Hybrid Power Management for Office Equipment

GANESH GINGADE, WENYI CHEN, YUNG-HSIANG LU, and JAN ALLEBACH,
Purdue University
HERNAN ILDEFONSO GUTIERREZ-VAZQUEZ, Hewlett-Packard Company

Office machines (such as printers, scanners, facsimile machines, and copiers) can consume significant amounts of power. Most office machines have sleep modes to save power. Power management of these machines is usually timeout-based: a machine sleeps after being idle long enough. Setting the time-out duration can be difficult: if it is too long, the machine wastes power during idleness. If it is too short, the machine sleeps too soon and too often—the wake-up delay can significantly degrade productivity. Thus, power management is a tradeoff between saving energy and keeping response time short. Many power management policies have been published and one policy may outperform another in some scenarios. There is no definite conclusion regarding which policy is always better. This article describes two methods for office equipment power management. The first method adaptively reduces power based on a constraint of the wake-up delay. The second is a hybrid method with multiple candidate policies and it *selects* the most appropriate power management policy. Using 6 months of request traces from 18 different printers, we demonstrate that the hybrid policy outperforms individual policies. We also discover that power management based on business hours does not produce consistent energy savings.

CCS Concepts: • **Software and its engineering** → **Power management**; • **Computer systems organization** → *Embedded systems;*

Additional Key Words and Phrases: Power management, office equipment, printer

ACM Reference Format:
Ganesh Gingade, Wenyi Chen, Yung-Hsiang Lu, Jan Allebach, and Hernan Ildefonso Gutierrez-Vazquez. 2016. Hybrid power management for office equipment. ACM Trans. Des. Autom. Electron. Syst. 22, 1, Article 14 (November 2016), 22 pages.
DOI: http://dx.doi.org/10.1145/2910582

1. INTRODUCTION

Office equipment may consume significant amounts of energy and the potential of energy savings has not been fully exploited. Today's office equipment is often set in "ready" mode to have short response time. However, the machine is idle for most part of the business hours [Kawamoto et al. 2004] and wastes a significant amount of power in the ready mode. As energy efficiency becomes increasingly important, power management is essential for office equipment. The most prevalent power management policy for office equipment is timeout: when the machine becomes idle, a timer is set. While waiting for the timer to expire, the machine stays in the ready mode. If

---

This work is sponsored by Hewlett-Packard Company.
Authors' addresses: G. Gingade, W. Chen, Y.-H. Lu, and J. Allebach, Purdue University, School of Electrical and Computer Engineering, Electrical Engineering Building, 465 Northwestern Ave., West Lafayette, Indiana 47907-2035; emails: {gpganesha, wenyichen1613}@gmail.com, {yunglu, allebach}@purdue.edu; H. I. Gutierrez-Vazquez, Hewlett-Packard, 11311 Chinden Blvd, Boise ID 83714; email: hernan_gutierrez@hp.com.

DOI: http://dx.doi.org/10.1145/2910582

the machine remains idle when the timer expires, the machine enters a lower-power "sleep" mode. The timeout value is set by either the user or the manufacturer in order to comply with regulations for Eco-labels such as Energy-Star [2014], Blue-Angel [2013], or Nordic-Swan [2013]. These regulations stipulate stringent energy requirements for certification (under 1W after a sufficient period of idle time) [Meier and LeBot 1999].

General power management policies may employ predictive- or stochastic-based approaches [Lu et al. 2000]. The former predicts the next machine activity based on exponential average, correlation, regression, or heuristics of past observed and predicted idle time [Lu et al. 2000]. Stochastic techniques model systems as Markov decision processes [Benini et al. 2000]. Several other policies employ machine learning techniques to learn the power management policy [Shen et al. 2013; Dhiman and Rosing 2006; Wang et al. 2011]. Some maintain a set of policies and perform online policy selection [Dhiman and Rosing 2009; Pettis and Lu 2009] based on user specified energy-performance criteria. In case of printers, an adaptive timeout policy is proposed [Ciriza et al. 2008, 2012] using the estimated distribution of past print requests.

Many dynamic power management studies focus on peripheral devices such as a hard disk drive (HDD) and a wireless local area network (WLAN) controller [Benini et al. 2000]. Very few studies are devoted to office equipment power management. Office equipment presents unique challenges wherein the wakeup energy and delay are significantly higher than the peripheral devices. The ready power of a printer can be 100W, while an HDD's ready power is around 5W [Pettis and Lu 2009]. The wake-up energy for a printer can be 2,000J compared to an HDD's 20J. The printer request is highly dynamic and depends on the location and time. Office equipment encounters a variety of request scenarios based on the offices' functions (business office, academic units, or student laboratories) or the time of the request (day, night, weekday, or weekend). The best power management policy for office equipment depends on hardware and workload. For frequent requests, a preferred power management policy would have a long timeout to prevent the printer from sleeping. For sparse requests, a better policy would sleep soon after each service. Hence, a fixed timeout cannot always achieve desirable energy savings and quick response. Some printers provide options to set custom schedules for sleep and wake-up times for each day of the week or holidays.

Using 6 months of request traces from 18 different printers deployed at Purdue University, we conclude that fixed timeout or preset schedules cannot achieve consistent energy savings for printers in different offices. We propose two policies for printer power management. The first policy divides the printer workload into distinct phases and provides a timeout for each phase. The timeouts are determined from the printer's past requests based on a constraint of the wake-up delay. The second selects the best power policy from a set of candidate policies based on the observed request pattern. We evaluate this hybrid policy and demonstrate that it is better than the individual policies. We also show that, contrary to intuition, a policy with scheduling information (such as weekdays vs. weekends) underperforms the policies without such knowledge about business hours.

This article has the following contributions:

(1) We propose a power management policy with multiple timeouts, each determined from the printer's past requests for balancing power and delay.
(2) We demonstrate that no single power management policy performs consistently for different printers in different offices.
(3) We propose a hybrid power policy that outperforms individual power management policies.
(4) We demonstrate that the hybrid power policy performs best with frequent policy updates and using short-term memory of past printer requests.

(5) We examine the policy with information about business hours (such as weekdays vs. weekends, daytime vs. night) and show that such knowledge does not produce consistent power savings.

2. RELATED WORK

Related work in power management can be found for peripheral devices like hard disk drives, displays, and WLAN controller [Benini et al. 2000]. Few studies have been devoted exclusively for office equipment. In the following sections, we discuss the related work and compare our policies with the existing policies.

2.1. Dynamic Power Management

Dynamic power management policies can be broadly classified as timeout-based, predictive, or stochastic [Benini and Micheli 2000; Lu et al. 2000; Irani et al. 2005]. In timeout-based policies, an idle machine is set to a low power state after the expiry of a timeout period. The timeout could be pre-configured to a fixed value based on expected future requests or changed dynamically. The value could be statically or adaptively determined [Douglis et al. 1995]. In a static timeout policy, a pre-configured value is used irrespective of the request conditions. In an adaptive timeout policy, the value is changed according to the workload changes. Ramanathan and Gupta [2000] describe an adaptive timeout policy based on a device's break-even time. Break-even time is the time period during which the device would incur no additional energy irrespective of the energy state it stays in after serving a request. Shih and Wang [2012] present a policy for mobile devices that can adapt to self-similar workloads exhibited by human interactions. Guo et al. [2008] take into account human behavior aspects to determine the best policy for the adaptive energy management. Prediction-based policies rely on past request patterns to predict a device's future requests and set the device's power state. The prediction of inter-arrival times is based on the observed patterns: exponential average, correlation, regression, or heuristics of past observed and predicted idle times [Benini et al. 2000; Lu et al. 2000; Golding et al. 1995; Hwang and Wu 2000]. For example, future inter-arrival times are predicted using an observed pattern like L-Shape [Srivastava et al. 1996], where it is assumed that a short busy period is followed by a long idle period. Stochastic policies model systems as Markov decision processes. Both Markov and semi-Markov models are considered. The policies are considered constrained optimization problems [Benini et al. 2000]. The models trade off between power and latency. They attempt to estimate the underlying request arrival distribution and globally optimize the expected power and latency. Both stationary and non-stationary requests are considered [Chung et al. 2002]. Discrete and continuous time stochastic models have been proposed [Qiu and Pedram 1999; Benini et al. 2000]. Several other policies employ machine learning techniques [Theocharous et al. 2006] to learn the power management policy from the device environment. Supervised and reinforcement learning have been explored [Shen et al. 2013; Dhiman and Rosing 2006; Wang et al. 2011; Khan and Rinner 2014; Tesauro et al. 2007; Tan et al.].

The competitive analysis technique can be used to compare online policies [Karlin et al. 1994]. In this approach, the performance of an online policy is compared against an optimal offline oracle policy. The policy is c-competitive if, for any sequence of requests, its worst-case performance is bounded by c times the performance of the offline oracle policy. The fixed timeout policy has been shown to be a 2-competitive [Ramanathan and Gupta 2000], if the timeout is set to the break-even time. The worst-case power dissipation expected from a fixed timeout policy is twice the oracle's power. The dual-timeout adaptive policy [Ramanathan and Gupta 2000] is 3-competitive, but shown to achieve better performance in practice. The competitive ratio's low bound is 1.58 [Karlin et al. 1994] for the best timeout policy. User annoyance is attributed to latency during system power management [Douglis et al. 1995]. Irani and Pruhs [2005] mention a need for a policy that allows a user or system to preferentially choose between optimizing one resource over another (e.g., optimizing power based on a delay constraint).

2.2. Power Management for Office Equipment

The most prevalent power management policy for office equipment is timeout. The timeout value is set by either the user or the manufacturer in order to comply with regulations for Eco-labels. Larson [2011] describes a policy of selecting power state for a peripheral by examining the activity packets. Ciriza et al. [2012] describe a timeout-based policy, wherein a fixed timeout is estimated to minimize a cost function. The cost function is based on the distribution of the past (week or month) observed inter-arrival times and penalty (wake-up delay and energy). The papers of Ciriza et al. [2008] and Durand et al. [2013] describe an adaptive timeout policy using a hidden Markov chain. These studies suggest using long-term observations to improve energy savings. In contrast, we show that the power management policy performs better using short-term past requests.

2.3. Power Policy Selection

Pettis and Lu [2009] introduce power policy selection in an operating system and show one policy outperforming another under some conditions. It may be difficult, or even impossible, to design the "best" policy for all conditions. A software framework called the Homogeneous Architecture for Power Policy Integration (HAPPI) is defined for system power management. The framework selects an active power management policy by choosing the best estimate for an evaluation metric, such as total energy consumption or energy-delay product. Experiments demonstrate that selecting policies can achieve better reduction in system energy. Different workloads are considered to demonstrate the notion of "no one policy fits all." Helmbold et al. [2000] maintain a set of "expert" policies and use multiplicative weights to make timeout predictions. Dhiman and Rosing [2009] describe an online learning policy to select among a set of possible policies and voltage-frequency settings. The online learning policy maintains a set of "experts" and selects one expert that has the best chance to perform well, based on the characterization of the current workload. An evaluation is performed at the end of each idle period to select among the experts. It demonstrates that the performance is at least as good as the best expert across different workloads. In addition to using the policy selection as described in the aforementioned two papers, we propose to perform policy evaluation and determine the period for better results. We also show that better performance is achieved using short-term past requests.

3. LASER PRINTER SYSTEM

3.1. Printer Mechanism

Office imaging equipment includes the following product types: printers, scanners, copiers, and facsimile machines. In this section, we discuss laser printers with a focus on the printer mechanism and performance metrics. Laser printers produce high quality text and graphics on plain paper. Laser printers have mechanical and thermal components [Hewlett-Packard 2008] that affect power management. Laser printers use electrically charged rotating drums. A laser beam is used to alter the charge components of the drum. Dry ink or toner is attracted to the charge altered areas of the drum. The toner is subsequently transferred to a paper and fused using heat and pressure. As shown in Figure 1, the laser printer image-formation system consists of the following parts: laser for scanning, print cartridge, imaging drums, intermediate transfer belt (ITB), and fuser. The process of image formation onto a paper involves the

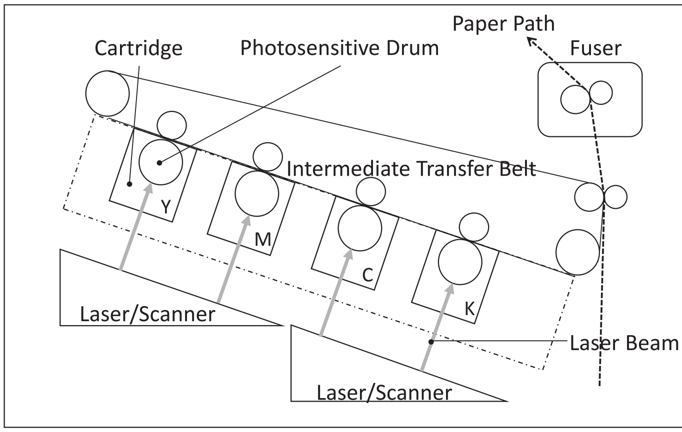

Fig. 1. Color laser printer image formation system. The desired image is formed on a negatively charged photosensitive drum using laser beams. Yellow (Y), magenta (M), cyan (C), and black (K) images are formed on four separate photosensitive drums. The charged toner gets attracted to the drum at the image areas and later transferred onto an intermediate transfer belt (ITB). The toner collected from different colors gets attached to a positively charged paper. Finally, the toner is fused and pressed onto a paper to generate a permanent image. (Source: HP color LaserJet service manual)

following steps [Hewlett-Packard 2008]: (i) latent image formation, (ii) development, (iii) transfer, and (iv) fusing.

Once the printer receives the data, a latent image is formed on the surface of the photosensitive drum. A laser beam is used to strike the surface of the photosensitive drum at those areas where an image is desired. The image is transferred onto a paper using the ITB. Likewise, toners of different colors: yellow, magenta, cyan, and black are transferred onto the ITB in sequence. The complete toner image gets transferred onto the paper. The paper is then passed through the heated and pressurized rollers to melt the toner and get the permanent image. The fuser accounts for the highest power usage in a laser printer. The fuser has to be at the right temperature before the printing can begin. The primary power saving feature available in most laser printers is to turn off the fuser. It takes a considerable amount of time (around 10 seconds) before the fuser returns to the required temperature after being turned off. Thus, any power management policy has to factor in the fuser property to balance power savings and delays.

3.2. Printer Power Cycle

The printer cycles through the following sequence each time there is a request, as illustrated in Figure 2. The printer at the sleep state enters a waiting period upon a request and starts heating the fuser. This takes a considerable amount of time. Upon heating the fuser to the right temperature, the printer enters the ready state. During the printing process, an image is formed on the photosensitive drum by striking a laser beam, which is eventually transferred using a charged toner onto a paper and fused. The printer then enters the ready state wherein the fuser is kept ready by constantly applying heat. After a specified length of inactivity, the printer enters the sleep state as a power saving feature. During the sleep state, energy is conserved by turning off many components, in particular, the fuser. Some printers have multiple sleep states and different sets of components are turned off. For example, the printer HP Color LaserJet CM3530 [Hewlett-Packard 2008] has the following power-saving modes: (i) power off, (ii) ready, (iii) shallow suspend, (iv) almost 1W suspend, and (v) deep suspend.

At each power state, a specific set of components is kept on with the rest turned off. Different printer manufacturers select different sets of components to achieve desired

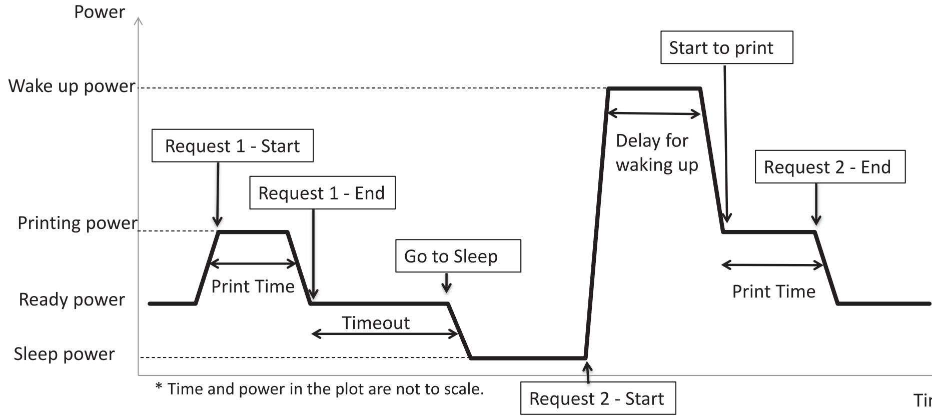

Fig. 2. Printer starts in the ready state, consuming ready power. Upon receiving a request, the printer transitions to a higher power state for serving the request. After serving the request, the printer returns to the ready state and starts a timer. If no request is received within this timeout period, the printer transitions to a low power state. Most of the components are turned off at the low power state. Waking up from the sleep state incurs a significant cost in terms of energy and time. The wake-up energy and delay is printer-specific.

energy conservation and swift response. Figure 2 shows a printer with multiple power states. There is an associated power dissipation at each state. The sleep state dissipates the lowest power among the power states. There is a cost in terms of wake-up energy, which is expended whenever the printer transitions from a low power state to a higher state. Also, there is a considerable latency involved in the transition. To conserve power, the printer has to transition to a low power state. As the printer transitions to a low power state, it also incurs additional energy and latency, while waking up to serve a request. Power management intends to conserve power with little wake-up delay (user experience impact).

3.3. Printer Performance Metrics

Printer performance is measured by the number of pages per minute (PPM). Each printer is classified based on PPM and has to meet a specific set of power regulations. For a printer, the performance metrics include Sleep-to-First-Page-Out and Sleep-to-First-Copy-Out. The latest printers can achieve Sleep-to-First-Page-Out in under 5 seconds using less than 5W. Most printers have Sleep-to-First-Page-Out in the range of 12 to 20 seconds [Knoder 2015]. Users often put higher priority on the performance (less than 10 seconds delay from low power mode) than energy efficiency and feel inconvenient while waiting for the printer to wake up. Energy efficiency and high performance are usually conflicting goals. To improve energy efficiency, a printer should sleep often. To improve performance, a printer should stay ready. There have been recent improvements in fuser heating technology and engine mechanical recovery, which constitute the main bottleneck in the printer's response time from sleep. Nevertheless, the delay due to wake-up is still noticeable. Better performance can be achieved with an intelligent power manager inside the printer, determining when to sleep. Thus, it is desirable to achieve more energy savings with less user inconvenience by balancing between the energy efficiency and performance.

4. POWER MANAGEMENT POLICY

Section 2 describes the existing studies on power management policies. Most of the studies focus on peripheral devices and very few are devoted to printer power management. Section 3 describes the printer system and unique challenges associated with printer power management. The printers have significantly higher wake-up energy

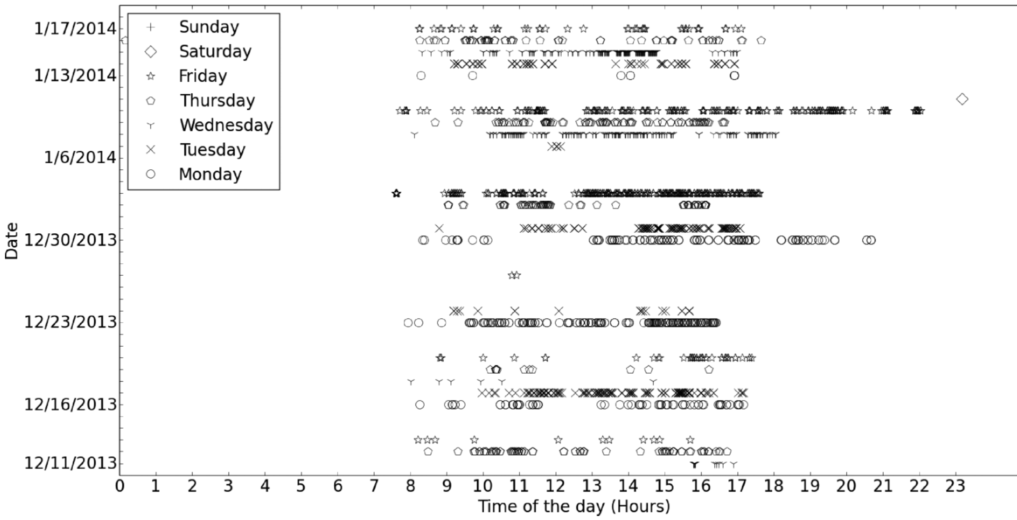

Fig. 3. Printer request log of an office printer. Each symbol represents a request. Several attributes are logged at each request like start and end times, job type, etc. Each row belongs to a day of the week plotted over 24 hours.

and delay compared with the peripheral devices. Hence, a better power management policy for printers is desirable. We study the request traces from printers installed at 18 different offices at Purdue University in search for a better power management policy for printers. We demonstrate two power management policies specific to printers: (1) adaptive multiphase power management and (2) hybrid power management. Adaptive multiphase power management relies on the past printer requests to classify the workload and derive different timeouts for the future. The hybrid power management selects policies, given a set of candidate power management policies. In Section 5, we perform evaluations and compare our new power management policies with the existing policies.

4.1. Printer Activity

As part of the solution towards better printer power management, 6 months of request traces from printers at 18 different offices were studied. The printers were located at different offices like a business office, administrative office, faculty room, and student lab. The traces contained the sequences of timestamps corresponding to the printer requests like print, scan, fax, and copy. The timestamps were logged at a resolution of one second for each of the request's start and end time. Figure 3 shows the request trace of an office printer over a month. Each symbol corresponds to a printer request. Each row belongs to a day of the week plotted over a course of 24 hours. We can see most requests during the weekdays in contrast to limited requests during weekends. Similarly, the requests are more frequent during the business hours starting from 8:00 a.m. to 6:00 p.m. compared with non-business hours. Similar patterns of requests were observed for most other office printers. Figure 4 depicts the cumulative distribution of inter-arrival times for printers placed at five different office rooms over 6 months. It can be observed that the request patterns are unique to the office printers. Some printers have more frequent usage than the others. For example, printer-3 has the most frequent usage with half of its inter-arrival times less than 3 minutes. While printer-1 has half of its inter-arrival times less than 15 minutes.

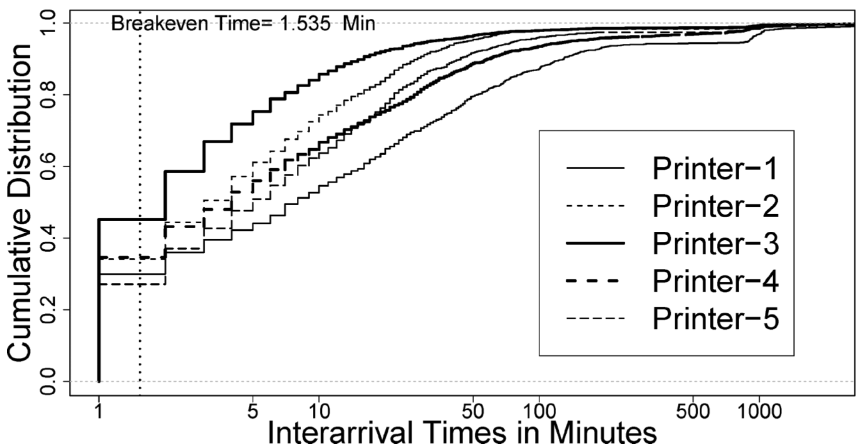

Fig. 4. Cumulative distribution of inter-arrival times for printers placed at different offices. It depicts the ratio of request occurrences with inter-arrival times less than a specific time to the total requests. Break-even time is the time period during which the printer would incur no additional energy irrespective of the power state it stays in after serving a request.

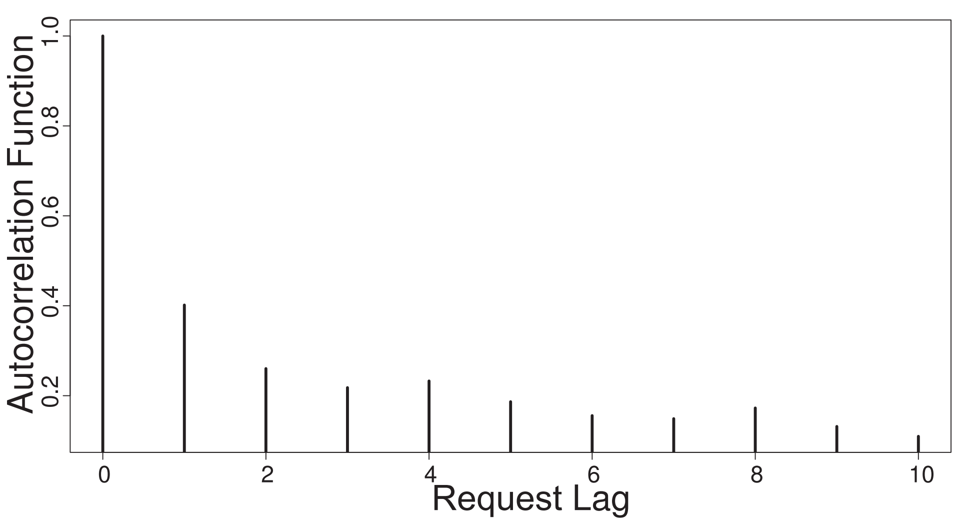

Fig. 5. Autocorrelation of inter-arrival times (binary) at different lags for printer-1. The inter-arrival times are classified using a threshold (break-even time). The inter-arrival times greater than the break-even time are assigned a value “1,” else a value “0” is assigned. The autocorrelation at different lags are computed on the binary sequence to verify any relation among printer requests. The printer requests exhibit a short-term correlation.

The printers are idle most of the time with 20% of the requests arriving no sooner than 30 minutes. Usually, the printers are set in ready mode to serve a request immediately. If the printers are idle most of the time, they waste considerable amounts of energy. From Figure 4, we also see a burst request among printers with 40% of inter-arrival times less than 5 minutes. Setting the printers to low power states would seriously degrade performance as printers would take considerable amounts of time waking up. From Figure 5, we see that there is a short-term correlation among printer requests. We compute the autocorrelation on the binary inter-arrival time sequence to verify any relation among printer requests. The binary sequence is computed using a threshold (break-even time). Any inter-arrival time greater than the break-even time is assigned "1," else a "0" is assigned. In summary, printers are idle most of the time with bursts of requests in between. The printer requests exhibit a short-term correlation. We use these properties to devise new power management policies specific to printers in the rest of this section.

4.2. Adaptive Multiphase Power Management

From the previous section, we observe the following: (1) The printer requests are unique to the offices. (2) The printer requests exhibit a unique pattern over a day and week. (3) The printer is idle most of the time with bursts of requests in between. (4) There is a short-term correlation among the printer requests. (5) Timeout is the most prevalent power saving feature among the printers. The existing single timeout policy is ineffective for a variety of printer workloads. Although a sleep and wake-up schedule is provided for the printer, it is manual and does not accurately characterize the printer workload. We propose an adaptive multiphase policy that divides the printer requests into multiple activity phases ranging from the busiest to the least busy. The activity phases are configurable to reflect the variations in the printer workload. For example, during the non-business hours with less variations in the printer requests, we can use dual activity phases. With more variations in the printer requests during the business hours, more phases can be used. Each activity phase reflects a set of printer workloads, and hence, a timeout could be used to set an appropriate power state. The busier the activity phase is, the longer the timeout is set, and vice versa. In a busier phase, the printer requests are more frequent, and hence, a longer timeout would prevent frequent shutdowns, thus providing a swift response. Similarly, in a less busy phase, a shorter timeout would shutdown the printer more often and save power. Thus, based upon the underlying requests, an appropriate activity phase is entered and a corresponding timeout is set. We restrict the phase transitions to adjacent phases to account for short-term correlations found among the printer requests. The phase transitions are based on the request inter-arrival times and a set of thresholds at each phase. The printer would incrementally transition to finally reach an activity phase corresponding to the underlying workload. For example during a typical weekday, the printer would start in the least busy phase with the shortest timeout. During business hours, the printer would incrementally transition to a busier phase, depending upon the workload, and transition back to the less busy phase at the end of the business hours. The parameters to switch the activity phase and the corresponding timeouts are computed from the past requests. The parameters are updated periodically to reflect the changes in the printer workload.

We illustrate the multiphase policy using the following example. Consider a printer with the break-even time of 1 minute and a wake-up delay of 10 seconds. Break-even time is the time period at which the printer serving a request would consume the same power irrespective of the power state it starts in. During the break-even time, the printer in the ready mode would consume the same energy as the wake-up energy. However, there is a delay when the printer starts in the sleep state. Consider a case where all the requests arrive with inter-arrival times equal to the break-even time. In this case, a printer in the ready mode is more desirable as it encounters no delay. We can use a single-phase policy with a timeout of greater than break-even time to keep the printer ready for all the requests. Similarly, a single-phase policy with a timeout of greater than break-even time encounters no delay for the case where the inter-arrival times are always less than the break-even time. For the aforementioned cases, the power consumption remains the same for the timeouts of zero minutes (immediate shutdown) and greater than break-even time. A single-phase policy with an immediate shutdown provides better energy savings for the case where the

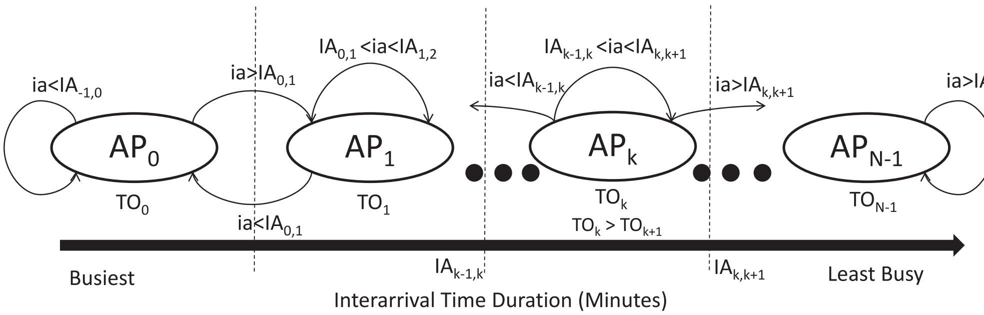

Fig. 6. Activity phases ranging from the busiest (AP_0) to the least busy (AP_{N-1}). Timeout (TO_k) and inter-arrival thresholds ($IA_{k-1,k}$, $IA_{k,k+1}$) for an activity phase (AP_k). The printer in the activity phase (AP_k) stays in the same phase (AP_k) or moves to its adjacent activity phase (AP_{k-1} or AP_{k+1}), based on the current inter-arrival time (IA).

inter-arrival times are always greater than the break-even time. In this case, the average delay is 10 seconds as the printer wakes up to serve every request. For the case with bursty requests followed by long idle periods, for example, with 90% of the inter-arrival times less than the break-even time and the rest greater than 1 hour, better energy savings and delay is achieved using a dual-phase policy with the timeouts of break-even time and immediate shutdown (zero minutes), and an inter-arrival threshold of break-even time. The policy applies a timeout of break-even time for the requests with the inter-arrival times less than the break-even time and shuts down immediately for the rest of the requests (10%) providing an average delay of 1 second. The dual-phase policy achieves better energy savings compared to the single-phase policy of always-on (1-hour timeout) and better delay compared to the single-phase policy of immediate shutdown (0 minutes). For the cases of printer requests as explained in Figure 4, we search for the timeouts and the inter-arrival thresholds of the multiphase policy that meets a desired input delay constraint with low power.

Figure 6 describes a general N-phase power management policy, with the activity phases ranging from the busiest (AP_0) to the least busy (AP_{N-1}). Each activity phase (AP_k) has an associated timeout (TO_k) and inter-arrival thresholds ($IA_{k-1,k}$, $IA_{k,k+1}$). The printer in an activity phase (AP_k) stays in the same phase (AP_k) or transitions to its adjacent phases (AP_{k-1} or AP_{k+1}) based upon the current inter-arrival time (ia). The number of phases, N, is configurable and can be set based on the variations in the workload. The timeout (TO_k) and the inter-arrival thresholds ($IA_{k-1,k}$, $IA_{k,k+1}$) are computed from the past printer requests and updated periodically. At each request, the inter-arrival time (ia) is computed as the difference in the time between the current request and the past request instance. The activity phase (AP_k) is updated based on the current inter-arrival time (ia). The printer at the activity phase (AP_k) moves to a busier phase (AP_{k-1}) for the inter-arrival time (ia) less than the inter-arrival threshold $IA_{k-1,k}$. It moves to a less busy phase AP_{k+1} for the inter-arrival time (ia) greater than the threshold $IA_{k,k+1}$. It stays in the same phase otherwise. The timeout corresponding to the new activity phase is applied for the next printer idle period until the next request arrives. The multiphase parameters are updated periodically (example every day, week, etc.) using the past requests to reflect any changes in the printer workload. Thus, given a set of requests and the printer model, the goal is to compute the parameters for an N-phase power management policy: TO_k, $IA_{k-1,k}$, and $IA_{k,k+1}$ for $0 < k < N-1$. The parameters should provide the best energy savings with swift response. However, the power and delay are dual in nature. Achieving a low power would incur a cost on delay

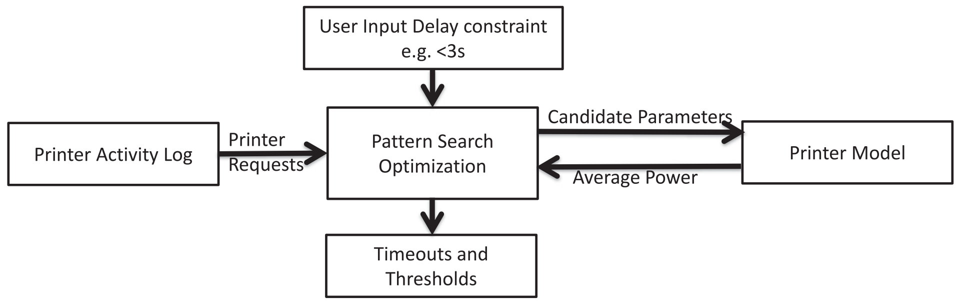

Fig. 7. Computing multiphase parameters using the pattern search optimizer.

and vice versa. Hence, we search for parameters that provide the best energy savings under a given delay constraint (for example, less than 3 seconds).

We employ an optimization method to search for the multiphase parameters under a given delay constraint. Pattern search [Kolda et al. 2003; Lewis et al. 2000; Davidon 1991; Torczon 1997] is one such optimization method that does not use derivatives. Figure 7 describes a method for searching the multiphase parameters (TO_k, $IA_{k-1,k}$, and $IA_{k,k+1}$) using the pattern search optimizer. We use the pattern search optimizer in MATLAB [MathWorks 2015] to find the multiphase parameters. We use a printer model with measured parameters (wake-up delay, wake-up energy, active power, and sleep power) that returns an average power for a given set of candidate multiphase parameters and the delay constraint. The pattern search uses the known request instances and the printer model to find the best parameters. The pattern search converges onto the best parameters that provide a low power under the given delay constraint in finite time. In order to prevent false convergence, the pattern search optimization is performed multiple times with random initial parameters.

The pattern search method uses derivative-free heuristic methods to search for the best parameters. The method can be illustrated using the following example of a compass search: given a function with two parameters, the pattern search starts with an initial step size and reduces it iteratively until a threshold is reached. It evaluates the function at each set of candidate parameters along the north, south, east, and west, by incrementing or decrementing the parameters by the step size. The candidate parameters that provide a reduction in function is selected for the next iteration. If no reduction is achieved along any direction, the step size is reduced by half and the search is continued. The pattern search is terminated when the step size reaches a lower threshold.

Figure 8 describes an example of the multiphase power management for a printer in a general office environment. During the non-working hours (before 6:00 a.m. and after 7:00 p.m.) a dual activity phase is used with a relaxed delay constraint of less than 5 seconds. During the work transition hours (6:00 a.m. – 8:00 a.m., Lunch hour, and 6:00 p.m. – 7:00 p.m.), a slightly tighter constraint of 4 seconds is used. Finally, during the working hours, four activity phases with a constraint of less than 2.5 seconds are used.

In summary, the multiphase power management policy divides the printer workload into multiple phases and provides a set of parameters that can be used to reach the relevant phase from any phase. The policy would transition to a relevant phase, based on the current workload, and an appropriate timeout is applied, balancing power and delay.

4.3. Hybrid Power Management

In the previous section, we described a power management policy that divides the printer requests into multiple phases. The multiphase policy's parameters are

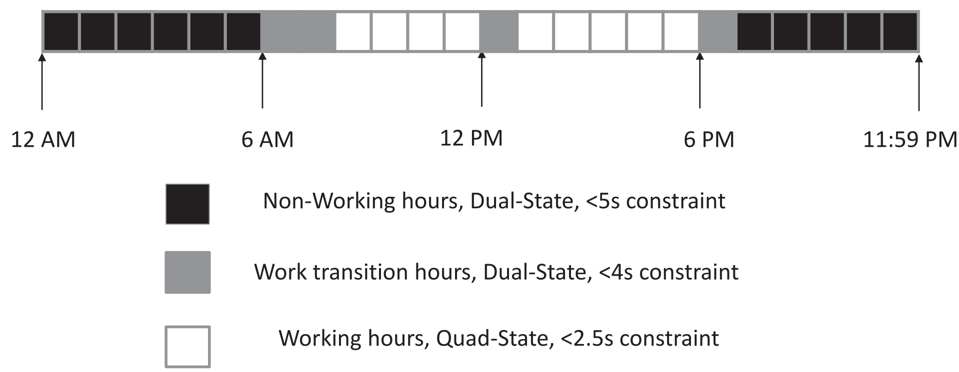

Fig. 8. Sample delay constraints during a weekday for a printer in a general office environment. The constraints are based on the assumption that a printer is used more often during the working hours from 8:00 a.m. to 6:00 p.m. A relaxed constraint is applied during the non-working hours with an intermediate constraint during the transition hours.

computed from the past requests and do not reflect the current workload. There is no conclusive evidence of a long-term correlation among the requests, although we observe a short-term correlation.

Based on the observations of the multiphase policy, we propose a second policy—hybrid policy that characterizes the current workload and selects a policy instead of a timeout. A policy that accurately characterizes the underlying workload is able to successfully perform power management achieving energy savings with acceptable delay. We show that each policy performs well for a different workload. We evaluate each policy's performance (energy savings) on multiple printer traces from different offices. We evaluate the policy's overall performance, as well as the performance during a short time period. We can associate a workload with a policy, and hence, with a combination of power management policies, we can achieve better performance. The hybrid policy is, thus, a combination of individual candidate policies that achieves better performance overall than a single policy. Instead of explicitly characterizing a printer's workload as a single pattern, we select a policy associated with the workload. As the workload changes, we select a different policy.

Figure 9 describes the hybrid policy that performs the policy selection. The hybrid policy selects policies regularly. The hybrid policy consists of the following components: (1) candidate power management policies, (2) an evaluator that performs the policy evaluation, and (3) the printer's parameters. The goal is to characterize the current printer workload and to select the best power policy. We perform policy selection periodically to reflect changes in the workload. Assuming the workload does not change drastically, we rely on a policy's performance during the immediate past to select it for the future. As shown in Figure 10, at current time t_C, an evaluation is performed on the just concluded workload. An evaluation window of time period t_E is used to find a policy that performs best during the evaluation window. Assuming the workload does not vary drastically, the best performing policy would also perform better for the future workload. The best performing policy is retained for the next freeze period t_F. The hybrid power management performs the workload characterization and, in turn, provides the best performance overall as compared with a single policy.

5. EVALUATION AND RESULTS

5.1. Printer Parameter Measurement

A multi-function printer is used to measure the parameters like wake-up delay, wake-up energy, average ready power, average sleep power, and printer break-even time,

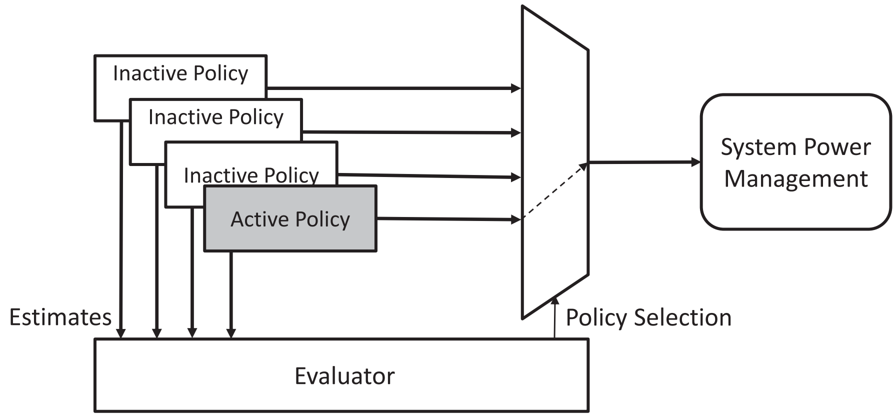

Fig. 9. Hybrid power management consists of a set of candidate power policies and an evaluator. Each candidate policy outperforms the others under different workloads. The candidate policies are evaluated based on a criteria of energy savings or delay during the policy evaluation. The best performing policy becomes active and manages the printer power for the next policy period.

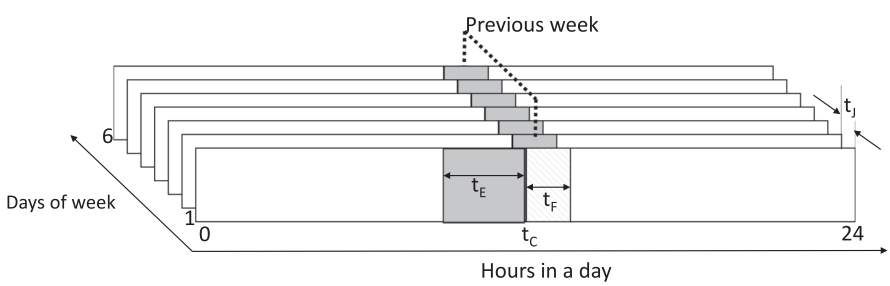

Fig. 10. Hybrid power manager with a policy evaluation window (t_E) and freeze window (t_F). The best performing policy during the evaluation window is applied for the next freeze window. The evaluation window can be also set from the previous day or week.

as shown in Figure 11. The printer is connected to the main power source through a power meter. The power meter with a sensitivity of 1mW logs the power dissipated at an interval of 1 second. The power meter is connected to a network enabling readings to be logged remotely. The printer sleep function is enabled with a factory default timeout of 30 minutes. Once the printer serves a request, it waits for a period of 30 minutes before transitioning to a low-power sleep state. The standard ISO/IEC 24712:2006 color test pages are used for printing and measuring the parameters. Figure 12 shows the power meter readings for a test page printing. From Figure 12, we observe that the printer consumes 12W of power, on average, until the first request. As explained in Section 3.2, a printer consumes a nominal power during the sleep state, with most of the components turned off. There is a surge in power consumption with a request. The power surge reaches as high as 1,000W. This surge is due to the transient current drawn by the fuser heating the rod and other mechanical components like fan and motors. The power consumption stabilizes at around 600W after the wake-up. The printer then enters the ready state consuming an average power of 70W waiting for the future requests. The printer waits until a preset timeout (30 minutes) before entering the sleep state. The energy consumed and the elapsed time is measured for both the cases of printing from the sleep and ready state. The aforementioned experiment is repeated with 1 page, 10 pages, and 50 pages per request. Table I lists the measured printer parameters.

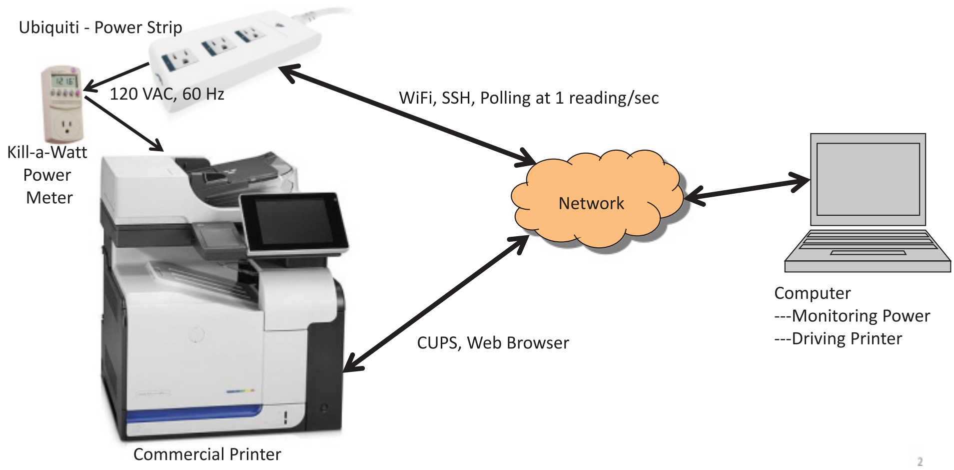

Fig. 11. Printer power measurement setup. The power meter and the printer are both connected to a network. A computer is used to control and access both the power meter and the printer. Common unix printing system (CUPS) standard is used to print to a network printer. Test pages are printed and the corresponding power readings are logged. The power meter refreshes the readings after each second. A log file is generated with the timestamps and power readings.

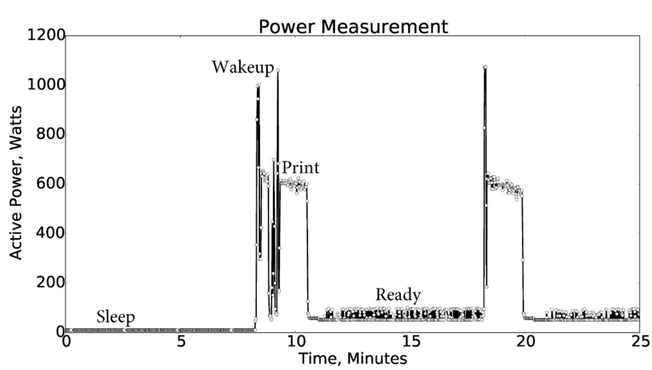

Fig. 12. Printer power measurement reading. A test page is printed while the printer is in (1) ready and (2) sleep state. Power readings are logged at each second. The printer starts in the sleep state consuming a low power. It serves a request consuming power at 650W. The printer enters the ready state, waiting for further requests, consuming an average ready power of 70W. It consumes more power while printing from the sleep state than from the ready state. The printer takes more time serving a request from the sleep state.

Table I. Measured Printer Parameters

SI.No	Parameter	Measured Values, Average
1	Wake-up delay	5 seconds
2	Wake-up energy	2,909J
3	Ready power	70W
4	Sleep power	12W
5	Break-even time	1.54 minutes

Table II. Quad-Phase Parameters Using Pattern Search (Refer to Figure 6 for the Definitions of TO and IA)

Printer	Timeouts (seconds)				Inter-arrival Thresholds (seconds)		
	TO_0	TO_1	TO_2	TO_3	$IA_{0,1}$	$IA_{1,2}$	$IA_{2,3}$
Printer-1	808	669	425	334	141	610	682
Printer-2	262	214	134	65	55	232	292
Printer-3	114	65	52	50	72	91	101
Printer-4	207	206	203	91	4	125	307
Printer-5	419	372	234	140	74	295	514

Table III. Dual-Phase Parameters Using Pattern Search (Refer to Figure 6 for the Definitions of TO and IA)

Printer	Timeouts (seconds)		Inter-arrival Thresholds (seconds)
	TO_0	TO_1	$IA_{0,1}$
Printer-1	227	137	1,155
Printer-2	121	72	91
Printer-3	52	0	1,256
Printer-4	141	73	76
Printer-5	240	103	115

5.2. Adaptive Multiphase Power Management

We compare the multiphase policy's performance with several other policies: fixed 1-minute timeout, factory pre-set 30-minute timeout, and immediate sleeping. We perform simulations in MATLAB with the measured printer parameters and the printer traces. The simulation setup consists of the following: (1) the printer trace, (2) the printer parameters (wake-up delay, wake-up energy, active power, and sleep power), and (3) the MATLAB pattern search optimizer to compute multiphase parameters. The trace provides the sequence of inter-arrival times from the timestamps associated with each request used by the the printer model to compute the cumulative energy and delay. Based on the policy, a corresponding timeout is applied for the simulation. The printer selects the states: ready, active, sleep, and accumulates the energy and delay. At the end of the simulation, the average power and delay is computed. The pattern search optimizer is used to compute the multiphase policy's timeout and inter-arrival thresholds, which achieve a target delay (e.g., less than 3 seconds). Tables II and III provide the parameters computed from the pattern search for the quad-phase and the dual-phase policies. The pattern search is run multiple times to prevent false convergence. The simulation is performed on the entire trace. However, in actual implementation, a sliding window can be adopted, whereby new sets of parameters (thresholds and timeouts) are computed periodically from the available requests. For example, every day a new set of parameters can be computed from the past week's requests. The quad-phase policy consists of four decreasing timeouts for phases ranging from the busiest to the least busy, and five increasing inter-arrival thresholds. Current simulation divides 1 hour into equal blocks based on the number of phases, and initializes the parameters, assigning random values from each block. The search is performed 256 times, and each time, the pattern search provides a set of parameters satisfying the delay constraint. The set of parameters that provide a low power is selected for the power management.

Figure 13 depicts the average power dissipated and the average delay per request encountered for different policies. The oracle policy provides the lower bound for power, as it knows the future and powers down appropriately. Fixed timeouts are poor at balancing both delay and power consumption. Immediate sleep and 1-minute timeout shuts down the printer very often causing increased delay. The 30-minute-longer timeout

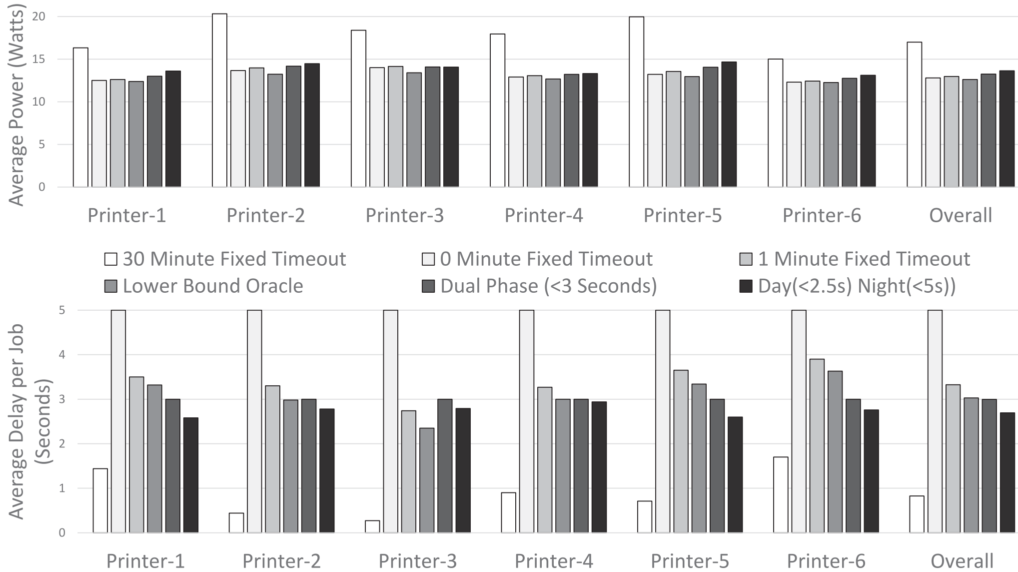

Fig. 13. Comparison of average power and average delay for different policies. The oracle policy saves the most energy, as it knows the future. The factory preset fixed 30-minute timeout keeps the printer awake most of the time and, hence, less delay per request, but has the highest power consumption. Immediate shutdown has to wake-up for each request and, hence, has the most delay per request. Multiphase policy balances delay with marginal penalty in power consumption.

rarely shuts down the printer and can serve a request immediately. However, it consumes the maximum power compared with other policies. As an example, for printer-5, the oracle policy expends on an average of 12.9 watts with 3.3 seconds delay. The fixed timeout policies (30 min, 1 min, and immediate shutdown) consume 19.9, 13.6, and 13.2W with 0.7s, 3.7s, and 5s delay, respectively. The multiphase policy consumes 14W with 3 seconds of delay. Thus, the multiphase policy achieves the performance goals with a marginal increase in power. The multiphase policy utilizes multiple timeouts to achieve the performance goals. The set of timeouts for a particular trace provides the best energy savings under a delay constraint. An oracle would choose a large timeout during busy periods and immediate shut down during reduced usage periods. The multiphase policy classifies the printer requests into finite activity phases. The printer incrementally moves to a relevant activity phase corresponding to the workload. Upon reaching the relevant activity phase, a corresponding timeout, determined using pattern search, is applied. The new set of thresholds and timeouts are computed again after a period with the consolidated requests.

5.3. Hybrid Power Management

The candidate power policies for the hybrid power management consists of the following:

(1) Fixed Timeout - 60 Seconds
(2) Adaptive Multiphase Power Management
(3) Adaptive Timeout
(4) 2-Competitive

A fixed timeout policy employs a single timeout, irrespective of the workload. Fixed timeouts of 1 minute and break-even time are considered. The adaptive multiphase

Table IV. Average Power (Watts) for the Candidate Power Policies

Printer Trace	Multiphase	Fixed-60	Adaptive	2-Competitive	Hybrid
Printer-1	12.89	12.61	12.54	12.70	12.46
Printer-2	13.94	13.96	13.96	14.20	13.59
Printer-3	14.98	15.22	15.86	15.42	14.95
Printer-4	12.91	12.69	12.68	12.77	12.56
Printer-5	13.88	13.59	13.48	13.81	13.23
Printer-6	14.44	14.49	14.39	14.76	14.00
Printer-7	13.07	12.72	12.66	12.80	12.58
Printer-8	12.62	12.40	12.38	12.45	12.33
Printer-9	12.31	12.37	12.36	12.40	12.31
Printer-10	12.73	12.49	12.44	12.56	12.37
Printer-11	13.21	12.85	12.80	12.92	12.72
Printer-12	13.04	12.44	12.42	12.46	12.39
Printer-13	13.96	12.85	12.79	12.94	12.69
Printer-14	13.91	13.50	13.41	13.71	13.16
Printer-15	13.33	12.74	12.67	12.85	12.56
Printer-16	14.53	14.56	14.50	14.69	14.33
Printer-17	14.27	14.03	13.95	14.29	13.60
Printer-18	12.48	12.34	12.30	12.39	12.26

No single policy provides low power for all the printers. The hybrid policy performs better than the best performing policy.

policy uses different phases each with an associated timeout. A set of inter-arrival thresholds are used to transition between phases based on the current inter-arrival time. Each multiphase parameter is computed from the past known requests, as explained in Section 4.2. The current simulation uses a dual-phase policy with a delay constraint of less than 3 seconds. A sliding window approach is used for computing the parameters. The parameters are computed every day using the past week's requests. The adaptive power policy uses a dual timeout: 0 and break-even time, based on the inter-arrival time. The simulation setup consists of the following: (1) the printer trace (2) the printer parameters, and (3) the policy selection. The sequence of inter-arrival times, busy times, day of the week, and the type of request (print, copy, etc.) are computed using the printer trace. Each of the policies are simulated using MATLAB, with the measured printer parameters generating accumulated energy, delay, number of events, total trace duration, and so on, and used for comparisons and analysis. Table IV lists the average power, and Table V lists the average delay for each of the candidate power policies and printers. It is observed that no single policy is effective for all the printers. For example, the multiphase policy provides better energy savings for printers 2, 3, and 9. Whereas the adaptive policy provides energy savings for printers 1, 5, 6, and 7. Similarly, low delay is achieved using the multiphase policy for printers 1, 5, and 7, and using 2-competitive policy for the printers 3, 6, and 9.

The hybrid policy switches the candidate policies and provides better performance than the best performing individual policy. Current simulation uses the energy savings as the policy selection criteria. The hybrid policy evaluates the candidate policies' energy consumption during an evaluation window to select the best policy. The best policy performs the power management for a duration called policy freeze window. At the end of the policy freeze window, another evaluation is performed. From Table IV, the hybrid policy provides more consistent energy savings than the best policy. The hybrid policy uses a combination of the individual policies to achieve better energy savings. Figure 14 shows the frequency of individual policy selection for printers at different offices. Depending upon the workload, each policy is selected to achieve better energy savings. For example, the hybrid policy selects the multiphase policy for 6.5%, the

Table V. Average Delay (Seconds) for the Candidate Power Policies

Printer Trace	Multiphase	Fixed-60	Adaptive	2-Competitive	Hybrid
Printer-1	2.84	3.26	3.47	3.08	3.14
Printer-2	3.07	3.05	3.39	2.80	2.87
Printer-3	2.78	2.23	2.44	1.96	2.50
Printer-4	2.80	2.97	3.23	2.77	2.81
Printer-5	3.19	3.44	3.76	3.22	3.30
Printer-6	2.94	2.82	2.98	2.50	2.53
Printer-7	3.15	3.47	3.76	3.26	3.38
Printer-8	2.67	3.19	3.47	3.04	3.03
Printer-9	1.35	1.15	1.25	1.08	1.12
Printer-10	3.13	3.65	3.95	3.47	3.47
Printer-11	2.74	3.28	3.57	3.09	3.09
Printer-12	2.67	3.37	3.73	3.21	3.19
Printer-13	2.77	3.55	3.85	3.41	3.45
Printer-14	2.96	3.31	3.60	3.13	3.03
Printer-15	2.97	3.58	3.80	3.41	3.46
Printer-16	3.44	3.36	3.72	3.15	3.25
Printer-17	3.10	3.38	3.69	3.09	3.18
Printer-18	3.08	3.54	3.75	3.30	3.31

No single policy provides low delay for all the printers.

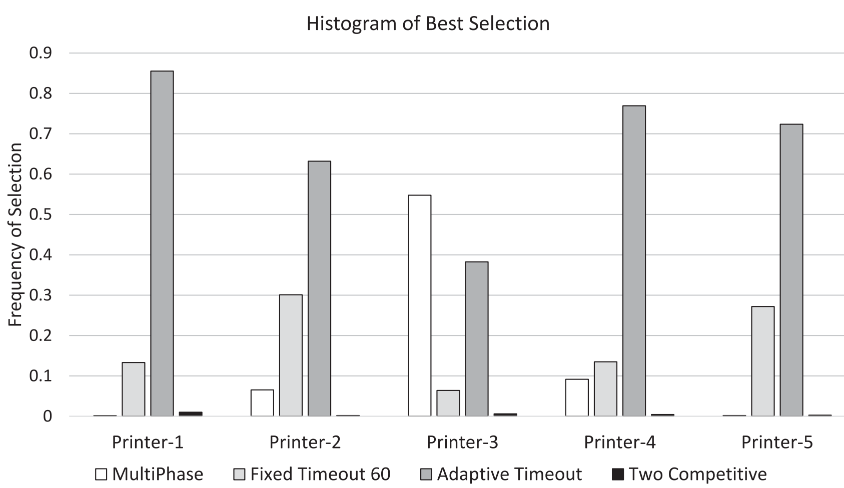

Fig. 14. Histogram of power management policies selected by the hybrid policy. This shows that the best policies vary with the underlying requests.

adaptive policy for 63%, and the fixed 1-minute timeout for 30% of the time for printer-2. Thus, there are instances during the printer activity wherein different policies provide better energy savings. Hence, a single policy is ineffective for the entire workload and better energy savings can be achieved using a combination of policies.

5.4. Evaluation Window Selection

We next perform policy selection simulations on printers from 18 different offices to find the evaluation and freeze windows that achieve the best energy savings. As explained in the previous section, the hybrid policy with four different policies is used for the simulation. We simulate the hybrid policy with different sets of time periods for the evaluation and freeze window. The best policy based on the energy saving criteria

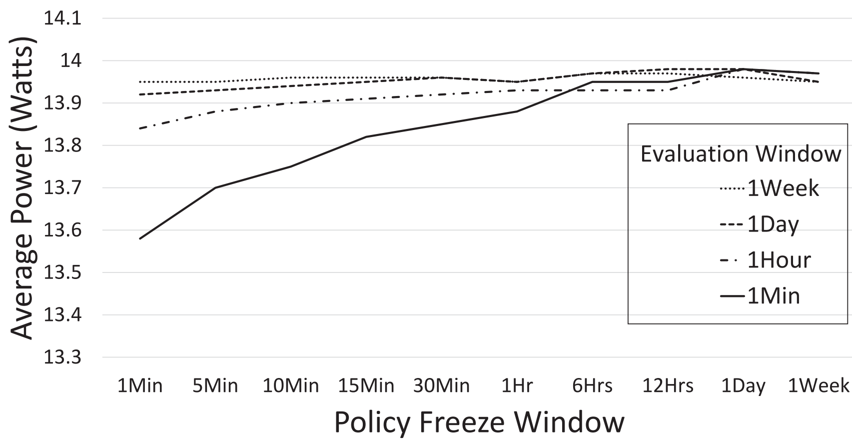

Fig. 15. Average power in watts for Printer-2 with hybrid policy using different evaluation and freeze windows.

during an evaluation window is selected and activated for the next freeze window. The following evaluation windows are considered for the simulation: 1 minute, 1 hour, 1 day, and 1 week. The following freeze windows are considered: 1, 10, and 30 minutes; 1, 6, and 12 hours, 1 day, and 1 week.

Figure 15 shows the average power for different evaluation and freeze windows. We see that using a short evaluation and freeze window, better energy savings can be achieved. For the current simulation, the hybrid policy using 1-minute evaluation and freeze window provides the best energy savings compared with the other combinations. The hybrid policy using a shorter evaluation window relies less on the distant past requests and, hence, the selected policy is more suited for the current workload. We see that using a longer evaluation window, the energy consumption approaches that of the best individual policy. A shorter freeze window enables the hybrid policy to update and switch policies more often based on the workload variation. However, using a longer freeze window, the hybrid policy is restricted in updating the policies with the workload variations. Also, any wrong selection of policy would take a longer time to be corrected and, hence, energy is wasted. Thus, the hybrid policy, using short-term memory and updating frequently, provides the best energy savings.

5.5. Week Analysis

We next evaluate energy saving using information from the previous week for policy selections. This is motivated by the printer request pattern, as explained in Section 4.1. Most of the printer traces display distinct request patterns—with significant requests during weekdays, followed by limited requests during weekends. Similarly, requests are frequent during working hours as compared with non-working hours. We verify any energy savings using this scheduling information. Table VI lists the simulation results for the hybrid policy with scheduling information from the previous week. We evaluate the candidate power policies from the previous week at the current time. For example, if the current time is 11 a.m., the candidate policies are evaluated from the previous week at exactly 11 a.m. The policy that provides the best energy savings for most number of days is selected. From Table VI, we observe that the regular hybrid policy without using any scheduling information performs better than the policy using scheduling information. The current workload cannot be characterized using the previous week's information, although we observe a pattern among printer requests. The current workload is different from the workload that existed during the previous

Table VI. Average Power in Watts for the Previous Week Schedule (Refer Figure 10 for the Definition of Freeze Window from the Past Week)

Printer trace	1Min	10Min	30Min	1 Hour	6 Hours	12 Hours	1 Day	Hybrid
Printer-1	12.56	12.56	12.55	12.55	12.55	12.55	12.54	12.46
Printer-2	13.98	13.98	13.98	14.00	13.99	13.99	13.98	13.59
Printer-3	15.32	15.30	15.28	15.23	15.12	15.11	15.11	14.95
Printer-4	12.68	12.68	12.68	12.68	12.68	12.68	12.68	12.56
Printer-5	13.50	13.50	13.52	13.52	13.51	13.51	13.48	13.23
Printer-6	14.39	14.38	14.36	14.37	14.37	14.38	14.40	14.00
Printer-7	12.67	12.67	12.67	12.67	12.67	12.67	12.67	12.58
Printer-8	12.41	12.41	12.41	12.41	12.41	12.41	12.39	12.33
Printer-9	12.36	12.36	12.36	12.36	12.35	12.35	12.33	12.31
Printer-10	12.44	12.44	12.44	12.44	12.44	12.44	12.44	12.37
Printer-11	12.83	12.83	12.83	12.83	12.82	12.82	12.83	12.72
Printer-12	12.42	12.42	12.42	12.42	12.42	12.42	12.43	12.39
Printer-13	12.79	12.79	12.79	12.79	12.79	12.79	12.79	12.69
Printer-14	13.46	13.46	13.46	13.46	13.45	13.46	13.43	13.16
Printer-15	12.68	12.68	12.69	12.68	12.68	12.68	12.69	12.56
Printer-16	14.47	14.48	14.48	14.47	14.47	14.47	14.47	14.33
Printer-17	14.00	13.99	13.99	13.98	14.02	14.00	13.97	13.60
Printer-18	12.31	12.31	12.31	12.31	12.31	12.31	12.31	12.26

The hybrid policy, using a short-term memory of past requests and frequent updates, performs better than the policies using scheduling information from the previous week.

week. Hence, scheduling information from the past requests (previous week) does not provide consistent energy savings. This finding suggests that power management based on working hour schedule is ineffective. To our knowledge, this is the first study showing schedule-based power management has no advantages.

6. SUMMARY

In this article, two power management policies are presented for office equipment: an adaptive multiphase policy and a hybrid policy. The multiphase policy divides the printer workload into finite phases. The printer transitions to a relevant phase, based on the current workload, and an appropriate timeout is applied, balancing power and delay. The hybrid policy selects the best performing policy to perform system power management. We obtain the appropriate policy evaluation and freeze windows and show that frequent updates with short-term memory provides the best performance. We also show that policies relying on scheduling information do not provide consistent energy savings. We perform the simulations measuring the printer parameters from a commercial printer and use the request traces from printers with contrasting usages.

REFERENCES

Luca Benini, Alessandro Bogliolo, and Giovanni De Micheli. 2000. A survey of design techniques for system-level dynamic power management. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 8, 3, 299–316.

Luca Benini and Giovanni de Micheli. 2000. System-level power optimization: Techniques and tools. *ACM Transactions on Design Automation of Electronic Systems (TODAES)* 5, 2, 115–192.

Blue-Angel. 2013. *Blue Angel - Basic Criteria for Award of the Environmental Label - Office Equipment with Printing Function*.

Eui-Young Chung, Luca Benini, Alessandro Bogliolo, Yung-Hsiang Lu, and Giovanni De Micheli. 2002. Dynamic power management for nonstationary service requests. *IEEE Transactions on Computers 51*, 11, 1345–1361.

Victor Ciriza, Christopher R. Dance, and Laurent Donini. 2012. Printer time-out. US Patent 8,230,248.

Victor Ciriza, Laurent Donini, Jean-Baptiste Durand, Stéphane Girard, et al. 2008. A statistical model for optimizing power consumption of printers. In *40èmes Journées de Statistique*.

William C. Davidon. 1991. Variable metric method for minimization. *SIAM Journal on Optimization* 1, 1, 1–17.

Gaurav Dhiman and Tajana Simunic Rosing. 2006. Dynamic power management using machine learning. In *Proceedings of the 2006 IEEE/ACM International Conference on Computer-aided Design*. ACM, 747–754.

Gaurav Dhiman and Tajana Simunic Rosing. 2009. System-level power management using online learning. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. IEEE, 676–689.

Fred Douglis, Padmanabhan Krishnan, Brian Bershad, et al. 1995. Adaptive disk spin-down policies for mobile computers. *Computing Systems* 8, 4, 381–413.

Jean-Baptiste Durand, Stéphane Girard, Victor Ciriza, and Laurent Donini. 2013. Optimization of power consumption and device availability based on point process modelling of the request sequence. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 62, 2, 151–165.

Energy-Star. 2014. *Energy Star Program Requirements for Imaging Equipment Partner Commitments*. https://www.energystar.gov/ia/partners/product_specs/program_reqs/Imaging_Equipment_Program_Requirements.pdf.

Richard Golding, Peter Bosch, and John Wilkes. 1995. Idleness is not sloth. In *Proceedings of the USENIX Winter Conference*, 201–212.

Ying Guo, Rongxin Li, Geoff Poulton, and Astrid Zeman. 2008. A simulator for self-adaptive energy demand management. In *Proceedings of the 2nd IEEE International Conference on Self-Adaptive and Self-Organizing Systems. SASO.* IEEE, 64–73.

David P. Helmbold, Darrell D. E. Long, Tracey L. Sconyers, and Bruce Sherrod. 2000. Adaptive disk spin-down for mobile computers. *Mobile Networks and Applications* 5, 4, 285–297.

Hewlett-Packard. 2008. *HP Color LaserJet CM3530 MFP Series Service Manual*. Hewlett-Packard.

Chi-Hong Hwang and Allen C.-H. Wu. 2000. A predictive system shutdown method for energy saving of event-driven computation. *ACM Transactions on Design Automation of Electronic Systems (TODAES)* 5, 2, 226–241.

Sandy Irani and Kirk R. Pruhs. 2005. Algorithmic problems in power management. *ACM SIGACT News* 36, 2, 63–76.

Sandy Irani, Gaurav Singh, Sandeep K. Shukla, and Rajesh K. Gupta. 2005. An overview of the competitive and adversarial approaches to designing dynamic power management strategies. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 13, 12, 1349–1361.

Anna R. Karlin, Mark S. Manasse, Lyle A. McGeoch, and Susan Owicki. 1994. Competitive randomized algorithms for nonuniform problems. *Algorithmica* 11, 6, 542–571.

Kaoru Kawamoto, Yoshiyuki Shimoda, and Minoru Mizuno. 2004. Energy saving potential of office equipment power management. *Energy and Buildings* 36, 9, 915–923.

Umair Ali Khan and Bernhard Rinner. 2014. Online learning of timeout policies for dynamic power management. *ACM Transactions on Embedded Computing Systems (TECS)* 13, 4, 96.

Jonathan Knoder. 2015. Best Laser Printer Reviews and Comparisons. Retrieved from http://printers.toptenreviews.com/laser/.

Tamara G. Kolda, Robert Michael Lewis, and Virginia Torczon. 2003. Optimization by direct search: New perspectives on some classical and modern methods. *SIAM Review* 45, 3, 385–482.

Bradley R. Larson. 2011. Managing a power state for a peripheral. US Patent 7,904,739.

Robert Michael Lewis, Virginia Torczon, and Michael W. Trosset. 2000. Direct search methods: Then and now. *Journal of Computational and Applied Mathematics* 124, 1, 191–207.

Yung-Hsiang Lu, Eui-Young Chung, Tajana Simunic, Luca Benini, and Giovanni De Micheli. 2000. Quantitative comparison of power management algorithms. In *Design, Automation and Test in Europe Conference and Exhibition. Proceedings*. IEEE, 20–26.

Inc. MathWorks. 2015. Matlab Global Optimization Toolbox - Patternsearch. Retrieved from http://www.mathworks.com/help/gads/patternsearch.html.

Alan K. Meier and Benoit LeBot. 1999. One watt initiative: A global effort to reduce leaking electricity. *Lawrence Berkeley National Laboratory*.

Nordic-Swan. 2013. *Nordic Ecolabelling of Imaging Equipment*.

Nathaniel Pettis and Yung-Hsiang Lu. 2009. A homogeneous architecture for power policy integration in operating systems. *IEEE Transactions on Computers* 58, 7, 945–955.

Qinru Qiu and Massoud Pedram. 1999. Dynamic power management based on continuous-time Markov decision processes. In *Proceedings of the 36th Annual ACM/IEEE Design Automation Conference*. ACM, 555–561.

Dinesh Ramanathan and Rajesh Gupta. 2000. System level online power management algorithms. In *Design, Automation and Test in Europe Conference and Exhibition. Proceedings*. IEEE, 606–611.

Hao Shen, Ying Tan, Jun Lu, Qing Wu, and Qinru Qiu. 2013. Achieving autonomous power management using reinforcement learning. *ACM Transactions on Design Automation of Electronic Systems (TODAES)* 18, 2, 24.

Hung-Cheng Shih and Kuochen Wang. 2012. An adaptive hybrid dynamic power management algorithm for mobile devices. *Computer Networks* 56, 2, 548–565.

Mani B. Srivastava, Anantha P. Chandrakasan, and Robert W. Brodersen. 1996. Predictive system shutdown and other architectural techniques for energy efficient programmable computation. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 4, 1, 42–55.

Ying Tan, Wei Liu, and Qinru Qiu. Adaptive power management using reinforcement learning. In *Proceedings of the 2009 International Conference on Computer-Aided Design*. ACM, 461–467.

Gerald Tesauro, Rajarshi Das, Hoi Chan, Jeffrey Kephart, David Levine, Freeman Rawson, and Charles Lefurgy. 2007. Managing power consumption and performance of computing systems using reinforcement learning. In *Advances in Neural Information Processing Systems*. 1497–1504.

Georgios Theocharous, Shie Mannor, Nilesh Shah, Prashant Gandhi, Branislav Kveton, Sajid Siddiqi, and Chih-Han Yu. 2006. Machine learning for adaptive power management. *Intel Technology Journal* 10, 299–312.

Virginia Torczon. 1997. On the convergence of pattern search algorithms. *SIAM Journal on Optimization* 7, 1, 1–25.

Yanzhi Wang, Qing Xie, Ahmed Ammari, and Massoud Pedram. 2011. Deriving a near-optimal power management policy using model-free reinforcement learning and Bayesian classification. In *Proceedings of the 48th Design Automation Conference*. ACM, 41–46.

Received September 2015; revised February 2016; accepted March 2016