Low-Power Buffer Management for Streaming Data

Jason Ridenour, *Student Member, IEEE,* Jianghai Hu, *Member, IEEE,* Nathaniel Pettis, *Student Member, IEEE,* and Yung-Hsing Lu, *Member, IEEE*

***Abstract*— In this paper, a method for reducing the power consumption in a pipeline of streaming data is presented. The pipeline is divided into multiple stages and buffer memories are inserted between adjacent stages. By dynamically changing the power state of each stage, the overall power consumption of the pipeline, including the power consumed by each stage and by each buffer, can be reduced. For example, when a buffer is full, the stage immediately before it can be turned off for a certain time period to save power. In this paper, the problem of finding the optimal strategy for managing the power state to minimize average power consumption is studied for a pipeline of data consisting of three stages with two buffers in between. The problem is formulated as an optimal control problem of a hybrid system, namely, a dynamical system with both continuous dynamics and discrete transitions. Several important properties of the optimal solutions are derived. Using these properties, the optimal periodic solutions are obtained analytically for the special case when each stage is allowed to turn on and off at most once during each period. Simulation results using practical data are presented. The results show substantial power savings of the proposed method over heuristic buffering strategies.**

***Index Terms*— Dynamic power management, streaming data, buffers, hybrid systems.**

I. Introduction

Minimizing the energy consumption of electronic systems is becoming an increasingly important problem as battery-powered portable systems gain popularity. Many methods have been proposed for reducing the energy consumption of individual electronic components such as processors, wireless network interface cards, hard disks, and displays. Existing methods predict the periods during which a component remains idle or under-utilized, and turn off (sometimes called "shut down" or "put to sleep") the component or scale down its performance to reduce its energy consumption. This is called dynamic power management [1]. However, changing the power state requires extra energy and time. Thus, it is not always preferable to turn off a component each time it becomes idle. Energy can be saved only if the hardware component can remain in the off state long enough to compensate for the extra energy incurred. This minimum amount of time in the off state is called the component's break-even time [2].

To avoid interrupting system operation when system parameters (e.g., network congestion) change, buffer memory is often added between interacting components in electronic systems. For example, when a user watches a video clip using a PDA (personal digital assistant), the PDA first downloads a sufficient amount of data into buffer memory so that the video can be played smoothly regardless of the variations of the network conditions. This process may consist of multiple components and buffers. The first component is the network card downloading the video data; the downloaded data are then stored in a local storage device such as a microdrive or flash memory, and later retrieved by the processor for decoding. The decoded video may then be stored in another buffer before finally appearing on the LCD display. These activities combined form a multi-stage pipeline for the streaming data.

To demonstrate the dynamic power management process for pipelines of streaming data, a simple example is shown in Fig. 1(a), where component X produces data for component Y to consume, and there is a single buffer inserted between them. For the PDA example above, X can be the network card to download the video and Y can be the processor to decode the data. The two components form a two-stage pipeline of streaming data. Since X typically produces data at a higher rate

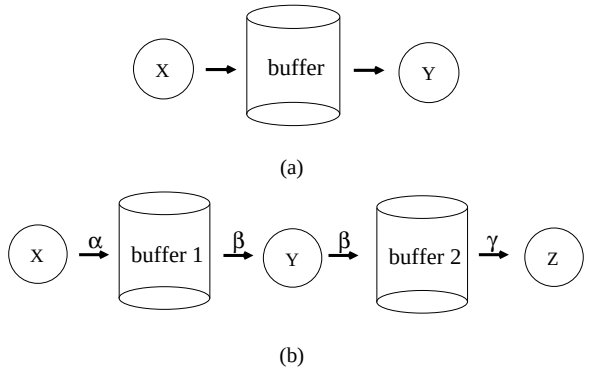

Fig. 1.

than Y can consume it, one can reduce the power consumption of the whole system without affecting its smooth operation by properly choosing the buffer size and turning X on and off at regular time intervals. X is turned on when the buffer is empty and turned off when the buffer is full. Since the buffer memory also consumes power, a large buffer may not actually save energy. In our previous works, we show how to determine the optimal size of a single buffer between two components to achieve maximal power reduction for fixed consumption rates [3], [4] and variable consumption rates [5] of Y.

In this paper, we extend our study to a pipeline of streaming data consisting of three components, X, Y, and Z, with two buffers in between, as is shown in Fig. 1(b). When X is on, it produces data at a constant rate α and stores the data in the first buffer; when X is off, no data are generated. When Y is on, it retrieves data from the first buffer, processes them, and stores the resulting data in the second buffer, all at the constant rate β. When Y is off, it does not retrieve any data from the first buffer and stores no new data in the second buffer. Finally, Z is always on and retrieves data from the second buffer at a constant rate γ. For the PDA example, α is the bandwidth of the network card; β represents the processor's decoding performance; and γ is the frame rate of the images. Since the data consumer Z cannot retrieve more data than those that are generated, γ must be smaller than both α and β, i.e., $\alpha > \gamma$ and $\beta > \gamma$. Although in practice, the data rates α, β, and γ may fluctuate with time, in this paper, we assume them to be constant for two reasons: it will greatly simplify the analysis and solution of the problem; the results obtained in the constant-rate case can serve as the basis for analyzing variable-rate case in future work.

Power management of the pipeline system consists of choosing the optimal sizes of the two buffers and the time sequence for turning on/off X and Y so that maximal power reduction of the whole system can be achieved. Compared with the single buffer case, managing two buffers is significantly more difficult because X and Y may need to be turned on and off at different times in a coordinated way to prevent overflowing or underflowing either of the two buffers.

This paper presents a solution to the optimal buffer management problem for the pipeline of streaming data in Fig. 1(b) using the hybrid system control method. A hybrid system is a dynamical system whose state consists of two sets of variables: one set can take only discrete values while the other set changes continuously over time [6], [7], [8], [9], [10]. For the buffer management problem, the power state of each component can take discrete values (either on or off), while the amounts of data stored in the two buffers change continuously over time at rates dependent on the components' power states. Thus, the whole system can be modeled by a hybrid system. The problem of optimal buffer management for maximal power reduction then becomes an optimal control problem for this hybrid system, i.e., finding the best shape of the state space of the hybrid system by properly choosing the sizes of the two buffers and a trajectory in the state space that minimizes the cost function of average power consumption. For simplicity, we focus on periodic solutions, which correspond to switching on/off components X and Y at regular time intervals so that the amounts of data stored in the two buffers rise and decline periodically. We establish several necessary conditions for optimal periodic solutions. Using these conditions, we then consider the special case when X and Y can be turned on and off at most once during each period, and derive the optimal solutions for the cases of $\alpha > \beta$ and $\alpha < \beta$. In particular, we show that, when $\alpha > \beta$, an optimal solution can never have X on and Y off at the same time. If $\alpha < \beta$, an optimal solution can never have X off and Y on at the same time.

This paper has the following contributions. (a) The optimal buffer management for power savings is extended to pipelines of streaming data consisting of two buffers and three components. We believe that this is the first study on managing two buffers simultaneously. (b) We demonstrate that the buffer management for power reduction can be effectively modeled as a hybrid system. (c) Several important properties of optimal periodic solutions are proved. (d) We obtain the optimal periodic solutions when X and Y can turn on and off at most once in a period. Simulations using data from commercial hardware components show that substantial amounts of power can be saved over heuristic buffering methods.

The rest of the paper is organized as follows. In Section II, we review the background of several topics relevant to this study. The problem of optimal buffer management is formulated in Section III. In Section IV, we focus on periodic solutions and derive necessary conditions for optimal solutions. These conditions enable us to obtain the optimal solutions in Section V for a constrained case. Simulation results are presented in Section VI. Finally, in Section VII, we outline the conclusions and some possible future directions.

II. Background

A. Dynamic Power Management

Dynamic power management reduces the power consumption of electronic systems by dynamically changing the components' power states [11]. For example, many computer users set timers to turn off displays when the computers are idle. A processor slows down when high performance is not required. Dynamic frequency/voltage scaling is another form of power management [12], [13], [14], [15]. Gurumurthi et al. [16] propose to slow down the rotational speed of a disk's platters for saving power. Among the studies in the literature, many are devoted to reducing hard disks' power [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27]. Disks attract the attention of the researchers for three reasons. First, disks are not always needed. When data are stored in memory, disks become idle and can be spun down. Second, a disk usually consumes more than 10% of a computer's overall power [28] and thus is a good candidate for power reduction. Third, spinning up (also called awakening) a disk consumes a significant amount of energy and time. Hence, a shutdown policy needs to predict the idle periods of the disk accurately to avoid incurring excessive energy overhead for spinning up the disk.

In some cases, a hardware component may not stay idle long enough to warrant a shut down as the energy savings do not compensate for the energy overhead incurred by the shut down and the awakening. For such a component, a data buffer is often inserted to create longer idle periods. For example, Bertozzi et al. [29] propose to add a data buffer for a network interface card so that the card can remain sleeping most of the time. The network card is awakened before the buffer is full, and starts retrieving the data in the buffer. When the buffer is emptied, the network card returns to a low-power sleeping state. They suggest using a buffer as large as possible to amortize the awakening energy. However, when the buffer's own power consumption is considered, a larger buffer is not necessarily better than a smaller one. In our previous works [3], [4], [5], we demonstrate how to compute the optimal buffer sizes by considering the awakening energy, the data rates, and the buffer's power per unit size, for one buffer between two components with constant data rates α and β (see Fig. 1(a)). In [5], we also describe how to handle variable consumption rate β with a given probability density function.

B. Memory Power Consumption

Dynamic random-access memory (DRAM) is organized into banks. Modern DRAMs provide low-power states where each individual bank may be put into a different mode. The size of the banks and the number of low-power states vary by technology. Bank sizes of 16 MB with at least one low-power mode are typical for both synchronous DRAM (SDRAM) and Rambus DRAM (RDRAM). A bank must be in the active state to provide data. Non-active banks may be put into states where data are retained or destroyed. Fig. 2 shows the power states for a 128 megabit RDRAM [30]. Valid state transitions are

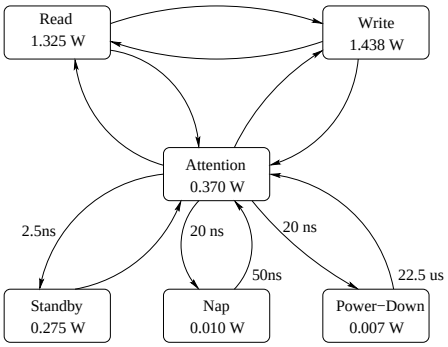

Fig. 2.

represented by arrows, with the latency for each state transition provided near the corresponding arrow. If a transition arrow does not have a latency, the transition is assumed to occur immediately. RDRAM has three active states: read, write, and attention. The attention state is the mode where the device is active but not currently serving a request. Upon receiving a read or a write request, the device transitions to the appropriate state. The device supports three low-power states: standby, nap, and power-down. All of these states retain data while consuming less power than the attention state. However, they all incur state transition latencies. The standby state has a 2.5 ns latency to enter from the attention state but responds immediately to requests. The nap and power-down states both require 20 ns to enter, and 50 ns and 22.5 μs respectively to respond to requests. Many memory devices are configured to enter the standby state immediately after serving requests. When transitioning between states, the power consumption is equal to that of the attention state. Because most memory buses are synchronous, a significant delay may cause bus errors, requiring operating system's attention to restart the operation [31]. These bus errors may occur whether memory power management is handled in software or in hardware. Hence, time and power overheads are associated with state transitions.

Research has been dedicated to creating opportunities for reducing memory power consumption. Lebeck et al. [32] create power-aware page allocation schemes that adjust the Direct RDRAM's power state based on the required amount of physical memory. Pisharath et al. [33] reduce energy consumption within the memory hierarchy by using "energy saver buffers." These buffers hold data waiting to be written back to the next higher level of memory until the memory is returned to an active state. Delaluz et al. [31] propose using a compiler to add instructions to change memory states. Hu et al. [34] propose a model called "cache decay," where cache lines are put into a sleep state after a preset timeout interval. Hu et al. note that cache lines are usually subject to bursty activities shortly after being loaded and remain inactive for an extended time before being overwritten by new data. Chen et al. [35] use garbage collectors within the Java Virtual Machine to reduce the amount of live memory within the heap. Ramachandran et al. [36] turn off sections of main memory while decoding MPEG-2 video to save power. In this paper, we first assume that the memory size can be adjusted at the bit level. Since practical memory sizes can be adjusted in discrete units only (usually in a 16 MB unit), the actual optimal memory size should be one of the discrete sizes closest to the one calculated by our approach, as explained in Section V-E.

C. Hybrid Systems

Proposed to model dynamical systems with both continuous and discrete dynamics, hybrid systems are finding increasing applications in a diverse range of scientific and engineering problems. Examples include transportation systems such as Air Traffic Management (ATM) systems [37] and automated highway systems [38], robotics [39], computer and communication networks [40], [41], and auto and industrial systems [42]. Besides engineering applications, hybrid systems are also useful in modeling biological systems [43], [44].

In our recent work [45], we present some preliminary results on applying a version of the hybrid systems model, called *multirate hybrid systems*, in the study of the dynamic power management problem for a pipeline of streaming data. In this model, the discrete state of the hybrid system consists of the power states of the individual components, and can assume a finite number of possible values (modes). The continuous state consists of the amounts of data stored in the buffers and can take continuous values. The continuous state changes at different constant rates dependent on the current discrete

mode, hence the name multirate hybrid systems. The data flow process in Fig. 1(b) can be modeled as a solution to this hybrid system, and the optimal buffer management problem reduces to finding the solution minimizing a certain cost function. A distinguishing feature of our method is that, while finding the optimal control strategy, we also study the optimal shape and size of the state space of the hybrid systems *at the same time*. The simultaneous design of the optimal control strategy and the state space makes the problem much more difficult. However, it also poses a potentially rewarding research direction.

III. Problem Formulation

Fig. 1(b) shows a pipeline of streaming data consisting of three components with two buffers in between. The first component X generates data at the rate α and stores the data in the first buffer. The second component Y retrieves data from the first buffer, processes them, and stores the resulting data in the second buffer, all at the rate β. The third component Z retrieves data from the second buffer at the rate γ. In this paper, we assume that α, β, and γ are constants. Possible extensions to the nonconstant rates case will be discussed in Section VII. In Section V-D, we will also briefly discuss the extension of our model to the more general case of compression or decompression by Y, i.e., Y retrieves data from the first buffer at a rate β different from the rate β' at which it stores the processed data into the second buffer.

The power consumed by each component consists of two parts: the dynamic power and the static power. For a component, its dynamic power includes the power to produce, process, and/or consume data, as well as the power for writing data into and retrieving data from the buffer memory. As the name suggests, the dynamic power depends on the data flow rate through the component. In contrast, the static power of the component is the part of power to sustain its normal operation, and is independent of the data rate. In this paper, we consider only the static power, as it is shown in [3], [4], [5] that the dynamic power does not affect the solution of the optimal buffer management problem to be defined later in this section. However, the dynamic power does affect the amount of possible power savings. In addition to component power, the two buffers also consume power that are proportional to their sizes. Denote by Q_1 and Q_2 the sizes of the two buffers. Then their power are m_1Q_1 and m_2Q_2, respectively, where the constants m_1 and m_2 are the power per unit size of the buffer memory (Watt per Mega byte). If the two buffers use the same technology, we have $m_1 = m_2$. For example, for Rambus direct RDRAM, the data for m_1 and m_2 can be looked up in the data sheet [30]. We remark here that the linear power model adopted in this paper is a simplified yet effective model commonly used in the dynamic power management literature (see, e.g., [31], [34], [32], [46]).

We assume that $\alpha > \gamma$ and $\beta > \gamma$, so that enough data can stream through the pipeline for Z to consume. This assumption makes it possible to implement dynamic power management for the pipeline system. During the process, component Z is assumed to be turned on all the time, while each of the components X and Y is turned on and off from time to time based on the amount of data stored in the buffer between the component and its downstream neighbor. For example, X may be turned on when the buffer between X and Y is empty, and turned off when the buffer is full. The optimal buffer management problem consists of finding the sequence of times for turning on and off the components X and Y so that the power consumed by the overall system is minimized, subject to the constraint that neither of the two buffers is underflowed or overflowed.

More precisely, each of the components X and Y has two power states: on and off. When X and Y are on, they consume the (static) power p_x and p_y, respectively. When they are off, they consume no power. Changing the power states of these two components may incur extra energy. For simplicity, we assume that the energy overheads for turning on X and Y are k_x and k_y, respectively, while their turn-off energies are both zero. We also assume that any delay in changing the power states of X and Y is amortized by an early wakeup policy because the system is deterministic. Since the last component Z is assumed to be always turned on, it consumes a constant

power no matter how X and Y change power states. Thus we can ignore Z's power when minimizing the overall system power consumption.

From the above description, the pipeline system operates in one of four possible modes: 00, 01, 10 and 11. The meanings of these four modes together with the corresponding total power consumed by the components in each mode are summarized below:

- Mode 00: X and Y are both off. The total component power is $p_{00} = 0$;
- Mode 01: X is off and Y is on. The total component power is $p_{01} = p_y$;
- Mode 10: X is on and Y is off. The total component power is $p_{10} = p_x$;
- Mode 11: X and Y are both on. The total component power is $p_{11} = p_x + p_y$.

Denote by σ the system mode, and by $S = \{00, 01, 10, 11\}$ the set of all possible modes.

Let q_1 be the amount of data stored in the buffer between X and Y, and q_2 be the amount of data stored in the buffer between Y and Z. Depending on the mode, q_1 and q_2 change at different constant rates. For example, when both X and Y are on, i.e., when the system is in mode 11, q_1 increases at the rate $\alpha - \beta$. Thus data are actually accumulating in the first buffer if $\alpha > \beta$, and depleting if $\alpha < \beta$. On the other hand, q_2 increases at the rate $\beta - \gamma$. Combining q_1 and q_2 into a single two dimensional vector $q = (q_1, q_2)$, we conclude that the rate of change of q is $(\alpha - \beta, \beta - \gamma)$ when the system is in mode 11. We denote this rate by v_{11}. Similarly, we can obtain the rates of change of q in other modes:

$$\begin{aligned} v_{00} &= (0, -\gamma), & v_{01} &= (-\beta, \beta - \gamma), \\ v_{11} &= (\alpha - \beta, \beta - \gamma), & v_{10} &= (\alpha, -\gamma). \end{aligned} \tag{1}$$

The four vectors in equation (1) are depicted in Fig. 3, with the subfigures (a) and (b) corresponding to the cases when $\alpha > \beta$ and $\alpha < \beta$, respectively.

The mode σ and the variable q together specify the system state of the pipeline of streaming data. Note that $q = (q_1, q_2)$ is a continuous variable taking values in the rectangle $[0, Q_1] \times [0, Q_2]$, while σ is a discrete variable taking values in S. Thus the pipeline system can be modeled as a hybrid system with state $z = (q, \sigma)$, where q is the continuous state and σ is the discrete state of the hybrid system. Let $[0, t_f]$ be the time interval during which the average system power consumption needs to be minimized. Then a power management scheme for the pipeline system is given by a mapping $\sigma : [0, t_f] \to S$ so that the power states of the components X and Y at time $t \in [0, t_f]$ are specified by the mode $\sigma(t)$. Typically, there exists a sequence of time intervals $[t_i, t_{i+1})$, $i = 0, \ldots, n-1$, with $t_0 = 0 \le t_1 \le \cdots \le t_n = t_f$, such that the system mode $\sigma(t)$ during the time interval $[t_i, t_{i+1})$ is a constant σ_i, and that immediately adjacent time intervals correspond to different modes. In other words, $\sigma(t)$ is piecewise constant with discrete jumps at a finite number of time epochs. Under this scheme $\sigma(t)$, the continuous state $q(t)$ follows a continuous trajectory given by the differential equation:

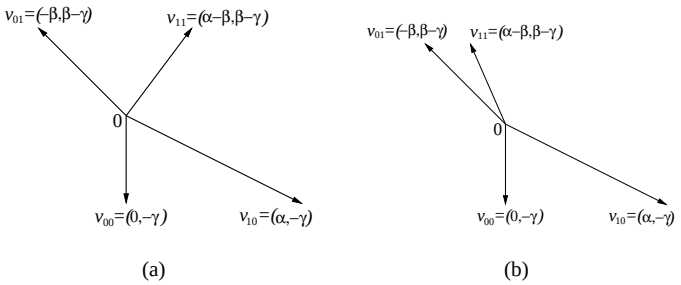

Fig. 3.

$$\frac{dq(t)}{dt} = v_{\sigma(t)}, \quad t \in [0, t_f]. \tag{2}$$

In other words, the instantaneous rate of change for q at time t is given by one of the four vectors defined in (1) corresponding to the system mode σ at time t. Note that in equation (2) the differentiation is defined for the duration $[0, t_f]$ except the finite number of time epochs t_i, $1 \le i \le n$. Together, $z(t) = (q(t), \sigma(t))$ forms a (hybrid) solution, or an execution, of the hybrid system.

Fig. 4 shows two examples of solutions. Note that only the continuous parts $q(t)$ of the solutions are plotted. In both cases, the system starts with both buffers empty, i.e., $(q_1, q_2) = (0, 0)$, and both X and Y off. In Fig. 4(a), at the beginning, both

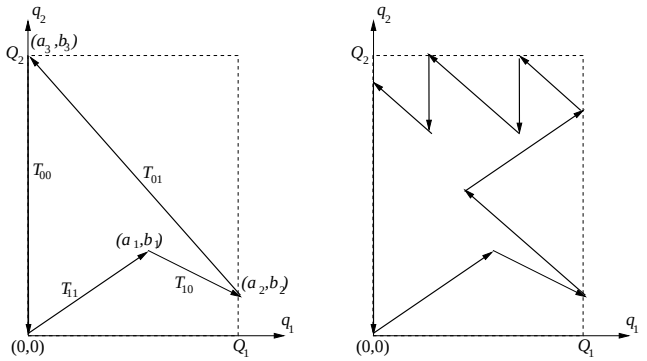

Fig. 4.

X and Y are turned on for a certain time, thus (q_1, q_2) changes along the arrow pointing towards the upper right direction. After (q_1, q_2) reaches the point (a_1, b_1), Y is turned off first while X remains on. Since X is still on, q_1 keeps increasing but q_2 begins to decrease, and (q_1, q_2) follows the direction of v_{10}. When (q_1, q_2) reaches the point (a_2, b_2) with $a_2 = Q_2$, i.e., when the first buffer is full, X is turned off and Y is turned on. Then (q_1, q_2) follows the direction of v_{01} until it reaches $(a_3, b_3) = (0, Q_2)$, in other words, until the first buffer becomes empty and the second buffer becomes full at the same time. Finally, Y is turned off (X remains off), and (q_1, q_2) returns to the starting point $(0, 0)$. During the process, X and Y are each turned on once. In Fig. 4(b), a more complicated solution is plotted where X is turned on twice while Y is turned on four times.

In general, for a solution $z(t) = (q(t), \sigma(t))$ during the period $[0, t_f]$ of interest, the energy consumption of the whole pipeline system consists of the following three parts.

- Static energy consumed by X and Y: $\int_0^{t_f} p_{\sigma(t)}\, dt$, where $p_{\sigma(t)}$ is the total component power in mode $\sigma(t)$;
- Energy for changing the power states of the components X and Y: $n_x k_x + n_y k_y$. Here n_x and n_y are integers denoting the numbers of times that X and Y are turned on, respectively, during $[0, t_f]$;
- Energy consumed by the buffer memory: $(m_1 Q_1 + m_2 Q_2) t_f$.

As a result, for the solution $z(t)$, the average power consumption, $\bar{P}$, of the overall system during $[0, t_f]$ is

$$\bar{P}(z; t_f, Q_1, Q_2) = \frac{1}{t_f}\Big[\int_0^{t_f} p_{\sigma(t)}\, dt + n_x k_x + n_y k_y + (m_1 Q_1 + m_2 Q_2) t_f\Big].$$

Note that the arguments of $\bar{P}$ are introduced to highlight the dependence of $\bar{P}$ on t_f, Q_1, and Q_2.

The dynamic power management problem is then to find $z(t)$ that can achieve the minimal average power. In particular, if the time horizon t_f is large enough, the problem becomes:

Problem 1: Find the hybrid trajectories $z(t)$ during $[0, \infty)$ and the buffer sizes Q_1 and Q_2 that minimize the average power $\lim_{t_f \to \infty} \bar{P}(z; t_f, Q_1, Q_2)$.

This is an optimal control problem for the hybrid system modeling the pipeline of streaming data.

Before we proceed to solve the above problem in the next few sections, for clarity, we summarize the key notations used in this paper in Table I. The first column denotes the symbol. The second column contains the section where the symbol is first defined. The third column holds the definition for the symbol within the context of the derivation. The last column states the physical units for the symbol.

Symbol	Page	Definition	Unit
α, β, γ	5	production/consumption rates of X, Y, and Z	MB/s
Q_1, Q_2	5	sizes of the two buffers	MB
$q_1(t)$, $q_2(t)$	6	amounts of data stored in the two buffers at time t	MB
m_1, m_2	5	power per unit size of the two buffers	Watt/MB
k_x, k_y	5	energies to turn on X and Y	Joule
S	6	four possible power states	-
s	8	a particular power state	-
p_s	8	total power of X and Y in power state s	Watt
$\sigma(t)$	6	power state at time t	-
n_x and n_y	7	numbers of times X, Y are turned on	-
$z(t) = (q(t), \sigma(t))$	6	a hybrid solution	-
*	8	(superscript) optimal value	-

TABLE I

IV. Periodic Solution

A hybrid trajectory $z(t) = (q(t), \sigma(t))$ over the time interval $[0, \infty)$ is called *periodic* with period T if it satisfies the condition $q(t+T) = q(t)$ and $\sigma(t+T) = \sigma(t)$ for all $t \geq 0$. For such $z(t)$, the corresponding power state switching strategy $\sigma(t)$ is uniquely determined by its *switching sequence* $(\sigma_0, \ldots, \sigma_{n-1})$ and *switching epochs* $(t_1, \ldots, t_{n-1})$ during the first period $[0, T]$. More precisely, $\sigma(t)$ switches from σ_{i-1} to σ_i at time t_i for $i = 1, \ldots, n-1$. Note that the periodic condition implies that $\sigma_0 = \sigma_n$. For periodic $z(t)$ with period T, the average power $\lim_{t_f \to \infty} \bar{P}(z; t_f, Q_1, Q_2)$ is equal to the average power during the first period $[0, T]$:

$$\begin{aligned}
&\bar{P}(z; T, Q_1, Q_2) \\
&= \frac{1}{T}\left[\sum_{s\in S} T_s p_s + n_x k_x + n_y k_y + (m_1 Q_1 + m_2 Q_2)T\right] \\
&= \frac{1}{T}\left(\sum_{s\in S} T_s p_s + n_x k_x + n_y k_y\right) + m_1 Q_1 + m_2 Q_2.
\end{aligned}$$

Here, $s \in S$ is one of the four possible power states (modes), p_s is the corresponding total component power, and T_s is the total amount of time during $[0, T]$ when the system is in power state s. The sum of all T_s is the length of the period:

$$\sum_{s\in S} T_s = T.$$

Since the solution is periodic, we must have $q(T) = q(0)$, implying that there is no net gain or net loss of the streaming data in one period: all data produced in the period will be consumed subsequently. As v_s is the speed of accumulation (or depletion) of the buffer data in the power state s, the following condition must hold.

Lemma 1 (Constraint on T_s): For periodic $z(t)$,

$$\sum_{s\in S} T_s v_s = 0.$$

This actually imposes two equality constraints on T_s as each v_s is a two dimensional vector.

If we focus on periodic solutions $z(t)$, then Problem 1 reduces to the following problem.

Problem 2: Find a periodic hybrid trajectory $z(t)$ with a proper period T and the proper buffer sizes Q_1 and Q_2 that minimize the average power $\bar{P}(z; T, Q_1, Q_2)$.

It is conjectured that the average power of the solutions to Problem 2 is the same as that of the solutions to Problem 1. Assuming this conjecture is true, we will try to solve Problem 2 in this paper. Another justification for this choice is that periodic strategies can be implemented practically.

In the following, we will derive several necessary conditions for the optimal solutions to Problem 2.

A. Scaling of Trajectories

Let $z(t) = (q(t), \sigma(t))$ be a periodic hybrid trajectory with period T for buffer sizes of Q_1 and Q_2. Let $\lambda > 0$ be a positive number. Then, it can be verified that $\lambda q(t/\lambda)$ is a continuous trajectory in $[0, \lambda Q_1] \times [0, \lambda Q_2]$ satisfying equation (2) with $\sigma(t)$ replaced by $\sigma(t/\lambda)$. In other words, $(\lambda q(t/\lambda), \sigma(t/\lambda))$ is a periodic hybrid trajectory with the period λT when the sizes of the two buffers become λQ_1 and λQ_2, respectively. The total time that the system spends in each power state s in a single period also scales by a factor of λ, and becomes λT_s. We call $z_\lambda(t) \triangleq (\lambda q(t/\lambda), \sigma(t/\lambda))$ the *scaling* of $z(t) = (q(t), \sigma(t))$ by λ. Compared with the original trajectory, the scaled one follows the same switching sequence. The average power consumed by following $z_\lambda(t)$ is

$$\begin{aligned}
&\bar{P}(z_\lambda; \lambda T, \lambda Q_1, \lambda Q_2) \\
=&\frac{1}{\lambda T}\left(\sum_{s\in S} \lambda T_s p_s + n_x k_x + n_y k_y\right) + m_1 \lambda Q_1 + m_2 \lambda Q_2 \\
=&\lambda(m_1 Q_1 + m_2 Q_2) + \frac{n_x k_x + n_y k_y}{\lambda T} + \frac{1}{T}\sum_{s\in S} T_s p_s \\
\geq&2\sqrt{\frac{1}{T}(m_1 Q_1 + m_2 Q_2)(n_x k_x + n_y k_y)} + \frac{1}{T}\sum_{s\in S} T_s p_s,
\end{aligned}$$

where in the last inequality the equality holds if and only if λ takes the value

$$\lambda^* = \sqrt{\frac{n_x k_x + n_y k_y}{(m_1 Q_1 + m_2 Q_2)T}}. \tag{3}$$

Note that in deriving the above inequality, we apply the arithmetic and geometric means inequality: for $a, b \geq 0$, $\frac{1}{2}(a+b) \geq \sqrt{ab}$ with equality if and only if $a = b$.

We call the value λ^* in (3) the optimal scaling factor. The optimality of λ^* implies that

$$\bar{P}(z_{\lambda^*}; \lambda^* T, \lambda^* Q_1, \lambda^* Q_2) \leq \bar{P}(z_\lambda; \lambda T, \lambda Q_1, \lambda Q_2),$$

for any $\lambda > 0$. In particular, if we choose $\lambda = 1$, then

$$\bar{P}(z_{\lambda^*}; \lambda^* T, \lambda^* Q_1, \lambda^* Q_2) \le \bar{P}(z; T, Q_1, Q_2),$$

where equality holds if and only if $\lambda^* = 1$.

Thus, for any hybrid trajectory z, unless the optimal scaling factor $\lambda^* = 1$, we can always scale it by λ^*, as well as the buffer sizes, to obtain a different hybrid trajectory z_{λ^*} whose average power is smaller than that of z. Thus a necessary condition for z to be an optimal solution is that $\lambda^* = 1$. By (3), this reduces to

Lemma 2 (Constraint on Q_1, Q_2, k_x, k_y and T): A solution $z(t)$, Q_1, and Q_2 to Problem 2 satisfies

$$n_x k_x + n_y k_y = (m_1 Q_1 + m_2 Q_2) T. \tag{4}$$

We can consider an extreme case when $k_x = k_y = 0$. Then condition (4) implies that the optimal buffer sizes are $Q_1 = Q_2 = 0$. In other words, since there is no penalty in turning on and off the two components, one can do so infinitely often so that their effective data rates are equal to γ. On the other hand, if k_x (or k_y) is set very high, then either n_x (or n_y) is zero so that the corresponding component does not switch at all, or in the case of nonzero n_x (or n_y), the optimal buffer size Q_1 (or Q_2) should be made very large, reducing the switching frequency of the corresponding component to very low.

As a result of (4), for any solution $z(t)$, Q_1, and Q_2 to Problem 2 with period T, we have

$$\begin{aligned}\bar{P}(z; T, Q_1, Q_2) &= \frac{2}{T}(n_x k_x + n_y k_y) + \frac{1}{T}\sum_{s\in S} T_s p_s \\ &= 2(m_1 Q_1 + m_2 Q_2) + \frac{1}{T}\sum_{s\in S} T_s p_s.\end{aligned}$$

B. Tightness of Optimal $q(t)$ in Q

Since a buffer consumes power proportional to its size, unused buffer memory will unnecessarily waste power. As a result, an optimal trajectory z should fill and empty each of the two buffers at least once sometime during the period (otherwise, the unused memory can be removed to save power). This is summarized by the following lemma.

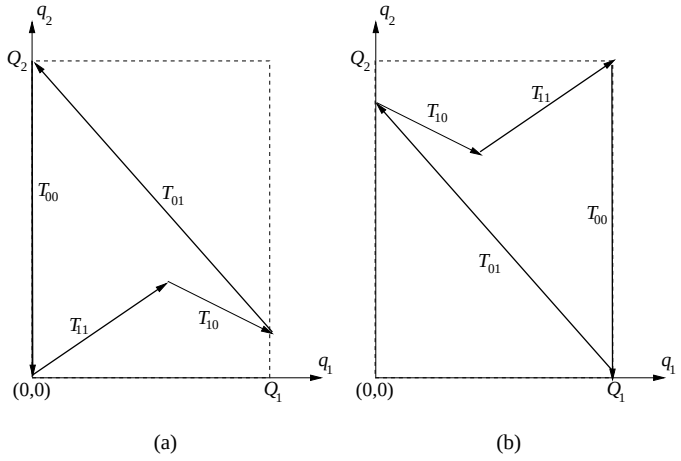

Fig. 5.

Lemma 3 (Tightness in Q): A solution $z(t) = (q(t), \sigma(t))$, Q_1, and Q_2 to Problem 2 satisfies that

$$\min_{t\in T} q_1(t) = 0, \quad \max_{t\in T} q_1(t) = Q_1,$$
$$\min_{t\in T} q_2(t) = 0, \quad \max_{t\in T} q_2(t) = Q_2.$$

Geometrically, Lemma 3 says that $q(t)$ as a curve in $Q = [0, Q_1] \times [0, Q_2]$ contacts all four edges of Q.

C. Reversing the Switching Sequence

Suppose that $z(t) = (q(t), \sigma(t))$ is a periodic hybrid trajectory with period T for buffer sizes Q_1 and Q_2, and with switching sequence $(\sigma_0, \ldots, \sigma_{n-1})$ and switching epochs $(t_1, \ldots, t_{n-1})$ in the first period $[0, T]$. We can obtain a new periodic hybrid trajectory $\hat{z}(t) = (\hat{q}(t), \hat{\sigma}(t))$ with the same period T and buffer sizes Q_1 and Q_2 but a reversed switching sequence $(\sigma_{n-1}, \ldots, \sigma_0)$ as follows:

$$\hat{q}(t) = (Q_1 - q_1(T-t), Q_2 - q_2(T-t)),\ \hat{\sigma}(t) = \sigma(T-t),$$

for $t \in [0, T]$. Fig. 5 illustrates two example trajectories that can be obtained from each other by reversing the sequences.

It is easily verified that $\hat{z}(t)$ is also a valid periodic trajectory. Note that $z(t)$ and $\hat{z}(t)$ may have different starting points. Indeed, $z(0)$ and $\hat{z}(0)$ are symmetric with respect to the center of Q. Moreover,

Lemma 4: Hybrid trajectories $z(t)$ and $\hat{z}(t)$ have the same average power: $\bar{P}(z; T, Q_1, Q_2) = \bar{P}(\hat{z}; T, Q_1, Q_2)$.

As a result, solutions to Problem 2 are not unique: for each solution $z(t)$, Q_1, and Q_2 to Problem 2, there is another

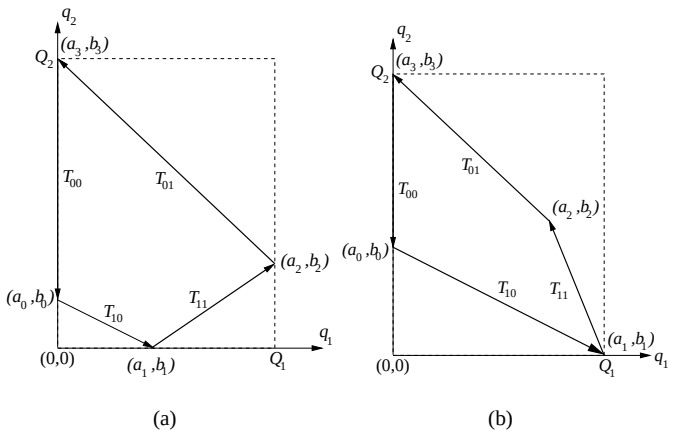

Fig. 6.

solution $\hat{z}(t)$, Q_1, and Q_2, where $\hat{z}(t)$ has the same period but the reversed switching sequence compared with $z(t)$.

V. Periodic Solution with $n_x = n_y = 1$

In Section IV, three properties of the optimal periodic solutions to the dynamic power management problem (Problem 2) are established. In this section, we consider the solutions under the constraint that X and Y each switches from off to on exactly once during one period, i.e., $n_x = n_y = 1$. This simplifying restriction is adopted so that we can obtain the analytic form of the optimal solutions. Optimal solutions with $n_x > 1$ or $n_y > 1$ will be discussed in future work.

Under the constraint $n_x = n_y = 1$, we can assume without loss of generality that the switching sequence of the optimal solutions $z(t)$ to Problem 2 is $(10, 11, 01, 00)$. Indeed, all the other sequences with $n_x = n_y = 1$ can be obtained from this sequence by one or more of the following operations.

1) Cyclic rotations, such as $(10, 11, 01, 00) \rightarrow (11, 01, 00, 10)$, which correspond to different phases of the same periodic hybrid trajectory;
2) Reversions, such as $(10, 11, 01, 00) \rightarrow (00, 01, 11, 10)$, as are described in Section IV-C;
3) Sub-sequences, for example, $(10, 11, 01, 00) \rightarrow (10, 11, 00)$, where the latter can be thought of as the degenerate case of the former with the segment representing the state 01 degenerating to zero.

Fig. 6 shows two generic periodic solutions with $n_x = n_y = 1$ and the switching sequence $(10, 11, 01, 00)$ for the cases $\alpha > \beta$ and $\alpha < \beta$, respectively. By Lemma 3, the rectangle $[0, Q_1] \times [0, Q_2]$ must bound $q(t)$ tightly. In the case $\alpha > \beta$, this implies that $a_0 = 0$, $b_1 = 0$, $a_2 = Q_1$ and $b_3 = Q_2$; while in the case $\alpha < \beta$, this implies that $a_0 = 0$, $b_1 = 0$, $a_1 = Q_1$ and $b_3 = Q_2$. The average power $\bar{P}$ to be minimized is

$$\bar{P} = \frac{T_{01}p_{01} + T_{11}p_{11} + T_{10}p_{10} + k_x + k_y}{T} + \sum_{i=1}^{2} m_i Q_i. \quad (5)$$

A. Optimal Solutions When $\alpha > \beta$

We first consider the case $\alpha > \beta$. By Lemma 3, we know that in Fig. 6(a), $a_0 = 0$, $b_1 = 0$, $a_2 = Q_1$ and $b_3 = Q_2$. In addition, $a_3 = 0$ since the segment corresponding to the power state 00 is vertical. The trajectory $q(t)$ is completely determined by the two parameters Q_1 and Q_2. In fact, the point $(a_2, b_2) = (Q_1, b_2)$ and the time T_{01} that $q(t)$ spends in power state 01 during one period can be determined from the relation $(Q_1, b_2) = (0, Q_2) - v_{01}T_{01}$ as

$$b_2 = Q_2 - \frac{\beta - \gamma}{\beta} Q_1, \quad T_{01} = \frac{Q_1}{\beta}.$$

Since $(Q_1, b_2) = (a_1, 0) + v_{11}T_{11}$, we deduce that

$$a_1 = \frac{\alpha}{\beta} Q_1 - \frac{\alpha - \beta}{\beta - \gamma} Q_2, \quad T_{11} = \frac{Q_2}{\beta - \gamma} - \frac{Q_1}{\beta}.$$

Furthermore, since $(a_1, 0) = (0, b_0) + v_{10}T_{10}$, we have

$$b_0 = \frac{\gamma}{\beta} Q_1 - \frac{\gamma(\alpha - \beta)}{\alpha(\beta - \gamma)} Q_2, \quad T_{10} = \frac{Q_1}{\beta} - \frac{(\alpha - \beta)Q_2}{\alpha(\beta - \gamma)}.$$

As a result of the above equation, since $T_{10} \geq 0$ is nonnegative, Q_1 must satisfy

$$Q_1 \geq \frac{\beta(\alpha - \beta)}{\alpha(\beta - \gamma)} Q_2, \quad (6)$$

which gives a lower bound of Q_1 for given Q_2 that will be used later in this section. Finally, from the relation $(0, b_0) = (0, Q_2) + v_{00}T_{00}$, we can compute the time that $q(t)$ spends in power state 00 during one period as:

$$T_{00} = -\frac{Q_1}{\beta} + \frac{\beta(\alpha - \gamma)Q_2}{\alpha\gamma(\beta - \gamma)}.$$

Combining the above results, the period of $q(t)$, $T = T_{00} + T_{01} + T_{11} + T_{10}$, is given by

$$T = \frac{\beta}{\gamma(\beta - \gamma)} Q_2,$$

and the average power $\bar{P}$ of $q(t)$ within one period as is given by (5) can be simplified to

$$\bar{P} = m_1 Q_1 + m_2 Q_2 + \frac{C_1}{Q_2} + C_2, \tag{7}$$

where C_1 and C_2 are positive constants defined by

$$C_1 = \frac{\gamma(\beta-\gamma)(k_x+k_y)}{\beta}, \quad C_2 = \frac{\gamma}{\alpha}p_x + \frac{\gamma}{\beta}p_y. \tag{8}$$

To find the optimal Q_1^* and Q_2^* that minimize $\bar{P}$ in (7), the optimization problem is solved in two steps. First, for each fixed Q_2, we find the value of Q_1 that minimizes $\bar{P}$. Since $\bar{P}$ is linear in Q_1, the best Q_1 is given by its smallest possible value, subject to the constraint (6). Then, after substituting the best value of Q_1 into $\bar{P}$, we find the optimal value of Q_2 that minimizes the resulting expression. Specifically,

$$\begin{aligned}\min_{Q_1,Q_2} \bar{P} &= \min_{Q_2} \min_{Q_1} \bar{P} = \min_{Q_2} \bar{P}\big|_{Q_1=\frac{\beta(\alpha-\beta)}{\alpha(\beta-\gamma)}Q_2} \\ &= \min_{Q_2} \left\{ \left[\frac{\beta(\alpha-\beta)}{\alpha(\beta-\gamma)} m_1 + m_2 \right] Q_2 + \frac{C_1}{Q_2} + C_2 \right\}.\end{aligned}$$

By applying the arithmetic and geometric means inequality, we deduce that the average power $\bar{P}$ is minimized when the value of Q_2 is chosen as $Q_2^* = \sqrt{\frac{C_1}{\frac{\beta(\alpha-\beta)}{\alpha(\beta-\gamma)}m_1+m_2}}$, i.e.,

$$Q_2^* = (\beta-\gamma)\sqrt{\frac{\alpha\gamma(k_x+k_y)}{\beta^2(\alpha-\beta)m_1 + \alpha\beta(\beta-\gamma)m_2}}, \tag{9}$$

where Q_2^* denotes the optimal value of Q_2. From (6), we can obtain the optimal value of Q_1 as $Q_1^* = \frac{\beta(\alpha-\beta)}{\alpha(\beta-\gamma)}Q_2^*$, i.e.,

$$Q_1^* = (\alpha-\beta)\sqrt{\frac{\beta\gamma(k_x+k_y)}{\alpha\beta(\alpha-\beta)m_1 + \alpha^2(\beta-\gamma)m_2}}. \tag{10}$$

Thus, the corresponding minimum average power $\bar{P}^*$ is

$$\begin{aligned}\bar{P}^* =& 2\sqrt{\frac{\gamma}{\alpha\beta}(k_x+k_y)\left[\beta(\alpha-\beta)m_1 + \alpha(\beta-\gamma)m_2\right]} \\ &+ \frac{\gamma}{\alpha}p_x + \frac{\gamma}{\beta}p_y,\end{aligned} \tag{11}$$

and the optimal period T^* is

$$T^* = \sqrt{\frac{\alpha\beta(k_x+k_y)}{\beta\gamma(\alpha-\beta)m_1 + \alpha\gamma(\beta-\gamma)m_2}}. \tag{12}$$

It is easily checked that Q_1^*, Q_2^*, and T^* satisfy condition (4) in Lemma 2. Since the optimal Q_1^* and Q_2^* are such that equality holds in (6), we conclude that $T_{10}^* = 0$. Hence, in the optimal solution z^*, the segment corresponding to the power state $s = 10$ vanishes. Fig. 7(a) illustrates such an optimal trajectory z^* that starts with both buffers empty (i.e., q^* starts from the origin) and both components X and Y turned on (i.e., the starting power state is $s = 11$). Since $\alpha > \beta > \gamma$, data will accumulate in both buffers. After T_{11}^* time, the first buffer reaches its capacity Q_1^*. Then, X switches off, and the data in the first buffer starts decreasing while data in buffer 2 keeps accumulating. After T_{01}^* time, the first buffer is emptied and the second buffer reaches its maximal capacity Q_2^*. Then Y switches off, and data in the second buffer are gradually fetched by Z until the buffer is empty. This process is then repeated every T^* time.

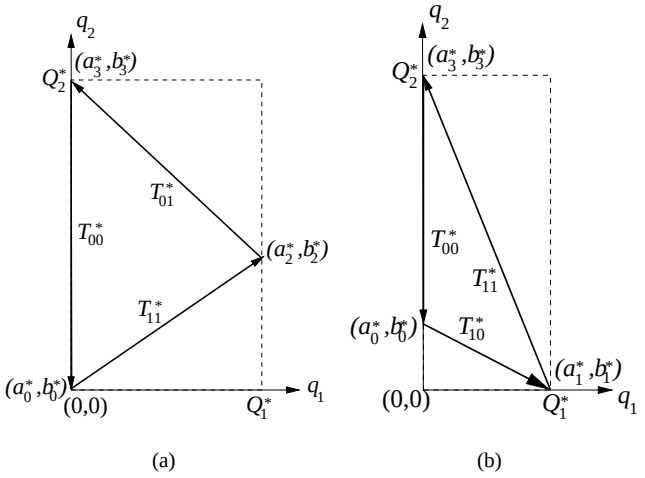

Fig. 7.

B. Optimal Solution When $\alpha < \beta$

We now consider the case $\alpha < \beta$. In this case, we have $a_0 = 0$, $b_1 = 0$, $Q_1 = a_1$ and $Q_2 = b_3$, and the generic trajectory $q(t)$ plotted in Fig. 6(b) is fully determined by the two parameters Q_1 and Q_2. Indeed, the vertex $(a_0, b_0) = (0, b_0)$ can be solved from the relation $(0, b_0) = (Q_1, 0) - T_{10}v_{10} = (0, Q_2) + T_{00}v_{00}$ as

$$b_0 = \frac{\gamma}{\alpha}Q_1, \qquad T_{00} = \frac{Q_2}{\gamma} - \frac{Q_1}{\alpha}, \qquad T_{10} = \frac{Q_1}{\alpha}.$$

On the other hand, solving the equations $(a_2, b_2) = (Q_1, 0) + T_{11}v_{11} = (0, Q_2) - T_{01}v_{01}$ yields

$$\begin{aligned} a_2 &= \frac{\beta}{\alpha}Q_1 + \frac{\beta(\alpha-\beta)}{\alpha(\beta-\gamma)}Q_2, \quad b_2 = -\frac{\beta-\gamma}{\alpha}Q_1 + \frac{\beta}{\alpha}Q_2, \\ T_{01} &= \frac{Q_1}{\alpha} + \frac{(\alpha-\beta)Q_2}{\alpha(\beta-\gamma)}, \quad T_{11} = -\frac{Q_1}{\alpha} + \frac{\beta Q_2}{\alpha(\beta-\gamma)}. \end{aligned}$$

Note that, since $T_{01} \geq 0$, we must have

$$Q_1 \geq \frac{\beta - \alpha}{\beta - \gamma} Q_2, \tag{13}$$

which gives a lower bound on Q_1 for fixed Q_2.

The period of $q(t)$ is $T = T_{00} + T_{11} + T_{01} + T_{10} = \frac{\beta}{(\beta-\gamma)\gamma} Q_2$. Thus the average power is reduced to

$$\bar{P} = m_1 Q_1 + m_2 Q_2 + \frac{C_1}{Q_2} + C_2, \tag{14}$$

where C_1 and C_2 are constants defined in (8). Note that $\bar{P}$ has the same expression as in the previous case, despite the different expressions for $T_{00}, T_{01}, T_{10}, T_{11}$, etc. Therefore, we can follow similar steps to find the optimal Q_1^* and Q_2^* that minimize $\bar{P}$. The only difference is that the lower bound on Q_1 for a given Q_2 is now given by (13) instead of (6). We omit the detailed derivation, and write the results as:

$$Q_1^* = (\beta - \alpha)\sqrt{\frac{\gamma(k_x + k_y)}{\beta(\beta-\alpha)m_1 + \beta(\beta-\gamma)m_2}},$$

$$Q_2^* = (\beta - \gamma)\sqrt{\frac{\gamma(k_x + k_y)}{\beta(\beta-\alpha)m_1 + \beta(\beta-\gamma)m_2}}.$$

The corresponding optimal average power $\bar{P}^*$ and optimal period T^* are

$$\bar{P}^* = 2\sqrt{\frac{\gamma(k_x + k_y)}{\beta}\left[(\beta-\alpha)m_1 + (\beta-\gamma)m_2\right]} + \frac{\gamma}{\alpha}p_x + \frac{\gamma}{\beta}p_y,$$

$$T^* = \sqrt{\frac{\beta(k_x + k_y)}{\gamma(\beta-\alpha)m_1 + \gamma(\beta-\gamma)m_2}}.$$

Since equality holds in (13), in Fig. 6(b), the segment corresponding to the power state $s = 01$ vanishes. See Fig. 7(b) for a plot of such an optimal solution z^*. For this z^*, at time 0, buffer 1 is empty while buffer 2 is partially filled; component X is on and component Y is off. Thus, data accumulates in buffer 1 and decreases in buffer 2. After exactly T_{10}^* time, buffer 1 reaches its capacity Q_1^* and buffer 2 is emptied. Then, component Y is switched on. Since $\alpha < \beta$, data in buffer 1 will start decreasing, and data in buffer 2 will start increasing as $\beta > \gamma$. After exactly T_{11}^* time, buffer 1 is emptied again while buffer 2 reaches its capacity Q_2^*. At this time, both X and Y switch off. Data in buffer 2 will be gradually fetched by Z until the amount of data left becomes b_0^* after T_{00}^* time. This process is then repeated every T^* time. In practice, since the periodic solution does not start with both buffers empty, a transient period is needed before it reaches steady state.

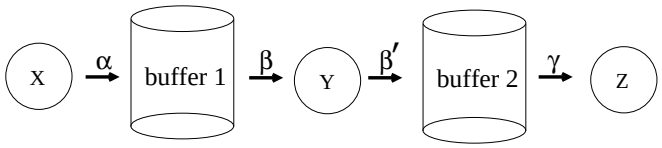

Fig. 8.

C. Optimal Solution When $\alpha = \beta$

This case can be treated as a limiting case of the previous two by letting $\beta \to \alpha$ from either below or above. In either case, we have $Q_1^* = 0$, and $Q_2^* = \sqrt{\frac{\gamma(\beta-\gamma)(k_x+k_y)}{\beta m_2}}$. Explicitly, according to this strategy specified, only the second buffer is needed with the size Q_2^*, and the first buffer has size 0, thus can be removed. At time 0, the second buffer is empty and both X and Y are switched on. Since $\alpha = \beta > \gamma$, data will accumulate in the second buffer until reaching its capacity after T_{11}^* time. Then, both X and Y switch off at that time, and data in buffer 2 will start decreasing until it is fully emptied after T_{00}^* time. Thus, when $\alpha = \beta$, the optimal strategy switches X and Y on (and off) at the same time in a coordinated way.

D. Data Compression and Decompression

So far we have assumed that Y retrieves data from the first buffer and fills data into the second buffer at the same rate β. This model represents no data increase or decrease by Y. In this section, we extend the model to consider the data compression and decompression effect of Y. In the new model, Y retrieves data from the first buffer at the rate β, performs some processing, and fills the data into the second buffer at the rate β', as shown in Fig. 8, where β and β' are known constants. Thus Y compresses data if $\beta > \beta$', and decompresses data if $\beta < \beta$'. To ensure that there are enough data for Z to consume, the constraint that $\alpha > \gamma$ in the original model has to be replaced by $\alpha\frac{\beta'}{\beta} > \gamma$ due to the compression/decompression effect of Y. In addition, the constraint that $\beta > \gamma$ now becomes $\beta' > \gamma$.

We can follow the same procedure presented in Section V-A to calculate the optimal buffer sizes and the length of the

period. The only difference in the procedure is to replace some β with β' in equation (1). Specifically, $v_{01} = (-\beta, \beta' - \gamma)$ and $v_{11} = (\alpha - \beta, \beta' - \gamma)$. When $\alpha > \beta$, the derived expressions for the optimal buffer sizes, the optimal period length, and the optimal average power are,

$$Q_1^* = \beta(\alpha-\beta)\sqrt{\frac{\gamma(k_x+k_y)}{\alpha\beta\beta'(\alpha-\beta)m_1+\alpha^2\beta'(\beta'-\gamma)m_2}},$$

$$Q_2^* = \alpha(\beta'-\gamma)\sqrt{\frac{\gamma(k_x+k_y)}{\alpha\beta\beta'(\alpha-\beta)m_1+\alpha^2\beta'(\beta'-\gamma)m_2}},$$

$$T^* = \sqrt{\frac{\alpha\beta'(k_x+k_y)}{\beta\gamma(\alpha-\beta)m_1+\alpha\gamma(\beta'-\gamma)m_2}},$$

$$\bar{P}^* = 2\sqrt{\frac{\gamma}{\alpha\beta'}(k_x+k_y)(\beta(\alpha-\beta)m_1+\alpha(\beta'-\gamma)m_2)} + \frac{\beta\gamma}{\alpha\beta'}p_x + \frac{\gamma}{\beta'}p_y.$$

Similarly, when $\alpha < \beta$, the optimal values are given by

$$Q_1^* = (\beta-\alpha)\sqrt{\frac{\gamma(k_x+k_y)}{\beta'(\beta-\alpha)m_1+\beta'(\beta'-\gamma)m_2}},$$

$$Q_2^* = (\beta'-\gamma)\sqrt{\frac{\gamma(k_x+k_y)}{\beta'(\beta-\alpha)m_1+\beta'(\beta'-\gamma)m_2}},$$

$$T^* = \sqrt{\frac{\beta'(k_x+k_y)}{\gamma(\beta-\alpha)m_1+\gamma(\beta'-\gamma)m_2}},$$

$$\bar{P}^* = 2\sqrt{\frac{\gamma}{\beta'}(k_x+k_y)((\beta-\alpha)m_1+(\beta'-\gamma)m_2)} + \frac{\beta\gamma}{\alpha\beta'}p_x + \frac{\gamma}{\beta'}p_y.$$

E. Discrete Memory Size

In the previous sections, a linear memory power model is assumed, i.e., it is assumed that the memories can take an arbitrary size, and that there is no static power. In practice, memories are available in only discrete block sizes (such as 16 MB banks), and always consumes static power that is independent of the size. In this section, we study the effect of these practical considerations on the optimal solutions.

Note that the expressions of the average power $\bar{P}$ in (7) and (14) are convex functions of the buffer sizes Q_1 and Q_2. Thus, when Q_1 and Q_2 can only take discrete values that are integer multiples of some unit sizes, or equivalently, when (Q_1, Q_2) takes values in the (scaled) integer grids on the plane, the minimal $\bar{P}$ is achieved at one of the grid points closest to the optimal value (Q_1^*, Q_2^*) as determined using the linear memory model with continuous memory size. For each real number Q, we use $\underline{Q}$ to denote the largest available discrete memory size less than Q and $\overline{Q}$ the smallest available discrete memory size greater than Q. Then, as shown in Fig. 9, the optimal buffer sizes for the discrete memory size model are specified by one of the four grid points closest to (Q_1^*, Q_2^*). Denote by $\mathcal{Q} = \{\{\underline{Q_1^*}, \overline{Q_1^*}\} \times \{\underline{Q_2^*}, \overline{Q_2^*}\}\}$ the set of these four grid points.

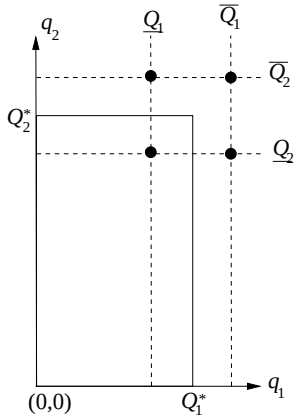

Fig. 9.

Let Q_1 and Q_2 be the optimal available buffer sizes in the discrete memory size model. Then, unless the ratio $\frac{Q_1}{Q_2}$ is the same as $\frac{Q_1^*}{Q_2^*}$, the optimal trajectory $q^*(t)$ obtained in the continuous memory size model can not fit into the rectangle $[0, Q_1] \times [0, Q_2]$ tightly, even after a proper scaling. In this case, the optimal trajectory $q(t)$ for the discrete memory size model can be obtained by scaling $q^*(t)$ so that it is contained in the rectangle $[0, Q_1] \times [0, Q_2]$ and has the largest possible size (though it may not necessarily touch all four edges of the rectangle). This is illustrated in Fig. 10 (a) and (b) for the cases $\frac{Q_1}{Q_2} < \frac{Q_1^*}{Q_2^*}$ and $\frac{Q_1}{Q_2} > \frac{Q_1^*}{Q_2^*}$, respectively. In the first case, the curve $q(t)$ never touches the top edge, implying that there are some unused memory for the second buffer. In the second case, $q(t)$ never touches the right edge of the rectangle, implying that the first buffer is not fully utilized.

Remark. For many pipeline systems already deployed in practice, the sizes of the buffers are pre-determined, and hence cannot be adjusted during the optimal buffer management process. The procedures outlined in the above paragraph on how to fit an optimal solution into a not-necessarily-optimal

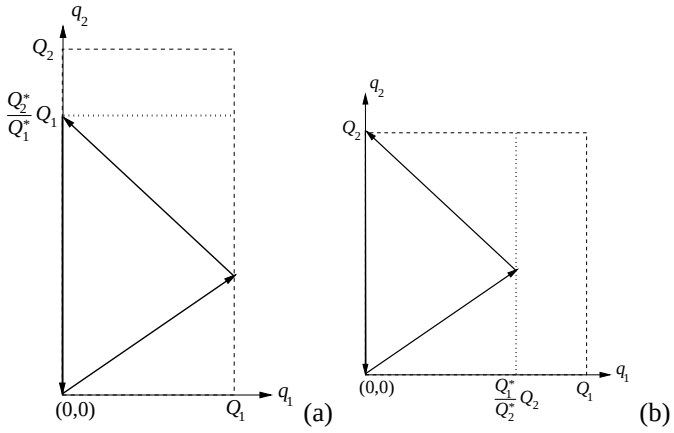

Fig. 10.

state space provides one way of finding power management schemes with reasonable performance in such situations.

To see more clearly which of the four grid points should be chosen as the optimal discrete buffer size for the discrete memory model, for each of the four grid points $(Q_1, Q_2) \in \mathcal{Q}$, we obtain the trajectory $q(t)$ from $q^*(t)$ as described in the previous paragraph. Depending on whether $\frac{Q_1}{Q_2} < \frac{Q_1^*}{Q_2^*}$ or not, the average power consumed by $q(t)$ can be calculated as:

$$\bar{P} = \begin{cases} m_1 Q_1 + m_2 Q_2 + \frac{C_1}{Q_2} + C_2 & \text{if } Q_2 \leq \frac{Q_2^*}{Q_1^*} Q_1, \\ m_1 Q_1 + m_2 Q_2 + \frac{Q_1^*}{Q_2^*}\frac{C_1}{Q_1} + C_2 & \text{otherwise.} \end{cases}$$

The optimal discrete buffer size $(\hat{Q}_2, \hat{Q}_2)$ can then be determined by

$$(\hat{Q}_1, \hat{Q}_2) \quad = \quad \underset{(Q_1,Q_2)\in\mathcal{Q}}{\text{argmin}} \ \bar{P}.$$

We now study the effect of the static power of the buffer memory on the optimal solutions. If the static power of the buffer memory is considered, some extra constants will appear in the C_2 term in (7) and (14) for the average power. As these are constants, they will not affect the optimization results. Thus the optimal buffer sizes and switching strategies remain unchanged. On the other hand, the static power terms will affect the power saving percentage: the higher the static power, the less power savings percentage can be achieved.

VI. Experiments

A. Simulation Setup

We present some simulations to compare the performance of our optimal solutions to heuristic buffer sizes and switching algorithms, and to analyze the effects of block memory sizing. The simulations use parameters from practical hardware components and different formats of streaming data. We consider two different producers (component X): an IBM 1GB Microdrive and an IBM 2.5" TravelStar hard disk, which will be referred to as "Microdrive" and "TravelStar," respectively. We also consider two through-putting elements (component Y): a local-area network (LAN) and a wide-area network (WAN). Both networks are accessed through a Linksys PCMCIA network card so that they have the same power characteristics, while the WAN realizes a slower data rate than the LAN due to more traffic and contention on the network. The parameters for the producing and through-putting elements are obtained from the manufacturer data sheets and physical measurements in [3] and can be found in Table II.

Device	p_p (W)	k (J)	DataRate (MB/s)
TravelStar	2.166	17.100	17.60
Microdrive	0.221	0.151	1.824
LAN	0.500	0.333	5.826
WAN	0.500	0.333	0.443
MPEG-1	—	—	0.1875
MPEG-2	—	—	3.125
MP3	—	—	0.016

TABLE II

Number	X	Y	Z	Q_1^* (MB)	Q_2^* (MB)
1	TravelStar	LAN	MPEG-1	37.658	54.728
2	TravelStar	LAN	MPEG-2	184.533	129.694
3	TravelStar	LAN	MP3	10.904	16.324
4	TravelStar	WAN	MPEG-1	56.566	33.462
5	TravelStar	WAN	MP3	14.780	14.614
6	Microdrive	LAN	MPEG-1	6.450	9.054
7	Microdrive	LAN	MP3	1.807	2.701
8	Microdrive	WAN	MPEG-1	7.894	6.013
9	Microdrive	WAN	MP3	2.030	2.585

TABLE III

We obtain the bandwidth of each disk using the LINUX command hdparm (`hdparm -t`). We first measure the power of the disk without applying any workload to determine the static power of the hard disk. To obtain the dynamic power, we measure the disk's power consumption under different data

read rates (MB/s). After subtracting the static power, we divide the differences by the rates, and use the average value of these quotients as the dynamic energy for reading one MB of data at a typical rate. Together, these two measures can be used to compute the producer's average power p_p when producing data. We determine k by measuring the energy consumed by the hard disk when it is turned on. The results of these measurements are summarized in Table II.

To determine the power characteristics for the Linksys PCMCIA network, we perform measurements with a National Instruments data acquisition card. We send a stream of packets over the network to measure the average power p_p. To find k, we measure the energy between the wake-up command and the first packet's transmission. We consider two applications for this network card: a LAN and a WAN. The production rate of each case and the hardware parameters of the network card are provided in Table II. In particular, to find the production rate of each case, we transfer many large files over the network and determine the average data transmission rate. The results of our measurements together with the hardware parameters of the network card are also shown in Table II. We use the memory model described in Section II-B for RDRAM. When banks of memory are idle, they turn into the nap state. We select the nap state instead of the power down state because transition back into the active state incurs little overhead compared with recovering from the power down state. The power per megabyte for buffers Q_1 and Q_2 can be looked up in the data sheet [30] as 6.258×10^{-4} W/MB. Thus the power consumed by a 16MB memory block is $6.258 \times 10^{-4} \times 16 = 0.010$ W. The data rate of the consumer depends on the type of data being consumed. We consider three kinds of media streams: MPEG-1, MPEG-2 and mp3. A typical data rate for an MPEG-1 video disk is 1.5 Mbps (0.1875 MB/s). MPEG-2 files come in a wide variety of data rates, with a typical rate of 25 Mbps (3.125 MB/s) for an HDTV broadcast. Music is often encoded in mp3 format at 0.128 Mbps (0.016 MB/s).

With all these choices of producing, throughputting, and consuming components, the combinations of components used in our simulations are shown in Table III. Note that we do not consider the cases when the Microdrive must source the MPEG-2 stream, or when the WAN must through-put the MPEG-2 stream, since the producing components do not have high enough data rates to support the stream.

B. Validation of Memory Power Models

In this section, we verify our method for selecting optimal buffer sizes when only discrete sizes are available. Each of the test cases listed in Table III is run for memory resolutions of 2 MB, 4 MB, 8 MB, 16 MB, 32 MB, as well as arbitrary memory size. In each case, the average power achieved by the buffer sizes and the switching strategy derived from the optimal solution as described in Section V-E is computed numerically for each of the memory resolutions. The results are shown in Fig. 11, which exhibit a graceful degradation in power savings as memory block size is increased. Note that the cases with smaller optimal buffer sizes are effected more by the discrete memory sizes than those with larger optimal buffer sizes. Compared with the unbuffered case (whose average power is not included in the plot), our solutions can achieve various degree of power saving. For example, in case 9 with a block size of 32 MB, our solutions achieve 61.1% power savings (.444 W); while in all other cases, the power savings achieved are even greater. In particular, in case 3 with a block size of 2 MB, our solutions achieve 98.5% power savings (2.628 W). This shows that our method is effective even when memory must be chosen in discrete block sizes.

C. Comparison with Heuristic Buffer Sizes

We now investigate how much power our approach can save versus different fixed buffer sizes. For our first buffer, we consider that hard disk caches typically come in sizes of 4 MB and 8 MB. For our second buffer, we consider the LINUX mplayer media player that uses a buffer size of up to 1 MB, and a player that buffers for the length of a long mp3, 8 MB. Since these buffer sizes are chosen heuristically and may not be optimal, we design a heuristic switching policy: each buffer is filled until it is full, then emptied until it is empty using the 3 vectors that the optimal policy uses. It is worth noting that

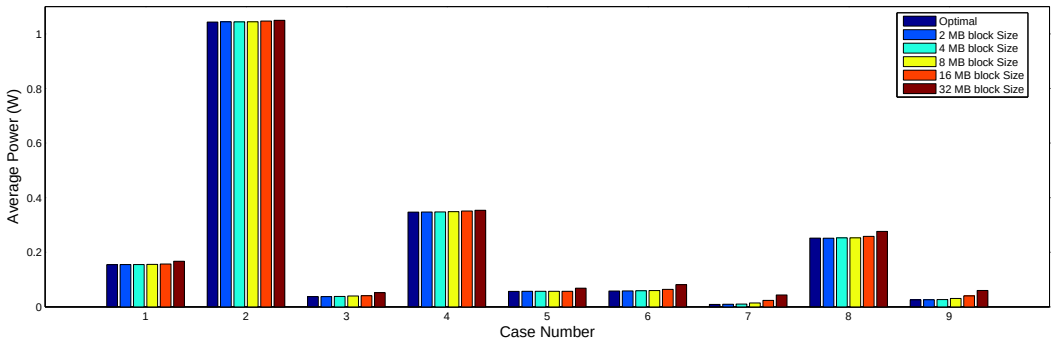

Fig. 11.

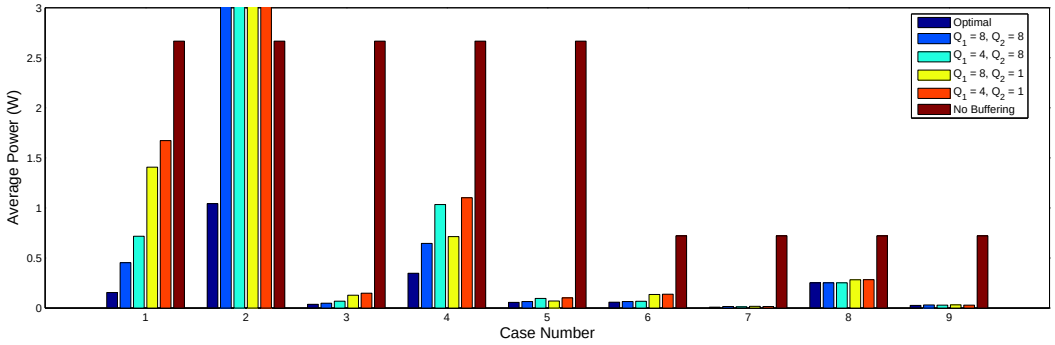

Fig. 12.

when optimal buffer sizes and equivalent initial conditions are chosen, this policy is reduced to our optimal one.

The results of the simulation are shown in Fig. 12. For each case, the average power are obtained for the unbuffered case, and for the optimal buffer sizes and optimal switching policy computed in Section V, as well as for the heuristic switching policy under four combinations of fixed buffer sizes in MB: (a) $Q_1 = 8, Q_2 = 8$, (b) $Q_1 = 4, Q_2 = 8$, (c) $Q_1 = 8, Q_2 = 1$, and (d) $Q_1 = 4, Q_2 = 1$. The results show that our optimal policy outperforms the heuristic policy and the unbuffered approach in all cases. For example, in case 2 where the media application is streaming an MPEG-2 movie from a TravelStar hard disk over a LAN, all four of the heuristic approaches actually consume more average power than if no buffers are used at all, while the optimal policy achieves a 60.9% power savings (1.623 W) over the case with no buffers.

D. Sensitivity Analysis

This section explores the sensitivities of the optimal buffer sizes Q_1^*, Q_2^*, optimal period T^*, and optimal power savings with respect to changes in various system parameters. Since the optimal buffer sizes are derived analytically in Section V, the effects of parameter variation can also be determined analytically. We define the sensitivity of a system output y with respect to a parameter x as the ratio of changes in y to incremental changes in x [43]. The (relative) sensitivity can be calculated by finding $S_x^y = \frac{\partial y}{\partial x}\frac{x}{y}$. For example, assuming $\alpha > \beta$, the sensitivity of Q_2^* to variations in the rate α can be computed from equation (9) as:

$$S_\alpha^{Q_2^*} = \frac{\partial Q_2^*}{\partial \alpha}\frac{\alpha}{Q_2^*} = \frac{-\beta^2 m_1}{\beta(\alpha-\beta)m_1 + \alpha(\beta-\gamma)m_2}. \qquad (15)$$

From (15) we see that the sensitivity of Q_2^* to α depends on α, β, γ, m_1, and m_2. By substituting in these values, we can determine the effect of a small variation in α has on the optimal buffer size Q_2^*. This equation reveals that, for small values of α, small changes in α tend to have large effects, whereas the sensitivity decreases as α increases. This ability to find the sensitivity of optimal solutions to parameter variations is a clear advantage of the analytic solution presented in this paper over a heuristic one.

Parameter	Value	Units
k_x	17.100	J
k_y	0.333	J
m_1	6.258×10^{-4}	W/MB
m_2	6.258×10^{-4}	W/MB
p_x	2.166	W
p_y	0.500	W
γ	0.016	MB/s

TABLE IV

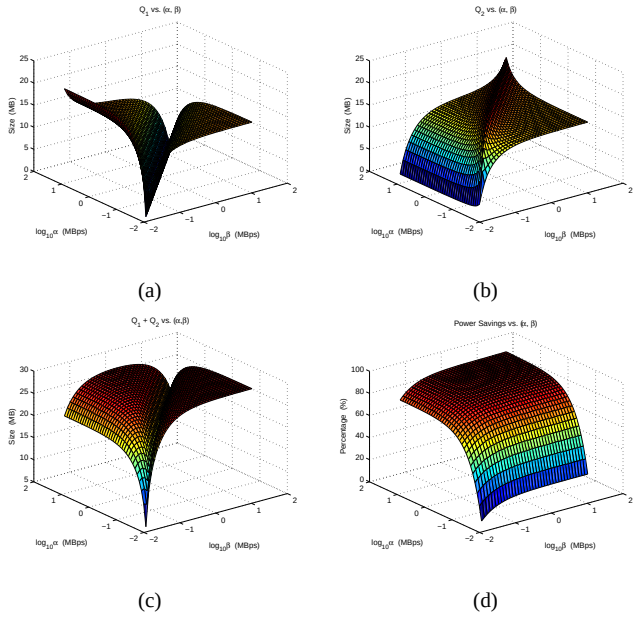

Fig. 13.

In Fig. 13, we plot the values of Q_1^*, Q_2^*, $Q_1^* + Q_2^*$, and the average power savings computed by our optimal solution using the parameters given in Table IV for α and β varying within the range $[0.017, 17.6]$ which represents the range of data rates found in our experimental hardware and for γ taking the value 0.016. Several insights can be obtained from these plots. From Fig. 13 (a) and (b), the values of Q_1^* and Q_2^* are relatively insensitive to variations of α and β when α and β are sufficiently larger than γ and α is sufficiently different from β. Fig. 13(c) shows that the total buffer size is not symmetric with respect to α and β. This implies that if X produces data at the rate β and Y through-puts the data at the rate α, the total memory required in the optimal solution would be different than if X produces data at the rate α and Y through-puts the data at the rate β, even if the physical characteristics of the two buffers are identical. Finally, we can see in Fig. 13(d) that when α and β are close to γ, power savings are minimal. On the other hand, when both α and β are more than two orders of magnitude larger than γ, the power savings almost level off when α and β increase further.

VII. Conclusion

This paper studies the dynamic power management problem of a pipeline of data streaming through three components and two buffers. By focusing on periodic solutions, properties of optimal solutions are derived. Using these properties, a method is presented for determining the optimal buffer sizes and the optimal switching strategies for periodic solutions where each device is switched on and off at most once during each period. This method is then generalized to find the optimal buffer sizes and switching strategies when one of the device compresses or decompresses data, and when the memory sizes are available only in discrete values. Simulation results are presented for some practical streaming data applications, which show a minimum power savings of 61.1% for the optimal solutions over the unbuffered case. The presented optimal solutions are also compared with favorable results to a natural heuristic switching strategy with various heuristic memory sizes.

The results of this paper can be extended in several ways. For example, the data rates of most data formats vary over time due to compression and framing as well as variations of network traffic. To handle variable and random data rates, the model and tools on the optimization of *stochastic hybrid systems* proposed in [47] can be applied. Some preliminary results in this direction are reported in [48] and [49]. For example, in [48], the optimal buffer size and random strategy are derived for a pipeline of streaming data with two components and a buffer in between whose incoming data rate α is modeled as switching randomly among several possible values according to a Markov process. We also intend to generalize our results to pipelines of streaming data consisting of multiple stages and with more complicated network topologies, and to the case of periodic solutions that allow each component to switch on multiple times during each period. The main issue

one faces when dealing with these cases is the exponential increase of problem complexity with the increase of number of stages and number of switchings allowed. To find scalable solution algorithms, we are currently studying the use of optimization tools such as mixed integer programming in finding exact/approximate solutions to the optimal buffer management problem in these general scenarios.

REFERENCES

[1] L. Benini and G. D. Micheli, "System-Level Power Optimization: Techniques and Tools," *ACM Transactions on Design Automation of Electronic Systems*, vol. 5, no. 2, pp. 115–192, April 2000.

[2] L. Benini, A. Bogliolo, and G. D. Micheli, "A Survey of Design Techniques for System-Level Dynamic Power Management," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 8, no. 3, pp. 299–316, June 2000.

[3] L. Cai and Y.-H. Lu, "Dynamic Power Management Using Data Buffers," in *Design Automation and Test in Europe*, 2004, pp. 526–531.

[4] ——, "Energy Management Using Buffer Memory for Streaming Data," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 2, pp. 141–152, February 2005.

[5] N. Pettis, L. Cai, and Y.-H. Lu, "Dynamic Power Management for Streaming Data," in *International Symposium on Low Power Electronics and Design*, 2004, pp. 62–65.

[6] R. Alur, S. L. Torre, and G. Pappas, "Optimal paths in weighted timed automata," in *Hybrid Systems: Computation and Control, 4th Int. Workshop*, ser. Lecture Notes in Computer Science, M. D. Benedetto and A. Sangiovanni-Vincentelli, Eds. Rome, Italy: Springer-Verlag, March 2001, vol. 2034, pp. 49–62.

[7] M. Egerstedt, Y. Wardi, and F. Delmotte, "Optimal control of switching times in switched dynamical systems," in *Proc. 42nd IEEE Int. Conf. on Decision and Control*, vol. 3, Maui, HI, December 2003, pp. 2138–2143.

[8] S. Hedlund and A. Rantzer, "Optimal control of hybrid systems," in *Proc. 38th IEEE Int. Conf. on Decision and Control*, Phoenex, AZ, December 1999, pp. 3972–3977.

[9] H. Sussmann, "A maximum principle for hybrid optimal control problems," in *Proc. 38th IEEE Int. Conf. on Decision and Control*, Phoenex, AZ, December 1999, pp. 425–430.

[10] X. Xu and P. Antsaklis, "Optimal control of switched systems based on parameterization of the switching instants," *IEEE Trans. Automatic Control*, vol. 49, no. 1, pp. 2–16, January 2004.

[11] L. Benini and G. D. Micheli, *Dynamic Power Management: Design Techniques and CAD Tools*. Kluwer, 1997.

[12] C. Im, H. Kim, and S. Ha, "Dynamic Voltage Scheduling Technique for Low-power Multimedia Applications Using Buffers," in *International Symposium on Low Power Electronics and Design*, 2001, pp. 34–39.

[13] T. Ishihara and H. Yasuura, "Voltage Scheduling Problem for Dynamically Variable Voltage Processors," in *International Symposium on Low Power Electronics and Design*, 1998, pp. 197–202.

[14] Y.-H. Lu, L. Benini, and G. D. Micheli, "Dynamic Frequency Scaling With Buffer Insertion for Mixed Workloads," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 21, no. 11, pp. 1284–1305, November 2002.

[15] J. Luo and N. Jha, "Static and Dynamic Variable Voltage Scheduling Algorithms for Real-time Heterogeneous Distributed Embedded Systems," in *Asia and South Pacific Conference on VLSI Design*, 2002, pp. 719–726.

[16] S. Gurumurthi, A. Sivasubramaniam, M. Kandemir, and H. Franke, "DRPM: Dynamic Speed Control for Power Management in Server Class Disks," in *International Symposium on Computer Architecture*, 2003, pp. 169–181.

[17] E. V. Carrera, E. Pinheiro, and R. Bianchini, "Conserving Disk Energy in Network Servers," in *International Conference on Supercomputing*, 2003, pp. 86–97.

[18] F. Douglis, P. Krishnan, and B. Bershad, "Adaptive Disk Spin-down Policies for Mobile Computers," in *USENIX Symposium on Mobile and Location-Independent Computing*, 1995, pp. 121–137.

[19] C.-H. Hwang and A. C.-H. Wu, "A Predictive System Shutdown Method for Energy Saving of Event-driven Computation," *ACM Transactions on Design Automation of Electronic Systems*, vol. 5, no. 2, pp. 226–241, April 2000.

[20] P. Krishnan, P. Long, and J. Vitter, "Adaptive Disk Spindown Via Optimal Rent-to-Buy in Probabilistic Environments," *Algorithmica*, vol. 23, no. 1, pp. 31–56, January 1999.

[21] K. Li, R. Kumpf, P. Horton, and T. E. Anderson, "A Quantitative Analysis of Disk Drive Power Management in Portable Computers," in *USENIX Winter Conference*, 1994, pp. 279–291.

[22] X. Li, Z. Li, F. David, P. Zhou, Y. Zhou, S. Adve, and S. Kumar, "Performance Directed Energy Management for Main Memory and Disks," in *International Conference on Architectural Support for Programming Languages and Operating Systems*, 2004, pp. 271–283.

[23] Y.-H. Lu and G. D. Micheli, "Adaptive Hard Disk Power Management on Personal Computers," in *Great Lakes Symposium on VLSI*, 1999, pp. 50–53.

[24] E. Pinheiro and R. Bianchini, "Energy Conservation Techniques for Disk Array-based Servers," in *International Conference on Supercomputing*, 2004, pp. 68–78.

[25] D. Ramanathan and R. Gupta, "System Level Online Power Management Algorithms," in *Design, Automation and Test in Europe*, 2000, pp. 606–611.

[26] Q. Zhu, A. Shankar, and Y. Zhou, "PB-LRU: A Self-tuning Power Aware Storage Cache Replacement Algorithm for Conserving Disk Energy," in *International Conference on Supercomputing*, 2004, pp. 79–88.

[27] Q. Zhu, F. M. David, C. Devaraj, Z. Li, Y. Zhou, and P. Cao, "Reducing Energy Consumption of Disk Storage Using Power-Aware Cache Management," in *International Symposium on High-Performance Computer Architecture*, 2004, pp. 118–129.

[28] J. R. Lorch and A. J. Smith, "Apple Macintosh's Energy Consumption," *IEEE Micro*, vol. 18, no. 6, pp. 54–63, November 1998.

[29] D. Bertozzi, L. Benini, and B. Ricco, "Power Aware Network Interface Management for Streaming Multimedia," in *Wireless Communications and Networking Conference*, 2002, pp. 926–930.

[30] *Rambus 128/144-Mbit Direct RDRAM data sheet*, Rambus Inc., June 2000, available at http://www.rambus.com.

[31] V. Delaluz, M. Kandemir, N. Vijaykrishnan, A. Sivasubramaniam, and M. J. Irwin, "Hardware and Software Techniques for Controlling DRAM Power Modes," *IEEE Transactions on Computers*, vol. 50, no. 11, pp. 1154–1173, November 2001.

[32] A. R. Lebeck, X. Fan, H. Zeng, and C. Ellis, "Power Aware Page Allocation," in *International Conference on Architectural Support for Programming Languages and Operating Systems*, 2000, pp. 105–116.

[33] J. Pisharath and A. Choudhary, "An Integrated Approach To Reducing Power Dissipation in Memory Hierarchies," in *International Conference on Compilers, Architectures and Synthesis for Embedded Systems*, 2002, pp. 89–97.

[34] Z. Hu, S. Kaxiras, and M. Martonosi, "Let Caches Decay: Reducing Leakage Energy Via Exploitation of Cache Generational Behavior," *ACM Transactions on Computer Systems*, vol. 20, no. 2, pp. 161–190, May 2002.

[35] G. Chen, R. Shetty, M. Kandemir, N. Vijaykrishnan, M. J. Irwin, and M. Wolczko, "Tuning Garbage Collection for Reducing Memory System Energy in An Embedded Java Environment," in *ACM Transactions on Embedded Computing Systems*, November 2002, pp. 27–55.

[36] A. Ramachandran and M. F. Jacome, "Xtream-Fit: An Energy-delay Efficient Data Memory Subsystem for Embedded Media Processing," in *Design Automation Conference*, 2003, pp. 137–142.

[37] C. Tomlin, G. Pappas, and S. Sastry, "Conflict resolution for air traffic management: A study in multi-agent hybrid systems," *IEEE Trans. on Automatic Control*, vol. 43, no. 4, pp. 509–521, April 1998.

[38] P. Varaiya, "Smart cars on smart roads: problems of control," *IEEE Trans. Automatic Control*, vol. 38, no. 2, pp. 195–207, February 1993.

[39] M. Egerstedt, "Behavior based robotics using hybrid automata," in *Hybrid Systems: Computation and Control, 3rd Int. Workshop (HSCC 2000)*, ser. Lecture Notes in Computer Science, N. Lynch and B. H. Krogh, Eds. Springer Verlag, March 2000, vol. 1790, pp. 103–116.

[40] J. Hespanha, "Stochastic hybrid systems: Application to communication networks," in *Hybrid Systems: Computation and Control, 7th Int. Workshop (HSCC 2004)*, ser. Lecture Notes in Computer Science, R. Alur and G. J. Pappas, Eds. Springer Verlag, March 2004, vol. 2993, pp. 387–401.

[41] R. M. Passos, C. J. N. C. Jr, A. A. F. Loureior, and R. A. F. Mini, "Dynamic Power Management in Wireless Sensor Networks: An Application-Driven Approach," in *Annual Conference on Wireless On-demand Network Systems and Services*, 2005, pp. 109–118.

[42] A. Balluchi, L. Benvenuti, M. D. D. Benedetto, G. M. Miconi, U. Pozzi, T. Villa, H. Wong-Toi, and A. L. Sangiovanni-Vincentelli, "Maximal safe set computation for idle speed control of an automotive engine," in *Hybrid Systems: Computation and Control, 3rd Int. Workshop (HSCC 2000)*, ser. Lecture Notes in Computer Science, N. Lynch and B. H. Krogh, Eds. Springer Verlag, March 2000, vol. 1790, pp. 32–44.

[43] K. Amonlirdviman, N. A. Khare, D. R. P. Tree, W.-S. Chen, J. D. Axelrod, and C. J. Tomlin, "Mathematical modeling of planar cell polarity to understand domineering nonautonomy," *Science*, vol. 307, no. 5708, pp. 423–426, January 2005.

[44] P. Lincoln and A. Tiwari, "Symbolic systems biology: hybrid modeling and analysis of biological networks," in *Hybrid Systems: Computation and Control, 7th Int. Workshop (HSCC 2004)*, ser. Lecture Notes in Computer Science, R. Alur and G. J. Pappas, Eds. Springer Verlag, March 2004, vol. 2993, pp. 660–672.

[45] J. Hu and Y.-H. Lu, "Buffer management for power reduction using hybrid control," in *Proc. IEEE Conference on Decision and Control and European Control Conference*, Seville, Spain, December 2005, pp. 6997–7002.

[46] V. D. L. Luz, I. Kadayif, M. Kandemir, and U. Sezer, "Access Pattern Restructuring for Memory Energy," *IEEE Transactions on Parallel and Distributed Systems*, vol. 15, no. 4, pp. 289–303, April 2004.

[47] R. Raffard, J. Hu, and C. Tomlin, "Adjoint-based optimal control of the expected exit time for stochastic hybrid systems," in *Hybrid Systems: Computation and Control, 8th Int. Workshop (HSCC 2005)*, ser. Lecture Notes in Computer Science, M. Morari, L. Thiele, and F. Rossi, Eds. Springer Verlag, March 2005, vol. 3414, pp. 557–572.

[48] J. Hu, "Application of stochastic hybrid systems in power management of streaming data," in *Proc. American Control Conference*, Minneapolis, MN, June 2006, pp. 4754–4759.

[49] N. Pettis, L. Cai, and Y.-H. Lu, "Statistically optimal dynamic power management for streaming data," *IEEE Trans. Computers*, vol. 55, no. 7, pp. 800–814, July 2006.

Jason Ridenour (S'02) received the MS degree in Electrical Engineering from Purdue University, in 2005. His research interests include power management of embedded systems and hybrid systems.

PLACE PHOTO HERE

Jianghai Hu (S'99-M'04) received the B.E. degree in automatic control from Xi'an Jiaotong University, P.R. China, in 1994, and the M.A. degree in Mathematics and the Ph.D. degree in Electrical Engineering from the University of California, Berkeley, in 2002 and 2003, respectively. He is currently an assistant professor at the School of Electrical and Computer Engineering, Purdue University. His research interests include hybrid systems, multi-agent coordinated control, control of systems with uncertainty, and applied mathematics and probability theory.

PLACE PHOTO HERE

Nathaniel Pettis (S99) received the BSEE degree from the University of Arkansas, Fayetteville, in 2002. He is currently pursuing the PhD degree in electrical and computer engineering at Purdue University, West Lafayette, Indiana. His research interests include operating systemsdirected power management and embedded systems. He is a student member of the IEEE.

PLACE PHOTO HERE

Yung-Hsing Lu (S'90-M'03) received the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, in 2002. He currently is an Assistant Professor with the School of Electrical and Computer Engineering, and by courtesy, the Department of Computer Science, at Purdue University, West Lafayette, IN. His research focuses on power management and energy conservation for computer systems, mobile robots, sensor networks, and image processing. In 2004, Dr. Lu received an NSF Career Award for studying energy conservation by operating systems.

List of Footnotes

Author Affiliation: Ridenour, Hu, Pettis, and Lu are with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, 47907, USA. Email: {ridenoja, jianghai, npettis, yunglu}@purdue.edu.

Acknowledgment of Financial Support: Ridenour and Hu are supported by the Purdue Research Foundation. Pettis is supported by the GAANN Fellowship. Lu is supported in part by NSF Career CNS 0347466. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.