Determining the Optimal Timeout Values for a Power-Managed System based on the Theory of Markovian Processes: Offline and Online Algorithms[1]

Peng Rong
Dept. of Electrical Engineering
University of Southern California
Los Angeles, CA 90089
e-mail : prong@usc.edu

Massoud Pedram
Dept. of Electrical Engineering
University of Southern California
Los Angeles, CA 90089
e-mail : pedram@ceng.usc.edu

ABSTRACT

This paper presents a timeout-driven DPM technique which relies on the theory of Markovian processes. The objective is to determine the energy-optimal timeout values for a system with multiple power saving states while satisfying a set of user defined performance constraints. More precisely, a controllable Markovian process is exploited to model the power management behavior of a system under the control of a timeout policy. Starting with this model, a perturbation analysis technique is applied to develop an offline gradient-based approach to determine the optimal timeout values. Online implementation of this technique for a system with dynamically-varying system parameters is also described. Experimental results demonstrate the effectiveness of the proposed approach. Introduction

Dynamic power management (DPM), which refers to selective shut-off or slow-down of components that are idle or underutilized, has proven to be a particularly effective technique for reducing power dissipation in such systems. In the literature, various DPM techniques have been proposed, from heuristic methods presented in early works [1][2] to stochastic optimization approaches [3][4].

Among the heuristic DPM methods, the timeout policy is the most widely used approach in industry and has been implemented in many operating systems. Examples include the power management scheme incorporated into the Windows system, the low-power saving mode of the IEEE 802.11a-g protocol for wireless LAN card, and the enhanced adaptive battery life extender (EABLE) for the Hitachi disk drive. Most of these industrial DPM techniques provide mechanisms to adjust the timeout values at the user level.

In the other direction, stochastic approaches have been proposed for DPM. Reference [3] proposed a power management model based on discrete-time Markovian decision chain. The discrete-time model requires policy evaluation at periodic time intervals and may thus consume a large amount of power dissipation even when no change in the system state has occurred. To overcome 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] also improved on the modeling technique of [3] by using a time-indexed semi-Markovian decision process to handle workloads with Pareto-distributed idle time durations. The most significant advantage of stochastic approaches is that, based on mathematically rigorous stochastic models of a power-managed system, these techniques can construct the optimum DPM policy.

This paper presents a timeout-based DPM technique, which is constructed based on the theory of Markovian processes and is capable of determining the optimal timeout values for an electronic system with multiple power-saving states. More precisely, in this paper, a continuous-time Markovian process based stochastic model is presented to capture the power management behavior of an electronic system under the control of a timeout policy. Perturbation analysis is used to construct an offline gradient-based approach to determine the set of optimal timeout values. Finally, online implementation of this approach is also discussed.

The motivation for this work comes from the following factors. The timeout policy is an industry standard that has been widely deployed in many real systems. A DPM technique based on timeout policies may thus be easier and safer for users to implement. At the same time, it helps them achieve a reasonably good energy-performance trade-off. To implement a more elaborate DPM technique requires the users to directly control the power-down and wake-up sequences of system components, which normally necessitates detailed knowledge of hardware and involves a large amount of low-level programming dealing with the hardware interface and device drivers. Notice also that the various system modules typically interact with each other implying that hasty power-down of a system module may cause the whole system to malfunction or become unstable. So it is a big responsibility to control directly over the state of a system module that should not be delegated unceremoniously. A DPM technique based on a simple and well-tested timeout policy and incorporated in the operating system will have none of the above concerns.

Previous stochastic DPM approaches with non-stationary input request generation rates [6][7] or variable system

[1] This project was funded in part by the NSF CNS grant no. 0509564.

parameters [8] are mostly based on online lookup or manipulation of policy tables that have in turn been constructed offline. This is because the calculation of the optimal DPM policy is very computation-intensive and hence cannot be done at runtime. The perturbation analysis-based DPM approach presented in this paper reveals a very different approach for online power management policy tuning. This approach does not require a pre-computed policy table yet it avoids the computational burden of solving complex mathematical programs to derive the optimal DPM policy online [7]. The key idea is to compute an optimal parameterized policy offline for the complete range of system input parameters. Next, at runtime, by exploiting perturbation analysis, the gradients of power dissipation and performance with respect to the control parameters are estimated and subsequently used to adjust the policy parameters, e.g., the timeout values or the probability that a decision is made by as part of policy.

In the literature, several works have been reported to optimize the timeout policies. For example, reference [10], based on the theory of competitive analysis, generalizes a 2-competitive algorithm [9] to power management of a system composed of components with multiple power saving states. The analysis shows that this generalization of the 2-competitive policy consumes at most twice as much energy dissipation as the optimum policy. Note that competitive analysis is a worst case analysis method which often provides pessimistic bounds on energy saving that is achievable by DPM. In particular, policies developed based on competitive analysis do not take advantage of knowledge about the workload. To surmount this deficiency, reference [10] also proposed a probabilistic analysis method to determine the optimal timeout values for a given input probability distribution with the objective of minimizing the average power consumption. A similar method is used in reference [11] to determine the optimal timeout value, assuming that the duration of idle periods between input requests follows a Pareto distribution. A key shortcoming of all of the abovementioned timeout value optimization techniques is that they do not consider any timing constraints that may be imposed on the service time of the input requests.

The remainder of this paper is organized as follows. Background for perturbation analysis is provided in Section 2. In Section 3, stochastic modeling of a system under the control of a timeout policy is presented. Offline solution approach is presented in Section 4 and online implementation issues are addressed in Section 5. Experimental results and conclusions are provided in Sections 6 and 7, respectively.

1 BACKGROUND OF PERTURBATION ANALYSIS

Perturbation analysis (PA) provides performance sensitivities for a discrete event system by analyzing the dynamic behavior of the system. In the literature, many PA methods have been developed [12] for various types of system descriptions, including differential equations, queuing networks, and Markovian processes. In this section, we will briefly introduce the PA method developed in [13] for Markovian processes, which is based on the concept of perturbation realization.

Consider a regular, positive recurrent, and irreducible continuous-time Markovian process (CTMP) $X = \{X_t, t \geq 0\}$ with a countable state space $S = \{s_1, s_2, \ldots\}$ and an infinitesimal generator $G = [g_{ij}]$, where g_{ij} represents the transition rate from state s_i to s_j. Let $p=(p_i)$ denote the steady-state probabilities of the Markovian process. Then it holds that $pG=0$ and $Ge=0$, $e=[1,1,\ldots]^T$. Let $f(s_i)$: $S \to R$ denote a performance function. The expected performance measure of the Markovian process is determined by

$$\eta = E_p(f) = \sum_{s_i \in S} p_i f(s_i) = pf \quad \text{(2-1)}$$

where $f = [f(s_1), f(s_2), \ldots]$ is a column vector.

Now suppose that G changes to $G_\delta=[g_{\delta, ij}]=G+\delta H$, where δ is a small real number and $He=0$, which results in the performance measure changing to $\eta_\delta=\eta+\Delta\eta$. Perturbation analysis studies the derivative of η in the direction of H, which is defined as follows

$$\frac{\partial \eta}{\partial H} \triangleq \lim_{\delta \to 0} \frac{\eta_\delta - \eta}{\delta} \cdot \quad \text{(2-2)}$$

The fundamental basis of perturbation realization is that the effect of a parameter change is the sum of the effects of many individual changes on a sample path. Consider the Markovian process X with state space S and generator G. Assume at time t_0, X is perturbed from state s_i to s_j, which means that X should transit to state s_i according to G, but it reaches state s_j because of the small change of generator G, i.e., G becomes G_δ at this instance. Let X_1 and X_2 denote the original and perturbed processes, respectively. It is assumed that after t_0 both X_1 and X_2 evolve based on generator G. Thus the effect of this single perturbation on the performance can be measured as the difference between the performances of X_1 and X_2.

We define potential vector $\varepsilon=(\varepsilon_i)$, where ε_i denotes the performance potential for state s_i and is determined as follows

$$\varepsilon_i = \lim_{T \to \infty} \{E[\int_0^T f(X_t^{(i)})dt] - \eta T\} \cdot \quad \text{(2-3)}$$

Here $X_t^{(i)}$ represents a sample path of Markovian process X with initial state s_i. Furthermore, ε satisfies the Poisson equation: $G\varepsilon=-f+\eta e$ [13]. Based on the concept of perturbation realization, the derivative of η is calculated as [13]

$$\frac{\partial \eta}{\partial H} = pH\varepsilon \cdot \quad \text{(2-4)}$$

2 SYSTEM MODELING

In this paper, we will focus our discussion on a power management framework which consists of a service requestor (SR), a service queue (SQ) and a single service provider (SP). The SP provides service for the service requests generated by the SR. The requests that cannot be serviced immediately are waiting in the SQ for the SP to become available. The SP has a working state and at least a low-power state. The SP uses less power in its low-power state, but a transition into or out of the low power state consumes additional energy and may increase the service delay for the requests. The state transition of the SP is controlled by a timeout policy, which assigns a timeout value to each state except the one with the lowest power. The SP will transit to the next highest power state from the current state if the corresponding timeout expires and no requests are generated. The objective of this paper is to find out the optimal timeout values that minimize the power consumption of the SP while satisfying the constraint on the average service delay.

Similar to the work in [4], the SR and the SQ can be modeled as CTMP models as depicted in Figure 1. The SR model consists of a state set $\mathbf{R} = \{r_i, i=1,2,\ldots,R\}$ and a generator matrix G_{SR}, where R is the number of the states of the SR. The SQ model consists of a state set $\mathbf{Q} = \{q_i, i=0,1,\ldots,Q\}$ 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.

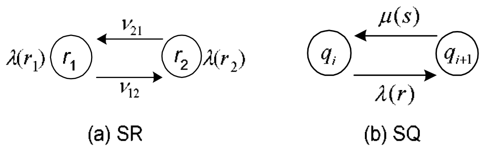


Figure 1. The CTMP models of the SR and the SQ.

The SP under the control of a timeout policy can be modeled as a Markovian process model which simulates the power management behavior of the SP. As an example, the CTMP model of an SP with a single low-power state is presented in Figure 2,. This model comprises of a state set $\mathbf{S}$={Work, Idle, $TO_1 \ldots TO_n$, Sleep} and a generator matrix $G_{SP}(\lambda)$, where λ is the parameter to be optimized. Notice that λ sets the duration of time that the SP spends in the Idle state before it can go to the low power Sleep state during an idle period (i.e., the timeout value.) Details of model states and state-transitions are explained next.

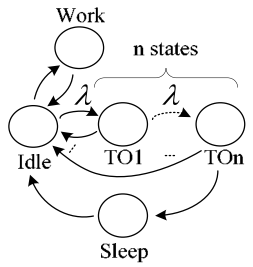


Figure 2. The CTMP model of the SP with a timeout policy.

Sleep: A low-power state. The SP transits to Idle state when the SQ is non-empty.

Work: A functional state, where the SSP provides service to the SR that is waiting in the SQ.

Idle: A non-functional state. If the SQ is non-empty, the SP goes to the Work state; otherwise, it goes to TO_1 state.

TO_i: i=1,2,…,n: One of n non-functional time-out states. These states are used to simulate the timeout policy. When the SQ is non-empty, the SP goes back to the Idle state; otherwise, the SP goes to TO_{i+1} state or Sleep state if the SP is in the TO_n state. Notice that the time for the SSP transferring from Idle to TO_n state is a random variable whereas in the timeout policy, the timeout value is fixed. Consequently, multiple TO states are used to improve the simulation accuracy based on the central limit theorem [14]. The transition rates between Idle to TO_1 and TO_i to TO_{i+1}, i=1,2,..,n-1, are all the same, i.e., λ.

This model can easily be extended to an SP with multiple low power states by introducing a TO state chain for each additional low power states. Next, we define the CTMP model SYS of the whole system comprising of the SR, SQ and SP. Let $\mathbf{X}$ denote the state space of SYS, thus $\mathbf{X}=\mathbf{R}\times\mathbf{Q}\times\mathbf{S}-$\{invalid states\}, where the invalid states include the states where the SP is in the Work state while the SQ is empty. Assume the SP has m low-power states s_i, i=1,2,…,m, sorted in a descending order of power consumptions. Let s_0 denote the full-power non-functional state, i.e., the Idle state in Figure 2. Let TO_{ij}, i=0,1,…,m−1, j=1,2,…,n, denote a TO state at the same power level as state s_i, where n denotes the number of TO states in a TO state chain. Let $\lambda=(\lambda_i)$, where λ_i determines the timeout value $T_{to,i}$ which should be exceeded before the SP can transit to low-power state s_{i+1}: $T_{to,i}= n/\lambda_i$. The generator of the SYS is a function of λ, and is denoted by $G(\lambda)$. Note that this model includes unknown continuous parameters, i.e. λ, to be determined.

3 PROBLEM FORMULATION AND SOLUTION

Let $\gamma=(\gamma_x)$ denote a column vector, where γ_x denotes the expected power consumption when the SYS is in state x. Let $w=(w_x)$ be a column vector, where w_x denotes the number of requests waiting in the SQ when the SYS is in state x. Let D denote an upper bound on the expected number of requests waiting in the SQ. Finally, let $p=(p_x)$ denote a row vector, where p_x represents the steady-state probability of state x. Now, we can formulate the timeout-policy based energy optimization problem as a constrained mathematical program as follows:

$$\text{Min}_{\{p,\lambda\}}\,(p\gamma) \tag{4-1}$$

which is solved over variables p and λ subject to:

$$pG(\lambda)=0, \tag{4-2}$$

$$pe=1,\quad e=[1,1,...,1]^{\mathrm{T}}, \tag{4-3}$$

$$pw\le D,\ \ p,\lambda\ge 0\,. \tag{4-4}$$

Equations (4-2) and (4-3) capture properties of a CTMP. Inequalities (4-4), which are based on the Little's theorem [15], impose constraints on the expected task delay of the SP.

This problem is a non-convex, constrained mathematical program (c.f. (4-2)). The optimal solution can be found by using standard stochastic search techniques, e.g., simulated annealing. However, we have developed a gradient-based optimization algorithm, which efficiently finds a good approximate solution based on the structural features of the CTMP model.

We next consider the derivative of performance measure η with respect to λ_i, i=0,1,…,m−1. From equation (2-4), the derivative is calculated as

$$\frac{\partial\eta}{\partial\lambda_i}=pH_i\varepsilon \quad\text{where}\quad H_i=[h_{i,xy}]=[\frac{\partial g_{xy}(\lambda)}{\partial\lambda_i}], \tag{4-5}$$

$g_{xy}(\lambda)$ denotes the (x, y) entry of the generator matrix $G(\lambda)$, $x,y\in X$. From an examination of the CTMP model, we find that λ_i affects the SYS state transition rate only if the SQ is in state q_0 and the SP is in state s_i or TO_{ij}. Thus H_i is a sparse matrix which has only $2nR$ non-zero entries, where n is the number of TO_i states and R of SR states. From the CTMP model SYS, it is seen that each non-zero entry of H_i only takes a value of either 1 or −1. Let $x(r, q, s)$ denote the fact that in a global SYS state x, the SR is in state r, the SQ is in state q, and the SP is in state s. From equation (4-5) and the observation about the sparsity of H_i, the derivative is

$$\frac{\partial\eta}{\partial\lambda_i}=\sum_r\left(\begin{array}{l} p_{x(r,q_0,s_i)}(\varepsilon_{x(r,q_0,TO_{i,1})}-\varepsilon_{x(r,q_0,s_i)})+ \\ \displaystyle\sum_{j=1}^{n-1}p_{x(r,q_0,TO_{i,j})}(\varepsilon_{x(r,q_0,TO_{i,j+1})}-\varepsilon_{x(r,q_0,TO_{i,j})})\end{array}\right)\cdot \tag{4-6}$$

From equation (4-6), we can see that, with $x=x(r, q_0, s)$, only a small number of p_x's and ε_x's are involved in the derivative calculation. Thus we may not have to calculate values for all p_x's and ε_x's.

Let indices of the generator matrix $G(\lambda)$ be sorted following the sequence of s, r, and q. Then, $G(\lambda)$ may be represented as

$$G(\lambda)=\begin{bmatrix} G_{11}(\lambda) & G_{12} \\ G_{21} & G_{22} \end{bmatrix}, \qquad (4\text{-}7)$$

where $G_{11}(\lambda)$ represents the state transitions between states in a sub-space $\mathbf{X}_0$ where $q=q_0$. This is a square matrix with a dimension of $m(n+1)+1$. G_{11}, G_{21} and G_{22} are matrices with constant entries, which are independent of the value of λ.

Lemma 1: If $G(\lambda)$ is the infinitesimal generator of an ergodic Markovian process, then the inverse of G_{22} will exist.

Based on this lemma, we can prove an important result, stating that to determine a derivative $\partial\eta/\partial\lambda_i$, we need only explore the sub-space $\mathbf{X}_0$, whose size is nearly $1/R$ of the whole state space $\mathbf{X}$. In the following, the superscripts "−1" and "T" denote the inverse and transpose of a matrix, respectively.

Theorem 1: Let $p_{X0}=(p_x)$ be a row vector, and $\varepsilon_{X0}=(\varepsilon_x)$ be a column vector, where $x\in\mathbf{X}_0$. p_{X0} and ε_{X0} can be calculated from the following equations

$$p_{X_0}\left(e_{X_0}e_{X_0}^{T}+G_{11}(\lambda)-G_{12}G_{22}^{-1}(e_{\bar{X}_0}e_{X_0}^{T}+G_{21})\right)=e_{X_0}^{T} \qquad (4\text{-}8)$$

$$\begin{aligned}&\left(e_{X_0}e_{X_0}^{T}+G_{11}(\lambda)-G_{12}G_{22}^{-1}G_{21}\right)\varepsilon_{X_0}=\\ &\quad((e_{X_0}-G_{12}G_{22}^{-1}e_{\bar{X}_0})p_{X_0}-I)(f_{X_0}-G_{12}G_{22}^{-1}f_{\bar{X}_0})\end{aligned} \qquad (4\text{-}9)$$

where $\bar{X}_0$ denotes the complement of $\mathbf{X}_0$ in the state space $\mathbf{X}$. Furthermore, η can be calculated based on p_{X0} as follows:

$$\eta=p_{X_0}\left(f_{X_0}-G_{12}G_{22}^{-1}f_{\bar{X}_0}\right) \qquad (4\text{-}10)$$

Before presenting our algorithm to solve the optimization problem (4-1), we first define two performance measures: the expected energy consumption rate $\eta_\gamma=p\gamma$ and the expected number of waiting requests in the SQ $\eta_w=pw$. The derivatives of the two performance measures $\partial\eta_\gamma/\partial\lambda_i$ and $\partial\eta_w/\partial\lambda_i$ can be determined from equations (4-6) to (4-10). However, it is awkward to directly use these two measures in gradient-based optimization search. The reason is that the absolute values of these derivatives become very large when λ_i approaches 0. To overcome this problem, we define variable τ_i, such that

$$\tau_i=\ln\lambda_i,\quad i=0,1,\cdots,m-1. \qquad (4\text{-}11)$$

We have

$$\frac{\partial\eta}{\partial\tau_i}=\lambda_i\frac{\partial\eta}{\partial\lambda_i},\quad i=0,1,\cdots,m-1. \qquad (4\text{-}12)$$

Our algorithm to solve the optimization problem (4-1) is then as follows.

<u>Offline Algorithm</u>

Input: A parameterized generator matrix $G(\lambda)$ which describes the power management system.

Output: Timeout values, $T_{to,i}=n/\lambda_i$, $i=0,1,\ldots,m-1$.

```
1.  Set initial values for $\tau_i$, denoted as $\tau_{i,0}$, and tolerances $\delta_u$, $\delta_D$>0.
2.  Calculate the constant parameter matrices used in equations (4-8) and (4-9).
3.  Set penalty factor $M>0$.
4.  Set initial step size $s_0$ and $k=0$.
5.  Calculate $\eta_\gamma$ and $\eta_w$ values at step $k$ by using equations (4-8) to (4-11).
6.  Calculate derivatives $\partial\eta_\gamma/\partial\tau_{i,k}$ and $\partial\eta_w/\partial\tau_{i,k}$ by using equations (4-6) and (4-12).
7.  Calculate the gradient of cost function $u=\eta_\gamma+M(\eta_w-D)$ at step $k$, denoted as $\nabla u_k$.
8.  If $\|\nabla u_k\|<\delta_u$, go to step 11.
9.  Set $\boldsymbol{\tau}_{k+1}=\boldsymbol{\tau}_k-s_k\nabla u_k$, where $\boldsymbol{\tau}_k=(\tau_{i,k})$, $i=0,1,\ldots,m-1$.
10. Let $k=k+1$, and go to step 5.
11. If $\eta_w-D<\delta_D$, go to END, else increase $M$ and go to step 4.
    END
```

The convergence of this algorithm is guaranteed based on the following key result.

Theorem 2: Given is a linear composition u of bounded, differentiable measures defined on an ergodic CTMP with parameterized generator $G(\tau)$ that is differentiable with respect to τ. If step size $s_k>0$ satisfies

$$\sum_{k=1}^{\infty}s_k=\infty,\ \sum_{k=1}^{\infty}{s_k}^2<\infty, \qquad (4\text{-}13)$$

then $u(\boldsymbol{\tau}_k)$ converges and $\lim_{k\to\infty}\|\nabla u(\boldsymbol{\tau}_k)\|=0$ with probability one, where $\boldsymbol{\tau}_k$ is a sequence defined in line 9 of the Offline Algorithm.

Note that, as an example, a sequence of step sizes that satisfy constraint (4-13) are $s_k=1/k$, $k=1,2,\ldots$.

4 ONLINE ALGORITHM

To develop online power management algorithms, we assume that initially, the optimal solution has been deployed. This optimal solution can be achieved by doing online statistical profiling followed by offline optimization to generate the optimal solution for the nominal values of the system parameters. However, at run time, the system's input parameters are subject to change. For example, the average service request generation rate can change over time. The goal of our online algorithm is to be able to adjust the initial optimal solution (i.e., the timeout values dictated by this solution) following changes in the system parameters, so as to maximize the power efficiency of the service provider while satisfying system-level performance constraints. For this purpose, Algorithm 1 can be used, however, although the matrix size has significantly been reduced by equations (4-8) and (4-9). Matrix multiplications and solving the system of equations may still be a heavy burden on the computation capability of the running system if such operations are performed online. Therefore, we estimate the values of p_{X0} and ε_{X0} through the real sample path. Let T_k be the k_{th} transition epoch of the Markovian processes $\{X_t\}$ which describes the behavior of system SYS; Let X_k be the state after the k_{th} transition, C_k be the sojourn time that process X_t stays in state X_k,. Then $C_k=T_{k+1}-T_k$. Now p_{X0} and ε_{X0} are estimated based on the following equations

$$p_x=\lim_{N\to\infty}\frac{1}{T_N}\left\{\sum_{k=0}^{N-1}\varsigma(X_k,x)C_k\right\}, \qquad (5\text{-}1)$$

$$\varepsilon_x = \lim_{N\to\infty} \frac{\sum_{k=0}^{N-l} \varsigma(X_k, x) \int_{T_K}^{T_K+T} f(X_t)dt}{\sum_{k=0}^{N-l} \varsigma(X_k, x)} \tag{5-2}$$

where $\varsigma(X_k, x) = 1$, if $X_k = x$; otherwise, it equals to 0; T is a properly chosen constant and l is selected satisfying that X_t is observed in the whole integration period in (5-2). Since $f(X_t)$ is a piecewise function of t, the integration in (5-2) can be implemented by accumulating the product of C_k and $f(X_k)$ at transition epoch T_{k+1}. In addition, we need to estimate η in order to check the constraint and convergence condition, which may be directly done by

$$\eta = \lim_{N\to\infty} \frac{1}{T_N} \int_0^{T_N} f(X_t)dt \tag{5-3}$$

For the online algorithm, it is important to capture the change in system parameters and guarantee that the performance constraints are met. Thus, we present an online algorithm, which adaptively adjusts the step size to speed up the convergence.

Online Algorithm

Input: Measured p_{X0}, ε_{X0}, η_γ and η_w along a sample path.

Output: Timeout values $T_{to,i}= n/\lambda_i$, i=0,1,…,m–1.

1. Set the initial step size s_0 and k=0. Let $\tau_{i,k}$ denote the values of τ_i at step k.
2. Calculate derivatives $\partial\eta_\gamma/\partial\tau_{i,k}$ and $\partial\eta_w/\partial\tau_{i,k}$ by using equations (4-6), (4-11), (4-12), (5-1) and (5-2).
3. Estimate η_γ and η_w at step k by using equation (5-3), and denoted by $\eta_{\gamma,k}$ and $\eta_{w,k}$, respectively.
4. If $\eta_{w,k}\le D$, then $\boldsymbol{\tau}_{k+1} = \boldsymbol{\tau}_k - (s_k \cdot \nabla\eta_\gamma)/\|\nabla\eta_\gamma\|$. If $\eta_{w,k}>D$, then $i^* = \arg\max_i \{(\partial\eta_\gamma/\partial\tau_{k,i})/(\partial\eta_w/\partial\tau_{k,i})\}$, $\tau_{k+1,i^*} = \tau_{k,i^*} - s_k$.
5. If $\eta_{w,k}\le D\le\eta_{w,k-1}$ or $\eta_{w,k-1}\le D\le\eta_{w,k}$, then set $s_{k+1} = s_k/\alpha$, where $\alpha>1$ is a constant factor.
6. If both $\eta_{w,k}, \eta_{w,k-1}>D$, then increment a counter nc by one. In this case, if nc becomes greater than a preset threshold, then set $s_{k+1} = \alpha s_k$ and clear nc.
7. If both $\eta_{w,k}, \eta_{w,k-1}\le D$ and $\nabla\eta_{\gamma,k}\cdot\nabla\eta_{\gamma,k-1}<0$, then set $s_{k+1} = s_k/\alpha$.
8. Go to step 2.

5 EXPERIMENTAL RESULTS

For this experiment, we used ten-hour traces of hard disk requests generated by two types of applications running on a Linux PC. The first application was a file manipulation program, which read some data file, edited the file, and wrote it back to the disk. The second application was a program which periodically reads data from another machine through a WLAN card, searches for some relevant information in the received data, and stores this information onto the disk. The request generation pattern of the first application was accurately modeled with a Poisson process with an average rate of 0.208 requests per second. The request generation statistics of the second program was adequately characterized by a two-state CTMDP model. The state transition rate and generation rates of SR to hard disk λ_{hd} are

$$\begin{bmatrix} 0 & 0.0415 \\ 0.0063 & 0 \end{bmatrix} (s^{-1}),\ \lambda_{hd} = [0.0826, 0.0187]\,(s^{-1}).$$

The average service time for a disk request is 67ms. For our experiment, we used the hard disk drive, Hitachi Travelstar 7K60, as the service provider, which has three low power states. Power dissipation and start-up energy and latency of the disk drive are reported in Table 1.

Table 1. Energy/transition data of hard disk driver

State	Power (w)	Start-up energy (J)	Wake-up time (s)
Active	2.5	--	--
Performance idle	2.0	0	0
Low power idle	0.85	1.86	0.4
Stand-by	0.25	10.5	2
Sleep	0.1	15.9	5

First, we examine the offline algorithm. The aforesaid traces are denoted as RT1 and RT2, respectively. In addition, we also generated two synthetic SR input traces: ST1 and ST2, where the SR generation rates were set to 1/10 and 1/15 per second, respectively. For comparison purposes, we considered two DPM techniques in addition to our proposed technique (which we denote by PA for perturbation analysis): a probabilistic analysis-driven timeout policy (PTO) [10] which determines the timeout value to minimize energy consumption while ignoring any timing constraints and a CTMDP-based DPM policy [4]. The simulation results of these policies for different input traces and under different delay constraints are reported in Figure 3.

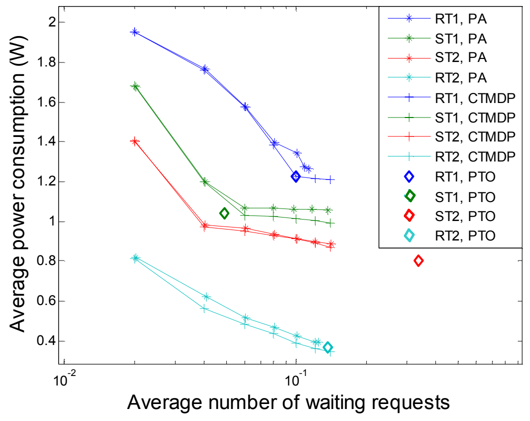
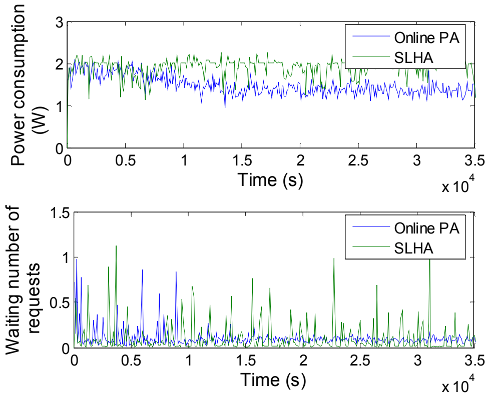


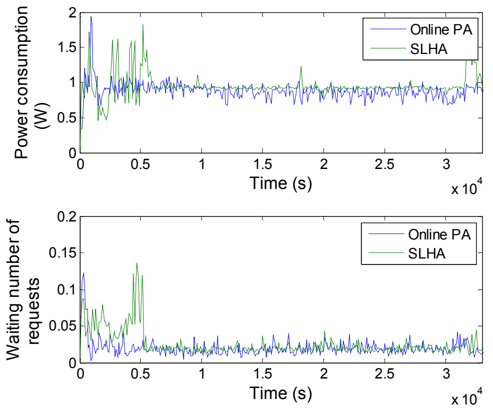
Figure 3. Comparison of simulation results of offline policies on a semi-log plot.

As we can see, the PTO approach yields well in terms of energy minimization. However, the corresponding SR delays are quite large except for ST1. Thus, in the case where constraining the SR latency is important, it is not suitable. The CTMDP-based DPM generates the provably optimal policy for minimizing energy under timing constraints. It is seen that the results generated by the PA technique are very close to the CTMDP-based technique. Recall that the key advantages of the PA technique lie in that (a) it is easier to implement the PA technique for a real system than the CTMDP-based techniques and (b) The PA technique does not require random variable generators, which are normally required by non-TO-based DPM.

We evaluated the proposed online algorithm. We compared our algorithm with Stochastic Learning Hybrid Automata (SLHA) approach presented in [16]. This model attempts to probabilistically learn the length of the future idle period and, accordingly adjust action the switching probabilities. The SLHA model was implemented according to our best of our ability and utilized for the remaining simulations. A linear reward-penalty scheme is used for feedback learning. The feedback parameters were selected to ensure that the timing constraints are eventually satisfied. We ran these two algorithms with three different input request generation patterns: ST1, RT1 and RT2. In Figure 4, two traces of the two algorithms for RT1 and RT2 are presented. The constraint on average waiting number of requests as set to 0.1 for (a) and 0.02 for (b). The reward-penalty coefficients used in SLHA were set as a=0.1 and b=0.01. Experimental results with different timing constraints D (c.f. (4-4)) are reported in Table 2. Parameter T in (5-2) is set to 100s for ST1 and RT1 and 200s for RT2. It is seen that the online PA technique achieves energy savings very close to the results of the offline optimal technique.



(a) RT1



(b) RT2

Figure 4. Comparison of simulation results of online policies.

Table 2. Results of power consumption for online algorithms

	D	ST1	RT1	RT2
SLHA	0.02	1.937	2.009	0.929
	0.1	1.386	1.779	0.457
Online PA	0.02	1.672	1.998	0.854
	0.1	1.102	1.437	0.443

6 CONCLUSION

This paper proposes a timeout-based DPM technique constructed on the theory of Markovian processes. This approach is designed to find out the optimal timeout values for a system with multiple power saving states in terms of energy dissipation while satisfying user defined performance constraints. First, a Markovian processes based stochastic model is proposed to model the power management behavior of a system under the control of a timeout policy. Next, based on this model, perturbation analysis technique is used to construct an offline gradient-based approach to search for the optimal timeout values. Online implementation of this approach is also proposed for a system with variable parameters.

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," *ICCAD*, 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. 1200-17, Oct. 2001.

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

[6] E.-Y. Chung, L. Benini, A. Bogliolo, Y.-H. Lu and G. De Micheli, "Dynamic power management for non-stationary service requests," *IEEE Trans. on Computers*, pp. 1345-61, Nov. 2002.

[7] Z. Ren, B.H. Krogh, and R. Marculescu, "Hierarchical adaptive dynamic power management," *DATE*, pp. 136-41, Feb. 2003.

[8] P. Rong and M. Pedram, "Extending the lifetime of a network of battery-powered mobile devices by remote processing: a Markovian decision-based approach," *DAC,* pp. 906-911, Jun. 2003

[9] A. Karlin, M. Manasse, L. McGeoch, and S. Owicki, "Competitive randomized algorithms for nonuniform problems," *Algorithmica*, vol. 11, no. 6, pp. 542–71, June 1994.

[10] S. Irani, S. Shukla, and R.Gupta, "Online strategies for dynamic power management in systems with multiple power saving states," *ACM Trans Embedded Systems*, vol. 2, no. 3, pp. 325-346, 2003

[11] L. Cai and Y-H Lu, "Joint power management of memory and disk," *Proc. of DATE* 2005, Vol. 1, pp. 86–91, 2005.

[12] J.F. Bonnans and A. Shapiro, "Perturbation analysis of optimization problems," *Springer*, New York, 2000.

[13] X-R Cao and H-F Chen, "Perturbation realization, potentials, and sensitivity analysis of Markov processes," *IEEE Trans Auto Control,* vol.42, no. 10, pp. 1382-93, Oct. 1997.

[14] http://mathworld.wolfram.com/CentralLimitTheorem.html.

[15] E. A. Feinberg, A. Shwartz, Handbook of Markov decision processes: methods and applications, Kluwer Academic, 2002.

[16] T. Erbes, S. K. Shukla, and P. Kachroo, Stochastic Learning Feedback Hybrid Automata for Dynamic Power Management in Embedded Systems, *Thesis*, 2004.