Battery-Aware Power Management Based on Markovian Decision Processes

Peng Rong and Massoud Pedram
Dept. of EE, University of Southern California
{prong,pedram}@usc.edu

Abstract - **This paper addresses the problem of maximizing capacity utilization of the battery power source in a portable electronic system under latency and loss rate constraints. First, a detailed stochastic model of a power-managed, battery-powered electronic system is presented. The model, which is based on the theories of continuous-time Markovian decision processes and stochastic networks, captures two important characteristics of today's rechargeable battery cells, i.e., the current rate-capacity characteristic and the relaxation-induced capacity recovery. Next, the battery-aware dynamic power management problem is formulated as a policy optimization problem and solved exactly by using a linear programming approach. Experimental results show that the proposed method outperforms existing methods by more than 20% in terms of battery service lifetime.**

I. INTRODUCTION

With the rapid progress in semiconductor technology, chip density and operation frequency have increased, making the power consumption in battery-operated portable devices a major concern. High power consumption reduces the battery service life. The goal of low-power design for battery-powered devices is, thus, to extend the battery service life while meeting a set of performance specifications. Dynamic power management (DPM) – which refers to a selective, shut-off or slow-down of system components that are idle or underutilized – has proven to be a particularly effective technique for reducing power dissipation in such systems.

Early work on DPM described predictive shutdown approaches [1][2] that make use of "time-out" based policies. A power management approach based on discrete-time Markovian decision processes was proposed in [3]. The discrete-time model requires policy evaluation at periodic time intervals and may thereby consume a large amount of power even when no change in the system state has occurred. To surmount this shortcoming, a model based on continuous-time Markovian decision processes (CTMDP) was proposed in [4]. The policy change under this model is asynchronous and, thus, more suitable for implementation as part of a real-time operating system environment. Reference [5] improved on the modeling technique of [3] by using time-indexed semi-Markovian decision processes.

Although the abovementioned DPM techniques may effectively reduce the system power consumption, they are not able to obtain the optimal policy for a battery-powered system. This is because the characteristics of battery power source are not properly modeled and exploited in these techniques. As demonstrated by research results in [6], the total energy capacity that a battery can deliver during its lifetime is strongly related to its discharge current rate. More precisely, as the discharge current rate increases, the deliverable capacity of the battery decreases. This phenomenon is called the (current) *rate-capacity characteristic*. Another important property of batteries, which was analyzed and modeled in [7][8], is often called the *recovery characteristic* (or relaxation effect.) This effect is due to the concentration gradient of active materials in the electrode and electrolyte that are formed during the battery discharge process. More precisely, the active material at the electrolyte-electrode interface is consumed by the electrochemical reactions during discharge. This material is replenished with new active materials through a diffusion process that is guided by the concentration gradient. Thus, the battery capacity is somewhat recovered in a state during which no current is drawn. Due to these two non-linear characteristics, a minimum power consumption policy does not necessarily result in the longest battery service life because the energy capacity of its power sources may be not fully exploited when the cut-off voltage of the battery is reached.

To the best of our knowledge, there has been no reported work on integrating the general model of a power-managed portable electronic system with that of its power source – i.e., batteries. Indeed, this is the contribution of the present paper. An integrated model makes it possible to wholly and correlatively consider the statistical behavior of incoming tasks, the electrical features (e.g. current levels) of the electronic system, and the electrochemical characteristics of the batteries. Consequently, a systematic approach, which optimally determines a power-management policy that exploits both the advantages of battery management and discharge-rate shaping techniques, can be developed on the top of this model.

In this paper, we present a novel battery-aware power management (BAPM) technique based on CTMDP. The BAPM technique presented in this paper targets to maximize the battery service lifetime while meeting the given service timing constraints. It operates as dynamically selecting the operating modes of the electronic system and concurrently choosing the working battery if multiple batteries are available, based not only on incoming task features and electronic system states, but also on the characteristics and state of the batteries powering the system. More precisely, we extend the work in [4] to achieve a complete model of a battery-powered portable system by introducing and incorporating a new CTMDP model of the battery source. This model correctly captures the two important battery characteristics, i.e., the recovery effect and current-capacity

curve. Furthermore, it considers the case of a multiple battery power source with a power switch that is controlled by the power management policy. Based on this model the battery-aware power management problem is formulated as a policy optimization problem based on the CTMDP theory and solved optimally by using linear programming (LP).

The remainder of the paper is organized as follows: The related work is discussed in Section 2. In Section 3, the formal statement of our problem is presented. The model of the battery-powered portable power-managed system is described in Section 4. The solution technique for the optimal problem is described in Section 5. In Section 6, we present the experiment results and we conclude in Section 7.

II. RELATED WORK

A number of battery models have been proposed. These can be divided into two categories: electrochemical model and stochastic model. The electrochemical models are based on diffusion equations and provide an accurate description of the underlying electrochemical process. A low level model for lithium-ion batteries and a high level model for the time-varying load were proposed in [8] and [9], respectively. A different high-level battery model based on discrete-time VHDL was presented [10]. Compared to the electrochemical ones, the high-level models are more efficient but less accurate. These models require a predetermined workload profile. However, in most real situations, the workload is not a priori known. In fact, the workload often evolves as a random process. In these cases, stochastic models become very useful. These models describe the battery behavior as a stochastic process whose parameters are extracted from the electrochemical characteristics of the simulated battery.

A number of stochastic models have been reported in the literature, e.g. a discrete-time Markovian chain model [11]. The stochastic model in [11] is a Markovian chain of the battery states of charge with forward and backward transitions corresponding to the normal discharge and recovery effect processes, respectively. The load is expressed as a stochastic demand on the charge units. This model is mainly focused on the recovery effect. In a later work [12], the authors extended this model to incorporate the rate-capacity effect. Both models are based on discrete time Markov chain construction and are merely developed to predict the battery lifetime. In contrast, we propose an integrated model of a power-managed system, which can be used to develop battery-aware power management techniques. More precisely, our work is based on a CTMDP-based stochastic model of a rechargeable battery and is suitable for development of system-level power management strategies.

To exploit the two battery characteristics in an attempt to extend the lifetime of a mobile battery-powered system, two classes of techniques have been proposed: *battery management* [14][15] and *discharge-rate shaping* [16][17][18]. Battery management refers to the class of techniques for selecting and scheduling the battery to discharge at any given time in a multiple-battery mobile system. A round robin policy was presented in [13]. Several battery-scheduling policies were studied and compared in [14]. Some of these policies are based on battery state-of-charge information, which can be measured with Smart Battery technologies.[1] Reference [15] considered a dual-battery power supply, which comprises of two batteries that have different rate-capacity characteristics. The authors employ the batteries in an interleaved manner in responding to the current requirements of the battery-powered electronic system. However, none of these techniques can guarantee the optimality of the online battery schedule.

The discharge-rate shaping techniques shape the discharge current profile of a single battery source in order to match the current draw of the system to the battery characteristics. In [16], the authors proved that minimizing the variance of the discharge current profile of a battery leads to the maximum battery lifetime. Based on this principle, reference [17] proposed a battery-aware variable-voltage scheduling technique for periodic tasks to minimize the peak power consumption. Reference [18], which is based on an analytical battery model of [9], proposed a number of battery-aware algorithms for task scheduling and voltage assignment, including idle time allocation to exploit the capacity recovery effect. The approaches in [17][18] require a complete a priori knowledge of the energy cost and execution time of all tasks that must be executed. A system-level communication-architecture based system execution regulation method, named CBPM, was proposed in [19]. The CBPM limits the instantaneous power consumption to a predetermined threshold level by delaying operations that have low timing criticality.

In [20], the authors made a first attempt to combine the two classes of techniques to extend the battery life of a dual battery-powered portable system. They proposed an open-loop policy (switching from one battery to next with a fixed frequency), a closed-loop policy (switching from a high quality factor system state to a low quality factor system state when the output voltage of the battery drops below some threshold), and a rather elaborate hybrid switching policy (which is in fact a combination of the fixed switching frequency and the voltage threshold-based quality factor scaling techniques.) Note that all three policies are only heuristic solutions. The closed loop policy increases the battery lifetime by 8% over the open-loop policy, but forfeits about 30% of the (sound) quality in the target audio player platform. We will compare our battery-aware power management policy with both the open-loop and hybrid switching policies of [20].

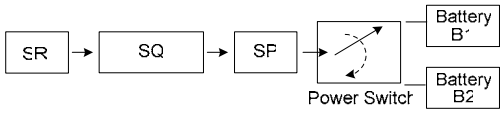
III. PROBLEM STATEMENT

This paper targets a portable, battery-powered electronic system. Fig. 1 depicts two example abstract models of such a system. The first model, called 2BAT, corresponds to that of a dual-battery, single-provider system. The 2BAT model

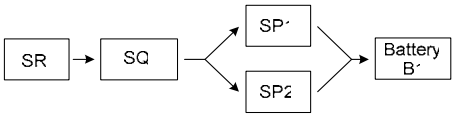
[1] Smart battery system [21] uses battery inside circuits to measure battery state and inner data, such as state of charge and temperature, and provides them to the operating system through SMBus. Using smart battery system the operating system, or power manager, can have instant and accurate information about battery available capacity.

consists of a service requestor (SR) to generate the tasks to be serviced, a single service provider (SP) to supply the required services, and a service queue (SQ) to store the tasks waiting for service. The SP is powered by two batteries (BAT) B1 and B2, which may have different current-capacity and recovery characteristics. The power switch (PS) selects either B1 or B2 to provide power for the SP at any given time. Note that only one of the batteries is used at any given time (the other is resting at that time.) The second model, called 2SP, corresponds to that of dual-SP, single-battery system. The 2SP model is different from the 2BAT model, only in that there are two SPs which may work in parallel, but only one battery.

This paper focuses on these two system models because they capture the key characteristics of many real systems that are in actual use. Multi-battery power supplies are employed in many portable systems. For example, laptops normally provide users an option to insert an additional battery bay instead of a floppy disk. A multi-processor structure is also widely used in portable electronic systems. A typical configuration is a high-end mobile device with two CPUs. Both models use up to two SP's or batteries. It is straightforward to extend the proposed framework and solution technique to handle multiple SP's or multiple batteries. In addition, notice that a common characteristic of both system models is that they employ one SR and one SQ. This is only for the sake of simplifying the presentation. Extension to include multiple SR's and/or multiple SQ's is straightforward and has been addressed in the published literature (e.g., see [22].) Based on the 2BAT and 2SP system models, we will show that an optimal management scheme can be obtained by solving a linear programming problem.



(a) Dual-battery, single-SP system model (2BAT)



(b) Dual-SP, single-battery system model (2SP)

Fig. 1. Two abstract models of battery-powered, portable systems.

IV. System Modeling

In this section, we will first present stochastic models of each component, i.e., SR, SQ, SP, PS and BAT, in the target electronic system. The models of the SR, SQ and SP are similar to those described in [4] and are only reviewed here. The models of PS and BAT are new. Finally, based on these component models, we build the complete model of the target power-managed, battery-powered system. The notation used in the model description is somewhat complex and cumbersome. To improve readability, we summarize this notation in the appendix.

A. Model of the Components

1. Model of the Service Requestor

The SR is modeled as a stationary, continuous-time Markovian decision process with a state set $\mathbf{R}=\{r_i, i=0,1,2,\ldots,R\}$ and a generator matrix G_{SR}, where R is the number of states of the SR. Each SR state r_i is associated with a request generation rate $\lambda(r_i)$. The notation $\nu_{i,j}$ represents the transition rate from state r_i to state r_j. An example of a two-state SR is shown in Fig. 2.

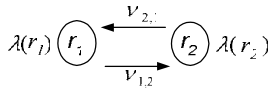


Fig. 2. CTMDP model of a two-state SR.

2. Model of the Service Provider

The SP is modeled as a stationary, continuous-time Markovian decision process with a state set $\mathbf{S}=\{s_i \ s.t. \ i=1,2,\ldots,S\}$, an action set A_s, and a parameterized generator matrix $G_{SP}(a_s)$, where $a_s \in A_s$. The SP can be described by a quadruple $(\chi, \mu(s), pow(s), ene(s_i, s_j))$, where χ is the transition speed matrix of the SP, $\mu(s)$ is the service speed of the SP when it is in state s, $pow(s)$ is the power consumption of the SP in state s, and $ene(s_i, s_j)$ is the energy required by the SP to transit from state s_i to s_j. There are two kinds of transitions: autonomous and command-driven. A command-activated transition may only occur when the SP receives a command from the controller when it asks the SP to make such a transition, e.g., the transition from state *idle1* to state *busy1* in Fig. 3. An autonomous transition is one that takes place without any command from the controller, e.g., the transition from state *busy1* to state *idle1* in Fig. 3 takes place autonomously as soon as the SP finishes the current service. An example of a six-state SP model is illustrated below.

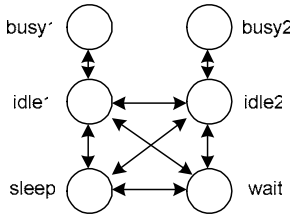


Fig. 3. CTMDP model of the SP.

The expected power consumption (cost rate) of the SP when it is in state s and action a_s is chosen is calculated as:

$$pow_{act}(s,a_s) = pow(s) + \frac{1}{\tau_s^{a_s}} \sum_{\substack{s' \neq s \\ s \in S}} P_{s,s'}^{a_s} ene(s,s')$$

where $\tau_s^{a_s}$ represents the expected time that the SP stays in state s when action a_s is chosen, whereas $P_{s,s'}^{a_s}$ represents the probability that the next state of the SP is s' when its present state is s and action a_s is chosen. $\tau_s^{a_s}$ and $P_{s,s'}^{a_s}$ are calculated as shown next:

$$\tau_s^{a_s} = 1/\sum_{s' \neq s} \sigma_{s,s'}^{a_s}, \ P_{s,s'}^{a_s} = \frac{\sigma_{s,s'}^{a_s}}{\sum_{s' \neq s} \sigma_{s,s'}^{a_s}}$$

where $\sigma_{s,s'}^{a_s}$ represents the transition rate of the SP from state s to state s' when a_s is chosen.

3. Model of the Service Queue

The SQ is modeled as a stationary, continuous-time Markovian decision process with a state set $\mathbf{Q}=\{q_i, i=0,1,2,\ldots,Q\}$ and a generator matrix $G_{SQ}(r,s)$, where Q is the maximum length of the queue, s denotes a state of the SP, and r denotes a state of the SR. When a service request is generated, the state of the SQ is autonomously incremented by one unless the SQ is full. When the SP services a service request, the index of the state of the SQ is autonomously decremented by one.

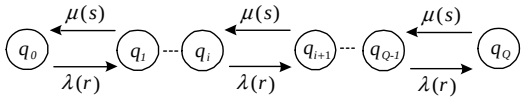


Fig. 4. CTMDP model of the SQ.

4. Model of the Power Switch

The Power Switch (PS) is modeled as a stationary, continuous-time Markovian process, with a state set $\mathbf{W}=\{w_i \; s.t. \; i=1,2,\ldots,W\}$, an action set $A_w=\{a_w(i) \; s.t. \; i=1,2,\ldots,W\}$, and a generator matrix G_{PS}. Here $a_w(i)$ means that the i_{th} battery source should be used next to power the system.

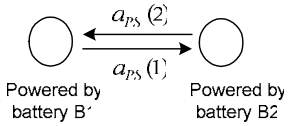


Fig. 5. CTMDP model of the PS.

5. Model of the Battery

The battery (BAT) is modeled as a stationary, continuous-time Markovian decision process with a state set $\mathbf{B}=\{b_i, rs_i\}$ $s.t.$ $i=0,1,2,\ldots,N$, a parameterized generator matrix $G_B(s,w,b)$, and a function $ene(b_i,b_j):N\times N\rightarrow R$.

The subscript i of state b_i in the state set **B**, denotes that in this state the remaining energy capacity of the BAT is $i/N\times 100\%$ of the full energy capacity. Therefore, b_0 implies that the battery has been completely discharged whereas b_N means that the battery is fully charged. State rs_i is the corresponding "stop recovery state" for state b_i. Function $ene(b_i,b_j)$ represents the battery energy-capacity difference between state b_i and b_j. Fig. 6 illustrates the CTMDP model of the BAT.

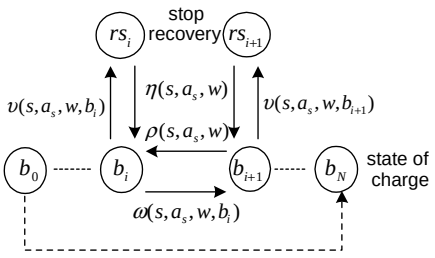


Fig. 6. CTMDP model of the BAT.

In this model, state b_i represents an "active" state, in which the battery may be discharged when it is used or can recover capacity when it is resting. State rs_i represents a "stable" state, in which neither the battery discharges; nor does it do any capacity recovery.

The transition from state b_{i+1} to b_i represents the discharge process of the battery. $\rho(s,a_s,w)$ denotes the corresponding transition rate. Notice that if the PS selects a battery and SP consumes power when it is in state s and action a_s is chosen, the value of $\rho(s,a_s,w)$ is calculated by equation (4-2) (cf. Table 1); otherwise, $\rho(s,a_s,w)$ is equal to 0.

The transition from state b_i to b_{i+1} represents the recovery process of the battery. $\omega(s,a_s,w,b_i)$ denotes the transition rate, which is a function of the SP state s and the battery state b_i. If the SP does not consume power when it is in state s and action a_s is chosen, or if the PS does not select the battery in question, then the value of $\omega(s,a_s,w,b_i)$ is determined by the battery state b_i (cf. Table 1); otherwise $\omega(s,a_s,w,b_i)$ is equal to 0. The transition from state b_i to rs_i can only occur when the battery is resting. $\upsilon(s,a_s,w,b_i)$ denotes the transition rate. If the SP does not consume power when it stays in state s and action a_s is chosen, or if the PS does not select this battery, then the value of $\upsilon(s,a_s,w,b_i)$ is determined by the battery state b_i (cf. Table 1); otherwise $\upsilon(s,a_s,w,b_i)$ is equal to 0. The transition from state rs_i to b_i may only occur when the battery is used again. Here $\eta(s,a_s,w)$ means that if the SP consumes power when it stays in state s and action a_s is chosen, and if the PS selects this battery, the battery goes from rs_i to b_i immediately.

The values of $\rho(s,a_s,w)$, $\omega(s,a_s,w,b_i)$, $\upsilon(s,a_s,w,b_i)$ and $\eta(s,a_s,w)$ are summarized in the following table. In the first row of Table 1, if the SP consumes power when it stays in state s and action a_s is chosen, (s,a_s) is set to 1; otherwise (s,a_s) is set to 0. If the PS selects this battery, w is set to 1; otherwise w is set to 0.

TABLE 1
PARAMETERS OF THE BAT

(s,a_s), w	0, 0	0, 1	1, 1	1, 0
$\rho(s,a_s,w)$	0	0	$\rho'(s,a_s)$ *	0
$\omega(s,a_s,w,b_i)$	$\omega'(b_i)$ **	$\omega'(b_i)$	0	$\omega'(b_i)$
$\upsilon(s,a_s,w,b_i)$	$\upsilon'(b_i)$ **	$\upsilon'(b_i)$	0	$\upsilon'(b_i)$
$\eta(s,a_s,w)$	0	0	∞	0

* $\rho'(s,a_s)$ denotes the transition rate corresponding to the battery discharge process. It will be precisely defined in section 4.B.5.a.

** $\omega'(b_i)$ and $\upsilon'(b_i)$ are functions defined by corresponding look-up tables indexed by b_i. The actual value of each entry in these tables is obtained from simulation results. The method is described in more detail in section 4.B.5.b.

The transition from state b_0 to b_N, denoted by the long wrap-around dashed arrow line, represents that an exhausted (fully discharged) battery is replaced with a fresh (fully charged) battery of the same type. This transition is added because without it, state b_0 becomes a trap. If transition from b_0 to b_N is not included in the model, then the battery will eventually arrive in state b_0 and cannot subsequently leave this state. Consequently, no feasible solution would be found when using the linear programming technique to solve the optimal policy problem.

The battery model is constructed based on the following three assumptions:

(a) During the discharge process of the battery, if the battery is in state b_{i+1}, only the transition from state b_{i+1}

to b_i is allowed, where i=0, 1, …, N-1,, which means that the battery discharges gradually.

(b) When the battery is resting (i.e. it is not being used), if it is in state b_i, then it may regain some of its capacity due to the recovery process or it may transit to state rs_i. When the battery is in state rs_i, it cannot recover capacity and will continue to remain in this state until it is used again to power up the system. As soon as this happens, the battery moves from state rs_i to b_i and then possibly to b_{i-1}, etc.

(c) During the recovery process of the battery, only a transition from state b_{i-1} to b_i, i=2,…,N, is allowed, which means that the battery recovers capacity gradually. State b_0 means that the battery capacity has been exhausted and the battery must be replaced.

Assumptions (a) and (c) are valid because of the continuous nature of the electrochemical processes. Assumption (b) is accurate because the energy recovery speed of a battery diminishes when the resting time increases. The simulation results depicted in Fig. 8 (in Section 4.B.5.a) empirically confirms this important observation. The data pointed by the markers are obtained by simulating an industrial Li-ion battery with a low-level battery simulator, DUALFOIL [7].

a. Determining $\rho'(s,a_s)$

As stated previously, $\rho'(s,a_s)$ represents the transition rate of the battery from state b_i to b_{i-1}, i=1,…,N, when the SP stays in state s and action a_s is chosen. It can be calculated as

$$\rho'(s,a_s)=\frac{pow_{act}(s,a_s)}{(1-\beta(s,a_s))\cdot C/N} \qquad (4\text{-}2)$$

where C is the nominal full energy capacity of the battery and $\beta(s,a_s)$ captures the rate-capacity characteristic of the battery, $0<\beta(s,a_s)<1$. The SP state s, and the selected action a_s, determine the electrical current drawn from the battery, i.e., determine the value of $\beta(s,a_s)$. As seen in Fig. 7, due to the rate-capacity effect, the deliverable capacity of a battery can be quite different under different discharge currents. In the figure, this is more pronounced for battery B1, therefore, $\beta(s,a_s)$ may assume very dissimilar values for different batteries.

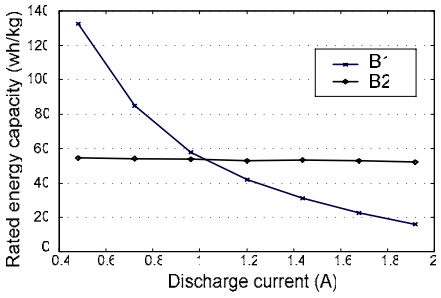


Fig. 7. Current-capacity relations of two different batteries.

b. Determining $\omega'(b_i)$ and $\upsilon'(b_i)$

Transition rates $\omega'(b_i)$ and $\upsilon'(b_i)$ can be obtained from battery simulation results. Founded on assumption (b), let $r_i(t)$, i=1,…,N, denote the expected recovered capacity during the time period from 0 to t, where 0 denotes the time instance when the battery begins its resting period, (during the period, the recovery process is not interrupted by the discharge requests, i.e., the battery is not selected by the PS to power the SP.) Furthermore, suppose the battery starts in state b_i at time instance 0. We define an $N\times1$ vector $\mathbf{r}(t)=[r_1(t)\ \ r_2(t)\ \ \dots\ \ r_N(t)]^T$, which satisfies the following equation:

$$\dot{\mathbf{r}}(t)=\mathbf{A}\mathbf{r}(t)+\mathbf{d}$$

where $\mathbf{A}$ is an $N\times N$ matrix with entries

$$a_{ij}=\begin{cases}-(\omega'(b_i)+\upsilon'(b_i)) & i=j\neq N\\ \omega'(b_i) & j=i+1,\ i\leq N-1\\ 0 & otherwise.\end{cases}$$

$\mathbf{d}$ is an $N\times1$ vector with entries

$$d_i=\begin{cases}\omega'(b_i)\cdot C/N & i\leq N-1\\ 0 & i=N.\end{cases}$$

Note that dot operator denotes differentiation with respect to t and C denotes the nominal full energy capacity of the battery (cf. equation (4-2).)

The boundary condition is:

$$\mathbf{r}(0)=0 \qquad (4\text{-}3)$$

$\omega'(b_i)$ and $\upsilon'(b_i)$ can be determined in a top-down manner as described next.

Since b_N represents a state of full-charge capacity, $r_N(t)\equiv0$. Thus $r_{N-1}(t)$ satisfies:

$$\dot{r}_{N-1}(t)=-(\omega'(b_{N-1})+\upsilon'(b_{N-1}))r_{N-1}(t)+\frac{C}{N}\omega'(b_{N-1}) \qquad (4\text{-}4)$$

After applying the boundary condition, we obtain the following solution:

$$r_{N-1}(t)=\frac{C}{N}\frac{\omega'(b_{N-1})}{\omega'(b_{N-1})+\upsilon'(b_{N-1})}(1-\mathrm{e}^{-(\omega'(b_{N-1})+\upsilon'(b_{N-1}))t}) \qquad (4\text{-}5)$$

We perform battery simulation as follows: we discharge the battery to (N-1)/N of its original capacity, let it rest for a time period t, then fully discharge the battery. Next, we change the value of t and repeat the above procedure. Proceeding in this way, we obtain the curve for the recovered capacity vs. the resting time when the battery starts in state b_{N-1}. We then choose $\omega'(b_{N-1})$ and $\upsilon'(b_{N-1})$ that force the curve determined by equation (4-6) to match the simulation curve. Since $r_{N-1}(t)$ is known, we can solve for $r_{N-2}(t)$ and determine $\omega'(b_{N-2})$ and $\upsilon'(b_{N-2})$ by using the same technique as in the case of $\omega'(b_{N-1})$ and $\upsilon'(b_{N-1})$. We repeat this procedure until $r_1(t)$ is obtained and $\omega'(b_1)$ and $\upsilon'(b_1)$ are determined.

The result of using our model to fit a DUALFOIL battery which simulates the commercial Bellcore technique is presented in Fig. 8. The markers represent the simulation results of battery capacities delivered at different discharge rates and with different resting time ratio. The solid colored curves reflect the fitting results by using our approach. This figure demonstrates that our model describing the recovered capacity of the battery as a function of the resting time t is very accurate when compared with detailed battery simulation results.

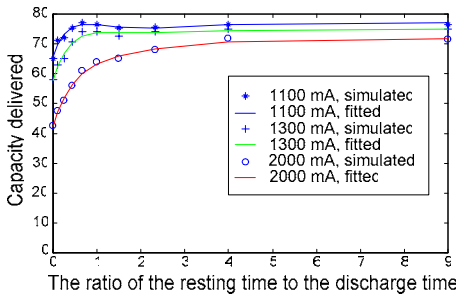


Fig. 8. Relationship between the capacity recovery effect and ratio of the resting time to the discharge time for a Li-ion battery.

A function $ene(b_i, b_j)$ is associated with each pair (b_i, b_j). This function is defined as follows. $ene(b_i, b_j)=\pm C/N$, where $j=i\pm1$. When $j=i-1$, $ene(b_i, b_j)>0$, represents the energy which is consumed when the battery transits from state b_i to b_j. When $j=i+1$, $ene(b_i, b_j)<0$, represents energy which is recovered in the battery due to the battery relaxation process.

Notice that for a two-battery system, the generator matrix of the two-battery model is calculated by $G_B=G_{B1}\otimes G_{B2}$, where $\otimes$ is the *tensor product* of the two generator matrixes of batteries B1 and B2 [25].

B. Model of the Battery-Powered System

In this section, we show in detail how to construct a stochastic model of the 2BAT system. The stochastic model of the 2SP system can be developed similarly and is omitted here to avoid duplication of details.

We use five components: SR, SQ, SP, PS, and BAT models. The state set is given by $X=R\times Q\times S\times W\times B-\{invalid\ states\}$. The invalid states include the states where the SP is busy and the SQ is empty. Thus the state of the complete battery-powered system (SYS) can be represented as a quintuple $x=(r,q,s,w,b)$, where $b=\{b^{(1)},b^{(1)}\}\in B$, $b^{(1)}\in B_1$, and $b^{(2)}\in B_2$ represent the state of battery B1 and B2 , respectively.

The system action set A_{sys} is the union of the action set A_s for the SP and the action set A_{ps} for the PS. We use $G_{SYS}(a)$ to represent the generator matrix of the system, where $a\in A_{sys}$. Since the service requester is assumed to be independent of the other components, the generator matrix $G_{SYS}(a)$ can be calculated as:

$$G_{SYS}(a)=G_{SR}(a)\otimes G_{SQ-SP-PS-BAT}(a)$$

Similarly, independence of the SP and the PS results in:

$$G_{SP-PS}(a)=G_{SP}(a)\otimes G_{PS}(a)$$

where the SQ-SP-PS-BAT denotes the joint CTMDP model of the SQ, SP, PS, and BAT, and the SP-PS denotes the joint CTMDP model of the SP and PS. Unfortunately, the Markovian processes of the SQ and the SP-PS, and the Markovian processes of the BAT and the SP-PS are both correlated. The SP-PS and the Battery are correlated in the sense that when the state of the SP-PS changes, the discharge rate of the Battery also changes. We therefore specify how to calculate each entry of the $G_{SQ-SP-PS-BAT}(a)$ below.

Let $\sigma_{x,x'}$ denote the transition rate of the system for going from state $x=(q,s,w,b)$ to $x'=(q',s',w',b')$, where $b=\{b^{(1)},b^{(1)}\}$ and $b'=\{b^{(1)}{}',b^{(1)}{}'\}$.

1). if $b'=b$, then $\sigma_{x,x'}$ is equal to $\sigma_{(q,s,w),(q',s',w')}$, which is the joint state transition rate of the SQ-SP-SW.

2). if $s'=s$ and $q'=q$, then

A). if $w=w_1$ and $b^{(1)}=b_i$ and $b^{(1)}{}'=b_{i-1}$ and $b^{(2)}{}'=b^{(2)}$, then $\sigma_{x,x'}$ is equal to $\rho_{B1}(s,a)$, which is the discharge transition rate of battery B_1 from state b_i to state b_{i-1}.

B). if $w=w_2$ and $b^{(2)}=b_i$ and $b^{(2)}{}'=b_{i-1}$ and $b^{(1)}{}'=b^{(1)}$, then $\sigma_{x,x'}$ is equal to $\rho_{B2}(s,a)$, which is the discharge transition rate of battery B_2 from state b_i to state b_{i-1}.

C). if $w=w_2$ or the SP is in the sleep state, then

a). if $b^{(1)}=b_i$ and $b^{(1)}{}'=b_{i+1}$ and $b^{(2)}{}'=b^{(2)}$, then $\sigma_{x,x'}$ is equal to $\omega_{B1}(s,a)$, which is the recovery transition rate of battery B_1 from state b_i to state b_{i+1}.

b). if $b^{(1)}=b_i$ and $b^{(1)}{}'=rs_i$ and $b^{(2)}{}'=b^{(2)}$, then $\sigma_{x,x'}$ is equal to $\upsilon_{B1}(s,w)$, which is the transition rate of battery B_1 from state b_i to state rs_i.

D). if $w=w_1$ or the SP is in the sleep state, then

a). if $b^{(2)}=b_i$ and $b^{(2)}{}'=b_{i+1}$ and $b^{(1)}{}'=b^{(1)}$, then $\sigma_{x,x'}$ is equal to $\omega_{B2}(s,a)$, which is the recovery transition rate of battery B_2 from state b_i to state b_{i+1}.

b). if $b^{(2)}=b_i$ and $b^{(2)}{}'=rs_i$ and $b^{(1)}{}'=b^{(1)}$, then $\sigma_{x,x'}$ is equal to $\upsilon_{B2}(s,w)$, which is the transition rate of battery B_2 from state b_i to state rs_i.

3). For any other transition, $\sigma_{x,x'}$ is equal to 0.

V. A MATHEMATICAL PROGRAMMING SOLUTION

The expected cost $\gamma_x^{a_x}$, which represents the expected energy delivered from the battery when the system is in state x and action a_x is chosen, is calculated as:

$$\gamma_x^{a_x}=\sum_{x'}P_{x,x'}^{a_x}\cdot ene(b,b')$$

Let $f_x^{a_x}$ denote the frequency that the system will be in state x and action a_x is chosen. Notice that $f_x^{a_x}\gamma_x^{a_x}$ is the expected power that the system expends in state x as a result of action a_x. Let $\tau_x^{a_x}$ denote the expected time that the system will stay in state x when action a_x is chosen. Then, $f_x^{a_x}\tau_x^{a_x}$, called *state-action probability*, is the probability that the system is in state x and action a_x is taken in a random observation. Let $lq_x^{a_x}$ denote the waiting cost in the queue. This cost can be calculated as $lq_x^{a_x}=q_x\cdot\tau_x^{a_x}$.

The goal is to find an optimal policy for minimizing the energy delivered from the batteries under constraints on the average number of waiting requests in the queue and the request loss rate. Notice that a request issued by the SR is lost (dropped) in the SQ if the queue is full when the request

comes in. We formulate this problem as a linear program (LP) as follows:

$$Minimize_{f_x^{a_x}} \sum_x \sum_{a_x} f_x^{a_x} \gamma_x^{a_x} \quad (5\text{-}2)$$

subject to

$$f_x^{a_x} \geq 0 \quad (5\text{-}3)$$

$$\sum_x \sum_{a_x} f_x^{a_x} \tau_x^{a_x} = 1 \quad (5\text{-}4)$$

$$\sum_{a_x} f_x^{a_x} - \sum_{x' \neq x} \sum_{a_{x'}} f_{x'}^{a_{x'}} P_{x',x}^{a_{x'}} = 0, \ \forall x \in X \quad (5\text{-}5)$$

$$\sum_x \sum_{a_x} f_x^{a_x} l q_x^{a_x} \leq D \quad (5\text{-}6)$$

$$\sum_x \sum_{a_x} f_x^{a_x} \tau_x^{a_x} \delta(q_x^{a_x}, Q) \leq P_{req\_block} \quad (5\text{-}7)$$

where $\delta(x, y) = \begin{cases} 1, & if\ x = y; \\ 0, & otherwise. \end{cases}$

An optimal policy is an assignment of the values of the variables $f_x^{a_x}$ such that the objective function is minimized and the constraints (5-3) to (5-7) are satisfied. These constraints can be explained as follows. The inequality (5-3) is implicit in the definition of the variable $f_x^{a_x}$. This is because $f_x^{a_x}$ is a frequency, which takes a nonnegative value. Equation (5-4) is a normalizing constraint which sets the summation of all *state-action probabilities* equal to one. It is known that if a Markovian process is stationary, then the input rate of each state will be equal to the output rate of that state (cf. equation (5-5).) The summation on the left-hand side of inequality (5-6) gives the expected number of the service requests waiting in the SQ. From Little's law [24], for a stationary process, this number is equal to the product of the average incoming rate of the requests and the expected delay experienced by a request. Inequality (5-6) limits the expected service delay seen by an incoming service request to *D*. Inequality (5-7) ensures that the probability that the queue becomes full is less than a preset threshold. This is our way of controlling the request loss rate in the system. For an interactive application, this probability may be viewed as a retransmission rate where a service request will be re-delivered (rather than dropped) if it arrives when the SQ is full. Constraints (5-6) and (5-7) capture QoS requirements for real-time applications. For a specific application, they can be enforced simultaneously or individually according to the user's requirement. Note that one may even choose other QoS criteria, such as variation in the service delay of incoming requests.

According to the proposed modeling technique, from any state, the system can only transit to a small number of other states. This feature makes the constraint matrix constructed from equations (5-5) a sparse matrix. The formulated LP problem is thus efficiently solved by LP solvers that make use of the sparse matrix calculus supporting operations such as sparse Cholesky factorization and multiple minimum-degree ordering. In particular, we use a Matlab-based software package, LIPSOL, which is a primal-dual interior-point method based on the predictor-corrector algorithm proposed by Mehrotra [26]. LIPSOL has a worst-case timing complexity of $O(n^{3.5})$, where n is the number of inequalities in the linear program. A typical system with a battery may have about 10 buffer slots for storing the incoming requests, 3 to 6 power states, and 5 to 20 battery states depending on the battery features. Thus totally a system may have around 800 states and 1600 actions. In practice, LIPSOL's performance is often much better than this worst-case analysis, especially for sparse linear programs. Experimental results related to the computational complexity are presented in section 6.D.

We propose a runtime approach for implementing the BAPM algorithm as follows. Its implementation requires support from both the system software and hardware. More precisely, the following hardware support is needed: 1) the system supports multiple working modes (i.e., power states); 2) A "smart" battery source is employed, which is capable of providing the required battery data (discharge rate, state-of-the-charge, etc.) online. The power manager is implemented as part of the system software. The block diagram of the BAPM in the Linux operating system is shown in Fig. 9. BAPM obtains information about the task generation rate from the scheduler and the battery data (voltage level, current, internal impedance) from the battery data interface. This info is then used by the battery state analyze to calculate the battery's state of charge info. The policy manager looks up a pre-designed and cached policy table to determine the optimal power management policy, and thereby, set the states of the system components and battery switch through appropriate device drivers. Notice that the BAPM policies are themselves computed *off-line*, yet the appropriate policy is chosen and applied based on runtime information.

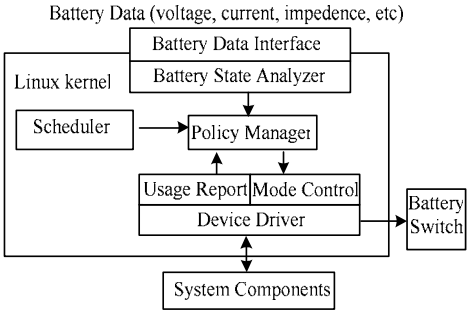


Fig. 9. Block diagram of the BAPM power manager implementation in Linux.

VI. Simulation Results

In this section, we present experimental results for the 2BAT and 2SP systems described in section 3. Experiments focus on demonstrating the effectiveness of the proposed approach in scheduling batteries for optimum energy discharge and in shaping the discharge rate of each battery, respectively. Note that this does not mean that the battery management and discharge-rate shaping are dealt with separately in our proposed approach. Indeed, in either case, the optimal power management policy, which is obtained by solving a mathematical program, determines the activities of the SPs. This in turn affects the discharge current profile as

well as the scheduling of the batteries, and thus, takes advantage of both techniques. In the third part, we show the results obtained by applying this technique to a real system.

In the experiments, we use an input trace file to realistically represent the statistical behavior of the SR. In particular, the distribution of the input requests is a combination of the exponential and Pareto distributions as observed in [5]. An augmented version of the low-level simulator called DUALFOIL [7] is used to simulate the batteries. The original DUALFOIL software only supports simulation in the batch mode, where the complete battery discharge profile is known before the simulation starts. We have modified the source code of this software program to enable interactive battery simulation. As a result, the modified DUALFOIL accepts the discharge requirement in every step and returns the battery state of charge data as output.

A. Simulations for the 2BAT System

It has been demonstrated that DPM techniques based on Markovian decision processes outperform heuristic policies (see [22].) We seek to evaluate and assess the effectiveness of the stochastic power management techniques on the system lifetime when the battery characteristics are taken into account. More precisely, to compare the effects of different power management policies on the battery service lifetime, in this experimental setup, we use the CTMDP-based optimal policy derived in [4] to determine the behavior of the SP under a number of heuristic methods, denoted by M1-M4 (see below.) Notice that none of these heuristic methods account for the battery effects as part of solving an integrated battery-aware power management problem.

As shown in Figure 1(a), the experimental system consists of an SR, an SP with its own SQ, and two batteries. The SP has six power states: {*busy1*, *busy2*, *idle1*, *idle2*, *wait*, *sleep*}. The *busy1* and *busy2* states are functional states where the SP services the requests waiting in the queue. In the *wait* or *sleep* states, the SP does not service any requests. The key differences between these two states are: 1) in the *wait* state, the SP consumes higher power than in the *sleep* state; 2) in the *wait* state, the SP can return to a working state much faster than in the *sleep* state. There is one idle state for each busy state. In fact, idle states are abstract states where new policy decisions are issued to the SP. Transition from the busy to idle state is autonomous and instantaneous. Since the DUALFOIL accepts current density as an input, in this experiment, we express χ, $pow(s)$, $ene(s_i, s_j)$ in terms of the current.

$$pow = [0.9 \quad 1.6 \quad 0.9 \quad 1.6 \quad 0.3 \quad 0] \quad (unit: A),$$

$$\chi = \begin{bmatrix} \infty & 0 & 0.2 & 0 & 0 & 0 \\ 0 & \infty & 0 & 0.33 & 0 & 9 \\ \infty & 0 & \infty & 1.68 & 1 & 0.5 \\ 0 & \infty & 1.68 & \infty & 1 & 0.5 \\ 0 & 0 & 0.454 & 0.454 & \infty & 1.5 \\ 0 & 0 & 0.166 & 0.166 & 1.5 & \infty \end{bmatrix}, \quad \underset{(unit: A \cdot s)}{ene} = \begin{bmatrix} 0 & \infty & 0 & \infty & \infty & \infty \\ \infty & 0 & \infty & 0 & \infty & \infty \\ 0 & \infty & 0 & 0.017 & 0.056 & 0.11 \\ \infty & 0 & 0.017 & 0 & 0.056 & 0.11 \\ \infty & \infty & 0.25 & 0.25 & 0 & 0.037 \\ \infty & \infty & 1.69 & 1.69 & 0.51 & 0 \end{bmatrix}.$$

The two batteries have different rate-capacity characteristics and recovery abilities. From Fig. 7, we can see that in the low current functional state, busy1, battery B1 can deliver more energy than B2, while in high current functional state, busy2, battery B2 can deliver more energy than B1. Fig. 10 shows that battery B1 exhibits a much stronger capacity recovery ability compared to battery B2.

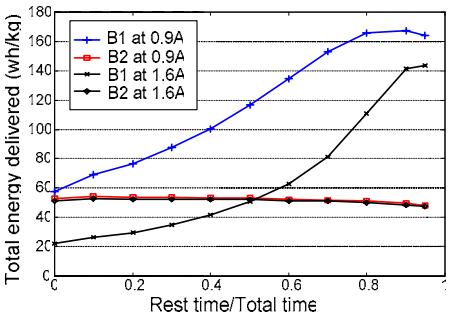


Fig. 10. Recovery abilities of battery B1 and battery B2.

We consider and compare four heuristic methods of battery management with our CTMDP-based policy (called BAPM, which stands for *Battery-aware Power Management*):

M1: As in [15], we account for the rate-capacity characteristics of the battery, but do not consider the recovery effect. In addition, we assign a pre-assigned battery when the SP is in a particular state, e.g., we use battery B1 when the SP is in state *busy1*, while we use battery B2 when the SP is in state *busy2*.

M2: As in [20], we account for the recovery effect in battery, but do not consider the rate-capacity characteristics of batteries. In addition, we switch between the two batteries (B1-B2) with a fixed frequency (0.1 Hz, as suggested in [20].) This is identical to the open-loop policy of [20].

M3: As in [20], we set a voltage threshold V_{th}. When the battery output voltage is larger than V_{th}, SP always works in *busy2* state. After the voltage drops below, SP use *busy1* state instead. In addition, we switch between the two batteries (B1-B2) with a fixed frequency (0.1 Hz). This policy is the hybrid switching policy of [20].

M4.1: We use two batteries of type B1, switching between them with a fixed frequency (0.1 Hz).

M4.2: We use two batteries of type B2, switching between them with a fixed frequency (0.1 Hz).

Furthermore, we consider two battery replacement policies:

P1: As soon as a battery is completely consumed, it is immediately replaced with a new battery of the same type.

P2: Both batteries are replaced together and only after both of them have been completely used up. If only one battery is used up, the other battery will be used continuously until it is also exhausted.

A two-state SR models are used in the simulation. In the first simulation, the SR parameters are: $\lambda(r_1)$=0.25, $\lambda(r_2)$=0, $\nu_{1,2}$=1/200, $\nu_{2,1}$=1/20. Actually the transition time from state r_1 to r_2 complies with Pareto distribution with shape parameter a=1.11 and scale parameter b=2. SR waiting number constraint D is set to 1.5. To meet this timing constraint, M3 policy has to set V_{th}=3.0, which means M3 is

identical to M2 in this case. As demonstrated in Fig. 16, BAPM provides as much as 16.9% improvement over the heuristic methods. In the second simulation, $\lambda(r_1)$ is changed to 0.2. The V_{th} equals to 3.1 for the M3 policy. Other parameters are the same. The simulation results are also shown in Fig. 16.

B. Simulations for the 2SP System

In this experimental setup, the system contains an SR, an SQ, two SPs, and a battery, as shown in Figure 1(b). The two SPs are identical and work in parallel and independently. Each SP can provide service for the incoming SRs. After an SR is serviced by any of the two SPs, it will leave the system. At any time instant, the total discharge-rate of the battery is the sum of the currents drawn by the SPs.

An SP has four states {*busy*, *idle*, *wait*, *sleep*}. The definitions of the states are the same as those in section 4.A.2. The parameters of the SPs are listed below, which are obtained from [22], and correspond to a Fujitsu hard disk:

$$pow = [0.43 \quad 0.19 \quad 0.07 \quad 0.026] \ (unit: A),$$

$$\chi = \begin{bmatrix} 0 & 12.5 & 0 & 0 \\ \infty & \infty & 2.5 & 1.49 \\ 0 & 0.833 & \infty & 3.33 \\ 0 & 0.625 & 1.67 & \infty \end{bmatrix}, \quad ene_{(unit: A \cdot s)} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0.0002 & 0.0134 \\ 0 & 0.4 & 0 & 0.006 \\ 0 & 1.2 & 1.02 & 0 \end{bmatrix}$$

BAPM is compared with the following three power management policies:

M5: This policy is a combination of the power-regulation method of [19] and the time-out policy. More precisely, we adopt a time-out policy for the power management of each SP and set a hard threshold P_{th} to limit the total discharge current.

M6: This policy is a modified version of M5 except that instead of restricting the discharge current to be less than the threshold P_{th} at all times, an instantaneous current exceeding the threshold is allowed. Pr_{th} denotes the acceptance probability.

M7: The policy is the optimal one derived from the CTMDP model of the 2SP system without considering the battery characteristics.

A battery of type B1 is used as the power source in this experiment. Experimental results are reported in Fig. 17. The task loss (request block) rate P_{block_req} is set to be less then 0.1%. This threshold value is chosen because it is in fact mentioned in many network service level agreements. M7 and BAPM both satisfy the target loss rate bound of 0.1%. However, in the case of the M5 and M6 policies which do not consider timing constraints, the actual task loss rates vary significantly. For example, when the average task generation interval is 0.2 s, it ranges from 0.01% (M5, P_{th} =1.0A) to 0.36% (M5, P_{th} =0.8A).

This figure shows that BAPM achieves an improvement of battery lifetime over the reference methods by 17% averaged over the 8 experiments. Furthermore, from, we can see that as compared to the heuristic methods, BAPM are more flexible to satisfy user's multi-dimensional design requirements and achieve desired performance.

An example of the battery discharge rate distributions under different policies, for the case where the average task delay is 0.4s in Fig. 17(a), is shown in Fig. 11. From a comparison of Fig. 11(a) and (b), we conclude that, compared to M7, BAPM reduces the variance of the discharge current by reducing the duration of time that the battery is providing very high discharge current. This can be seen by noting that the M7 policy results in a battery discharge current of greater than 1.3A about 1% of the time and greater than 0.6A about 44% of the time whereas the BAPM policy results in no discharge current greater than 0.9A and about 33% of the time with a discharge current greater than 0.6A. From Fig. 11(c), it can be seen that M5 (or M6) can indeed limit the maximum discharge current, but they are incapable of shaping the current profile while meeting the given threshold P_{th}. These figures and the table illustrate that by scheduling activities of the two SPs and shaping the total current draw to match the battery characteristics, BAPM is capable of simultaneously minimizing the average discharge rate and the variance of current profile. This is a major advantage of BAPM over the reference methods. The discharge profiles of the battery in a 2SP system under M5-M7 and BAPM policies are shown in Fig. 12. The battery cut-off voltage was set to 3.0V [7].

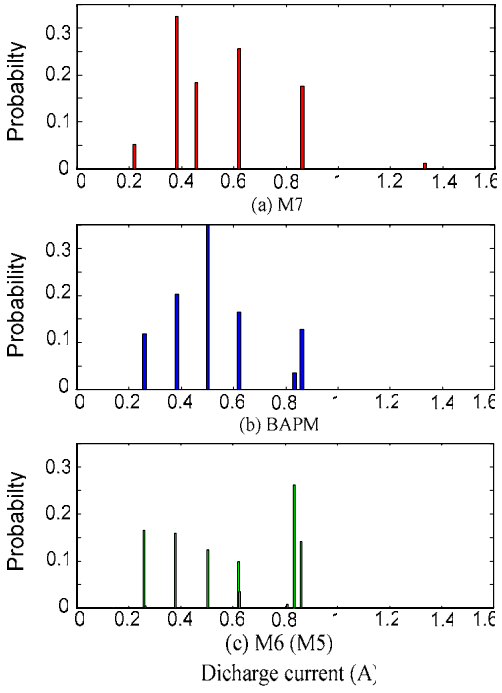


Fig. 11. Discharge current distributions with different policies.

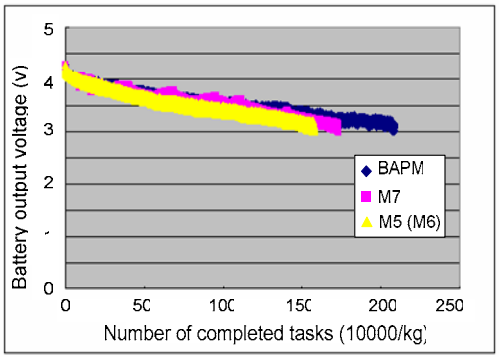


Fig. 12. Discharge profiles of the 2SP system model under different policies.

C. *Simulations based on the BitsyX System*

In this simulation, we consider an application scenario where a mobile computer collects and stores batches of sampled raw data for subsequent processing. A desktop workstation continuously sends requests to the mobile computer to process the raw data and return the result. A BitsyX system [27] with a Socket low-power Compact-Flash WLAN card [28] was used as the mobile computer. The BitsyX system supports a low power sleep mode and has multiple working modes corresponding to different CPU voltage/frequency settings. For this experiment, the BitsyX system was configured to allow the WLAN card to be powered up during the sleep mode. More precisely, during the sleep mode, if a valid data packet is received from the wireless link, the BitsyX system will be awakened by an interrupt signal, which is generated by the WLAN card. A software program that performs pattern recognition was used as the data processing algorithm. The hardware-measured average power dissipation and average data processing times of the BitsyX in different working modes are given in Table 2. The averaging is done over different input data as well as over time for a given input data. More precisely, we calculate the mean value of power dissipation of the application running in state y of Bitsy X and average this over 100 different data values to generate the corresponding entry in the table. In addition, this table provides the average transition times between the various working modes. We next set up a simulation experiment in which the mobile computer is powered by a simulated battery based on the commercial Bellcore battery technology [28]. The simulated battery has a c-rate of 0.962A. We assumed that the Smart-Battery technology is available so that the battery state can be probed online. The workstation runs the BAPM policy and determines the operation mode of the mobile computer based on the battery state of charge that is reported by a concurrently running battery simulator.

TABLE 2.
MEASURED POWER AND TRANSITION TIMES OF THE BITSYX

Operation mode	CPU/Bus frequency setting	Average power (mW)	Average program running time (S)
Busy1	100MHz/100MHz	1060	0.89
Busy2	200MHz/100MHz	1390	0.64
Busy3	400MHz/100MHz	2190	0.38
Idle	100MHz/100MHz	801	--
Sleep	--	96	--
The frequency adjusting time (between *Busy1*, *Busy2* and *Busy3*)		0.5 ms	
The wake-up time (between Sleep to *Idle*)		2.0 s	
The go-sleep time (between *Idle* to *Sleep*)		0.3 s	

In this simulation, the service queue has 20 slots to hold the waiting tasks. The battery has 18 states of charge. In the following figures, the BAPM policy is compared with three time-out policies and a CTMDP policy which does not consider the battery characteristics (M7). Each time-out policy is named Timeout-Busy*x* for *x*=1,2,3. This in turn means that the operation mode *Busy1, Busy2, or Busy3* is used when processing the raw data. The timeout threshold value is adjusted to meet different service delay requirements. The simulation results are presented in Fig. 18. In this figure, each time-out policy is presented by three points. The leftmost point corresponds to an infinite time-out threshold value whereas the rightmost point corresponds to a zero threshold value.

From Fig. 18, it is observed that BAPM always results in the longest battery lifetime for a given task delay constraint. Time-out policies do not work well when the required delay for processing a task is rather large, because these policies are incapable of using the rates of task generation and the delay constraints, and therefore, tend to waste a significant amount of energy by waking up the system and then shutting it off. M7 is an optimal policy in terms of the total energy consumption, but it does not consider the battery characteristics. For example, in the case where the average task generation interval is 1.2s and the average delay is 9.6s, the M7 policy prefers the *Busy3* mode to the other two busy states, because the *Busy3* mode consumes the least amount of energy to perform the task. However, the *Busy3* mode also requires the largest current to be drawn from the battery source which subsequently reduces the deliverable capacity of the battery. The recovery effect which allows the battery to regain capacity during the sleep mode also affects the operation mode selection when calculating the BAPM policy. It makes a trade-off between the sleep time and the discharge current of the working modes. For example, for the same that was mentioned above, if the number of waiting tasks is larger than 11 and the battery state of charge is 50% of the total capacity, then the BAPM policy will select *Busy2* as the working mode instead of *Busy1*. However, if the recovery effect is not considered, the *Busy2* mode will never be used. An example of the energy dissipation rate distributions under different policies is depicted in Fig. 13, where the average task generation interval is 1.2s and delay is 9.6s. Notice that for the timeout police, a threshold value that maximizes the battery lifetime is applied.

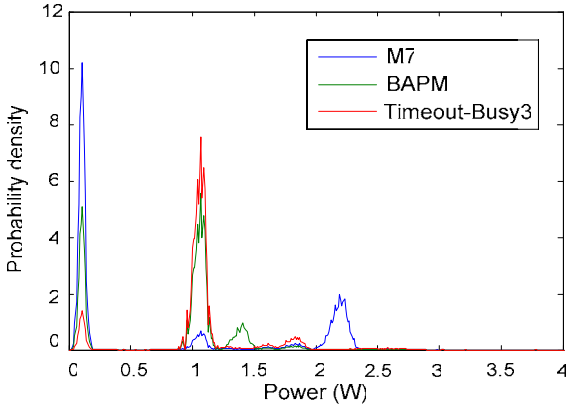


Fig. 13. Energy dissipation rate distributions with different policies.

D. *Effect of the State Count of the Battery*

The number of charging states of the battery impacts the lifetime that can be achieved by the BAPM policy. The increase in the state count of the battery tends to produce a more accurate description of the battery's characteristics and a more fine-grained control policy. However, this benefit is acquired at the cost of the increased policy computation time and a larger memory space for storing the policy table. In this section, we present experimental results to evaluate the

influence of the number of battery states on the quality of solutions that are obtained by BAPM.

The simulation in section 6.C is repeated by changing the state count of the battery model. Here, the size of the service queue is 20, the average task generation interval is set to 1.2s, and the average task latency (delay) is set to 9.6s. The simulation results are shown in Fig. 14. When the number of the states of the battery is small, the lifetime that is achieved by BAPM is rather low mainly because of our inability to properly capture the battery characteristics with only two states. This can be achieved by using a larger number of battery states, say a number between 3 and 10. Increasing the number of the battery states above 10 only results in a marginal improvement in the battery lifetime.

The policy computation time is measured by using a PIII 800-MHz PC machine. For example, as shown in the figure, when the battery model has 10 states, the computation time is about 37 seconds. Notice that for this example, the whole system model has 1200 states and 2534 actions.

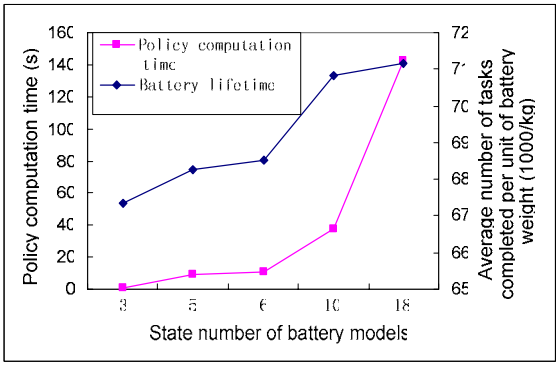


Fig. 14. Effect of the number of battery states on the BAPM policies for a system powered with a single battery.

To assess the effect of the state count of the battery on a system which is powered by two different batteries, we repeat the BAPM simulation with battery replacement policy P2 as presented in section 6.A, where the simulation parameters as in Fig. 16 are used. In addition, $\lambda(r_1)$=0.25. Since the recovery effect of battery B2 is almost negligible, we can model it by using a two-state battery model. However, B1 exhibits a strong recovery effect, and therefore, it is advisable to use a battery model with a large number of states of charge. To avoid excessive computation burden that is associated with the large number of battery states of charge, we use a battery model with non-uniformly distributed battery states. Generally speaking, fine-grained control is more helpful when a battery gets closer to its zero remaining charge state. Thus, for the battery model of B1, from 0 to 100 percentage point for the remaining battery capacity, each state interval represents twice the range of the previous one. For example, with 9 states, we use 0, 1, 2, 4, …, 32, 64, 100 percentile points. The parameters of this model can be obtained by using an approach similar to the one presented in section 4.B.5. As demonstrated in the Fig. 15, when the battery model has more than 14 states, the BAPM policy achieves desirable battery lifetime improvement.

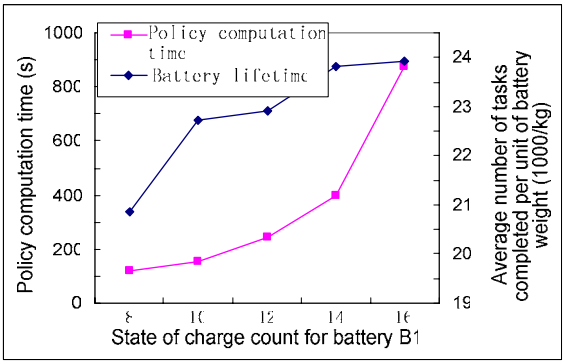


Fig. 15. Effect of the number of battery states of charge on the quality of solutions generated by BAPM for a two-battery system.

The simulation in section 6.B uses battery B1. However, there is no other battery to help provide the required energy for the system. This condition in turn limits the recovery effect for B1. In the case where the average task delay is set to 0.4s, an 8-state model of battery B1 generates almost optimal results.

VII. Conclusions

In this paper, a novel battery-aware power management (BAPM) technique is proposed, which is based on a Continuous-Time Markovian Decision Process (CTMDP) model that integrates the models of the electronic system and its power source. The BAPM technique presented in this paper attempts to maximize the battery service lifetime while meeting the given service timing constraints. It operates by determining the operating mode of an electronic system and concurrently choosing the battery that is used as the power source of the system when multiple batteries are available. This goal is achieved based on the characteristics of the incoming tasks, operation modes of the electronic system, and electrochemical properties and state of charge information about the batteries that are powering the system.

To develop the integrated battery-microelectronics system model, a properly constructed and characterized CTMDP model of the battery source is introduced. This model correctly captures the two important battery characteristics, i.e., the recovery effect and current-capacity curve. Based on this integrated model, the BAPM policy optimization problem is formulated and solved by using a Linear Programming formulation. Experimental results demonstrate the effectiveness of BAPM.

Appendix

Notations used in this paper are summarized below:

r, q, s, w, b, and x: respectively represent the state of the SR, SQ, SP, PS, BAT and SYS.

a_s, a_w, a_x: respectively denote the action of the SP, PS and SYS. When such a notation is used as a superscript, it describes the condition "when the action is chosen".

$\tau_s^{a_s}$ ($\tau_x^{a_x}$): denotes the average time that the SP (SYS) stays in state s (x) when the action a_s (a_x) is chosen.

$\lambda(r_i)$: denotes the service request generation rate when the SR is in state r_i.

$\nu_{i,j}$: denotes the transition rate of the SR from state r_i to r_j.

$\mu(s)$: denotes the service speed of the SP when it is in state s, i.e. the service request departure rate from the SQ.

$P_{s,s'}^{a_s}$: denotes the probability that the next state of the SP is s' when its present state is s and action a_s is chosen.

$\sigma_{s,s'}^{a_s}$: denotes the transition rate of the SP from state s to state s' when a_s is chosen.

$ene(b_i, b_j)$: denotes the energy-capacity difference of the battery between state b_i and b_j.

ρ, ρ': denotes the battery transition rate regarding to the discharge process.

ω, ω': denotes the battery transition rate regarding to the recovery process.

υ, υ': denotes the battery transition rate regarding to the recovery stop phenomenon.

$\gamma_x^{a_x}$: represents the expected energy delivered from the battery when the system is in state x and action a_x is chosen

$f_x^{a_x}$: denotes the frequency that the system will be in state x and action a_x is chosen.

lq: denotes the waiting cost in the queue.

REFERENCES

[1] M. Srivastava, A. Chandrakasan, and R. Brodersen, "Predictive system shutdown and other architectural techniques for energy efficient programmable computation," IEEE Trans. VLSI Systems, Vol. 4, pp. 42–55, Mar. 1996.

[2] C-H. Hwang and A. Wu, "A predictive system shutdown method for energy saving of event-driven computation," Proc. of Int'l Conf. on Computer-Aided Design, pp. 28–32, Nov. 1997.

[3] L. Benini, G. Paleologo, A. Bogliolo, and G. De Micheli, "Policy optimization for dynamic power management," IEEE Trans. Computer-Aided Design, Vol. 18, pp. 813–33, Jun. 1999.

[4] Q. Qiu, Q Wu and M. Pedram, "Stochastic modeling of a power-managed system-construction and optimization," IEEE Trans. Computer-Aided Design, Vol. 20, pp. 1200-1217, Oct. 2001.

[5] T. Simunic, L. Benini, P. Glynn, G. De Micheli, "Event-driven power management," IEEE Trans. Computer-Aided Design, Vol. 20, pp. 840-857, Jul. 2001.

[6] M. Doyle, J. Newman, "Analysis of capacity-rate data for lithium batteries using simplified models of the discharge process," Journal of Applied Electrochemistry, Vol. 27, Iss. 7, 846-856, 1997.

[7] M. Doyle, T.F. Fuller and J. Newman, "Modeling of galvanostatic charge and discharge of the lithium/polymer/insertion cell", Journal of Electrochemical Society, Vol. 141, No. 1, pp. 1-9, Jan. 1994.

[8] T.F. Fuller, M. Doyle and J. Newman, "Relaxation phenomena in lithium-ion-insertion cells," Journal of Electrochemical Society, Vol. 141, No. 4, Apr. 1994.

[9] D. Rakhmatov and S. Vrudhula, "An analytical high-level battery model for use in energy management of portable electronic systems," Proc. of Int'l Symp. on Low Power Electronics and design, pp. 88-91, Oct. 2001.

[10] L. Benini, G. Castelli, A. Macii, E. Macii, M. Poncino, and R.Scarsi, "A discrete-time battery model for high-level power estimation," Proc. of Design Automation and Test in Europe, pp. 35-39, Mar. 2000.

[11] C. Chiasserini, R. Rao, "Importance of a Battery Pulsed Discharge in Portable Radio Devices", Proc. of Int'l Conf. on Mobile Computing and Networking, pp. 88-95, August 1999.

[12] D. Panigrahi, C. Chiasserini, S. Dey, R. Rao, A. Raghunathan, K. Lahiri, "Battery Life Estimation for Mobile Embedded Systems," Proc. of Int'l Conf. on VLSI Design, pp. 55-63, January 2001.

[13] Carla-Fabiana Chiasserini, Ramesh R. Rao, "Energy Efficient Battery Management," IEEE Journal of Selected Areas in Communications, Vol. 19, No.7, Jul. 2001.

[14] L. Benini, G. Castelli, A. Macii, E. Macii, M. Poncino, R. Scarsi, "Extending lifetime of portable systems by battery scheduling," Proc. of Design Automation and Test in Europe, pp. 197-201, Mar. 2001.

[15] Q. Wu, Q. Qiu and M. Pedram, "An Interleaved Dual-battery Power Supply for Battery-Operated Electronics," Proc. of Design Automation Conference, pp. 387-390, Jun. 2000.

[16] M. Pedram and Q. Wu, "Design Considerations for battery-power electronics," Proc. of Design Automation Conference, pp. 861-866, Jun. 1999.

[17] J. Luo and N.K. Jha, "Battery-aware static scheduling for distributed real-time embedded systems," Proc. of Design Automation Conference, pp. 444-449, Jun. 2001.

[18] D. Rakhmatov, S. Vrudhula and C. Chakrabarti, "Battery-conscious task sequencing for portable devices including voltage/clock scaling", Proc. Of Design Automation Conference, pp. 10-14 Jun. 2002.

[19] K. Lahiri, S. Dey and A. Raghunathan, "Communication-based power management," IEEE Design & Test of Computers, Vol. 19, pp. 118 -130, July-Aug. 2002.

[20] L. Benini, G. Castelli, A. Macii, R. Scarsi, "Battery-driven dynamic power management," IEEE Design & Test of Computers, Vol. 18, pp. 53-60, 2001.

[21] I. Buchmann, Batteries in a Portable World, URL: http://www.buchmann.ca/toc.asp.

[22] Q. Qiu, Stochastic Modeling of a Power-Managed System: Construction and Optimization, Ph.D. Dissertation, Dept. of Electrical Engineering, Univ. of Southern California, Dec. 2000.

[23] M. Kijima, Markov Processes for Stochastic Modeling, New York: Chapman & Hall, 1997.

[24] J. Medhi, Stochastic Models in Queueing Theory, Amsterdam: Academic Press, 2003.

[25] U. N. Bhat, Elements of Applied Stochastic Processes. New York: Wiley, 1984.

[26] S. Mehrotra, "On the implementation of a primal-dual interior point method," SIAM Journal on Optimization, Vol. 2, pp. 575-601, 1992.

[27] http://www.applieddata.net/developers/documents/110115-00033_BitsyX_Users_Manual_prelim.pdf

[28] http://www.socketcom.com/product/WL6004-322.asp

[29] J.M. Tarascon, A.S. Gozdz, C. Schmutz, F. Shokoohi, and P.C. Warren, "Performance of bellcore's plastic rechargeable Li-ion batteries," Solid State Ionics, pp 49-54, Month: 1996.

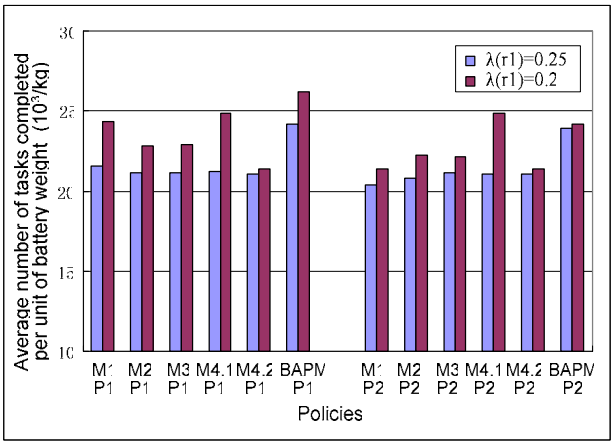
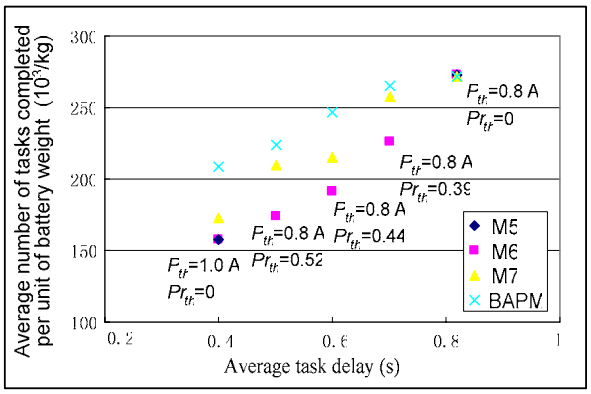
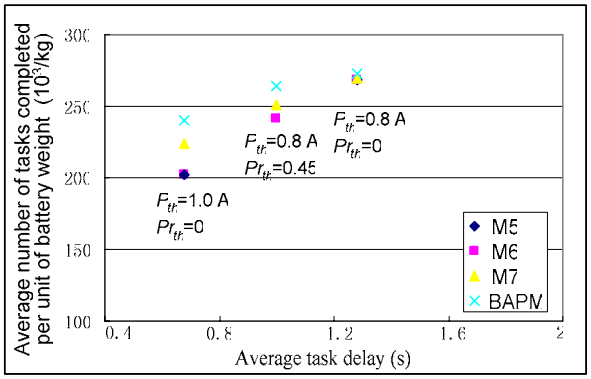


Fig. 16. Simulation results of BAPM for the 2BAT system model.

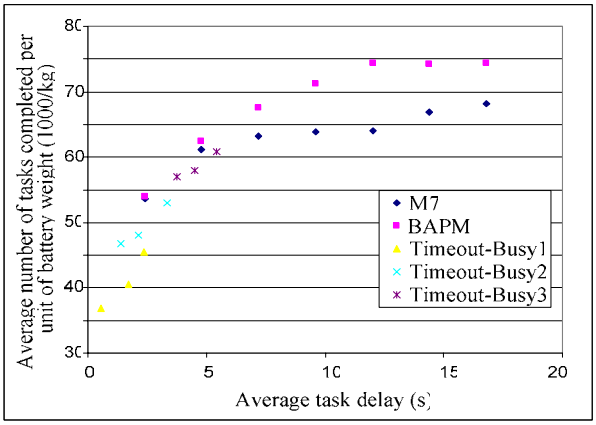


(a) The average task generation interval is 0.2 s.

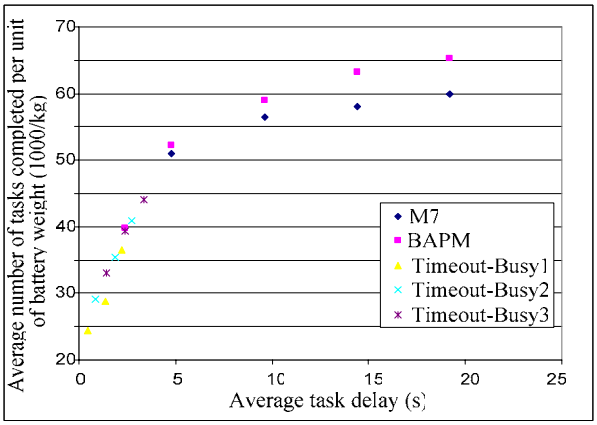


(b) The average task generation interval is 0.4 s.

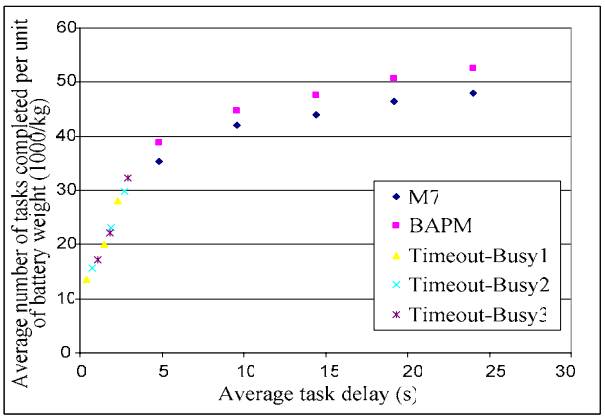
Fig. 17. Simulation results of BAPM for the 2SP system model.



(a) The average task generation interval is 2.4 s.



(b) The average task generation interval is 2.4 s.



(c) The average task generation interval is 4.8 s.

Fig. 18. Simulation results for the BitsyX system.