11 $\quad$ Let M' be the mode with the highest $\frac{B'}{I'}$ in $\boldsymbol{M}'$;
12 $\quad$ Push τ_i onto STK;
13 **else**
14 $\quad$ Discard τ_i;
15 **end**
16 **if** $STK \neq \varnothing$ **then**
17 $\quad$ Execute tasks in STK in an earliest-deadline-first order with operating mode M' and update $\boldsymbol{S}$ accordingly;
18 **end**

[1] Only τ_i needs to be considered when there is no active task since there is no task in STK if STK is empty. The verification could be done by simply comparing the amount of data of all tasks with the provided bandwidth according to $\boldsymbol{S}'$ over time. Note that the provided bandwidth of a 3G module in practice could not be predicted perfectly because of factors such as interference.

Algorithm 8. DSMBU-Completion Algorithm

```
    Input: M, S', STK, WTL, t_idle, t_cur
1   Pop the completed task from STK and remove tasks
    which miss their deadlines from WTL;
2   while WTL ≠ ∅ do
3       Let τ_i be the task with the earliest deadline in
        WTL;
4       if STK ≠ ∅ then
5           Let τ_h be the task with the earliest deadline
            in STK;
6               if D_h < D_i then
7               Goto Step 20;
8           end
9       end
10      Remove τ_i from WTL;
11      M' ← {M_j | M_j ∈ M and T_interval(V_0, S, M_j) · B_j
        is no less than the total amount of remaining data
        of τ_i and all tasks in STK};
12      if M' ≠ ∅ then
13          Let M' be the mode with the highest B'/I' in M';
14          Push τ_i onto STK and goto Step 20;
15      else
16          Discard τ_i;
17      end
18  end
19  if STK ≠ ∅ then
20      Execute tasks in STK in an earliest-deadline-first
        order with operating mode M' and update S acco-
        rdingly;
21  else
22      Let t_idle ← t_cur and switch to operating mode M_N;
23  end
```

4 Performance Evaluation

The purpose of this section is to verify the correctness of the presented battery and 3G models and to evaluate the proposed algorithms for different application behaviors and hardware performance. We shall also explore insights in the maximization of the amount of data in transmission, that could be closely related to the battery lifespan issue.

4.1 Environment Setup

Fig.1 shows the environment setup of our experiments, where HTC magic mobile phones of Android 1.5 were adopted in the evaluation. In the experiments, an application with needed functionality was implemented upon HTC magic mobile phones. In addition to the application, the policy administrator program was also implemented to monitor the application and to send mode switching commands to the power control integration based on the proposed algorithms and the status information of the battery provided by the battery status monitor.

Each HTC magic mobile phone under experiments was powered by a Li-ion battery (BA S350) and equipped with a 3G module which was integrated with the Qualcomm MSM7200A. A profiling device, i.e., the power monitor by the Monsoon Solution Inc., was adopted to measure the current and voltage of the battery and then to derive the battery parameters. The profiling device acquired data in a sampling way of a period equal 200 microseconds. Based on the same method in [15], we derived the following parameters to model the battery BA S350: $\gamma = 0.479$, $V_0 = 4.17$, $V_{cut\_off} = 3.52$, $\phi = 0.0737$, $\alpha_n = 56$, and $\alpha_p = 815$. The operating modes supported by HTC magic mobile phones were $\boldsymbol{M} = \{(256, 450.18)$, $(192, 360.56)$, $(128, 306.05)$, $(64, 243.84)$, $(32, 224.11)$, $(0, 1)\}$, where

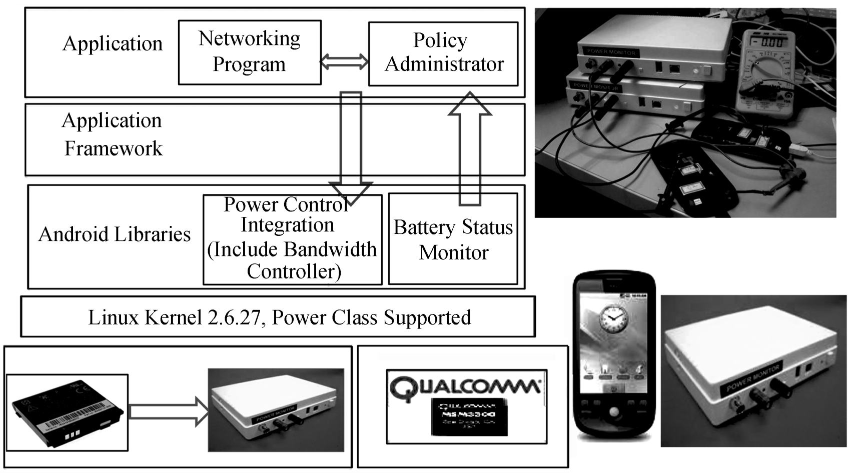
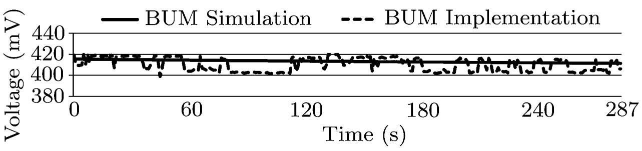


Fig.1. Experiment setup.

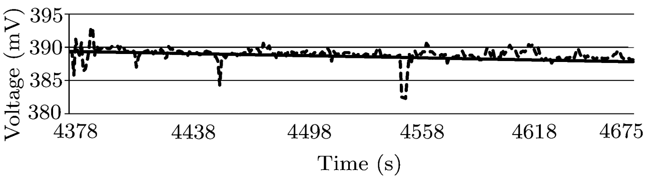
the units of each bandwidth B_i and its corresponding current I_i were Kbps and mA, respectively. Note the last mode $M_6 = (0, 1)$ was the sleep mode which we could switch to while the 3G module was idle for power saving with the cost of 539.76 mA in one second due to the switching overhead.

4.2 Experimental Results

Fig.2 shows the energy consumption in terms of the voltage drop of a battery when we sent data with the 3G module according to Algorithm BUM, where the x-axis and y-axis stand for the time (second) and voltage (mV), respectively. The experimental results of the intervals that could not transmit data, due to interference, were not listed. The actual amount of data that the on-board 3G module could send was slightly less than the specified bandwidth, i.e., 256 Kbps, because of the bit error resulting from the signal degradation and potential interference. The difference of the voltage drops between the simulation and the on-board implementation was less than 4%. It shows that the adopted battery and 3G models provided reasonable information in the designs of energy-efficient algorithms in data transmissions.



Time Interval 1	0∼59	60∼119	120∼179	180∼239	240∼287
Average Kbps (BUM-Implementation)	210.432	212.48	174.08	235.776	208.896



Time Interval	4378∼4437	4438∼4497	4498∼4557	4538∼4617	4588∼4675
Average Kbps (BUM-Implementation)	187.904	209.152	248.064	233.472	228.096

Fig.2. Voltage drop and the average bandwidth of Algorithm BUM.

Two sets of experiments were also conducted to evaluate Algorithm BUM in the maximization of the amount of data transmission based on the behavior of the Li-ion battery (BA S350), especially on the efficiency of the 3G module versus the potential energy saving. Note that Algorithm BUM is an optimal algorithm in terms of maximizing the amount of data transmission.

In the first set of the experiments, the potential energy saving was evaluated by increasing the current in each bandwidth of the operating modes of a 3G module. Fig.3 shows the increased amount of data transmission, i.e., the y-axis, with different ratio increments of the currents, i.e., the x-axis. Note that when the increased ratio of the current was 110%, the current of each corresponding bandwidth was its original value multiplied by 110%. For example, the set of the original operating modes of the 3G module was $\boldsymbol{M} = \{(256, 736)$, $(192, 576)$, $(128, 392)$, $(64, 198)$, $(32, 100)\}$. When the increased ratio of the current was set as 110%, the set of the operating modes became $\boldsymbol{M} = \{(256, 809.6)$, $(192, 633.6)$, $(128, 431.2)$, $(64, 217.8)$, $(32, 110)\}$. It was shown that the potential in energy saving for data transmission would become more significant when operating currents increased or, more specifically, when the difference between the power consumption of a high bandwidth and that of a low bandwidth increased for a 3G module.

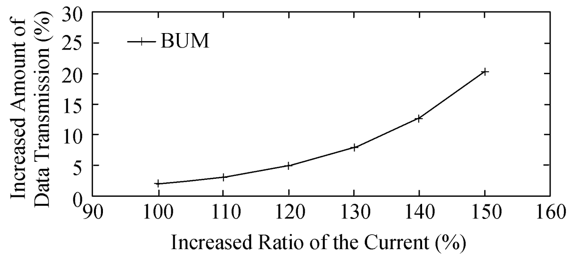


Fig.3. Increased amount of data transmission with different 3G-device currents.

The second set of experiments further evaluated the potential energy saving with respect to the efficiency in data transmission. As pointed out by Theorems 1 and 2, the system should favor a bandwidth with a higher ratio of the bandwidth to the current, i.e., $(\frac{B_i}{I_i})$, in the maximization of data transmission. When a 3G module has a high efficiency level for a high bandwidth, the system should try to transmit data with a large

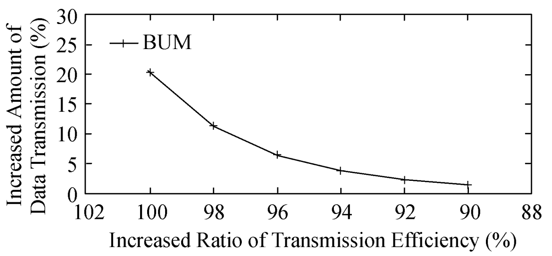


Fig.4. Increased amount of data transmission with different efficiency levels of data transmission.

current and delay its switching to a low bandwidth mode until the battery capacity approaches the cutoff voltage. In the second set of experiments, the potential energy saving was evaluated for possible solutions (even though Algorithm BUM was adopted as an example study) with different efficiency-level possibilities. Fig.4 shows the increased amount of data transmission, i.e., the y-axis, when the system had a different increased ratio of transmission efficiency, i.e., the x-axis. When the increased ratio of transmission efficiency was set as 98%, the current of the i-th mode was multiplied by $(98\%)^{(5-i)}$. In other words, the transmission efficiency increased with the decreasing of the ratio. Suppose that the original operating mode set was $\boldsymbol{M} = \{(256, 1104), (192, 865), (128, 588), (64, 297), (32, 150)\}$. When the increased ratio of transmission efficiency was 98%, the currents of the modes became $1104(\frac{98}{100})^4$, $865(\frac{98}{100})^3$, $588(\frac{98}{100})^2$, $297(\frac{98}{100})$, and 150, respectively. The experimental results show that the potential saving in energy would decrease for solutions when the efficiency of a high bandwidth increased. In some extreme case, the system should just try to use a high bandwidth greedily to maximize the total amount of data transmission.

To demonstrate the effectiveness of the proposed algorithms for the BUMM problem, we further conducted a set of experiments to evaluate the real-time performance and energy efficiency. The battery model of BA S350 and the 3G module of HTC Magic $\boldsymbol{M} = \{(256, 450.18), (192, 360.56), (128, 306.05)\}, (64, 243.84), (32, 224.11), (0, 1)\}$ were adopted for the following experiments. Algorithm FIFO was implemented for reference, where the tasks were served according to their arrival time, and no preemption was allowed. In this implementation, the system never considered the deadline of each sending job and run as a batch processing server to send all the data. The *deadline satisfaction ratio* which was defined as the number of tasks that meet their deadlines under Algorithms BUMM-Arrival and BUMM-Completion divided by that under Algorithm FIFO was adopted as the performance metrics. A larger value of deadline satisfaction ratio means a greater improvement of Algorithms BUMM-Arrival and BUMM-Completion over Algorithm FIFO. In our experiments, each task set consisted of 10 tasks, and we evaluated 100 task sets independently for each value of *slack*. The experimental data of each task was generated by a normal distribution function, where the mean and the standard deviations were 4096 KByte (4 MByte) and 1024, respectively. The value of *slack* was defined as the ratio of the relative deadline to the sending time under the highest bandwidth mode, i.e, 256 Kbps, for each task. The average improvement of deadline satisfaction ratio when *slack* is from 1 to 6 is shown in Fig.5. In our experimental results, Algorithms BUMM-Arrival and BUMM-Completion could finish much more tasks without violating their deadline constraints, compared to that under the Naïve FIFO approach.

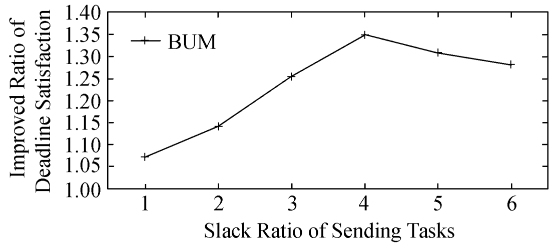


Fig.5. Deadline satisfaction ratio of Algorithm BUMM.

5 Conclusion

This paper explores the best battery utilization problems under different configurations over a 3G module. We exploit a battery model and the characteristics of a 3G module in data transmissions. We first propose optimal algorithms to derive bandwidth schedules with and without the consideration of a receiving task, with an objective to maximize the amount of transmitted data. We later extend the work with the consideration of the deadline constraint of a real-time sending task. The presented models were validated by a series of experiments, and insights in energy-efficient data transmission were drawn with different configurations in transmission efficiency. When multiple networking applications are considered in the system scenario, two power-aware bandwidth scheduling algorithms are proposed to favor the amount of data transmission and the deadline satisfaction ratio, respectively. The experimental results also illustrate the flexibility of the presented models.

For future work, we shall explore the joint resource scheduling of tasks with both needs in computing and data transmissions. More research on application behaviors would be very important in energy-efficient designs of mobile devices.

References

[1] Zhang T, Chen Y J, Chang C W, Yang C Y, Kuo T W, Chen T. Power management strategies in data transmission. In *Proc. Asia and South Pacific Design Automation Conference*, Yokohama, Japan, Jan. 25-28, 2011, pp.668-675.

[2] Science and Technology Policy Research and Information Center. http://cdnet.stpi.org.tw/techroom/market/eetelecom_mobile/2009/eetelecomm_mobile_09_008.htm, Jul. 1, 2010.

[3] Ahn J, Min J, Cha H, Ha R. A power management mechanism for handheld systems having a multimedia accelerator. In *Proc. IEEE International Conference on Pervasive Computing and Communications*, Hong Kong, China, Mar. 17-21, 2008, pp.663-668.

[4] Ashwini H S, Thawani A, Srikant Y N. Middleware for efficient power management in mobile devices. In *Proc. the 3rd Int. Conf. Mobile Technology, Applications & Systems*, Bangkok, Thailand, Oct. 25-27, 2006, Article No.49.

[5] Cornea R, Mohapatra S, Dutt N, Nicolau A, Venkatasubramanian N. Integrated power management for video streaming to mobile handheld devices. In *Proc. ACM Multimedia*, Berkeley, USA, Nov. 2-8, 2003, pp.582-591.

[6] Lu Y H, Simunic T, De Micheli G. Software controlled power management. In *Proc. the IEEE Hardware/Software Co-Design Workshop*, Rome, Italy, May 3-5, 1999, pp.157-161.

[7] Qiu Q, Wu Q, Pedram M. OS-directed power management for mobile electronic systems. In *Proc. Power Source Conference*, pp.506-509.

[8] Yuan W, Nahrstedt K. Practical voltage scaling for mobile multimedia devices. In *Proc. the 12th Annual ACM International Conference on Multimedia*, New York, USA, Oct. 10-16, 2004, pp.924-931.

[9] Chandra R. MultiNet: Connecting to multiple IEEE 802.11 networks using a single wireless card. In *Proc. IEEE INFOCOM*, Hong Kong, China, Mar. 7-11, 2004, pp.882-893.

[10] Chiasserini C F, Rao R R. A model for battery pulsed discharge with recovery effect. In *Proc. Wireless Communications and Networking Conference*, New Orleans, USA, Sept. 21-24, 1999, pp.636-639.

[11] Nam M, Choi N, Seok Y, Choi Y. Wise: Energy-efficient interface selection on vertical handoff between 3G networks and WLANs. In *Proc. the 15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications* (*PIMRC 2004*), Barcelona, Spain, Sept. 5-8, 2004, pp.692-698.

[12] Yang S R. Dynamic power saving mechanism for 3G UMTS system. *Mob. Netw. Appl.*, 2007, 12(1): 5-14.

[13] Baey B, Dumas M, Dumas M C. QoS tuning and resource sharing for UMTS WCDMA multiservice mobile. *IEEE Transactions on Mobile Computing*, 2002, 1(3): 221-235.

[14] Yuan W, Nahrstedt K. Energy-efficient soft real-time CPU scheduling for mobile multimedia systems. In *Proc. the Nineteenth ACM Symposium on Operating Systems Principles*, The Sagamore, USA, Oct. 19-22, 2003, pp.149-163.

[15] Rakhmatov D. Battery voltage modeling for portable systems. *ACM Trans. Des. Autom. Electron. Syst.*, 2009, 14(2): pp.1-36.

[16] Rakhmatov D, Vrudhula Sa. Energy management for battery-powered embedded systems. *ACM Trans. Embed. Comput. Syst.*, 2003, 2(3): 277-324.

[17] Zhuo J, Chakrabarti C, Chang N. Energy management of DVS-DPM enabled embedded systems powered by fuel cell-battery hybrid source. In *Proc. the 2007 International Symposium on Low Power Electronics and Design*, Portland, USA, Aug. 27-29, 2007, pp.322-327.

[18] Rao R, Vrudhula S, Rakhmatov D. Analysis of discharge techniques for multiple battery systems. In *Proc. the 2003 International Symposium on Low Power Electronics and Design*, Seoul, Korea, Aug. 25-27, 2003, pp.44-47.

[19] Aydin H, Melhem R, Mosse' D, Meji'a-Alvarez P. Dynamic and aggressive scheduling techniques for power-aware real-time systems. In *Proc. RTSS 2001*, London, UK, Dec. 3-6, 2001, pp.95-105.

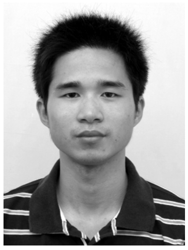
[20] Chen J J, Kuo T W. Procrastination determination for periodic real-time tasks in leakage-aware dynamic voltage scaling systems. In *Proc. 2007 IEEE/ACM Int. Conf. Computer-Aided Design*, San Jose, USA, Nov. 5-8, 2007, pp.289-294.

[21] Jejurikar R, Gupta R. Dynamic slack reclamation with procrastination scheduling in real-time embedded systems. In *Proc. the 42nd Annual Design Automation Conference*, San Diego, USA, Jun. 13-17, 2005, pp.111-116.

[22] Yang C Y, Chen J J, Kuo T W. An approximation algorithm for energy-efficient scheduling on a chip multiprocessor. In *Proc. the Conference on Design, Automation and Test in Europe*, Munich, Germany, Mar. 7-11, 2005, pp.468-473.

[23] Yang C Y, Chen J J, Kuo W. Energy-efficiency for multi-frame real-time tasks on a dynamic voltage scaling processor. In *Proc. the 7th IEEE/ACM International Conference on Hardware/Software Codesign and System Synthesis*, Grenoble, France, Oct. 11-16, 2009, pp.211-220.

Che-Wei Chang received his B.S. degree in computer science from "National Chiao Tung University", Hsinchu, Taiwan, China, in 2006. He is currently working toward the Ph.D. degree in computer science and information engineering in "National Taiwan University" and supervised by Prof. Tei-Wei Kuo. His currently research interests include energy-efficient scheduling, fast-booting designs and multicore management.



Tie-Fei Zhang received his B.S. degree in control science and engineering from Zhejiang University, Hangzhou, China, in 2007. He is currently pursuing the Ph.D. degree in College of Computer Science, Zhejiang University. His research interests include energy-efficient design and power-aware designs.

Chuan-Yue Yang received his B.S. degree in information and computer education from "National Taiwan Normal University", Taipei, Taiwan, China, in 2003. He received his M.S. and Ph.D. degrees in computer science and information engineering from "National Taiwan University", in 2005 and 2010, respectively. While he studied in "National Taiwan University", he was supervised by Prof. Tei-Wei Kuo. His research interests include energy-efficient scheduling, power-aware designs, and temperature-aware scheduling. He currently works for Genesys Logic, Inc.

Ying-Jheng Chen received his B.S. degree from Department of Computer Science of "National Chiao Tung University", China, in 2008. He received his M.S. degree in computer science and information engineering from "National Taiwan University" in 2010. While he studied in "National Taiwan University", he was supervised by Prof. Tei-Wei Kuo. His research interests include energy-efficient scheduling, and mobile system like embedded Linux and Android.

Shih-Hao Hung joined the Department of Computer Science and Information Engineering at "National Taiwan University" as assistant professor in 2005. His research interests include cloud computing, parallel processing, embedded systems, and pervasive applications. He worked for the Performance and Availability Engineering group at Sun Microsystem Inc. in Menlo Park, California (2000∼2005) after completing his post doctoral work (1998∼2000), Ph.D. (1994∼1998) and M.S. (1992∼1994) degrees at the University of Michigan, Ann Arbor. He graduated from "National Taiwan University" with a B.S. degree in electrical engineering in 1989.

Tei-Wei Kuo received his B.S.E. and Ph.D. degrees in computer science from the "National Taiwan University" and the University of Texas at Austin in 1986 and 1994, respectively. He is a distinguished professor of the Department of Computer Science and Information Engineering, "National Taiwan University", where he served as the chairman from August 2005 to July 2008. Prof. Kuo is an IEEE Fellow. He serves in the editorial board of the Journal of Real-Time Systems and IEEE Transactions on Industrial Informatics. He was a program chair and a general chair of the IEEE Real-Time Systems Symposium (RTSS) in 2007 and 2008, respectively. Prof. Kuo serves as an independent board director of the Genesys Logic and the MStar Semiconductor, which are leading companies in IC designs for display, smartphones, wireless communication, USB, and flash-memory controllers. He has over 180 technical papers published or been accepted in international journals and conferences and more than 6 patents in USA and Taiwan on the designs of flash-memory storage systems.

Tian-Zhou Chen is a professor of computer science at Zhejiang University. His current research interests include computer architecture, embedded system, power-aware computing.