Lassila, Pasi; Aalto, Samuli

# On energy-aware M/G/1-LAS queue with batch arrivals

*Published in:*
Operations Research Letters

*DOI:*
10.1016/j.orl.2019.10.008

Published: 01/11/2019

*Document Version*
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

*Please cite the original version:*
Lassila, P., & Aalto, S. (2019). On energy-aware M/G/1-LAS queue with batch arrivals. *Operations Research Letters*, *47*(6), 611-617. https://doi.org/10.1016/j.orl.2019.10.008

# On energy-aware M/G/1-LAS queue with batch arrivals

Pasi Lassila[a,*], Samuli Aalto[a]

[a] *Department of Communications and Networking, Aalto University, Finland*

---

## Abstract

We analyze an energy-aware $\mathrm{M}^X$/G/1 queue under LAS scheduling with a setup delay and an idle timer that controls the delay before the server enters a sleep state. Through a classical busy period analysis, the expression for the mean conditional delay is derived, which generalizes the earlier well-known result for the ordinary M/G/1-LAS queue. We also analyze the performance-energy tradeoff and show that two well-known cost metrics, weighted sum and product of the mean delay and mean power, are minimized by setting the timer equal to zero or infinite, i.e., a finite idle timer is never used.

*Keywords:* $\mathrm{M}^X$/G/1 queue with setup delay, Least Attained Service, mean delay analysis, performance-energy tradeoff

---

## 1. Introduction

Energy-aware queueing models have been recently developed in order to study the performance-energy tradeoff inherent in modern data servers supporting sleep states. The considered queueing models are typically variations of the single-server M/G/1 queue with a setup delay, where the setup delay reflects the delay penalty from waking up a server once it has been put to sleep to save energy, see [3, 6, 7, 8, 10]. Also, multisever variants of the models have been studied, see [2, 3, 4, 11].

In a number of recent papers the single-server M/G/1 model with setup delays also includes an idle timer, which allows postponing the decision to go to sleep until after the timer expires. The model has been analyzed for different scheduling disciplines, including FIFO [7, 10], PS [6] and SRPT [8]. In this paper, we consider the same model but assume the Least Attained Service (LAS) scheduling discipline, sometimes also referred to as Foreground Background (FB), which always serves the job with the least amount of attained service.

The conditional mean delay of a job of size $s$ in an ordinary M/G/1 queue with LAS scheduling and Poisson arrivals has the well-known form, see, e.g., [9],

$$\mathrm{E}[T_{\mathrm{M/G/1-LAS}}(s)] = \frac{\lambda \mathrm{E}[X_s^2]}{2(1-\rho_s)^2} + \frac{s}{1-\rho_s}, \qquad (1)$$

where $\lambda$ is the arrival rate of the jobs, $X_s$ is the random variable for the service times $X$ truncated to $s$ and $\rho_s$ is the fraction of time the server is busy serving jobs with service times $X_s$.

As our main result, by applying ideas from [1], [8] and [9], we generalize (1) to the energy-aware $\mathrm{M}^X$/G/1 queue with LAS scheduling receiving batches that arrive according to a Poisson process with rate $\lambda_b$. As an important side result we obtain that for two common cost metrics characterizing the performance-energy tradeoff, the so-called ERWS (Energy Response time Weighted Sum) and ERP (Energy Response time Product), the metrics are minimized by setting $I = 0$ or $I = \infty$. This is the same result as has been already earlier proved for FIFO [7], PS [6] and SRPT [8], and gives further evidence that the result holds generally for any work conserving discipline.

The paper is organized as follows. Section 2 introduces the system model. The mean delay analysis is in Section 3 and numerical examples are in Section 4. Finally, Section 5 concludes the paper.

## 2. System model

An energy-aware server supporting processor sleep states can be modeled reasonably as a single server queue with appropriate energy-aware features. We consider a single server system under the following assumptions. New jobs arrive to the queue according to a batch Poisson process with arrival rate $\lambda_b$ and the batch size $\beta$ is an i.i.d random variable. The service time requirement of a job from the server is characterized by the i.i.d. continuous random variable $X$ with cumulative distribution function denoted by $F(t)$ and density $f(t)$. Thus, our model corresponds to the $\mathrm{M}^{\mathrm{X}}$/G/1 queue. The load of the queue $\rho$ is given by $\rho = \lambda_b \mathrm{E}[\beta]\mathrm{E}[X]$. The system is stable if $\rho < 1$.

The job processing and energy-aware controls operate as follows. When the server is *busy*, jobs are processed according to the LAS scheduling discipline giving always service to the job with the least attained service, and if there are multiple jobs with the same least amount of attained service, they are served according to the PS discipline. Upon completing the last job in a busy period, the server becomes *idle* and a timer $I$ is initiated with $I$ being an i.i.d. random variable with a general distribution. If a new batch of jobs arrives before the timer expires, the processor becomes immediately busy again and starts serving the jobs. If, on the other hand, the timer expires, the server enters the *sleep state*, where it can not anymore process jobs.

---

[*] Corresponding author, postal address: COMNET Department, Aalto University, P.O. Box 15400, FIN-00076 Aalto, Finland

*Email addresses:* `Pasi.Lassila@aalto.fi` (Pasi Lassila), `Samuli.Aalto@aalto.fi` (Samuli Aalto)

The duration of the sleep time is controlled by a parameter $k$, which measures the number of batches to arrive until the server is started again. As soon as $k$ batches have accumulated, the server is started and the server enters the *setup* state, but the service of the jobs will only start after a setup delay, which is denoted by $D$. The setup delay $D$ is an i.i.d random variable with a general distribution.

As described above, the server has clearly four states: busy, idle, sleep and setup. In each of these states the power consumption is denoted by $P_{\text{busy}}, P_{\text{idle}}, P_{\text{sleep}}$ and $P_{\text{setup}}$ with a natural ordering

$$P_{\text{busy}} \geq P_{\text{setup}} > P_{\text{idle}} > P_{\text{sleep}} \geq 0.$$

We denote by $\mathrm{E}[T]$ the overall mean delay of the jobs and by $\mathrm{E}[T(s)]$ the conditional mean delay of a job with size $s$. Similarly, the mean power consumption of the system is denoted by $\mathrm{E}[P]$. Note that $\mathrm{E}[P]$ is independent of the scheduling discipline, as long as it is work-conserving, and can be found in [8]. In our analysis, we will focus on deriving the mean conditional delay $\mathrm{E}[T(s)]$.

## 3. The energy-aware M$^X$/G/1-LAS queue

Consider the energy-aware M$^X$/G/1 queue as described in Section 2. We follow the classic approach and study the system at the arrival instant of a test job of size $s$. We also refer to this as a type-$s$ job for short.

A fundamental observation already made in [9] is that under LAS scheduling from the type-$s$ job point of view the system behaves as a priority queue: all jobs are served until they have attained service up to $s$, and if the original size of an arbitrary job exceeds $s$, after having attained service up to $s$ the job has no bearing on the test job with size $s$. Thus, from the point of view of the type-$s$ job the server only experiences a workload characterized by the *truncated service time* $X_s = \min\{X, s\}$ with the first two moments given by

$$\mathrm{E}[X_s] = \int_0^s t f(t)\, dt + s(1 - F(s)) \quad \text{and}$$
$$\mathrm{E}[X_s^2] = \int_0^s t^2 f(t)\, dt + s^2(1 - F(s)).$$

Thus, the fraction of time $\rho_s$ that the server is busy serving jobs with service times $X_s$ is clearly

$$\rho_s = \lambda_b \mathrm{E}[\beta]\mathrm{E}[X_s]. \tag{2}$$

The conditional mean delay of a type-$s$ job, $\mathrm{E}[T(s)]$, consists of two components, see also (1),

$$\mathrm{E}[T(s)] = \mathrm{E}[W(s)] + \mathrm{E}[R(s)], \tag{3}$$

where $\mathrm{E}[W(s)]$ is called the *mean conditional waiting time* and $\mathrm{E}[R(s)]$ is the *mean conditional residence time*. The mean conditional waiting time $\mathrm{E}[W(s)]$ is in the case of LAS scheduling defined as the time it takes for the server to serve all other jobs in the system up to $s$, and the mean conditional residence time $\mathrm{E}[R(s)]$ is the delay from serving the test job itself with size $s$. With the modified definition of $\rho_s$, see (2), the mean conditional residence time $\mathrm{E}[R(s)]$ remains same as in (1),

$$\mathrm{E}[R(s)] = \frac{s}{1 - \rho_s}. \tag{4}$$

However, the derivation of the mean conditional waiting time $\mathrm{E}[W(s)]$ requires a detailed busy period analysis.

### 3.1. Busy period description

We begin by introducing the structure of the busy period. A central role in our analysis is played by the notion of a *type-s busy period*, which is defined as a busy period during which all arriving jobs are served until their attained service reaches $s$ or they complete since their original service time requirement was less than $s$.

An illustration of the regenerative cycle is given in Figure 1. The regenerative cycle is defined such that it begins the moment that the idle timer expires and the server goes to sleep state. In the figure, this is marked as a cross on the left. The time that the server is in the sleep state is called *period 1.* It ends when $k$ batches have accumulated in the system, at which point the setup delay begins. In the figure, new arriving batches are indicated by arrows and we have $k = 2$. The time that the server is in the setup state is called *period 2*. After a random delay characterized by the random variable $D$, the server becomes busy and starts processing according to PS the jobs that accumulated during periods 1 (sleep) and 2 (setup), since none of the jobs has received service so far.

From the point of view of the test job with size $s$ that arrives at a random time instant, the first busy period that starts after period 2 begins with two sub-busy periods, where (i) the server is working on jobs until their attained service reaches $s$ and (ii) when the server is serving those jobs for which attained service $s$ was not enough. Of these, the first sub-busy period is a type-$s$ busy period that was started by all the jobs that accumulated during sleep and setup, and we refer to this as *period 3*. We denote by $B_3(s)$ the length of the associated type-$s$ busy period. Note that in a complete regeneration cycle, there is only one $B_3(s)$ busy period. After $B_3(s)$ is completed the server is still busy, but it is working on jobs for which reaching attained service $s$ was not enough. In the figure, we denote this time by $B_{>s}$. In our analysis, this period is referred to as *period 4*.

As shown in Figure 1, new arrivals during period 4 interrupt the on-going sub-busy period $B_{>s}$ and trigger a new type-$s$ busy period. However, now the type-$s$ busy period corresponds to a type-$s$ busy period in the ordinary M$^X$/G/1 queue since it is started by a single batch of arrivals. The length of one such type-$s$ busy period is denoted by $B_5(s)$. IF there are no new arrivals, after completing $B_5(s)$ and the following $B_{>s}$ the system becomes idle and the idle timer $I$ is sampled. Unless the timer expires, a new busy period begins with the arrival of a new batch, which initiates a new $B_5(s)$ busy period since the server is still in idle state. In the figure, we have one such additional busy period starting with a $B_5(s)$ busy period with two arrivals and ending with the subsequent $B_{>s}$. The idle timer $I$ is

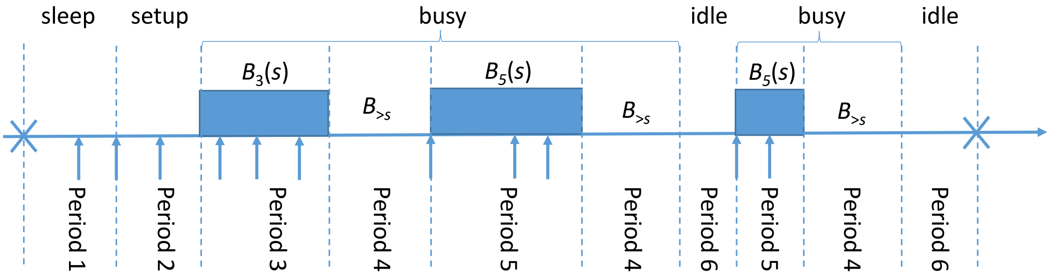

Figure 1: Illustration of the regenerative cycle.

sampled again, and in the figure the cycle completes when the idle timer expires before any new arrival, indicated by the cross on the right in Figure 1. As can be seen, during a regeneration cycle there can be several $B_5(s)$ type-$s$ busy periods and collectively they are referred to as *period 5*. The periods when the server is working on the remaining workload from the type-$s$ busy periods all together constitute the earlier introduced period 4. Similarly, jointly all idle periods during a regeneration cycle are referred to as *period 6*. Finally, independent of when the type-$s$ job arrives, there will be an associated type-$s$ busy period due to the other jobs that arrive at the same time in the same batch as the type-$s$ job.

*Remark*: Note that the regenerative cycle definition is identical with our earlier papers [7, 6, 8]. However, without any major modifications to the analysis the regeneration point could also be selected as the point where the queue empties.

### 3.2. *Busy period analysis*

The mean conditional waiting time can be expressed as

$$\mathrm{E}[W(s)] = \sum_{i=1}^{6} p_i \mathrm{E}[W_i(s)], \tag{5}$$

where $p_i$ is the probability that the type-$s$ job arrives during period $i$ and $\mathrm{E}[W_i(s)]$ is the mean waiting for an arrival during period $i$. Next we derive the probabilities that the type-$s$ job arrives during the different periods.

In general, the probabilities $p_i$ are given by the ratio of the mean number of arrivals during each period $i$ to the mean total number of arrivals in a cycle, denoted by $\mathrm{E}[N]$. The mean number of arrivals in a cycle is independent of the scheduling policy and is given by, see [6, 8],

$$\mathrm{E}[N] = \mathrm{E}[\beta]\frac{k + \lambda_b \mathrm{E}[D] + \lambda_b \mathrm{E}[I^{\mathrm{tot}}]}{1-\rho}, \tag{6}$$

where $\mathrm{E}[I^{\mathrm{tot}}]$ denotes the mean cumulative idle time during a cycle. The number of idle periods during a cycle obeys a geometric distribution with success probability $\mathrm{P}\{I < A\}$, where $A$ denotes a random variable for interarrival times. Thus, we have

$$\mathrm{E}[I^{\mathrm{tot}}] = \frac{\mathrm{E}[\min\{I, A\}]}{\mathrm{P}\{I < A\}}.$$

The mean number of jobs that arrive during period 1 (sleep) and period 2 (setup) equal $\mathrm{E}[\beta]k$ and $\mathrm{E}[\beta]\lambda_b\mathrm{E}[D]$, respectively. Thus, we have

$$p_1 = \frac{\mathrm{E}[\beta]k}{\mathrm{E}[N]}, \qquad p_2 = \frac{\mathrm{E}[\beta]\lambda_b\mathrm{E}[D]}{\mathrm{E}[N]}. \tag{7}$$

Periods 3 and 5 together correspond to the fraction of time that the server is processing jobs until their attained service reaches $s$, i.e., $p_3 + p_5 = \rho_s$, and also $p_4 + \rho_s = \rho$. Of these, $p_3$ depends on the mean length of the type-$s$ busy period $B_3(s)$, and can be expressed as

$$p_3 = \frac{\mathrm{E}[\beta]\lambda_b\mathrm{E}[B_3(s)]}{\mathrm{E}[N]}. \tag{8}$$

Thus, we get for $p_4$ and $p_5$ the following

$$p_4 = \rho - \rho_s, \qquad p_5 = \rho_s - \frac{\mathrm{E}[\beta]\lambda_b\mathrm{E}[B_3(s)]}{\mathrm{E}[N]}. \tag{9}$$

Finally, period 6 is the total idle period, for which we have

$$p_6 = \frac{\mathrm{E}[\beta]\lambda_b\mathrm{E}[I^{\mathrm{tot}}]}{\mathrm{E}[N]}. \tag{10}$$

In our analysis of the conditional waiting times $\mathrm{E}[W_i(s)]$ in (5), we need the first and second moments of the type-$s$ busy periods $B_3(s)$ and $B_5(s)$. They are analyzed next.

### 3.3. *Moments of $B_3(s)$ and $B_5(s)$*

We consider first $B_3(s)$, i.e., the type-$s$ busy period that starts immediately after the setup delay is over. Its first and second moments are given below.

**Proposition 1.** *The first two moments of $B_3(s)$ are given by*

$$\begin{aligned}
\mathrm{E}[B_3(s)] &= \frac{\mathrm{E}[S_0]}{1-\rho_s},\\
\mathrm{E}[B_3^2(s)] &= \frac{\mathrm{E}[S_0^2]}{(1-\rho_s)^2} + \lambda_b\mathrm{E}[S_0]\frac{\mathrm{E}[\beta](\mathrm{E}[X_s^2] + b\,\mathrm{E}[X_s]^2)}{(1-\rho_s)^3},
\end{aligned}$$

*where*

$$\begin{aligned}
\mathrm{E}[S_0] &= \mathrm{E}[\beta](k + \lambda_b\mathrm{E}[D])\mathrm{E}[X_s] \quad \textit{and}\\
\mathrm{E}[S_0^2] &= (k + \lambda_b\mathrm{E}[D])\mathrm{E}[\beta](\mathrm{E}[X_s^2] + b\,\mathrm{E}[X_s]^2)\\
&\quad + (\mathrm{E}[\beta]\mathrm{E}[X_s])^2(k(k-1) + 2k\lambda_b\mathrm{E}[D] + \lambda_b^2\mathrm{E}[D^2]).
\end{aligned}$$

*Proof.* The type-$s$ busy period $B_3(s)$ has been started by all arrivals during periods 1 (sleep) and 2 (setup) each having service time $X_s$. During the busy period new arrivals each with service times $X_s$ enter from batches at rate $\lambda_b$. The length of the type-$s$ period $B_3(s)$ can be characterized as

$$B_3(s) = S_0 + \sum_{n=1}^{N_3} B_{3,n},$$

where $S_0$ represents the total service time from all jobs in the batches that accumulated into the system during sleep and setup, $N_3$ is the number of new batches that arrived during $S_0$ and $B_{3,n}$ are type-$s$ sub-busy periods in an ordinary M/G/1 queue receiving jobs at rate $\lambda_b$ and service times

$$Y_s = \sum_{i=1}^{\beta} X_s,$$

i.e., the total workload from a batch. Thus, $B_3(s)$ corresponds to an M/G/1 queue with arrival rate $\lambda_b$ and service times $Y_s$ with an exceptional initial workload $S_0$, for which the first two moments are given by, see [12],

$$\begin{aligned} \mathrm{E}[B_3(s)] &= \frac{\mathrm{E}[S_0]}{1-\rho_s}, \\ \mathrm{E}[B_3^2(s)] &= \frac{\mathrm{E}[S_0^2]}{(1-\rho_s)^2} + \lambda_b \mathrm{E}[S_0]\frac{\mathrm{E}[Y_s^2]}{(1-\rho_s)^3}. \end{aligned}$$

The properties of $Y_s$ are considered first, and its first and second moments are given by

$$\begin{aligned} &\mathrm{E}[Y_s] = \mathrm{E}[\beta]\mathrm{E}[X_s] \quad \text{and} \\ &\mathrm{E}[Y_s^2] = \mathrm{E}[\beta]\mathrm{E}[X_s^2] + \mathrm{E}[X_s]^2(\mathrm{E}[\beta^2] - \mathrm{E}[\beta]). \end{aligned} \tag{11}$$

Then conditioning on the length of the setup delay $D$, the amount of work that accumulates during sleep and setup can be expressed as

$$S_0 \,|\, D = \sum_{i=1}^{k+N_D} Y_s,$$

where $N_D$ represents the number of batches that arrive during the given setup delay, which obeys $N_D \sim \mathrm{Poi}(\lambda_b D)$, i.e., the Poisson distribution with parameters $\lambda_b D$. Thus, we find, given $D$, the first two moments

$$\begin{aligned} &\mathrm{E}[S_0|D] = \mathrm{E}[k+N_D]\mathrm{E}[Y_s] \quad \text{and} \\ &\mathrm{E}[S_0^2|D] = \mathrm{E}[k+N_D]\mathrm{E}[Y_s^2] + \mathrm{E}[Y_s]^2(\mathrm{E}[(k+N_D)^2] - \mathrm{E}[k+N_D]). \end{aligned}$$

Utilizing (11) and unconditioning on $D$, we finally arrive at the result. ☐

Next we consider the type-$s$ busy period $B_5(s)$, which corresponds to normal type-$s$ busy periods that starts after $B_3(s)$ is over and the server has become idle. Recall that during a complete regeneration cycle there can be several $B_5(s)$ busy periods until the cycle ends when the idle timer expires. The moments of $B_5(s)$ are given below.

**Proposition 2.** *The first two moments of $B_5(s)$ are given by*

$$\begin{aligned} \mathrm{E}[B_5(s)] &= \frac{\mathrm{E}[\beta]\mathrm{E}[X_s]}{1-\rho_s}, \\ \mathrm{E}[B_5^2(s)] &= \frac{\mathrm{E}[\beta](\mathrm{E}[X_s^2] + b\,\mathrm{E}[X_s]^2)}{(1-\rho_s)^3}. \end{aligned}$$

*Proof.* The type-$s$ busy period $B_5(s)$ has been started by a batch of arrivals with service times $X_s$ and during the busy period new arrivals have the same properties. Thus, $B_5(s)$ can be characterized as a normal busy period in an M/G/1 queue with arrivals rate $\lambda_b$ and service times $Y_s$ with first two moments given by (11). Thus, by standard busy period results the first two moments of $B_5(s)$ are

$$\mathrm{E}[B_5(s)] = \frac{\mathrm{E}[Y_s]}{1-\rho_s} \quad \text{and} \quad \mathrm{E}[B_5^2(s)] = \frac{\mathrm{E}[Y_s^2]}{(1-\rho_s)^3},$$

from which the result follows. ☐

### 3.4. Conditional mean waiting times and final result

Now we begin the analysis of the components of the mean conditional waiting times in (5). Due to the batch arrival process, no matter when the type-$s$ job arrives, it will experience a waiting time due to other jobs that arrived in the same batch. The mean number of other jobs in the batch in addition to the type-$s$ job is denoted by $b$ and is given by, see, e.g., [1],

$$b = \frac{\mathrm{E}[\beta^2]}{\mathrm{E}[\beta]} - 1.$$

The mean conditional waiting time due to these other jobs in a batch is denoted by $\mathrm{E}[W^b(s)]$ and is given below.

**Proposition 3.** *The mean waiting time of a type-$s$ job due to other jobs in its batch is given by*

$$\mathrm{E}[W^b(s)] = \frac{b\,\mathrm{E}[X_s]}{1-\rho_s}. \tag{12}$$

*Proof.* Consider a modified M/G/1 queue with arrival rate $\lambda_b$ and service times $Y_s$ having $\mathrm{E}[Y_s] = \mathrm{E}[\beta]\mathrm{E}[X_s]$, i.e., total truncated service time requirement of a batch. The mean conditional waiting time $\mathrm{E}[W^b(s)]$ is the same as the mean length of a busy period of such a modified queue having an initial workload of size $Z_0$ with mean equal to $\mathrm{E}[Z_0] = b\mathrm{E}[X_s]$. Thus, we can express $\mathrm{E}[W^b(s)]$ as, see [12],

$$\mathrm{E}[W^b(s)] = \frac{\mathrm{E}[Z_0]}{1-\lambda_b \mathrm{E}[Y_s]} = \frac{b\,\mathrm{E}[X_s]}{1-\rho_s}.$$

☐

Next we derive the expressions for the mean conditional delays $\mathrm{E}[W_i(s)]$ in (5). If the type-$s$ job arrives during period 1 (sleep), it is one of the $k$ initial batches. Similarly as in [8], the type-$s$ job must wait until the end of period 1, the setup delay, then a modified $B_3(s)$ busy period, where the number of batches during sleep equals $k-1$, and finally the delay due to other jobs in its own batch. Thus, Corollary 1 in [8] and Proposition 1 give

$$\mathrm{E}[W_1(s)] = \frac{k-1}{2\lambda_b} + \mathrm{E}[D] + \frac{\mathrm{E}[\beta](k-1+\lambda_b \mathrm{E}[D])\mathrm{E}[X_s]}{1-\rho_s} + \mathrm{E}[W^b(s)]. \quad (13)$$

where the third term corresponds to the mean length of a $B_3(s)$ busy period with $k-1$ batches.

As analyzed in [8], if the type-$s$ job arrives during period 2 (setup), the job must wait until the end of the remaining setup delay with mean $\mathrm{E}[D^2]/(2\mathrm{E}[D])$, then a modified $B_3(s)$ busy period, where the mean setup delay equals $\mathrm{E}[D^2]/\mathrm{E}[D]$ (elapsed and remaining setup delay), and the delay due to other jobs in the same batch. Thus, Corollary 2 in [8] and Proposition 1 yield

$$\mathrm{E}[W_2(s)] = \frac{\mathrm{E}[D^2]}{2\mathrm{E}[D]} + \frac{\mathrm{E}[\beta](k+\lambda_b \mathrm{E}[D^2]/\mathrm{E}[D])\mathrm{E}[X_s]}{1-\rho_s} + \mathrm{E}[W^b(s)], \quad (14)$$

where the second term corresponds to the mean length of a $B_3(s)$ busy period with mean setup delay $\mathrm{E}[D^2]/(2\mathrm{E}[D])$.

If the type-$s$ job arrives during periods 3 or 5, the type-$s$ job needs to wait until the end of the on-going type-$s$ busy period, either $B_3(s)$ or $B_5(s)$, and the delay due to other jobs in its own batch. Thus, we have

$$\mathrm{E}[W_i(s)] = \frac{\mathrm{E}[B_i^2(s)]}{2\mathrm{E}[B_i(s)]} + \mathrm{E}[W^b(s)], i = \{3,5\}. \quad (15)$$

In (15), recall that the moments of $B_5(s)$ are the same given in Proposition 2.

Finally, if the type-$s$ job arrives during periods 4 or 6, a delay is only incurred due to the other jobs in the batch and thus

$$\mathrm{E}[W_4(s)] = \mathrm{E}[W_6(s)] = \mathrm{E}[W^b(s)]. \quad (16)$$

Now we have all the elements ready from the analysis and below we give the expression for the mean conditional delay $\mathrm{E}[T(s)]$ in the energy-aware $\mathrm{M}^X$/G/1-LAS queue.

**Theorem 1.** *For an energy-aware $M^X$/G/1-LAS queue, the mean conditional delay* $\mathrm{E}[T(s)]$ *is given by*

$$\mathrm{E}[T(s)] = \mathrm{E}[T_{M^X/G/1\text{-}LAS}(s)] + \frac{\mathrm{E}[\beta]}{\mathrm{E}[N]}\left(\frac{k(k-1)}{2\lambda_b} + k\mathrm{E}[D] + \frac{\lambda_b}{2}\mathrm{E}[D^2]\right)\frac{1}{(1-\rho_s)^2}, \quad (17)$$

*where* $\mathrm{E}[N]$ *is given by (6) and* $\mathrm{E}[T_{M^X/G/1\text{-}LAS}(s)]$ *refers to the conditional mean delay in the ordinary $M^X$/G/1-LAS queue given by*

$$\mathrm{E}[T_{M^X/G/1\text{-}LAS}(s)] = \frac{\mathrm{E}[\beta]\lambda_b\mathrm{E}[X_s^2]}{2(1-\rho_s)^2} + \frac{b\,\mathrm{E}[X_s](2-\rho_s)}{2(1-\rho_s)^2} + \frac{s}{1-\rho_s}. \quad (18)$$

*Proof.* By applying equations (7)-(10), (13)-(16) and Propositions 1-3 in the general expression for the mean conditional waiting time $\mathrm{E}[W(s)]$, see (5), we obtain after simplifications

$$\mathrm{E}[W(s)] = \frac{\mathrm{E}[\beta]\lambda_b\mathrm{E}[X_s^2]}{2(1-\rho_s)^2} + \frac{b\,\mathrm{E}[X_s](2-\rho_s)}{2(1-\rho_s)^2} + \frac{\mathrm{E}[\beta]}{\mathrm{E}[N]}\left(\frac{k(k-1)}{2\lambda_b} + k\mathrm{E}[D] + \frac{\lambda_b}{2}\mathrm{E}[D^2]\right)\frac{1}{(1-\rho_s)^2}.$$

Then by combining this with $\mathrm{E}[R(s)]$ from (4) in the general expression of $\mathrm{E}[T(s)]$ in (3), we obtain the final expression. In the expression, only term relating to the energy-aware features is the last term in $\mathrm{E}[W(s)]$ above, which vanishes as $\mathrm{E}[I^{\mathrm{tot}}] \to \infty$ resulting in the mean conditional delay of the ordinary $\mathrm{M}^X$/G/1-LAS queue with only idle and busy states given by (18). ☐

Apparently, the explicit expression of $\mathrm{E}[T_{\mathrm{M}^x/\mathrm{G}/1\text{-}\mathrm{LAS}}(s)]$ in Theorem 1 is not directly available easily in the literature, as far as we know. The $z$-transform of the distribution of the number of jobs in the ordinary $\mathrm{M}^X$/G/1-LAS queue can be found in [13], but it is in a somewhat implicit form and obtaining, e.g., the mean is not straight forward. The result can be elegantly derived directly also by observing that $\mathrm{E}[T_{\mathrm{M}^x/\mathrm{G}/1\text{-}\mathrm{LAS}}(s)]$ equals the work present in the system (from the Pollaczek–Khinchin formula), the work brought by the batch containing the size-$s$ job and the extra work that arrives during its sojourn time, i.e.,

$$\mathrm{E}[T_{\mathrm{M}^x/\mathrm{G}/1\text{-}\mathrm{LAS}}(s)] = \lambda_b\mathrm{E}[Y_s^2]/2(1-\rho_s) + (b\mathrm{E}[X_s] + s) + \lambda_b\mathrm{E}[T_{\mathrm{M}^x/\mathrm{G}/1\text{-}\mathrm{LAS}}(s)]\mathrm{E}[Y_s].$$

Applying (11) above gives the desired result.

Observe that when the mean total idle time goes to infinity, i.e., $\mathrm{E}[I^{\mathrm{tot}}] \to \infty$ following from selecting $I = \infty$, and setting $\mathrm{E}[\beta^2] = \mathrm{E}[\beta] = 1$, corresponding to the normal Poisson arrival process with $b = 0$, the expression for $\mathrm{E}[T(s)]$ in Theorem 1 yields the result of the ordinary M/G/1-LAS queue (1).

Finally, the decomposition of the delay $\mathrm{E}[T(s)]$ into the non-energy-aware component and an additional energy-aware cost term in (17) has a similar form as in the corresponding expression of the delay in the SRPT queue, see Theorem 4 in [8].

The overall mean delay $\mathrm{E}[T]$ is readily obtained from

$$\mathrm{E}[T] = \int_0^\infty \mathrm{E}[T(s)]f(s)\,ds. \quad (19)$$

*Exponential service times*: Assuming that the service times are exponential with $\mathrm{E}[X] = 1/\mu$, it can be verified from (17) and (19) that the mean delay $\mathrm{E}[T]$ is given by

$$\mathrm{E}[T] = \frac{1+\frac{b}{2}}{\mu(1-\rho)} + \frac{\frac{k(k-1)}{2\lambda_b} + k\mathrm{E}[D] + \frac{\lambda_b}{2}\mathrm{E}[D^2]}{k + \lambda_b\mathrm{E}[D] + \lambda_b\mathrm{E}[I^{\mathrm{tot}}]}. \quad (20)$$

The result (20) is identical with the corresponding expression for PS in [6] and FIFO in [5], which is intuitive due to the memoryless property of the exponential service times.

### *3.5. Application to performance-energy trade-off*

Next we consider the implications of Theorem 1 and the possibility of selecting the timer $I$ to optimize the performance energy-tradeoff of the system. Two popular metrics to characterize the tradeoff include ERWS and ERP. ERWS is the weighted sum of the mean delay $\mathrm{E}[T]$ and the mean power $\mathrm{E}[P]$ and ERP their product, i.e.,

$$\mathrm{ERWS} = w_1 \mathrm{E}[T] + w_2 \mathrm{E}[P] \quad \text{and} \quad \mathrm{ERP} = \mathrm{E}[T]\,\mathrm{E}[P],$$

where $w_1$ and $w_2$ are weights.

To optimize the performance-energy tradeoff, we have the following problem. The objective is to minimize the ERWS or ERP by appropriately selecting the idle timer $I$ distribution and its parameters. Below we state the result for this.

**Corollary 1.** *The optimal policy for selecting $I$ to minimize ERWS or ERP is to select $I = 0$ or $I = \infty$.*

*Proof.* In the ERWS and ERP cost metrics, the mean delay $\mathrm{E}[T]$ is obtained from Theorem 1 and by integration with respect to the service time distribution in (19). As mentioned earlier, the mean power $\mathrm{E}[P]$ is independent of the scheduling policy and is given in [8]. The form of the factor in the mean conditional delay $\mathrm{E}[T(s)]$, see Theorem 1, containing the total mean idle time $\mathrm{E}[I^{\mathrm{tot}}]$ has exactly the same form as in the corresponding energy-aware $\mathrm{M}^X$/G/1-SRPT queue, see Theorem 4 in [8]. Thus, by Proposition 7 in [8] the optimal selection for $I$ is deterministic with $I = 0$ or $I = \infty$. ☐

The result above shows that the optimum is always either to immediately switch off $I = 0$ or never to switch off $I = \infty$. The optimal idle timer control policy remains the same for LAS scheduling as has been already shown for FIFO in [7], PS in [6] and SRPT in [8]. Thus, Corollary 1 provides further evidence that the optimal timer selection is independent of the scheduling policy, at least for typical work conserving policies.

## 4. Numerical examples

Next we illustrate our results through numerical examples. The following power consumption values are used: $P_{\mathrm{busy}} = P_{\mathrm{setup}} = 200$ W, $P_{\mathrm{idle}} = 120$ W and $P_{\mathrm{sleep}} = 15$ W. The setup delay is deterministic with $D = 10$ s. These values reflect capabilities of modern servers and are also used in, e.g., [8]. The batch size is geometrically distributed with $\mathrm{E}[\beta] = 2$ and the mean service times are $\mathrm{E}[X] = 1$ s.

### *4.1. Mean delay with LAS compared with PS and FIFO*

We first consider the delay performance with idle timer $I = 0$ (server goes to sleep immediately when queue empties) and $k = 1$ (server wakes up when first batch arrives). The mean delay of LAS, see Theorem 1 and eq. (19), is compared against the corresponding performance under PS and FIFO, see [6] and [5] for the exact formulae. Figure 2 depicts the relative mean delay of LAS (blue curves), PS (green curves) and FIFO (red curves) for Pareto distributed sizes with shape parameter 2.5 (solid lines) and hyperexponential sizes with two phases $\mu_1 = 2$ and $\mu_2 = 0.2$ (dashed lines). For both distributions the remaining parameters have been fixed such that $\mathrm{E}[X] = 1$. The relative delay is taken as the ratio of the mean delay for the corresponding scheduling discipline and the given distribution to the mean delay with exponential service times. For the exponential service times all scheduling disciplines give the same result, shown as the horizontal line with black in Figure 2. FIFO clearly gives highest delays as the variability increases (note that in this case Pareto and hyperexponential end up having the same 2nd moments and hence there is only one solid red curve for FIFO). On the other hand, LAS performs still clearly better than PS also in this system with setup delays and batch arrivals.

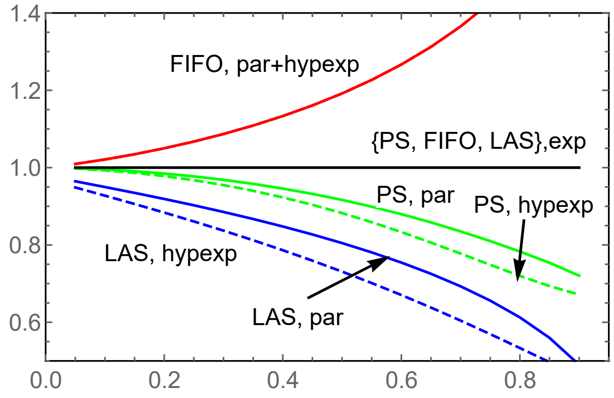

Figure 2: Relative mean delay as a function of the load for FIFO, PS and LAS disciplines with different service time distributions.

### *4.2. Performane-energy tradeoff*

Here we first illustrate the implications of Corollary 1 on the performance energy-tradeoff, which states that $I = 0$ or $I = \infty$ is the optimal timer value for the ERWS and ERP cost metrics, but a priori it is not known which one is the case.

We first consider the ERWS cost metric with $w_1 = w_2 = 1$. Figure 3 (upper panel) depicts the relative ERWS cost, defined as the ratio of the ERWS cost with $I = 0$ to the ERWS cost with $I = \infty$ corresponding to the ordinary $\mathrm{M}^X$/G/1-LAS queue, as a function of the load for the same distributions as earlier, i.e., exponential (red curve), hyper-exponential (green curve) and Pareto (blue curve). Note that the results for Pareto and hyper-exponential distributions are practically indistinguishable. As can be seen, at low loads selecting $I = 0$ is optimal, which is intuitive, and with the given parameters the situation changes roughly at load $\rho = 0.2$. Interestingly, the distribution has very little impact on this.

The relative ERP cost, defined analogously as for the ERWS cost, is given in Figure 3 (lower panel). In this case, we observe that the ERP cost is always higher in the energy-aware system with $I = 0$ in the considered load region than in the non-energy-aware system with $I = \infty$. This is due to the very long setup delay relative to the service times, in our case. If the setup delay could be reduced to, say $D = 1$ s, the ERP cost at lower loads would be lower with $I = 0$ than $I = \infty$. The sensitivity to the service time distribution appears to be somewhat higher for the ERP cost metric than for the ERWS cost metric.

The previous results highlighted the optimality result with respect to specific forms of the cost function, namely ERWS and ERP. However, the performance-energy tradeoff can also be analyzed by considering how the idle timer $I$ affects the mean power $\mathrm{E}[P]$ and mean delay $\mathrm{E}[T]$ separately. Figure 4 displays the mean delay $\mathrm{E}[T]$ as a function of the mean power $\mathrm{E}[P]$ as

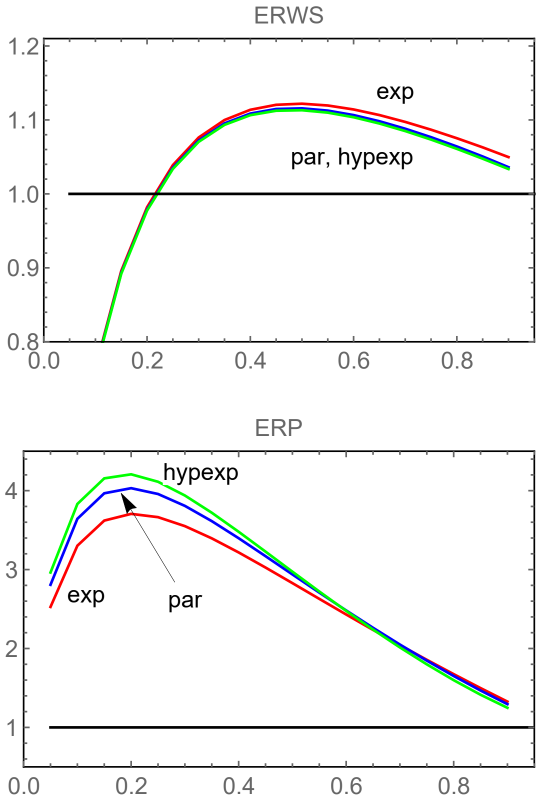

Figure 3: The relative ERWS (upper panel) and ERP (lower panel) costs under LAS scheduling with different distributions as a function of the load.

the idle timer value is varied from $I = 0$ to $I = \infty$ (ordinary M/G/1-LAS queue without setup delay) for four load values $\rho = \{0.1, 0.2, 0.4, 0.6\}$. In the figure, the results for exponentially distributed service times are shown with solid lines and the results with Pareto distributed service times are shown with dashed lines. As is expected, the mean delay E[$T$] is always lower with Pareto distributed service times for a given value of the mean power E[$P$] (recall that the mean power is insensitive to the service time distribution) and the difference is greater at higher loads. The main observation in Figure 4 is that at lower loads, see results for $\rho = \{0.1, 0.2\}$, we can observe a tradeoff between E[$T$] and E[$P$]: by increasing $I$, E[$T$] can be reduced at the expense of a higher E[$P$]. However, at higher loads, see results for $\rho = \{0.4, 0.6\}$, there is no tradeoff: both E[$T$] and E[$P$] can be simultaneously minimized by selecting $I = \infty$, i.e., the optimal idle time selection is to never switch off the server. With the used parameter values in this example, the load where the tradeoff disappears is between $\rho = [0.2, 0.4]$.

## 5. Conclusions

We analyzed an energy-aware M$^X$/G/1-LAS queue with a setup delay and idle timer, which is used to delay the server from entering the sleep state too quickly. The expression for the conditional mean delay of a test job with size $s$ was derived by applying classical busy period analysis. The expression interestingly has a convenient form consisting of the mean conditional delay in the ordinary M$^X$/G/1-LAS queue and an additive factor containing the effects of the setup delay and idle timer. As a corollary of our result, the idle timer control problem to minimize the ERWS and ERP cost functions had a simple solution: either the timer is set equal to zero or infinite.

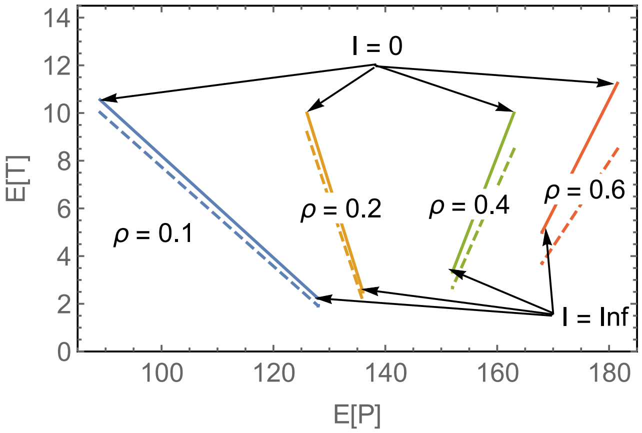

Figure 4: The mean power vs. mean delay plot as a function of $I$ for different values of the load $\rho$ for exponentially distributed service times (solid lines) and Pareto distributed service times (dashed lines).

The above optimality result on the idle timer has been previously observed also for FIFO, PS and SRPT scheduling. The optimality in these cases, including LAS in this paper, follows from the particular structure of the mean delay and mean power consumption for a given scheduling policy. It is plausible that the result holds generally for any work-conserving scheduling discipline. However, it remains as an elusive open problem how this could be formally proven without the precise assumption of the scheduling policy.

## Acknowledgements

We gratefully acknowledge the fruitful comments from Prof. Benny Van Houdt and the insightful remarks from the anonymous reviewers.

## References

[1] K. Avrachenkov, U. Ayesta, and P. Brown. Batch arrival processor-sharing with application to multi-level processor-sharing scheduling. *Queueing Systems*, 50(4):459–480, Aug 2005.

[2] A. Gandhi, S. Doroudi, M. Harchol-Balter, and A. Scheller-Wolf. Exact analysis of the M/M/K/Setup class of markov chains via recursive renewal reward. *Queueing Systems*, 77(2):177–209, June 2014.

[3] A. Gandhi, V. Gupta, M. Harchol-Balter, and M. A. Kozuch. Optimality analysis of energy-performance trade-off for server farm management. *Performance Evaluation*, 67(11):1155 – 1171, 2010.

[4] A. Gandhi, M. Harchol-Balter, and I. Adan. Server farms with setup costs. *Performance Evaluation*, 67(11):1123 – 1138, 2010.

[5] M. E. Gebrehiwot. Optimal control for energy-aware server farms. Doctoral Dissertation, Aalto University, 2018. available at `https://aaltodoc.aalto.fi/handle/123456789/34071`.

[6] M.E. Gebrehiwot, S. Aalto, and P. Lassila. Energy-performance trade-off for processor sharing queues with setup delay. *Operations Research Letters*, 44(1):101 – 106, 2016.

[7] M.E. Gebrehiwot, S. Aalto, and P. Lassila. Optimal energy-aware control policies for FIFO servers. *Performance Evaluation*, 103:41 – 59, 2016.

[8] M.E. Gebrehiwot, S. Aalto, and P. Lassila. Energy-aware SRPT server with batch arrivals: Analysis and optimization. *Performance Evaluation*, 115:92 – 107, 2017.

[9] L. Kleinrock. *Queueing Systems: Volume 2: Computer Applications*. John Wiley & Sons New York, 1976.

[10] V.J. Maccio and D.G. Down. On optimal policies for energy-aware servers. *Performance Evaluation*, 90:36 – 52, 2015.

[11] V.J. Maccio and D.G. Down. Structural properties and exact analysis of energy-aware multiserver queueing systems with setup times. *Performance Evaluation*, 121-122:48 – 66, 2018.

[12] L. Takács. *Introduction to the Theory of Queues*. Oxford University Press, 1962.

[13] S.F. Yashkov. Processor-sharing queues: Some progress in analysis. *Queueing Systems*, 2(1):1–17, Mar 1987.