

Optimizing Energy-Performance Trade-Offs in Solar-Powered Edge Devices

Peter G. Harrison

Imperial College London
pgh@doc.ic.ac.uk

Naresh M. Patel

NetApp Inc., Sunnyvale, California
naresh@netapp.com

ABSTRACT

Power modes can be used to save energy in electronic devices but a low power level typically degrades performance. This trade-off is addressed in the so-called EP-queue model, which is a queue depth dependent M/G/1 queue augmented with power-down and power-up phases of operation. The ability to change service times by power settings allows us to leverage a Markov Decision Process (MDP). We illustrate this approach by using a simple fully solar-powered case study with finite states representing levels of battery charge and solar intensity.

ACM Reference Format:

Peter G. Harrison and Naresh M. Patel. 2018. Optimizing Energy-Performance Trade-Offs in Solar-Powered Edge Devices. In *ICPE '18: ACM/SPEC International Conference on Performance Engineering, April 9–13, 2018, Berlin, Germany*. ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/3184407.3184426>

1 INTRODUCTION

We consider an edge device that is fully powered by solar energy and connected to a cellular data network for communicating with the core services. Although the energy generated is not under our control and can vary by time of day (and time of year), the energy consumed can be altered by adjusting the processor clock rate which in turn impacts the application response time. This creates a trade-off between the energy consumed and the response time delivered by the cognitive system at the edge. We would like to adjust the clock rates judiciously throughout the day so that acceptable quality of service is delivered without running out of battery power given arrival rates and battery charging rates that depend on the time of day.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

ICPE '18, April 9–13, 2018, Berlin, Germany

© 2018 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-5095-2/18/04.

<https://doi.org/10.1145/3184407.3184426>

To illustrate one particular type of application operating on the edge, consider a camera endpoint that sends a video stream to a compute engine running a deep learning algorithm to identify gender and age of people passing by. The processing needs to be fast enough to classify the video clip and display age/gender-appropriate adverts and information. The inferred output data has some transient value when a person is in the line of sight of the display but that value vanishes as the person walks away. For example, this happens to Tom Cruise in the movie “Minority Report” as he is identified by a retina scan and immediately shown targeted adverts.

An arrival stream of people can often be modeled realistically as a Poisson process, where multiple people walking together at the same time can be modeled as a batch with specified batch-size probability distribution at each arrival instant. The service time will depend on the speed of inference of the learning algorithm, and we assume that the processor clock frequency can be adjusted downward to save energy. In this model, the service times have a general probability distribution but also the service rates vary with the queue depth. The EP-queue provides precisely these features, and more, for example power-up and power-down periods that have their own power demands [6]. In this paper, we derive expressions for the amount of energy consumed as well as standard performance metrics during a non-idle period, and create penalty metrics to make trade-offs at regular periods during the day. The objective is to find the policy or power settings for each period that minimizes the accumulated penalty at the end of the day.

2 ENERGY CONSUMPTION

2.1 Model definition

We extend the generalized M^B/G/1 queue defined in [6], with Poisson batch-arrivals and service times that are state-dependent when the queue length i (including the task in service, if any) at the start of a service period is less than some threshold $n \geq 1$; the service time random variable is denoted S_i when $1 \leq i < n$ and S_n when $i \geq n$. The Poisson batch-arrivals have rate λ , the batch size is an integer random variable B with probability generating function $G(z) = \sum_{i=1}^{\infty} b_i z^i$ and the service discipline is first come first served (FCFS). In addition, the server has to be powered-up when an arrival occurs in its idle state (queue length 0) and powered-down when a departure leaves the queue empty. Power-up and

power-down times are independent random variables, denoted by U and D respectively. If an arrival occurs during power-down, the power-down continues unaffected and is immediately followed by a power-up period, immediately after which the first task to have arrived commences service. Thus the non-idle, (partially or fully) powered-up period is elongated beyond the regular busy period, which is simply a maximal time period throughout which there is at least one task in the queue. This queue was called an EP_n^G -queue in [6]. It reduces to a standard M/G/1 queue when $U = D = 0$ with probability one, $G(z) = z$, $n = 1$.

Both power levels and service times are functions of the clock frequency of a device, which we take to be ν_i when the queue length is $i > 0$ at the start of service of a task, remaining constant until the end of the task's service period. The power level, or rate of energy consumption, $\omega_i(\nu_i)$ at queue length $i > 0$ is a function of the clock frequency; to a coarse approximation, $\omega_i \propto \nu_i^2$, but more accurate functions can be found through profiling. The energy ϵ_i used during a service time S_i that started when the queue length was $i > 0$ may then be approximated by $\epsilon_i = S_i \omega_i(\nu_i)$, although this too is an approximation due to the fact that a processor is not necessarily running at full power throughout an instruction cycle. Similarly, a common approximation is that the rate at which a task receives service is proportional to the clock frequency, so we may write $S_i = \Delta/\nu_i$, where the random variable Δ is the number of clock cycles required by a task. Under these approximations, the service times and energy units consumed by a task may be parameterized in terms of the power levels by $S_i \sim \sqrt{\frac{\omega_n}{\omega_i}} S_n$ and $\epsilon_i \sim \omega_i S_i$, where we use the symbol \sim to signify "has the same distribution as". Like the service times, the power levels ω_i have a threshold at $i = n$ so that $\omega_i = \omega_n$ for all $i \geq n$. Thus reducing the power level by a factor of four results in doubling the service times, which may be reasonable when queue depths are low.

Of course, if all power levels ω_i are now replaced by the value 1, after first resetting the distributions of S_i as above, ϵ_i is the same as the service time S_i . This is just a mathematical device that conveniently uses one general expression, rather than having to work with multiple expressions: one with energy units and the other, almost identical, with time units. This observation is used to find the moments of the length of the non-idle cycle (either busy, powering up or powering down), required in the next section.

We use the following notation regarding random variables. The cumulative distribution function of a continuous random variable X is denoted by $X(t) = \mathbb{P}(X \leq t)$ and its Laplace-Stieltjes transform (LST) by $X^*(\theta) = \mathbb{E}[e^{-\theta X}]$. The probability density function (PDF) of X is $x(t) = X'(t)$, the derivative of the distribution function. The m^{th} moment of X is written $X_{[m]} = \mathbb{E}[X^m] = (-1)^m X^{*(m)}(0)$, where the parenthesized superscript denotes differentiation m times with respect to θ . Correspondingly, the probability generating function (pgf) of a discrete random variable Y is written $G_Y(z) = \mathbb{E}[z^Y]$ (so that $G(z)$ is an abbreviation for $G_B(z)$).

2.2 Energy usage in non-idle periods

2.2.1 Underlying recurrence formula. Let the energy used between the start of service of a task at queue length i and the first subsequent instant at which the server becomes fully powered down, or idle, be denoted by the random variable W_i , for $i \geq 1$. Further, let W_0 denote the energy used between an instant at which the queue becomes empty and the end of the current non-idle period. The time period associated with W_0 is not just the power-down period D since it may be that new tasks will arrive during this period, starting a busy period after the power-down period has been completed¹. Then the energy used in the non-idle period, $H = \omega_U U + W_{B+N_U}$, where B is the number of tasks in the batch that started the non-idle period and N_U is the number of task-arrivals during the powering-up period U ; N_D is defined similarly with respect to the power-down period D . Let the power levels during power-up and power-down be ω_U and ω_D , respectively. Then, for $j \geq 1$, we have the recurrence $W_j = \omega_j S_j + W_{N_{S_j}+j-1}$, where $W_0 = \omega_D D + (\omega_U U + W_{N_U+N_D}) \mathbf{I}_{N_D > 0}$, N_{S_j} is the number of task-arrivals during the service time S_j and \mathbf{I} is the indicator function. For example, when there are no arrivals during the service time that started with j tasks in the queue ($N_{S_j} = 0$), the recurrence is simply the sum of the energies used during that service time and in the non-idle period that starts with $j-1$ tasks.

2.2.2 Energy PDF in non-idle periods. The LSTs $W_i^*(\theta)$ for $i = 1, 2, \dots$ are given by the following series of lemmas and propositions, culminating in Theorem 1. The proofs of the lemmas are generalizations to corresponding results in [6] for time delay distributions' LSTs.

LEMMA 1. For $i \geq n-1$,

$W_i^*(\theta) = (V^*(\omega_n \theta))^{i-n+1} W_{n-1}^*(\theta)$ where the function V^* is the fixed point of the equation $v^*(\theta) = S_n^*(\theta + \lambda(1 - G(v^*(\theta))))$. For $1 \leq i < n-1$, omitting the arguments θ from W^* for brevity,

$$W_i^* = \left[S_i^*(\omega_i \theta + \lambda(1 - G(V^*(\omega_n \theta)))) (V^*(\omega_n \theta))^{i-n} - \sum_{j=0}^{n-i-1} s_{ij}(\theta) (V^*(\omega_n \theta))^{j+i-n} \right] W_{n-1}^* + \sum_{j=0}^{n-i-1} s_{ij}(\theta) W_{j+i-1}^*$$

where $W_0^*(\theta)$ is to be determined and, for $1 \leq i \leq n, 0 \leq j \leq n$,

$$s_{ij}(\theta) = \frac{1}{j!} \left. \frac{\partial^j S_i^*(\omega_i \theta + \lambda(1 - G(z)))}{\partial z^j} \right|_{z=0}.$$

¹We assume that once started, a power-down period must complete fully before a new startup period can begin. Other modi operandi are possible, e.g., the startup period could begin immediately.

LEMMA 2.

$$W_0^*(\theta) = D^*(\lambda + \omega_D\theta) + \sum_{k=1}^{n-2} \left[W_k^*(\theta) - W_{n-1}^*(\theta) V^*(\omega_n\theta)^{k-n+1} \right] \sum_{j=1}^k u_{k-j}(\theta) d_j(\theta) + W_{n-1}^*(\theta) V^*(\omega_n\theta)^{-n+1} U^*(\omega_U\theta + \lambda(1 - G(V^*(\omega_n\theta)))) \times \left[D^*(\omega_D\theta + \lambda(1 - G(V^*(\omega_n\theta)))) - D^*(\lambda + \omega_D\theta) \right]$$

$$\text{where } u_j(\theta) = \frac{1}{j!} \frac{\partial^j U^*(\omega_U\theta + \lambda(1 - G(z)))}{\partial z^j} \Big|_{z=0} \text{ and } d_j(\theta) = \frac{1}{j!} \frac{\partial^j D^*(\omega_D\theta + \lambda(1 - G(z)))}{\partial z^j} \Big|_{z=0}.$$

The next result defines an algorithm for the computation of $\{W_i^*(\theta) \mid i \geq 0\}$.

PROPOSITION 1. For each θ and $i \geq 1$,

$W_i^*(\theta) = W_0^*(\theta) \tau_i(\theta) / \tau_0(\theta)$, where, omitting the arguments $(\omega_U\theta + \lambda(1 - G(V^*(\omega_n\theta))))$ from U^* and $(\omega_D\theta + \lambda(1 - G(V^*(\omega_n\theta))))$ from D^* ,

$$W_0^*(\theta) = D^*(\lambda + \omega_D\theta) \tau_0(\theta) / \left[\tau_0(\theta) - V^*(\omega_n\theta)^{-n+1} U^* [D^* - D^*(\lambda + \omega_D\theta)] - \sum_{k=1}^{n-2} (\tau_k(\theta) - V^*(\omega_n\theta)^{k-n+1}) \sum_{j=1}^k u_{k-j}(\theta) d_j(\theta) \right] \\ \tau_i(\theta) = V^*(\omega_n\theta)^{i-n+1} \text{ for } i \geq n-1; \\ \tau_i(\theta) = \frac{1}{s_{i+1,0}(\theta)} \left[\tau_{i+1}(\theta) - \sum_{j=1}^{n-i-2} s_{i+1,j}(\theta) \tau_{i+j}(\theta) - \left(V^*(\omega_n\theta)^{i-n+1} S_{i+1}^*(\omega_{i+1}\theta + \lambda(1 - G(V^*(\omega_n\theta)))) - \sum_{j=0}^{n-i-2} s_{i+1,j}(\theta) V^*(\omega_n\theta)^{i+j-n+1} \right) \right] \text{ for } 0 \leq i < n-1.$$

PROOF. The set of values $\tau_0(\theta), \tau_1(\theta), \dots$ are proportional to $W_0^*(\theta), W_1^*(\theta), \dots$ and given by Lemma 1 up to a constant of proportionality, which is chosen to be such that $\tau_{n-1}(\theta) = 1$ for all θ . Thus, for $i > 0$, $W_i^*(\theta) = W_0^*(\theta) \tau_i(\theta) / \tau_0(\theta)$ and the expression for $W_0^*(\theta)$ is obtained by plugging into Lemma 2. \square

The required LST of the probability distribution of the energy used in a non-idle period – i.e. a contiguous, partially or fully powered-up period – is given by the following:

THEOREM 1. The LST of the probability distribution function of the energy used in a non-idle period, H , is:

$$H^*(\theta) = G(V^*(\omega_n\theta)) U^*(\omega_U\theta + \lambda(1 - G(V^*(\omega_n\theta)))) \times V^*(\omega_n\theta)^{-n+1} W_{n-1}^*(\theta) + \sum_{i=1}^{n-2} b_i \sum_{j=0}^{n-i-2} u_j(\theta) (W_{i+j}^*(\theta) - V^*(\omega_n\theta)^{i+j-n+1} W_{n-1}^*(\theta))$$

PROOF.

$$H^*(\theta) = \mathbb{E}[\mathbb{E}[e^{-\theta(\omega_U U + W_{B+N_U})} \mid U]] \\ = \mathbb{E} \left[\sum_{i=1}^{\infty} \sum_{j=0}^{\infty} b_i \frac{1}{j!} \frac{\partial^j e^{-\lambda U(1-G(z))}}{\partial z^j} \Big|_{z=0} e^{-\theta(\omega_U U + W_{i+j})} \right] \\ = \mathbb{E} \left[\sum_{i=1}^{\infty} \sum_{j=0}^{\infty} b_i \frac{1}{j!} \frac{\partial^j e^{-U(\omega_U\theta + \lambda(1-G(z)))}}{\partial z^j} \Big|_{z=0} e^{-\theta W_{i+j}} \right]$$

since $\frac{1}{j!} \frac{\partial^j e^{-\lambda U(1-G(z))}}{\partial z^j} \Big|_{z=0}$ is the probability that there are j task arrivals during time U . The rest of the proof is now similar to that of lemma 2. \square

Finally, if energy is consumed during the exponentially distributed idle periods, at power level ω_I , the LST of the distribution of the energy used, C say, in a complete idle-busy cycle is $C^*(\theta) = \lambda H^*(\theta) / (\lambda + \omega_I\theta)$.

The moments of C, H and the W_j depend on the moments of V , which has its Laplace transform defined as a fixed point. When we know *a priori* that only a given number p of moments are required, we only need the function $V^*(\theta) = 1 + \sum_{i=1}^p v_i \theta^i$ to order p . The coefficient $v_i = V^{*(i)}(0) / i! = (-1)^i V_{[i]} / i!$ in terms of moments, and it is routine to compute any finite number of these symbolically, using standard mathematical software.

2.3 Energy used in a given time interval

Idle-busy delay-cycles are usually short relative to wall clock times, e.g., energy generation or battery discharge times. We consider the case where a given time period t is many times greater than the mean delay-cycle time c . Suppose that there are n complete cycles that comprise the period t , i.e. $n \simeq \lfloor t/c \rfloor$ to a good approximation at large n . Since cycles are independent and identically distributed (iid), the probability distribution of the combined length of a large number, n , of cycles is well approximated by a Normal distribution with mean nm and variance nv , where m and v are respectively the mean and variance of the length of a single cycle, calculable by the procedure defined in the previous section when all power levels ω_x ($x \in \{U, D, i\}$) are replaced by the value one.

Now, if the random variable N_t is the number of cycles in a given length of time t and T_n is the time elapsed until the end of the n^{th} cycle from the beginning of that time period, then at

large n ,

$$\begin{aligned} p_n(t) &\stackrel{\text{def}}{=} \mathbb{P}(N_t = n) = \mathbb{P}(T_n \leq t) - \mathbb{P}(T_{n+1} \leq t) \\ &\simeq \Phi\left(\frac{t - nm}{\sqrt{nv}}\right) - \Phi\left(\frac{t - (n+1)m}{\sqrt{(n+1)v}}\right) \end{aligned}$$

Thus, the energy used up to time t has distribution function $F_t(x)$ with Laplace-Stieltjes transform $F_t^*(\theta)$ that can be approximated by (ignoring part-cycles, which are insignificant at large n) $F_t^*(\theta) = \sum_{n=1}^{\infty} p_n(C^*(\theta))^n$, where $C^*(\theta)$ is the LST of the distribution of the energy consumed in a single cycle, as in the previous section. At large n we can also apply the CLT to the energy used over n cycles, X_n , to obtain $C_n(x) = \mathbb{P}(X_n \leq x) \simeq \Phi\left(\frac{x - nm_e}{\sqrt{nv_e}}\right)$, where m_e, v_e are the mean and variance of the energy used in one cycle, simply obtained from $C^*(\theta)$ in the previous section. Thus we arrive at the approximation:

$$F_t(x) = \sum_{n=1}^{\infty} p_n(t) \Phi\left(\frac{x - nm_e}{\sqrt{nv_e}}\right) \quad (1)$$

Notice that when t is small and $[0, t]$ contains few cycles, p_n could be computed exactly (up to numerical approximations) by inverting the Laplace transform $T^*(\theta)^n - T^*(\theta)^{n+1}$ and using $C^*(\theta)$ directly. We would then also have to be concerned with part-cycles. However, as already remarked, such small time periods do not arise in the analysis that follows.

3 ENERGY-LATENCY MANAGEMENT

We seek a control system that sets the device power levels so as to provide the best performance possible, according to a given Quality-of-Service (QoS) metric, throughout the daylight hours, subject to the device remaining powered up through the night, i.e. giving 24-hour availability. We consider a finite horizon, discrete time Markov Decision Process (MDP), which samples the prevailing weather conditions every 15 minutes and sets the power-mode of the device so as to maximize a reward (actually, minimize a certain energy-latency metric) whilst achieving a given minimum battery charge level at the end of the daylight hours [2]. First, therefore, we need to model the charge and discharge rates of the battery under different conditions.

3.1 Battery charge and discharge rates

A brightness intensity function $i(t)$ gives the power harvested by the battery from the lumens of the sun (in watts) at time t (hours) of the day. Observation shows that this function is roughly parabolic. For example, the function $i(t) = \max[0, k_2(1 - k_1(13 - t)^2)]$, where k_1, k_2 are positive constants, gives a peak at 1pm, which is appropriate for the summer time. Setting $k_1 = 0.025$ and $k_2 = 500$ gives a peak output of 500 watts, which is typical for a one meter square PV panel, over a productive day running from about 7am to 7pm. This fits a sunny

climate and simply scaling down the parabolic output represents hazy or cloudy conditions well. We also use a more accurate intensity function obtained from empirical data in section 4.1.2. Changes in the weather may be modeled as a Markov chain, which can be adjusted before running the MDP according to the local weather forecast. However for simplicity, we consider a uniform day with approximately parabolic power input throughout the daylight hours.

Battery discharge rate is determined by the power used by the device, which is approximated by the probability distribution $F_t(x)$, neglecting leakage. We assume that the efficiency of the battery is 100%, so that no more energy is consumed than that required by the application. Obviously we can adapt this to cope with any observed lower efficiency if we take this as constant over output power and charge level. Moreover, we can extend the model to allow efficiency to be a function of the state. We use three battery charge-level states: f , or “full”, which means above a (high) threshold, set a little below 100% full; h , or “high”, which means below this high threshold but above a low threshold, set a little above 0%; and ℓ , or “low”, which means less than the low threshold. Similar to battery charge boundaries used in [7], our model also sets low and high thresholds to 25% and 75% respectively.

3.2 Markov Decision Process

An MDP consists of a finite set of states, a set of actions for each state, transition probabilities for each pair of states dependent on the action taken, an immediate reward for each transition, a goal and, possibly, a discount factor (e.g., to handle infinite time horizons). In our case, we consider the daylight hours of one day with a finite time-horizon, corresponding to the end of the day, with 9 states (3 weather-states \times 3 battery states). The state of the system is sampled every 15 minutes to yield a discrete state, discrete time Markov chain that drives the MDP. This Markov chain is defined by the state-transition probabilities $q_{(i_1, j_1), (i_2, j_2)}(t) = q_{i_1 i_2}^w(t) q_{j_1 j_2}^b(t)$, with $i_1, i_2 \in \{s, z, c\}$ (weather states sunny, hazy, or cloudy) and $j_1, j_2 \in \{f, h, \ell\}$ (battery charge states full, high, or low), where (omitting the argument t for brevity where there is no confusion) q^w, q^b are the state-transition probability matrices for the independent weather and battery charge level Markov chains, respectively. The former can be parameterized from the weather forecast, but in our initial model we assume uniform days so the states $\{z, c\}$ do not arise. Hence we only need to estimate the parameters of q^b , giving a three-state model. Clearly it would be straightforward to extend this model to incorporate transitions in the weather, at greater complexity but consequent loss of clarity.

Let $y_s(t) = \int_t^{t+1/4} i_s(u) du$ denote the energy harvested from the sun in the quarter-hour interval beginning at time t hours; in the untruncated parabolic area this would give $y(t) = -3.125t^2 + 81.25t - 403.125$ watt-hours. Let the “low”, “high” and “full” battery states have charge-levels in the ranges $[0, b_1]$, $[b_1, b_1 + b_2]$ and $[b_1 + b_2, b_1 + b_2 + b_3]$ respectively; i.e. the low band has width b_1 , the half-full band has width b_2

and the high band has width b_3 , so that level $b_1 + b_2 + b_3$ represents 100% full. We assume that at the beginning of a quarter hour time slot, the battery charge level random variable is uniformly distributed over the band with which its current state is associated. Then it is straightforward to obtain the parameters listed below, abbreviating the distribution function $F_{0,25}(\cdot)$ for the energy consumed by the device in a quarter-hour interval to $F(\cdot)$. For example, the probability that the state reduces from “full” to “high”, given current charge level $x + b_1 + b_2$ with $x \in [0, b_3]$, is the probability that the device consumes more than $y(t) + x$ watt-hours, but less than $y(t) + x + b_2$, which would result in a transition to state “low”. Since x is uniformly distributed over $[0, b_3]$, we get the first equation in the following list:

$$q_{fh}^b(t) = \frac{1}{b_3} \int_0^{b_3} [F(y(t) + x + b_2) - F(y(t) + x)] dx$$

$$q_{fl}^b(t) = 1 - \frac{1}{b_3} \int_0^{b_3} F(y(t) + x + b_2) dx$$

$$q_{hf}^b(t) = \frac{1}{b_2} \int_0^{b_2} F(y(t) - x) dx$$

$$q_{hl}^b(t) = 1 - \frac{1}{b_2} \int_0^{b_2} F(y(t) + x) dx$$

$$q_{lf}^b(t) = \frac{1}{b_1} \int_0^{b_1} F(y(t) - x - b_2) dx$$

$$q_{lh}^b(t) = \frac{1}{b_1} \int_0^{b_1} [F(y(t) - x) - F(y(t) - x - b_2)] dx$$

The diagonal entries in the matrix q^b are set so as to make the rows all sum to 1.

4 APPLICATION

The key recurrence, Proposition 1, for computing the LST of the distribution of the energy used during a powered-up cycle with fixed power settings has been implemented in Wolfram’s Mathematica, and this LST can be implemented numerically by any of several known algorithms. However, the CLT-based approximations are more applicable for our problem.

4.1 Model Parameters

4.1.1 Foot-traffic arrival rate. In practical applications the arrival rate often varies with the time of day. For our purposes, we use Google Maps data of popularity times for a particular location (Ghirradelli Square in San Francisco), from which we can create a function that gives the instantaneous arrival rate at any time of day using interpolation. We use 15 minute intervals and assume the system has many busy cycles during that interval and reaches steady-state quickly. This gives 96 arrival rates for the 24 hour period. Figure 1 shows Google’s “popularity function” over the 24-hour period of December 21, 2016. We normalize this function so that the average rate over 24 hours is one arrival per second, and then use a parameter to scale up the arrival rate as needed to account for batch arrivals. This way the time-dependent arrival rate is modeled by a single parameter λ , the batch-arrival rate of our model.

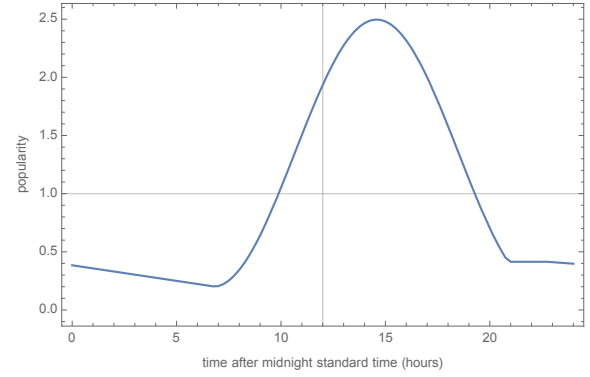


Figure 1: Foot traffic arrival rate at Ghirradelli Square over the 24 hours of the winter solstice, 2016

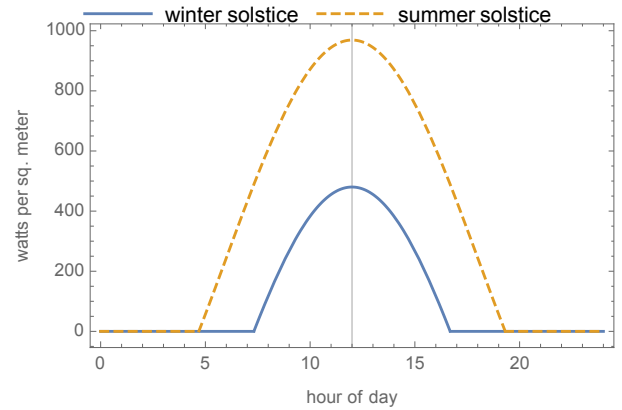


Figure 2: Expected solar intensity at Ghirradelli Square over the 24 hours of the winter and summer solstices

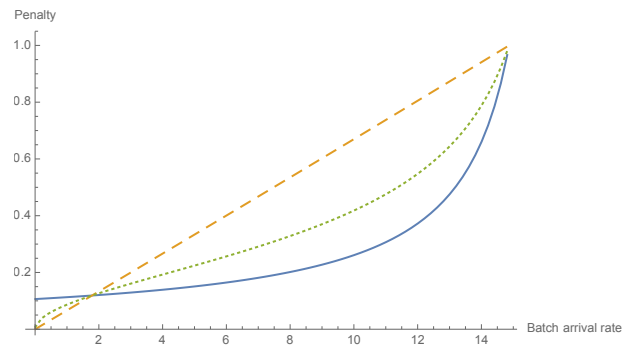


Figure 3: Latency, energy and hybrid penalty functions for the high power level, shown in solid blue, dashed orange and dotted green lines, respectively

4.1.2 Solar power harvesting. The amount of solar power harvested per square meter depends on many factors. We use a simple model that captures the parabola-like shape of the sun's intensity during a particular day – the winter solstice, December 21, which has solar declination angle of -23.5 degrees (due to the tilt of the earth). Ghirradelli Square at 37.8059 degrees North latitude is our chosen location. The solar constant is about 1000 W per square meter but typical solar cells are capable of converting only a fraction of this. Using the equation from [3] the sun insolation at time t hours on a 24 hour clock is the maximum of 0 and $1000(\cos(x) \cos(y) \cos(h(t - 12)) + \sin(x) \sin(y))$, where x is the latitude (in radians), y is the declination angle (in radians), h is the “hour angle” ($\pi/12$ in radians). Figure 2 shows a chart of the sun insolation equation for the winter solstice and the summer solstice at Ghirradelli Square. In the model we consider in section 4.3, we assume a quarter square meter solar panel.

4.1.3 Device service times. The device may be performing a variety of tasks and service times should be parameterized by fitting to the specific applications being run. Somewhat arbitrarily, we chose to parameterize all our models with respect to a storage system from which we have data; see [6]. Accordingly, service times have gamma distributions with means (9.808, 7.930, 6.053) milliseconds and coefficients of variation (0.64, 0.401, 0.218) for queue lengths of 1, 2 and 3 or more, respectively; i.e. the threshold of the EP-queue is 3 – so the service time distribution is the same for all queue lengths 3 and above. This way the service time is characterized by just 3 service time distributions, each with 2 parameters. Note that the mean service times and variability decrease as the queue depth increases, which is a nice property because it implies the system becomes more efficient with more load.

4.1.4 Penalty function. The goal of the MDP is to identify a policy that chooses power levels for the device over each quarter hour interval so as to minimize a certain aggregated penalty, whilst meeting a goal at the end of a specified finite period. We define the *penalty function* $R_{a,t}(s, s')$ to be a suitable energy-performance metric for the 15-minute slot starting at time t achieved by taking action a when in state s that leads to next state s' . This metric, which is state-dependent, could be the expectation of user-response time, r , (if energy is not a major concern), or the energy used, e , (if this is the only major concern), or of the product of the two if a trade-off is required, as in [6] for example. These metrics are normalized by their values, R and E , obtained at 90% utilization. Our *primary* metric is then the vector $(\frac{r}{R}, \sqrt{\frac{re}{RE}}, \frac{e}{E})$ corresponding to the battery states (f, h, ℓ) . Note we used the geometric mean of the first and last value to obtain the middle value. Graphs of these penalty functions are shown in Figure 3 for the model with base power level of 100 watts. To minimize the possibility of a temporarily flat battery and so loss of data, we can instead choose to deter the system from visiting the “low” state. We create this *detering* metric by setting the penalty in the “low” state to a fixed quantity equal to a multiple of the maximum

power at which the device is run – so that the normalized metric becomes that constant multiple.

4.2 MDP definition

For a given state, actions a_1 or a_2 set the device power to 50 or 100 watts, respectively. The power settings can be changed in the MDP by appropriate choice of action a_1, a_2 at the beginning of a time slot, based on the current state of the system and the predicted energy supply and drain going forward. Once a policy is chosen, the state transition probabilities are fixed and the process is Markovian because the next state only depends on the current state, and not earlier states. Even with binary actions for each state, the number of policies to enumerate is large. Our goal is to find the optimum *policy at the beginning of each quarter-hour time slot* $\pi_t(s) \in \{a_1, a_2\}$, over states s in the state-space $\mathcal{S} = \{s, z, c\} \times \{f, h, \ell\}$, that yields the minimum penalty over the remainder of a day such that the battery is left full (or above a specified charge level) at the end of the daylight hours: at time T hours, say.

This leads to the following conventional MDP specification: Determine the *value function* $V(s, t)$ at discrete time t and policy $\pi(s, t)$ for all $s \in \mathcal{S}$ given by the iteration $V_T(s) = 0$ if $s = f$, $V_T(s) = 1$ if $s \neq f$ and

$$V_t(s) = \min_{a \in \{a_1, a_2\}} \left[\sum_{s' \in \mathcal{S}} P_{a,t}(s, s') (R_{a,t}(s, s') + V_{t+1}(s')) \right],$$

$$\pi_t(s) = \arg \min_{a \in \{a_1, a_2\}} \left[\sum_{s' \in \mathcal{S}} P_{a,t}(s, s') (R_{a,t}(s, s') + V_{t+1}(s')) \right]$$

for $t = T - 1, T - 2, \dots, 0$. The terms $P_{a,t}(s, s')$ are the transition probabilities from state s to state s' corresponding to discrete times t and $t + 0.25$ hours, when action a_1 or a_2 is chosen, i.e. when the power mode is set to either ω_1 or ω_2 . They are computed as the matrix $q^b(t)$, parameterized with the power level defined by the actions a .

4.3 Numerical experiments

In order to reduce the number of parameters and simplify the results, we assume the weather is fixed at “sunny”. This reduces the unexpected variability in power generation from occasional cloudy and hazy states but we retain the variability of the sun intensity during the day. With this simplification we consider two model parameterizations:

- (1) The baseline, or “Parabolic” model, which uses the quadratic sun intensity function given in section 3.1 and assumes the arrival rate to be constant at 2 batches per quarter-hour time-slot, with mean batch size 10.
- (2) “Ghirradelli” model with arrival rate and sun intensity functions given in sections 4.1.1 and 4.1.2. Mean batch size is again 10.

Both models use the same gamma service time distributions, described in section 4.1.3. For each of these parameterizations, we use either the primary metric or the detering metric, which gives 4 scenarios for numerical comparison. For each scenario,

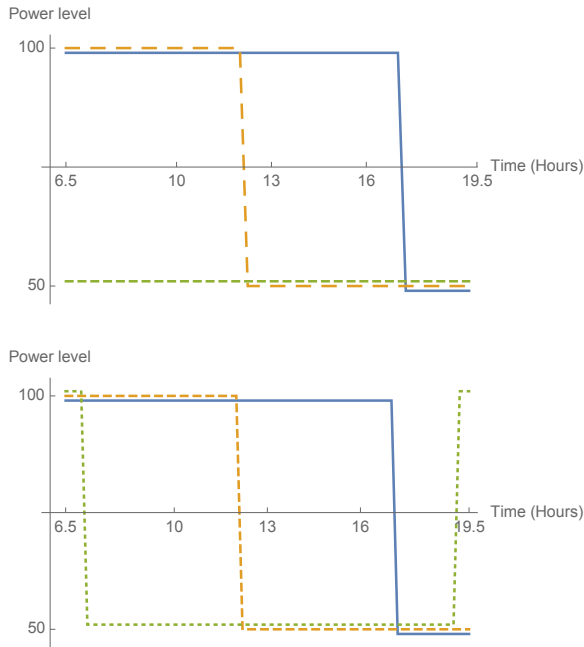


Figure 4: Policy choices made in the MDP for the Parabolic model with the primary (top) and deterring (bottom) metrics. The optimal power levels for states $\{\ell, h, f\}$ are shown in dotted green, dashed orange and solid blue, respectively, at each quarter-hour time point.

we want the MDP policy to select the next operating power level (50 or 100 watts) for each battery state in $\{\ell, h, f\}$ at every quarter-hour time point.

4.3.1 Parabolic model. Figure 4 shows the optimal policy under the primary and deterring (higher penalty in state ℓ) metrics, respectively. In the former case, we see how the high power mode is used in the full battery state for most of the day up to a little after 5pm. Similarly the high battery state can use maximum power until about midday. However, the low battery state ℓ cannot transit to a higher charge level due to insufficient net energy coming from the sun. At a lower arrival rate of work for the device, the sun would supply enough energy to make transition from the low state possible, especially near the middle of the day. Then the ultimate penalty of being in the low state at the end of the day might be avoided.

The situation is similar with the deterring metric under our model parameterization. Notice that the unexpected visits of the low state to the high power mode at the beginning and end of the day are artifacts of equal value functions for each choice of power level: we choose the maximum in the event of such a tie.

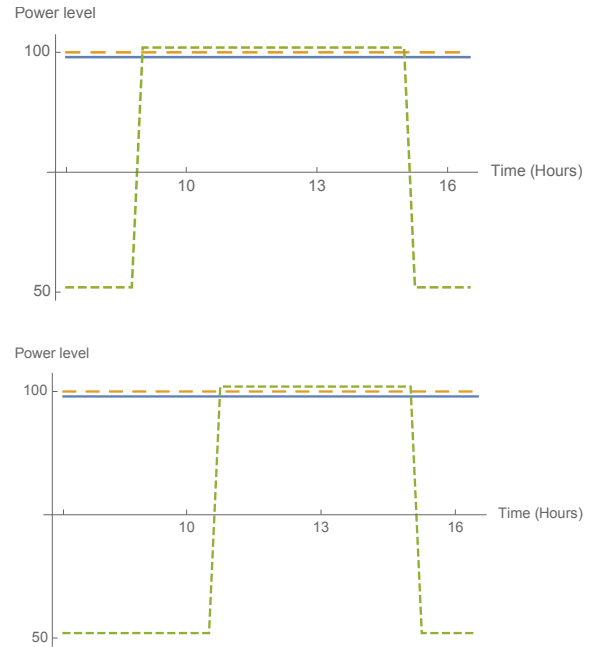


Figure 5: Policy choices made in the MDP for the Ghirradelli model with the primary (top) and deterring (bottom) metrics. The optimal power levels for states $\{\ell, h, f\}$ are shown in dotted green, dashed orange and solid blue, respectively, at each quarter-hour time point.

4.3.2 Ghirradelli model. Similarly to the parabolic case, Figure 5 shows the optimal policy under the primary and deterring (higher penalty in state ℓ) metrics, respectively. These charts are somewhat less interesting since the system is very underloaded for most of the day. However, as solar intensity increases during the day, the low battery state is able to use the high power level and is open to the possibility of transition to a higher charge state. Further experiments allowing for alternate parameterizations and other workloads are in progress.

5 RELATED WORK

The work described in this paper builds on previous research from modeling energy and performance in queues, optimizing energy-performance metrics, and applying MDP for specific domains. Much of the research in energy-performance has been for wireless sensor networks where energy is a scarce resource [1]. Typically, device power can be changed and the focus is about reducing energy consumption so that the batteries last longer. For a survey of energy efficient wireless communication see [9]. Also in the wireless sensor application context, [7] uses MDPs to analyze energy-constrained sensors that can be recharged, and suggests optimal policies for recharging sensors based on battery charge levels and delay

times. Energy-performance metrics are optimized for server farms in [4]. MDPs have been used extensively for many applications in other areas, including road maintenance, scheduling policies and reinforcement learning. The idea of adjusting a server's power, with nonlinear impact on the energy used, to increase service rate has been used in discrete-event simulation software to model interference or arbitrary blocking [5, 8]. However, we believe there have been few analytical models in the context of optimizing energy usage to meet some QoS requirement on performance.

6 CONCLUSION

In this paper we have explicitly incorporated energy consumption into the EP-queue model, facilitating a joint investigation of energy and performance. The numerical calculations involved are costly and we have introduced efficient approximations that allow the model's output to be used effectively in a real-time optimizer. This sets the power level at regular time points over the day to achieve optimum aggregated values of a chosen energy-performance metric such that a battery-charge level goal is met at the end of the day. By profiling real-world applications, the task service times at different queue lengths could be obtained accurately and used for finding optimal settings. We developed an MDP framework in a simple example with three sun states (which we reduced to one) and three battery charge levels. The two non-controllable factors are the amount of energy used by arrivals and the amount of energy supplied by the sun. As it stands, the model is highly simplified – in terms of the number of states, use of the power-up and down feature, small number of alternate power levels and efficiency of the implementation. The output of the model shows that the methodology is viable and that the best power setting can be identified and implemented in a timely fashion for dynamic optimization. A significant factor in model-design is always the particular metric that one wants to optimize. We considered two variants of an energy-performance metric – many more are possible, the “best” depending on *what* the application is and *who* is interested in optimizing it. Indeed it is often of most benefit to consider families of metrics; for example, in our case we might consider (in the notation of section 4.1.4) $\{(r^\alpha e^\beta)^{1/(\alpha+\beta)}\}$ for a range of α and β . The geometric mean that we used corresponds to $\alpha = \beta = 1$, and if response time or energy were considered the only relevant metric, we would have $\alpha = 1, \beta = 0$, respectively.

REFERENCES

- [1] P. Agrawal, A. Kumar, J. Kuri, M. K. Panda, V. Navda, R. Ramjee, and V. N. Padmanabhan. Analytical models for energy consumption in infrastructure wlan stas carrying tcp traffic. In *Proceedings of the 2Nd International Conference on COMmunication Systems and NETworks, COMSNETS'10*, pages 10–19, Piscataway, NJ, USA, 2010. IEEE Press.
- [2] Richard J. Boucherie and Nico van Dijk. *Markov Decision Processes in Practice*. International Series in Operations Research and Management Science. Springer International, 2017.
- [3] Ryan Ferguson. *How to Calculate Solar Insolation*, 2017 (accessed January 11, 2018).
- [4] Anshul Gandhi, Varun Gupta, Mor Harchol-Balter, and Michael A. Kozuch. Optimality analysis of energy-performance trade-off for server farm management. *Perform. Eval.*, 67(11):1155–1171, November 2010.
- [5] Anshul Gandhi, Varun Gupta, Mor Harchol-Balter, and Michael A. Kozuch. Optimality analysis of energy-performance trade-off for server farm management. *Perform. Eval.*, 67(11):1155–1171, November 2010.
- [6] Peter G. Harrison, Naresh M. Patel, and William J. Knottenbelt. Energy-performance trade-offs via the ep queue. *ACM Trans. Model. Perform. Eval. Comput. Syst.*, 1(2):6:1–6:31, June 2016.
- [7] Sudip Misra, Rashmi Ranjan Rout, T. Raghu Vamsi Krishna, Patel Manish Kumar Manilal, and Mohammad S. Obaidat. Markov decision process-based analysis of rechargeable nodes in wireless sensor networks. In *Proceedings of the 2010 Spring Simulation Multiconference, SpringSim '10*, pages 97:1–97:7, San Diego, CA, USA, 2010. Society for Computer Simulation International.
- [8] Priya Sehgal, Vasily Tarasov, and Erez Zadok. Optimizing energy and performance for server-class file system workloads. *Trans. Storage*, 6(3):10:1–10:31, September 2010.
- [9] Iñaki Ucar, Carlos Donato, Pablo Serrano, Andres Garcia-Saavedra, Arturo Azcorra, and Albert Banchs. Revisiting 802.11 rate adaptation from energy consumption's perspective. In *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, MSWiM '16*, pages 27–34, New York, NY, USA, 2016. ACM.