# Analysis and Optimization of Sleeping Mode in WiMAX via Stochastic Decomposition Techniques

Amar Prakash Azad, Sara Alouf, Eitan Altman

## To cite this version:

Amar Prakash Azad, Sara Alouf, Eitan Altman. Analysis and Optimization of Sleeping Mode in WiMAX via Stochastic Decomposition Techniques. IEEE Journal on Selected Areas in Communications, 2011, 29 (8), pp.1630 - 1640. 10.1109/JSAC.2011.110912 . hal-01445314

HAL Id: hal-01445314
https://inria.hal.science/hal-01445314

Submitted on 24 Jan 2017

# Analysis and Optimization of Sleeping Mode in WiMAX via Stochastic Decomposition Techniques*

Amar Prakash Azad, Sara Alouf, Eitan Altman
INRIA Sophia Antipolis, B.P. 93, 06902, Sophia Antipolis Cedex, France
email: {amar.azad, Sara.alouf, eitan.altman}@sophia.inria.fr

**Abstract**

The paper establishes a general approach for analyzing queueing models with repeated inhomogeneous vacations. The server goes on for a vacation if the inactivity prolongs more than the vacation trigger duration. Once the system enters in vacation mode, it may continue for several consecutive vacations, possibly with a *different* probability distribution. We study a simple $M/G/1$ queue, which has the advantage of being tractable analytically. The theoretical model is applied to the problem of power saving for mobile devices in which the sleep durations of a device correspond to the server vacations. Various system performance metrics such as the frame response time and the economy of energy are derived. A constrained optimization problem is formulated to maximize the economy of energy in power save mode with QoS constraints. An illustration of the proposed methods is shown with a WiMAX system scenario to obtain design parameters for better performance. Our analysis allows us not only to optimize the system parameters for a given traffic intensity but also to propose parameters that provide the best performance under worst case conditions.

**Index Terms:** gain optimization, M/G/1 queue with repeated vacations, power save mode, system response time.

## 1 Introduction

Power save/sleep mode is the key point for energy efficient usage in recent mobile technologies such as WiFi, 3G, and WiMAX. Sleep mode operation enhances lifetime but it forces a trade off in terms of delay for various QoS services e.g. voice and video traffic. The mobility extension of WiMAX [1] is one of the most recent technologies whose sleep mode operation is discussed in detail and is being standardized.

The IEEE 802.16e standard [1] defines several types of power saving classes. Type I classes are recommended for connections of Best-Effort and Non-Real Time Variable Rate traffic. Under the sleep mode operation, sleep and listen windows are interleaved as long as there is no downlink traffic destined to the node. During listen windows, the node checks with the base station whether there is any buffered downlink traffic destined to it in which case it leaves the sleep mode. Each sleep window is twice the size of the previous one but it is not greater than a specified final value. A node may awaken in a sleep window if it has uplink traffic to transmit. Type II classes are recommended for connections of Unsolicited Grant Service and Real-Time Variable Rate traffic. All sleep windows are of the same size as the initial window. The operational parameters including the initial and maximum sleep window sizes can be negotiated between the mobile node and the base station.

The sleep mode operation of IEEE 802.16e, more specifically the type I power saving class, has received an increased attention recently. In [2], the base station queue is seen as an $M/GI/1/N$ queueing system with multiple vacations; an embedded Markov chain models the successive (increasing in size) sleep windows. Solving for the stationary distribution, the dropping probability and the mean waiting time of downlink packets are computed. Analytical models for evaluating the performance in terms of energy consumption and frame response time are proposed in [3, 4] and supported by simulation results. While [3] considers incoming traffic solely, both incoming and outgoing traffic are considered in [4]. In [5], the authors evaluate the performance of the type I power saving class of IEEE 802.16e in terms of packet delay and power consumption through the analysis of a semi-Markov chain.

In this paper, we propose a queueing-based model that is general enough to study many of the power save operations described in standards and in the literature. In particular, our model enables the characterization of the performance of type I and type II power saving classes as defined in the IEEE 802.16e standard [1]. The system composed of the base station, the wireless channel and the mobile node is modeled as an $M/G/1$ queue with repeated inhomogeneous vacations. Traffic

*This is an author version of the 11-page 2-column paper that has appeared in IEEE JSAC 9(8):1630–1640, Sept. 2011. The published paper lists only the first author. This has been corrected in IEEE JSAC 30(4):846, May 2012.

destined to the mobile node awaits in the base station as long as the node is in power save mode. When the node awakens, the awaiting requests start being served on a first-come-first-served basis. The service consists of the handling of a frame at the base station, its successful transmission over the wireless channel and its handling at the node. Analytical expressions for the distribution and/or the expectation of many performance metrics are derived yielding the expected frame transfer time and the expected gain in energy. We formulate an optimization problem so as to maximize the energy efficiency gain, constrained to meeting some QoS requirements. We illustrate the proposed optimization scheme through four application scenarios.

Although we have motivated our modeling framework using power saving operation in wireless technologies, it is useful whenever the system can be modeled by a server with repeated vacations. The structure of the idle period is general enough to accommodate a large variety of scenarios.

There has been a rich literature on queues with vacations, see e.g. the survey by Doshi( [6]). Our model resembles the one of server with repeated vacations: a server goes on vacation again and again until it finds the queue non-empty. To the best of our knowledge, however, all existing models assume that the vacations are identically distributed whereas our setting applies to inhomogeneous vacations and can accommodate the case when the duration of a vacation increases in the average if the queue is found empty.

Stochastic decomposition property of M/G/1 type queueing system with server vacation is one of the most remarkable results shown by [7]. We use this property to derive the main results. In [7], the stationary queue length distribution at a random point in time is decomposed in two or more parts where one part corresponds to stationary queue length distribution of M/G/1 system without vacation. This type of decomposition was first observed by [8], and subsequently by [9], [10], [11], [12]. Most of the references can be found in two excellent review articles by [6] and [13] in (1986).

Our model differs from the vacation model of [7] due to the presence of inhomogeneous repeated vacation and warm-up time and vacation trigger time. However, the decomposition property is still applicable to our model since it holds the required assumptions stated as in [14]. Stochastic decomposition property allows us to obtain various distribution through highly simplified approach which in turns yields more insight of the system. The rest of the paper is organized as follows. Section 2 describes our system model whose analysis is presented in Sect. 3. Our modeling framework is applied to the power saving mechanism in a WiMAX standard through four scenarios in Sect. 4. Section 5 formulates several performance and optimization problems whose results are shown and discussed in Sect. 6. Section 7 VII concludes the paper with some perspectives. Some of the proofs are omitted in this paper, are available at [15].

## 2 System Model and Notation

Consider an $M/G/1$ queue in which the server goes on vacation for a predefined period once the queue is observed empty for a vacation trigger duration. At the end of a vacation period, a new vacation initiates as long as no request awaits in the queue. We consider the exhaustive service regime, i.e., once the server has started serving customers, it continues to serve the queue until the queue empties. Request arrivals are assumed to form a Poisson process, denoted $N(t), t \geq 0$, with rate $\lambda$. Let $\sigma$ denote a generic random variable having the same (general) distribution as the queue service times.

Note that the queue size at the beginning of a busy period impacts the duration of this busy period and is itself impacted by the duration of the last vacation period. Because arrivals are Poisson (a non-negative Lévy input process would have been enough), the queue regenerates each time it empties and the cycles are i.i.d. Each regeneration cycle consists of: $(i)$ *Vacation Trigger* time; Failing any arrival during trigger time, denoted by $T_t$, activate the vacation mode. However vacation is deferred if there is an arrival during $T_t$ which mimics the standard $M/G/1$ queue. Time of first arrival, denoted by $t_f$, is the idle duration , if the vacation is not triggered; $(ii)$ an *idle* period; let $I$ denote a generic random variable having the same distribution as the queue idle periods, a generic idle period $I$ consists of $\zeta$ vacation periods denoted $V_1, \ldots, V_\zeta$ and vacation trigger time $T_t$; $(iii)$ a *warm-up* period; it is a fixed duration denoted $T_w$ during which the server is warming up to start serving requests. Note that $T_w$ is a part of system only when vacation mode is triggered, i.e., $T_{\tilde{w}} = T_w \mathbb{I}\{t_f > T_t\}$; $(iv)$ a *busy* period; let $B$ denote a generic random variable having the same distribution as the queue busy periods. The distribution of $V_i$ may depend on $i$, so the repeated vacations are *not* identically distributed. They are however assumed to be independent.

Let $X(t)$ denote the queue size at time $t$. It will be useful to define the following instants relatively to the beginning of a generic cycle (in other words, $t = 0$ at the beginning of the generic cycle): $(i)$ $\hat{V}_i$ refers to the end of the $i$th vacation period, for $i = 1, \ldots, \zeta$; observe that the idle period ends at $\hat{V}_\zeta$; we have $\hat{V}_i = \sum_{j=1}^{i} V_j$ and $I = \hat{V}_\zeta = \sum_{i=1}^{\zeta} V_i$; $(ii)$ $T_N$ refers to the beginning of the busy period $B$; we define $N := X(T_N)$ as the queue size at the beginning of a busy period; $(iii)$ $T_i$ refers to the first time the queue size *decreases* to the value $i$ (i.e. $X(T_i) = i$) for $i = N-1, \ldots, 0$; observe that the cycle ends at $T_0$.

The times $\{T_i\}_{i=N,N-1,\ldots,0}$ delimit $N$ subperiods in $B$, as can be seen in Fig. 1. We can write $B = \sum_{i=1}^{N} B_i$ where $B_i = T_{i-1} - T_i$. The random variable $N$ is in fact the number of arrivals from $t = 0$ until time $T_N$, even though all of the arrivals occur between $\hat{V}_{\zeta-1}$ and $T_N$. Introduce $N_I$ as the number of requests that have arrived up to time $\hat{V}_\zeta$ (i.e. during

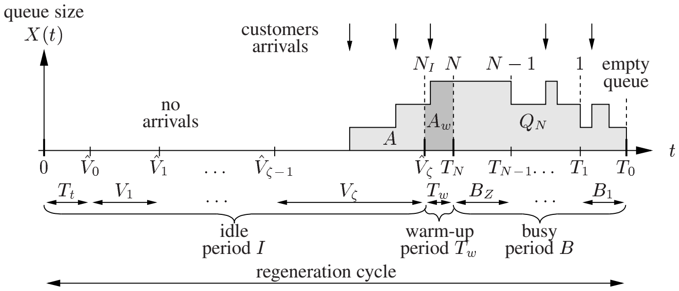


Figure 1: Sample trajectory of the queue size during a regeneration cycle.

period $I$) and $N_{T_w}$ as the number of arrivals during the warm-up period $T_w$. Hence $N = N_I + N_{T_w}$. Note that the queue size at the end of idle duration is $X(I) = N_I$.

# 3 Analysis

This section is devoted to analyze the queueing system presented in Sect. 2. We characterize the distributions of $\zeta$ and $N$, derive the expectations of $\zeta$, $I$, $N$, $B$ and $X(t)$ and the second moments of $I$ and $N$, and lastly compute the system response time. The gain from idling the server is introduced in the special case when the model is applied to study the power save operation in wireless technologies; see Sect. 4.

## 3.1 The Number of Vacations

To compute the distribution of $\zeta$, the number of vacation periods during an idle period, we first observe that the event $\zeta \geq i$ is equivalent to the event of no arrivals during $\hat{V}_{i-1} = \sum_{k=1}^{i-1} V_k$. Note that $\zeta \geq 1$ reflects that the first arrival occurred after the trigger wait time $T_t$. Equivalently, the event $\zeta = 0$ reflects that the atleast one arrival occurred during the vacation trigger time $T_t$. When $\zeta = 0$, let the arrival time of the first customer be denoted by $t_f$ conditioned that $t_f < T_t$.

Let $A_{T_t}$ denote the event of no arrival during $T_t$ and $L_{T_t}(\lambda)$ denotes $e^{\lambda T_t}$. Let $A_k$ denote the event of no arrivals during the period of time $V_k$, and let $A_k^c$ denote the complementary event. Denoting by $L_k(s) := \mathbb{E}[\exp(-sV_k)]$ and $L_{\widehat{i-1}}(s) := \mathbb{E}[\exp(-s\hat{V}_{i-1})]$ the Laplace Stieltjes transform (LST) of $V_k$ and $\hat{V}_{i-1}$ respectively, we can readily write

$$P(\zeta = 0) \quad = \quad 1 - L_{T_t}(\lambda), \text{ and } P(\zeta = 1) = L_{T_t}(\lambda)(1 - L_1(\lambda)), \tag{1}$$

and for $i > 1$, we have

$$P(\zeta = i) \quad = \quad L_{T_t}(\lambda)\left(\prod_{k=1}^{i-1} L_k(\lambda)\right)(1 - L_i(\lambda)), \tag{2}$$

$$P(\zeta \geq i) \quad = \quad L_{T_t}(\lambda)\prod_{k=1}^{i-1} L_k(\lambda) = L_{T_t}(\lambda)L_{\widehat{i-1}}(\lambda), \tag{3}$$

where we have used the fact that arrivals are Poisson with rate $\lambda$. The product $\prod_{k=a}^{b} L_k(\lambda)$ is defined as equal to 1 for any $b < a$. Let $\mathcal{L}_{T_t}(s) := \exp(-sT_t)$. Using (3), the expected number of vacations in an idle period is given by

$$\mathbb{E}[\zeta] = \sum_{i=1}^{\infty} P(\zeta \geq i) = L_{T_t}(\lambda)\sum_{i=1}^{\infty} L_{\widehat{i-1}}(\lambda). \tag{4}$$

## 3.2 The Idle Period

The system goes on vacation only if the inactivity duration is more than the trigger time $T_t$. Therefore, if the vacation is triggered the idleperiod is the sum of all the vacation durations including vacation trigger time. Otherwise, Idle period is only the duration of first arrival (idle period of standard $M/G/1$ queue). The idle period is thus given by

$$I = \min[T_t, t_f] + \sum_{i=1}^{\zeta} V_i 1\!\!1_{\{\zeta \neq 0\}}. \tag{5}$$

Using the equality $\sum_{i=1}^{\zeta} V_i = \sum_{i=1}^{\infty} V_i \mathbb{1}_{\{\zeta \geq i\}}$, the expected idle period is

$$\begin{aligned} \mathbb{E}[I] &= \mathbb{E}[\min(T_t, t_f)] + \mathbb{E}\left[\left(\sum_{i=1}^{\infty} V_i \mathbb{1}_{\{\zeta \geq i\}}\right) \mathbb{1}_{\{\zeta \neq 0\}}\right] \\ &= \frac{1}{\lambda} L_{T_t}^c(\lambda) + T_t L_{T_t}(\lambda) + L_{T_t}(\lambda) \sum_{i=1}^{\infty} \mathbb{E}[V_i] \mathcal{L}_{\widehat{i-1}}(\lambda) \end{aligned} \tag{6}$$

## 3.3 The Initial Queue Size Distribution in Busy Period

The number of requests/packets waiting in the queue at the beginning of busy period is $N = N_I + N_{T_{\tilde{w}}}$, where $T_{\tilde{w}} = T_w(\mathbb{1}\{(t_f > T_t)\})$. The indicator function simply indicates that the warmup comes in picture only when sleep mode is triggered. Since the number of arrival during the idle period is independent of the arrival during the warmup period $T_w$, the p.g.f. of $N$ is the product of p.g.f. of $N_I$ and $N_{T_w}$. Therefore, we have the p.g.f. of $N$ given as

$$N(z) = N_I(z)\, N_{T_{\tilde{w}}}(z). \tag{7}$$

The queue size p.g.f. during the idle period $N_I(z)$ is given by

$$N_I(z) = \sum_{i=0}^{\infty} z^i\, \mathbb{P}(N_I = i) = \sum_{i=1}^{\infty} z^i\, \mathbb{P}(N_I = i). \tag{8}$$

The last equality ensures at least one arrival is sure during idle period, $\mathbb{P}(N_I = 0) = 0$. The number of arrivals during the idle period is the sum of arrivals during each vacation periods $V_i$'s. To derive the distribution of $N_I$, we first compute the joint distribution $(N_I, \zeta)$. Note, that $N_I$ takes value in $\mathbb{N}^*$. We have

$$P(N_I = j, \zeta = i) \quad = \quad \mathbb{E}\left[\exp(-\lambda V_i) \frac{(\lambda V_i)^j}{j!}\right] \prod_{k=1}^{i-1} L_k(\lambda) L_{T_t}(\lambda),$$

and,

$$P(N_I = j, \zeta = 0) = \begin{cases} 1 - L_{T_t}(\lambda), & \text{if } j = 1 \\ 0, & \text{otherwise.} \end{cases}$$

The second term depicts when vacation is not triggered (case of an arrival before the vacation trigger time $T_t$), which is nothing but a standard M/G/1 queue without vacation.

Let $L_{T_t}^c(\lambda)) = (1 - L_{T_t}(\lambda))$. The $z$-transform of initial queue size $Z(.)$ using Eq. (8) can be expressed as

$$N_I(z) \quad = \quad L_{T_t}^c(\lambda) z + \sum_{i=1}^{\infty} \mathcal{L}_i(\lambda(1-z)) \mathcal{L}_{\widehat{i-1}}(\lambda) L_{T_t}(\lambda).$$

Since the arrival is a poisson process, the p.g.f. of arrival during the fixed warm up period $T_w$ is given as

$$N_{T_{\tilde{w}}}(z) \quad = \quad \mathcal{L}_{T_t}(\lambda) \mathcal{L}_{T_w}(\lambda(1-z)) + \mathcal{L}_{T_t}^c(\lambda), \tag{9}$$

where the Laplace transform of the arrivals during the warm up period $T_w$ is given as $N_{T_w}(z) = e^{-\lambda T_w(1-z)} := \mathcal{L}_{T_w}(\lambda(1-z))$. The above equations combine to yield $N(z) = N_I(z) N_{T_{\tilde{w}}}(z)$, given by

$$N(z) = \left( z L_{T_t}^c(\lambda) + \sum_{i=1}^{\infty} \mathcal{L}_i(\lambda(1-z)) \mathcal{L}_{\widehat{i-1}}(\lambda) L_{T_t}(\lambda) \right) \left( \mathcal{L}_{T_t}(\lambda) \mathcal{L}_{T_w}(\lambda(1-z)) + \mathcal{L}_{T_t}^c(\lambda) \right). \tag{10}$$

Observe that $T_t = 0$ corresponds to a forced vacation scenario (the model presented in [16]), while $T_t = \infty$ corresponds to the simple M/G/1 queue.

Noting that, $z$ transform is one of well known tool to obtain moments by using the relation $N^{(n)}(1) = \mathbb{E}[N(N-1)\ldots(N-i+1)]$, which simply means the evaluation of $n$th derivative of $N$, denoted as $N^{(n)}(,)$, at $z = 1$, we derive the first, second and third derivatives of $N(z)$ (which will be required in latter sections). The first derivative of $N(z)$ is given by

$$N^{(1)}(z) \quad = \quad N_I(z) N_{T_{\tilde{w}}}^{(1)}(z) + N_I^{(1)}(z) N_{T_{\tilde{w}}}(z). \tag{11}$$

where

$$\begin{aligned} N^{(1)}_{T_{\tilde{w}}}(z) &= \mathcal{L}_{T_t}(\lambda)\lambda T_w \mathcal{L}_{T_w}(\lambda(1-z)), && (12)\\ N^{(1)}_I(z) &= L^c_{T_t}(\lambda) + \sum_{i=1}^{\infty} \frac{d\mathcal{L}_i(\lambda(1-z))}{dz} \mathcal{L}_{\widehat{i-1}}(\lambda) L_{T_t}(\lambda). && (13) \end{aligned}$$

From the definition of $z$ transform we know that $N_{T_{\tilde{w}}}(1) = N_I(z) = 1$, thus $N^{(1)}_{T_{\tilde{w}}}(1) = \lambda T_w L_{T_t}(\lambda)$. Further substituting Eq. (6) in $N^{(1)}_I(.)$, we can obtain

$$\mathbb{E}[N_I] = \lambda \left[\mathbb{E}[I] - T_t L_{T_t}(\lambda)\right] = \lambda \mathbb{E}[\widetilde{I}] \tag{14}$$

where $\mathbb{E}[\widetilde{I}] := \mathbb{E}[I] - T_t \mathcal{L}_{T_t}(\lambda)$. Combining all together we obtain the first moment,

$$\mathbb{E}[N] = N^{(1)}(1) = \lambda T_w \mathcal{L}_{T_t}(\lambda) + \lambda \mathbb{E}[\widetilde{I}]. \tag{15}$$

Similarly for the second moment, from (10) we have

$$N^{(2)}(z) = N_I(z) N^{(2)}_{T_{\tilde{w}}}(z) + N_{T_{\tilde{w}}}(z) N^{(2)}_I(z) + 2N^{(1)}_{T_{\tilde{w}}}(z) N^{(1)}_I(z), \tag{16}$$

where, using eq. (13) and eq. (12), we have

$$N^{(2)}_I(z) = \sum_{i=1}^{\infty} \frac{d^2\mathcal{L}_i(\lambda(1-z))}{dz^2} \mathcal{L}_{\widehat{i-1}}(\lambda) \mathcal{L}_{T_t}(\lambda),$$

$$N^{(2)}_{T_{\tilde{w}}}(z) = (\lambda T_w)^2 \mathcal{L}_{T_t}(\lambda) \mathcal{L}_{T_w}(\lambda(1-z)).$$

Evaluating at $z = 1$ and doing some simple calculus, we obtain

$$N^{(2)}(1) = \lambda^2 \left( T_w^2 \mathcal{L}_{T_t}(\lambda) + 2T_w \mathcal{L}_{T_t}(\lambda) \mathbb{E}[\widetilde{I}] + \mathbb{E}[I_a] \right). \tag{17}$$

Therefore the second moment is given by

$$\mathbb{E}[N^2] = \mathbb{E}[N] + N^{(2)}(1) = \lambda(\mathcal{L}_{T_t}(\lambda) T_w + \mathbb{E}[\widetilde{I}]) + \lambda^2(\mathcal{L}_{T_t}(\lambda) T_w^2 + 2T_w \mathcal{L}_{T_t}(\lambda) \mathbb{E}[\widetilde{I}] + \mathbb{E}[I_a]). \tag{18}$$

Note that the expected initial queue size $\mathbb{E}[N]$ and its second moment $\mathbb{E}[N^2]$ obtained in [16, Eqs. 11 and 12] are a special case of Eqs. (15) and (17) when $T_t = 0$. Note that, setting the parameter $T_t = 0$ essentially means that the vacation must be triggered whenever the system is idle, which is a particular case.

## 3.4 Queue Size Distribution

Queue size is given by the number of requests/packets in the queue seen by any random arriving packet. From "Poisson Arrival See Time Average" (PASTA) ([17]) the stationary queue size is equivalent to the number of requests waiting in the queue to be served left behind by a random departure. Further more, from [18], it is equivalent to the queue size using the workload decomposition or stochastic decomposition approach (refer to [7]. The p.g.f. of the stationary distribution of the number of request/packets left behind by a random departing customer is given by

$$X(z) = \frac{1 - N(z)}{\mathbb{E}[N](1-z)} X_{M/G/1}(z). \tag{19}$$

where $Z(.)$ denotes the p.g.f. of queue length at the beginning of busy period, $X_{M/G/1}(.)$ denotes the p.g.f. of the number of customers left behind in a standard $M/G/1$ queue. In stationary regime, the distribution of queue length can also be given by $X(z)$. From [19, p. 210], the p.g.f. of a standard $M/G/1$ queue is given by

$$X_{M/G/1}(z) = \frac{(1-\rho)(1-z)\sigma(\lambda - \lambda z)}{\sigma(\lambda - \lambda z) - z}, \tag{20}$$

where $\sigma(.)$ is the service time distribution.

### 3.4.1 Expected queue length

The moments of queue length can be directly obtained from $X(z)$ from its derivatives at $z = 1$. We double differentiate Eq. (19) w.r.t. $z$ and do some simple calculus to obtain the expectation of $X(.)$ which is given by

$$\begin{aligned}&\mathbb{E}[N]\left((1-z)X^{(2)}(z) - 2X^{(1)}(z)\right)\\&= (1-N(z))X^{(2)}_{M/G/1}(z) - 2N^{(1)}(z)X^{(1)}_{M/G/1}(z) - N^{(2)}(z)X_{M/G/1}(z).\end{aligned} \tag{21}$$

Using the relations $[1 - Z(1)] = 0$ (from eq. (19)), we obtain

$$\mathbb{E}[X] = X^{(1)}(1) = \frac{N^{(2)}(1)}{2\mathbb{E}[N]} + \mathbb{E}[X_{M/G/1}]\,. \tag{22}$$

Substituting $N^{(2)}(1)$ from (17) and $X^{(1)}_{M/G/1}(1) = \mathbb{E}[X_{M/G/1}] = \rho + \frac{\lambda^2\mathbb{E}[\sigma^2]}{2(1-\rho)}$ from [18, Sec. 5.6] (Pollaczek-Khinchin mean value formula in Eq. (21), we finally obtain the expected queue length

$$\mathbb{E}[X] = \frac{\lambda(T_w^2\mathcal{L}_{T_t}(\lambda) + 2T_w\mathcal{L}_{T_t}(\lambda)\mathbb{E}[\tilde{I}] + \mathbb{E}[I_a])}{2(T_w\mathcal{L}_{T_t}(\lambda) + \mathbb{E}[\tilde{I}])} + \left(\rho + \frac{\lambda^2\mathbb{E}[\sigma^2]}{2(1-\rho)}\right). \tag{23}$$

## 3.5 Busy period

The length of the busy period, denoted by $B$, depends on the number of customers/packets are waiting at the end of the vacation interval. If there are $Z$ number of packets requests are waiting, the subsequent busy period will consists of $Z$ independent busy periods, each of which is denoted by $B_1$ which mimics the single request service time as in $M/G/1$ queue. Therefore, we have

$$\mathcal{B}^*(s) = \sum_{i=1}^{\infty} N_I[\mathcal{B}_1^*(s)]^i = N[\mathcal{B}_1^*(s)]$$

Thus the expected busy period can be given by

$$\mathbb{E}[\mathcal{B}] = \mathbb{E}[N]\mathbb{E}[\mathcal{B}_1] = \frac{\mathbb{E}[N]\mathbb{E}[\sigma]}{1-\rho} \tag{24}$$

## 3.6 Sojourn time

Assume the waiting time of a customer is independent of the part of the arrival process that occurs after the customer's arrival epoch, which is easy to show. Our policy, which is FCFS discipline, falls in this category. The waiting time of an arbitrary customer in a queue is exactly the number of customer ahead of the tagged customer in the queue under FCFS scheme. The number of customers left behind by the tagged customers is precisely the number of arrivals during the sojourn time (waiting+service) of the tagged customers, denoted its pgf by $N(z)$. Since Poisson arrival see time average (PASTA), the p.g.f. of the number of cutomers ahead of a random customer has the same p.g.f. as $N(z)$. Therefore we can express the Laplace Stieltjes transform of the waiting time $W^*(s)$ of a random customer in the queue as (from [7])

$$W^*(s) \;=\; \frac{\lambda[1 - N(1 - s/\lambda)]}{s\mathbb{E}[N]}W^*_{M/G/1}(s), \tag{25}$$

where $W^*_{M/G/1}(s)$ is the LST of waiting time of an arbitrary request in the queue of a standard $M/G/1$ queue. From [20, 1.45], we have $W^*_{M/G/1}(s) = \frac{s(1-\rho)}{s-\lambda+\lambda\sigma^*(s)}$, where LST of service time is given by $\sigma^*(s) = \mathbb{E}[e^{-s\sigma}]$. Thus, we have

$$W^*(s) = \frac{\lambda[1 - N(1 - s/\lambda)]}{s\mathbb{E}[N]}\frac{s(1-\rho)}{s-\lambda+\lambda\sigma^*(s)} = K\frac{[1 - N(1 - s/\lambda)]}{s-\lambda+\lambda\sigma^*(s)}. \tag{26}$$

where $K = \frac{(1-\rho)\lambda}{\mathbb{E}[N]}$. The moments of the $W(.)$ can be obtained from its LST by simply evaluating its derivatives at $s = 0$, i.e., $E[W^{*n}] = (-1)^n W^{*(n)}(0)$ as follows,

$$\mathbb{E}[W] = \frac{N^{(2)}(1)}{2\lambda\mathbb{E}[N]} + \frac{\lambda\mathbb{E}[\sigma^2]}{2(1-\rho)}\,. \tag{27}$$

### 3.7 Message Response Time

The *message response time* $T$ is defined as the time interval from the arrival time of an arbitrary message to the time when it leaves the system after the service completion. The mean message response time is said to be the *single most important performance measure without blocking* [18, p. 162].

The response time of a message consists of the *waiting time* $W$ and the service time $\sigma$. Since the waiting time and service times are independent, we can express the LST of response time $T$ and mean waiting time directly as follows

$$T^*(s) = W^*(s)\sigma^*(s) \Rightarrow \mathbb{E}[T] = \mathbb{E}[W] + \mathbb{E}[\sigma]. \tag{28}$$

Thus, we can express

$$\mathbb{E}[T] = \frac{N^{(2)}(1)}{2\lambda\mathbb{E}[N]} + \frac{\lambda\mathbb{E}\sigma^2}{2(1-\rho)} + \mathbb{E}[\sigma] = \frac{T_w^2\mathcal{L}_{T_t}(\lambda) + 2T_w\mathcal{L}_{T_t}(\lambda)\mathbb{E}[\tilde{I}] + \mathbb{E}[I_a])}{2(T_w\mathcal{L}_{T_t}(\lambda) + \mathbb{E}[\tilde{I}]} + \left(\frac{\lambda\mathbb{E}[\sigma^2]}{2(1-\rho)} + \mathbb{E}[\sigma]\right). \tag{29}$$

The last term in bracket above is the mean response time of standard $M/G/1$ queue, denoted by $\mathbb{E}[T_{M/G/1}]$, (refer [20]), while the first part is the additional contribution due to vacation (which includes warm up and trigger time). Therefore, the expected sojourn time can be rewritten

$$\mathbb{E}[T] \quad = \quad \frac{T_w^2\mathcal{L}_{T_t}(\lambda) + 2T_w\mathcal{L}_{T_t}(\lambda)\mathbb{E}[\tilde{I}] + \mathbb{E}[I_a]}{2(T_w\mathcal{L}_{T_t}(\lambda) + \mathbb{E}[\tilde{I}])} + \mathbb{E}[T_{M/G/1}] \tag{30}$$

**Remark 1** *One can also obtain the expected time a customer spends in the queue using Little's formula as follows,* $\mathbb{E}[T] = \frac{\mathbb{E}[X]}{\lambda}$. *Substituting* $\mathbb{E}[X]$ *in* (23) *yields* (30) *back.*

As the rate $\lambda \to 1/\mathbb{E}[\sigma]$ (recall that the stability condition enforces that $\lambda\mathbb{E}[\sigma] < 1$), we must have $P(\zeta = 1) \to 1$ (thus $L_1(\lambda) \to 0$) whatever the distribution of the vacations. There will then be only one vacation period in most idle periods. Therefore, at large input rates, the largest contribution to the sojourn time is expected to come from the waiting time when the server is active (queueing delays).

## 4 Application to Power Saving

The model analyzed in Sect. 3 can be used to study energy saving schemes used in wireless technologies. Consider the system composed of the base station, the wireless channel and the mobile node. When the energy saving mechanism is disabled, the system can be seen as an $M/G/1$ queue; and when it is enabled, the system can be modeled as an $M/G/1$ queue with vacations. The server goes on vacations repeatedly until the queue is found non-empty. This models the fact that the mobile node goes to sleep by turning off the radio as long as there are no packets destined to it.

In practice, the mobile needs to turn on the radio to check for packets. The amount of time needed is called the *listen* window and is denoted $T_l$. During a listen window, the mobile can be informed of any packet that has arrived *before* the listen window. Any arrival during a listen window can only be notified in the following listen window. To comply with this requirement, we will make all but the first vacation periods start with a listen window $T_l$. The last listen window is included in the warm-up period $T_w$ (we use $T_w = T_l$.

Let $S_i$ be a generic random variable representing the time for which a node is sleeping during the $i$th vacation period. We then have $V_1 = S_1$ and $V_i = T_l + S_i$ for $i = 2, \ldots, \zeta$. In this paper, we are assuming $T_l$ to be a constant. As for the $\{S_i\}_i$, four cases will be considered as detailed further on. Figure 4 maps the state of an $M/G/1$ queue with repeated vacations to the possible states of a mobile node.

### 4.1 The Energy Gain under Power Saving

The performance metric defined in this section complements the ones derived in Sect. 3, but is specific to applications in wireless networks, and more precisely, to energy saving mechanisms. In this section, we will derive the gain in energy at a node should the power save mechanism be activated.

Having noted the possible node states, we can distinguish between four possible levels of energy consumption, that are, from highest to lowest,

- $C_{\text{high}}$: experienced during exchanges of packets which includes the busy period($B$),
- $C_{\text{listen}}$: experienced when checking for downlink packets which includes the listening periods $T_l$,

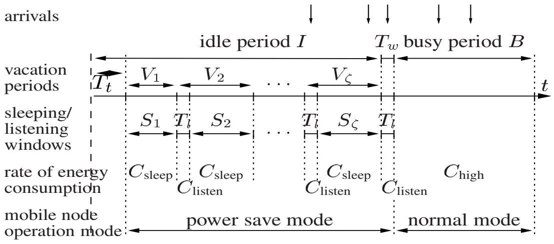


Figure 2: Mapping the $M/G/1$ queue with repeated vacations to the possible states of a mobile node.

- $C_{\text{low}}$: the lowest level observed when the mobile node is inactive but not in sleep state which includes the duration $I - \sum_{i=1}^{\zeta} V_i$,
- $C_{\text{sleep}}$: the lowest level observed when the mobile node is in sleep state which includes the total vacation duration given by $\sum_{i=1}^{\zeta} V_i$.

When the power save mechanism is not activated, the energy consumption per unit of time is $C_{\text{low}}$ in idle periods (whose expectation is $1/\lambda$) and is equal to $C_{\text{high}}$ during the busy periods (whose expectation is $\mathbb{E}[B_1]$). The energy consumption rate can be written

$$E_{\text{no sleep}} := \rho\, C_{\text{high}} + (1-\rho)C_{\text{low}} \tag{31}$$

where $\rho = \lambda\mathbb{E}[\sigma] = \mathbb{E}[B_1]/(1/\lambda + \mathbb{E}[B_1])$ (loss free system).

Consider now the case when the power save mechanism is activated. During busy periods, the energy consumption per unit of time is $C_{\text{high}}$. During idle periods, the consumption is $C_{\text{listen}}$ in listen windows (whose length is $T_l$) and is equal to $C_{\text{sleep}}$ the rest of the idle period, except the average vacation triggering duration $\widetilde{T_t} = \mathbb{E}[I] - \sum_{i=1}^{\zeta} V_i$ when the consumption is $C_{low}$. Observe that there are on average $\mathbb{E}[\zeta]$ listen windows (first listening starts after trigger time is elapsed) in each idle period; see Fig. 4. The energy consumption rate is

$$\begin{aligned} E_{\text{sleep}} \quad &:= \tfrac{1}{\mathbb{E}[C]}\left(\mathbb{E}[I\mathbb{1}\{t_f > T_t\}]C_{\text{sleep}} - \mathbb{E}[I\mathbb{1}\{t_f \le T_t\}]C_{\text{low}} + T_t\mathbb{E}[\mathbb{1}\{t_f > T_t\}](C_{\text{low}} - C_{\text{sleep}})\right. \\ &\quad \left. + \mathbb{E}[T_l\,(\zeta-1)\mathbb{1}\{t_f > T_t\}](C_{\text{listen}} - C_{\text{sleep}}) + \mathbb{E}[T_w\,\mathbb{1}\{t_f > T_t\}]C_{\text{listen}} + \mathbb{E}[B]C_{\text{high}}\right) \end{aligned} \tag{32}$$

where the average cycle duration can be computed as $\mathbb{E}[C] := \mathbb{E}[I + T_w\mathbb{1}\{t_f > T_t\} + B] = \mathbb{E}[I] + T_w\mathcal{L}_{T_t}(\lambda) + \mathbb{E}[B]$. Evaluating the above terms we obtain

$$E_{\text{sleep}} = \frac{(\mathbb{E}[I] - \mathbb{E}[\tilde{I}])C_{\text{sleep}} + \mathbb{E}[\tilde{I}]C_{\text{low}} + T_w\mathcal{L}_{T_t}C_{\text{listen}} + \mathbb{E}[B]C_{\text{high}} + \mathbb{E}[\zeta - 1]T_l\mathcal{L}_{T_t}(\lambda)(C_{\text{listen}} - C_{\text{sleep}})}{\mathbb{E}[C]}$$

where $\mathbb{E}[I]$ is in (6) and $\mathbb{E}[\tilde{I}] := \frac{1}{\lambda}\mathcal{L}_{T_t}(\lambda)$. Thus,

$$\begin{aligned} \mathbb{E}[\tilde{I}] \quad &:= \quad \mathbb{E}[I\mathbb{1}\{t_f \le T_t\}] = \mathbb{E}[\min(T_t, t_f)\mathbb{1}\{t_f \le T_t\}] \\ &\quad + \mathbb{E}\left[\left(\sum_{i=1}^{\infty} V_i\mathbb{1}_{\{\zeta \ge i\}}\right)\mathbb{1}_{\{\zeta \ne 0\}}]\mathbb{1}\{t_f \le T_t\}\right]\mathbb{E}[t_f]\mathcal{L}_{T_t}(\lambda), \end{aligned} \tag{33}$$

and

$$\begin{aligned} \mathbb{E}[I\mathbb{1}\{t_f > T_t\}] \quad &= \quad \mathbb{E}[\min(T_t, t_f)\mathbb{1}\{t_f > T_t\}] + E\left[\sum_{i=1}^{\infty} V_i\right]\mathcal{L}_{T_t}(\lambda) \\ &\quad + \mathbb{E}\left[\left(\sum_{i=1}^{\infty} V_i\mathbb{1}_{\{\zeta \ge i\}}\right)\mathbb{1}_{\{\zeta \ne 0\}}\mathbb{1}\{t_f > T_t\}\right]T_t\mathcal{L}_{T_t}(\lambda) \end{aligned} \tag{34}$$

Observe that $\mathbb{E}[B]/\mathbb{E}[C] = \rho = \lambda\mathbb{E}[\sigma]$ because we have assumed an unlimited queue (no overflow losses). The economy in energy per unit of time should a node enable its power saving mechanism is $E_{\text{no sleep}} - E_{\text{sleep}}$. The *energy gain* is defined as

$$\begin{aligned} G \quad := \quad \frac{E_{\text{no sleep}} - E_{\text{sleep}}}{E_{\text{no sleep}}} &= \frac{(1-\rho)\frac{C_{\text{low}}}{C_{\text{high}}}}{\rho + (1-\rho)\frac{C_{\text{low}}}{C_{\text{high}}}} - \frac{\rho/\mathbb{E}[B]}{(\rho + (1-\rho)\frac{C_{\text{low}}}{C_{\text{high}}})}\left((\mathbb{E}[I] - \mathbb{E}[\tilde{I}] - T_l(\mathbb{E}[\zeta] - 1))\frac{C_{\text{sleep}}}{C_{\text{high}}}\right. \\ &\quad \left. + \frac{\mathcal{L}_{T_t}(\lambda)}{\lambda}\frac{C_{\text{low}}}{C_{\text{high}}} + (T_l(\mathbb{E}[\zeta] - 1) + T_w)\mathcal{L}_{T_t}(\lambda)\frac{C_{\text{listen}}}{C_{\text{high}}}\right) \end{aligned} \tag{35}$$

We expect the battery lifetime to increase by the same factor. In practice $C_{\text{sleep}} \ll C_{\text{high}}$ so that terms in multiplication with $\frac{C_{\text{sleep}}}{C_{\text{high}}}$ can be neglected. Letting $T_w = T_l$, the lifetime gain reduces to

$$G = \frac{(1-\rho)\frac{C_{\text{low}}}{C_{\text{high}}} - \rho/\mathbb{E}[B]\left(T_l\mathbb{E}[\zeta]\mathcal{L}_{T_t}(\lambda)\frac{C_{\text{listen}}}{C_{\text{high}}} + \frac{\mathcal{L}_{T_t}(\lambda)}{\lambda}\frac{C_{\text{low}}}{C_{\text{high}}}\right)}{\rho + (1-\rho)\frac{C_{\text{low}}}{C_{\text{high}}}}. \tag{36}$$

All performance metrics found so far have been derived as functions of

- *network* parameters: such as the load $\rho$, the input rate $\lambda$, and the first and second moments of the service time ($\mathbb{E}[\sigma]$ and $\mathbb{E}[\sigma^2]$);
- *physical* parameters: such as the consumption rates $C_{\text{low}}$, $C_{\text{high}}$ and $C_{\text{listen}}$, neglecting $C_{\text{sleep}}$;
- *combined* physical and network parameters: such as the listen window $T_l$ and warm-up period $T_w$;
- the LSTs of the vacation periods and their first and second moments.

In the following we will specify the distribution of the sleep windows $\{S_i\}_i$ so as to compute explicitly $\{L_i(s)\}_i$, $\{\mathbb{E}[V_i]\}_i$ and $\{\mathbb{E}[{V_i}^2]\}_i$.

## 4.2 Sleep Windows are Deterministic

We will first consider that the sleep windows $\{S_i\}_i$ are deterministic. More precisely, let

$$S_i = a^{\min\{i-1,l\}}T_{\min}, \quad i = 1, 2, \ldots,$$

where $T_{\min}$ is the initial sleep window size, $a$ is a multiplicative factor, and $l$ is the final sleep window exponent or equivalently the number of times the sleep window could be increased. We call $T_{\min}$, $a$ and $l$ the *protocol* parameters. The LSTs of the vacations periods and their first and second moments can be rewritten

$$L_i(s) = \begin{cases} \exp(-T_{\min}s), & i = 1 \\ \exp(-(a^{\min\{i-1,l\}}T_{\min} + T_l)s), & i = 2, 3, \ldots, \end{cases}$$

$$\mathbb{E}[{V_i}^n] = \begin{cases} T_{\min}^n, & i = 1 \\ (a^{\min\{i-1,l\}}T_{\min} + T_l)^n, & i = 2, 3, \ldots, \end{cases}$$

for $n = 1, 2$.

We will study two cases so as to model type I and type II saving classes as defined in the IEEE 802.16e standard (see Sect. 1).

### Scenario D-I

This scenario is inspired by type I power saving classes. We consider $a > 1$ which implies that the first $l + 1$ sleep windows are all distinct. In particular, the value $a = 2$ is consistent with IEEE 802.16e type I power saving classes.

### Scenario D-II

In order to mimic the type II power saving classes of the IEEE 802.16e, we set $a = 1$ in this scenario. Letting $a = 1$ equates the length of all sleep windows. Observe that we could have alternatively let $l = 0$; the resulting sleep windows would then be the same, namely $S_i = T_{\min}$ for any $i$. Recall from Sect. 1 that in type II classes, a node may send or receive traffic during listen windows if the requests handling time is short enough. Hence, our model applies to these classes only if we assume that no request is sufficiently small to be served during a listen window $T_l$.

## 4.3 Sleep Windows are Exponentially Distributed

As an alternative to deterministic sleep windows, we explore in this section the situation when the sleep window $S_i$ is exponentially distributed with parameter $\mu_i$, for $i = 1, 2, \ldots$. Similar to what was done in Sect. 4.2, we let

$$\mathbb{E}[S_i] = \frac{1}{\mu_i} = a^{\min\{i-1,l\}}T_{\min}, \quad i = 1, 2, \ldots. \tag{37}$$

The LSTs of the $\{V_i\}_i$ and their first and second moments are

$$L_i(s) = \begin{cases} \dfrac{1}{1 + T_{\min}s}, & i = 1, \\ \dfrac{\exp(-sT_l)}{1 + a^{\min\{i-1,l\}}T_{\min}s}, & i = 2, 3, \ldots, \end{cases} \qquad \mathbb{E}[V_i] = \begin{cases} T_{\min}, & i = 1, \\ a^{\min\{i-1,l\}}T_{\min} + T_l, & i = 2, 3, \ldots, \end{cases}$$

$$\mathbb{E}[{V_i}^2] = \begin{cases} 2T_{\min}^2, & i = 1 \\ 2a^{2\min\{i-1,l\}}T_{\min}^2 + 2a^{\min\{i-1,l\}}T_{\min}T_l + T_l^2, & i = 2, 3, \ldots. \end{cases}$$

Like in Sect. 4.2, we consider two cases inspired by the first two types of IEEE 802.16e power saving classes.

### Scenario E-I

Similarly to what is considered in scenario D-I, we consider multiplicative factors that are larger than 1, in other words, the values $\{\mu_i\}_{i=1,\ldots,l+1}$ are different. When $a > 1$, the sleep windows increase in average over time. For $T_l = 0$ we can find closed-form expressions for all metrics derived in Sect. 3.

### Scenario E-II

The last case considered in this paper is when the sleep windows are i.i.d. exponential random variables. This can be achieved by letting either $a = 1$ or $l = 0$ in (37). Hence $\mu_i = 1/T_{\min}$ for any $i$. The LSTs of the $\{V_i\}_i$ and their first and second moments simplify to

$$L_i(s) = \begin{cases} \dfrac{1}{1 + T_{\min} s}, & i = 1, \\ \dfrac{\exp(-sT_l)}{1 + T_{\min} s} & i = 2, 3, \ldots \end{cases} \qquad \mathbb{E}[V_i] = \begin{cases} T_{\min}, & i = 1, \\ T_{\min} + T_l, & i = 2, 3, \ldots \end{cases}$$

$$\mathbb{E}[{V_i}^2] = \begin{cases} 2T_{\min}^2, & i = 1, \\ 2T_{\min}^2 + 2T_{\min}T_l + T_l^2, & i = 2, 3, \ldots \end{cases}$$

# 5 Exploiting the Analytical Results

Our model is useful for evaluating performance measures as a function of various network parameters (such as the input rate), and allows us to identify the protocol parameters that mostly impact the system performance. Instances of the expected system response time $T$ and the expected energy gain $G$ are provided in Sect. 6.1.

Beside performance evaluation, we will use our analytical model to solve a large range of optimization problems. Below we propose some optimization problems adapted to various degrees of knowledge on the parameters defining the traffic statistics.

1. **Direct optimization** This approach is useful when the traffic parameters information (e.g. the arrival rate) are directly available, or when they can be measured or estimated. An optimization problem can thus be formulated to maximize the system performance (e.g. the energy gain); see Sect. 5.1 for details.
2. **Average performance.** Given that we know the probability distribution of the traffic parameters then we may obtain the protocol parameters that optimize the expected system performance. This optimization analysis is detailed in Sect. 5.2.
3. **Worst case performance.** In the case where we do not have knowledge of even the statistical distribution of the network parameters, then we can formulate the worst case optimization problem which aims at guaranteeing the optimal performance under worst choice of network parameter. Though this is a more robust optimization approach, it yields a quite pessimistic selection of protocol parameters. Even if we do have knowledge of the statistical distribution, we may have to use a worst case performance in the case that that there is a strict bound on the value of some performance measure. The worst-case analysis will be further detailed in Sect. 5.3.

We propose a multiobjective formulation of the optimization problem, where the performance objectives are the energy consumption (or performance measures directly related to the energy consumption) and the response time. We formulate the multiobjective problem as a constrained optimization one: the energy related criterion will be optimized under a constraint on the expected sojourn time. When the traffic parameters are not directly known, two types of constraints on the expected sojourn time will be considered; in the first case the constraint is with respect to the average performance, and in the second case, it is on the worst case performance.

## 5.1 Constrained Optimization Problem

The objective is to optimize the protocol parameters defined earlier, namely, the initial window $T_{\min}$, the multiplicative factor $a$, and the exponent $l$. We define the following generic non-linear program:

$$\text{maximize} \;\; G, \quad \text{subject to} \;\; T \le T_{\text{QoS}} \tag{38a}$$

or equivalently (recall (35))

$$\text{minimize} \;\; E_{\text{sleep}}, \quad \text{subject to} \;\; T \le T_{\text{QoS}} \tag{38b}$$

where $G$ is given in (36), $E_{\text{sleep}}$ is given in (33) and $T$, the system response time, is given in (29). The program (38) maximizes the energy gain, or equivalently, minimizes the expected energy consumption rate, conditioned on a maximum system response time $T_{\text{QoS}}$. The value of $T_{\text{QoS}}$ is application-dependent; it needs to be small for interactive multimedia whereas larger values are acceptable for web traffic.

The decision variables in the above optimization will correspond to one or more protocol parameters. For a given distribution of the sleep windows $\{S_i\}_i$, the expected number of vacations $\mathbb{E}[\zeta]$, the expected idle period $\mathbb{E}[I]$, and subsequently the gain $G$ and the expected energy consumption rate $E_{\text{sleep}}$ will depend on the protocol parameters $T_{\min}$, $a$ and $l$ and on the physical parameters $C_{\text{low}}$, $C_{\text{high}}$ and $C_{\text{listen}}$ (assumed fixed).

We propose four types of applications of the mathematical program (38).

1. In the first, denoted $\mathcal{P}_1$, the decision variable is the initial expected sleep window $T_{\min}$. The parameters $a$ and $l$ are held fixed.
2. The second mathematical program, denoted $\mathcal{P}_2$, has as decision variable the multiplicative factor $a$ whereas $T_{\min}$ and $l$ are given.
3. The decision variable of the third program, denoted $\mathcal{P}_3$, is the exponent $l$. The parameters $T_{\min}$ and $a$ are given.
4. In the fourth program, denoted $\mathcal{P}_4$, all three protocol parameters are optimized. The corresponding energy gain $G$ is the highest that can be achieved.

These four mathematical programs will be solved considering (i) deterministic and (ii) exponentially distributed sleep windows $\{S_i\}_i$. Instances are provided in Sect. 6.2.

## 5.2 Expectation Analysis

Assume that the statistical distribution of the arrival process is known. Then we may obtain the protocol parameters that optimize the *expected* system performance. One may want to optimize either the expected energy consumption in power save mode or the economy of energy achieved by activating the power save mode. These problems are not equivalent as was the case in (38) since the energy consumption in normal mode itself also depends on the arrival process.

As already mentioned, we consider two different constraints on the expected sojourn time corresponding to the situations in which the application is sensitive either to the worst case value (hard constraint) or the average value (soft constraint).

### Hard Constraints

Here, the application is very sensitive to the delay, so we need to ensure that the constraint on the expected sojourn time is always satisfied no matter the value of $\lambda$.

The problem is to find the protocol parameter $\theta$ that achieves

$$\min_\theta \textstyle\sum_\lambda p(\lambda) E_{\text{sleep}}(\lambda, \theta), \quad \text{subject to } T(\lambda, \theta) \leq T_{\text{QoS}} \ \forall \lambda. \tag{39}$$

Another problem is to find the protocol parameter $\theta$ that achieves

$$\max_\theta \textstyle\sum_\lambda p(\lambda) G(\lambda, \theta), \quad \text{subject to } T(\lambda, \theta) \leq T_{\text{QoS}} \ \forall \lambda. \tag{40}$$

The problems (39) and (40) are not equivalent because $G$ depends also on $E_{\text{no sleep}}$ which itself depends on $\lambda$; recall (31). Instances of (40) will be provided in Sect. **??**.

### Soft Constraints

In this optimization problem it is assumed that the application is sensitive only to the expected sojourn time rather than to its worst case value. The objective is to find $\theta$ that achieves

$$\min_\theta \textstyle\sum_\lambda p(\lambda) E_{\text{sleep}}(\lambda, \theta), \quad \text{subject to } \textstyle\sum_\lambda p(\lambda) T(\lambda, \theta) \leq T_{\text{QoS}}. \tag{41}$$

Alternatively, one may want to find $\theta$ that achieves

$$\max \theta \textstyle\sum_\lambda p(\lambda) G(\lambda, \theta), \quad \text{subject to } \textstyle\sum_\lambda p(\lambda) T(\lambda, \theta) \leq T_{\text{QoS}}. \tag{42}$$

Instances of (42) will be provided in Sect. **??**.

## 5.3 Worst Case Analysis

When the actual input rate is unknown, then a worst case analysis can be performed to enhance the performance under the considered time constraint. Let $\theta$ represent the protocol parameter(s) over which we optimize.

### Hard Constraints

Assume the constraint on the expected sojourn time has to be satisfied for any value of $\lambda$. The problem then is to find $\theta$ that achieves

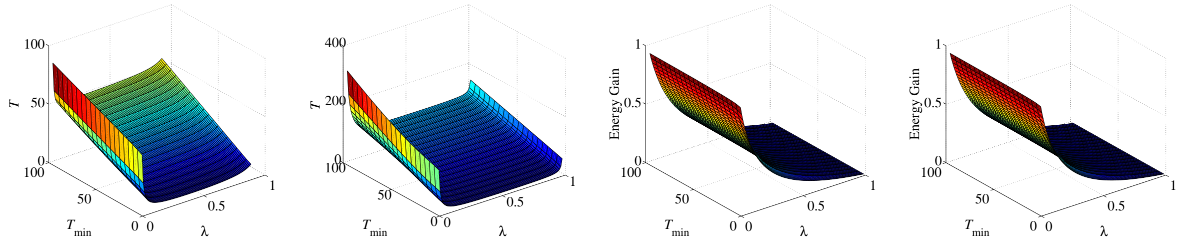
$$\min_\theta \max_\lambda E_{\text{sleep}}(\lambda, \theta), \quad \text{subject to } T(\lambda, \theta) \leq T_{\text{QoS}} \ \forall \lambda. \tag{43}$$

In other words, we want to find the value of $\theta$ that improves the worst possible energy consumption.

A different problem consists of finding $\theta$ that improves the worst possible gain, namely,

$$\max_\theta \min_\lambda G(\lambda, \theta), \quad \text{subject to } T(\lambda, \theta) \leq T_{\text{QoS}} \ \forall \lambda. \tag{44}$$

Observe that the worst possible gain is the one obtained when the traffic input rate tends to $1/\mathbb{E}[\sigma]$. Thus $\min_\lambda G(\lambda, \theta) \approx 0$. Therefore, the above problem is meaningful only for a restricted range of small values of $\lambda$ for which the worst energy gain is far above 0. Instances of (44) will be provided in Sect. **??**.



(a) sojourn time, deterministic sleep windows (b) sojourn time, exponential sleep windows (c) energy gain, deterministic sleep windows (d) energy gain, exponential sleep windows

Figure 3: Impact of $T_{\min}$ on $T$ and $G$ in type I like power saving classes.

### Soft Constraints

Here, the application is not very sensitive to the delay, so it is acceptable that the constraint is respected by the average performance. The statistical distribution of the input rate, denoted $p(\lambda)$, is assumed to be known. The problem is to find $\theta$ that achieves

$$\min_\theta \max_\lambda E_{\text{sleep}}(\lambda, \theta), \quad \text{subject to } \textstyle\sum_\lambda p(\lambda) T(\lambda, \theta) \leq T_{\text{QoS}}. \tag{45}$$

Again, a different objective can be desired, namely to maximize the worst gain. Like what was mentioned in the previous section, the problem is meaningful only when the rate $\lambda$ is small.

$$\max_\theta \min_\lambda G(\lambda, \theta), \quad \text{subject to } \textstyle\sum_\lambda p(\lambda) T(\lambda, \theta) \leq T_{\text{QoS}}. \tag{46}$$

Instances of (46) will be provided in Sect. **??**.

# 6 Results and Discussion

We have performed an extensive numerical analysis to evaluate the performance of the system in terms of the expected system response time $T$ given in (29) and the expected energy gain $G$ given in (36); cf. Sect. 6.1. In addition we have solved the problems $\mathcal{P}_1$–$\mathcal{P}_4$ for given values of the protocol parameters held fixed; cf. Sect. 6.2. Instances of the problems (40), (42), (44) and (46) are also provided; cf. Sect. **??**. We first consider vacation trigger time $T_t = 0$, i.e. when ever the system is idle it is bound to go for atleast one vacation. Latter we illustrate the the impact of vacation trigger time depicting one of the case.

Physical and network parameters have been selected as follows:

$$C_{\text{low}}/C_{\text{high}} = C_{\text{listen}}/C_{\text{high}} = 0.2, \quad \mathbb{E}[\sigma] = 1, \quad \mathbb{E}[\sigma^2] = 2, \quad T_l = T_w = 1, \quad T_{\text{QoS}} = 50/100.$$

Unless otherwise specified, the protocol parameters are set to the *default* values: $T_{\min} = 2$, $a = 2$, $l = 9$ and $T_t = 0$ in scenarios D-I and E-I, and $T_{\min} = 2$, $a = 1$, $l = 0$ and $T_t = 0$ in scenarios D-II and E-II.

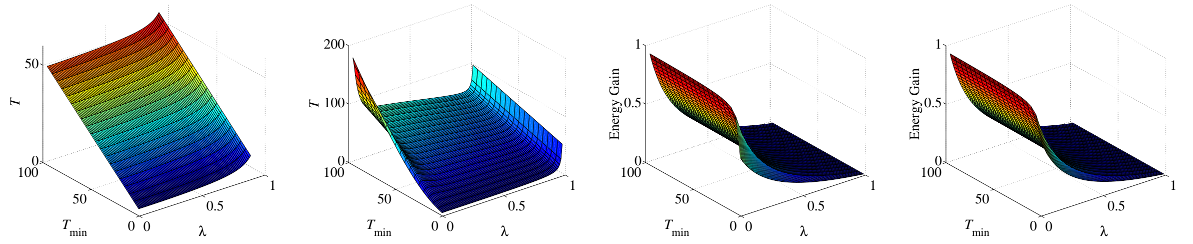
We have varied $\lambda$ in the interval $(0, 1)$, $T_{\min}$ in $(1, 100)$, $a$ in $(1, 10)$, and $T_t$ in $(0, 10)$. The parameter $l$ takes integer values in the interval $(0, 10)$.

## 6.1 Performance Evaluation

We have evaluated numerically the expected sojourn time $T$ and the expected energy gain $G$ in all four scenarios defined in Sects. 4.2 and 4.3, varying the input rate $\lambda$ and the three protocol parameters $T_{\min}$, $a$ and $l$. Our results will be presented in the following sections. First, we discuss the impact of each of the three parameters on the performance of the system in terms of $T$ and $G$: impact of $T_{\min}$ in Sect. 6.1.1, impact of $a$ in Sect. 6.1.2, and impact of $l$ in Sect. 6.1.3. Then, we comment on each of the performance metrics: comments on $T$ are in Sect. 6.1.5, and comments on $G$ are in Sect. 6.1.6.

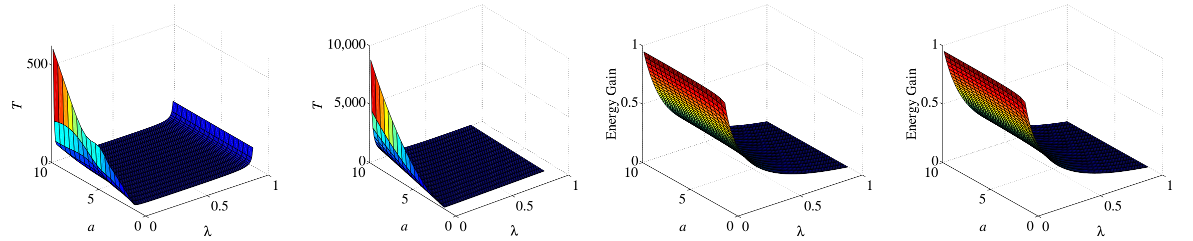
### 6.1.1 Impact of the initial window size $T_{\min}$

We will first investigate the impact that the initial window size $T_{\min}$ has on the performance of the system. For reasons that will be made clear later, this parameter is foreseen to be the most important parameter in type I like power saving classes (scenarios D-I and E-I) and it is the unique parameter in type II like power saving classes (scenarios D-II and E-II).



(a) sojourn time, deterministic sleep windows (b) sojourn time, exponential sleep windows (c) energy gain, deterministic sleep windows (d) energy gain, exponential sleep windows

Figure 4: Impact of $T_{\min}$ on $T$ and $G$ in type II like power saving classes.



(a) sojourn time, deterministic sleep windows (b) sojourn time, exponential sleep windows (c) energy gain, deterministic sleep windows (d) energy gain, exponential sleep windows

Figure 5: Impact of $a$ on $T$ and $G$ with either deterministic or exponential $\{S_i\}_i$.

**Type I like power saving classes** We set $a = 2$, $T_t = 0$ and $l = 9$ in scenarios D-I and E-I. The results are graphically reported in Fig. 3.

Figures 3(a) and 3(b) respectively depict the expected sojourn time $T$ against the traffic input rate $\lambda$ and the initial sleep window size $T_{\min}$ when sleep windows are deterministic and exponentially distributed. The energy gain under the same conditions is depicted in Figs. 3(c) and 3(d).

The size of the initial sleep window has a large impact on $T$ for any value of $\lambda$. More precisely, $T$ increases linearly with an increasing $T_{\min}$ for any $\lambda$; see Figs. 3(a), 3(b). As for the gain $G$, it is not impacted by $T_{\min}$, except for a small degradation at very small values of $T_{\min}$, hardly visible in Figs. 3(c) and 3(d).

**Type II like power saving classes** We set $a = 1$, $T_t = 0$ and $l = 0$ in scenarios D-II and E-II. The results are graphically reported in Fig. 4.

Figures 4(a) and 4(b) respectively depict the expected sojourn time $T$ against the traffic input rate $\lambda$ and the initial sleep window size $T_{\min}$ when sleep windows are deterministic and exponentially distributed. The energy gain under the same conditions is depicted in Figs. 4(c) and 4(d).

About the impact of $T_{\min}$ on $T$ and $G$, we can make similar observations to those made for type I like power saving classes, to the only exception that here the degradation of $G$ at very small values of $T_{\min}$ is more visible, especially in Fig. 4(c).

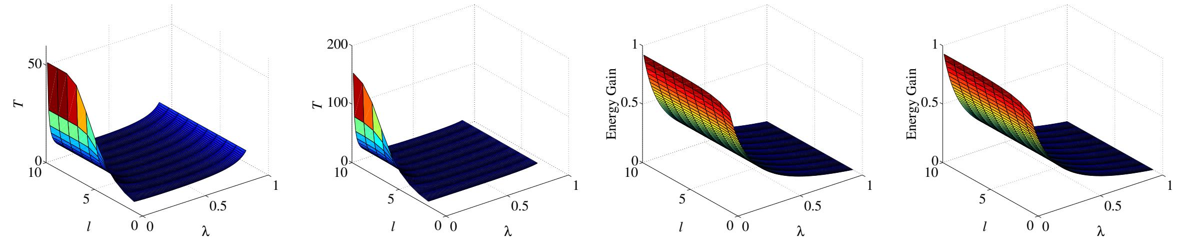
Observe that a larger $T_{\min}$ yields a larger sleep time but it also reduces $\mathbb{E}[\zeta]$ which together explains why the impact on the energy gain is not significant.

### 6.1.2 Impact of the multiplicative factor $a$

The second parameter used in type I like power saving classes (scenarios D-I and E-I) is the multiplicative factor $a$. In order to assess the impact of $a$ on the performance of the system, we perform a numerical analysis in which the initial window size is $T_{\min} = 2$, the vacation trigger time $T_t = 0$, the exponent is $l = 9$ and the multiplicative factor $a$ is varied from 1 to 10. We evaluate the expected sojourn time $T$ and the energy gain $G$ both for deterministic (scenario D-I) and exponentially distributed (scenario E-I) sleep windows. We report the results in Fig. 5.

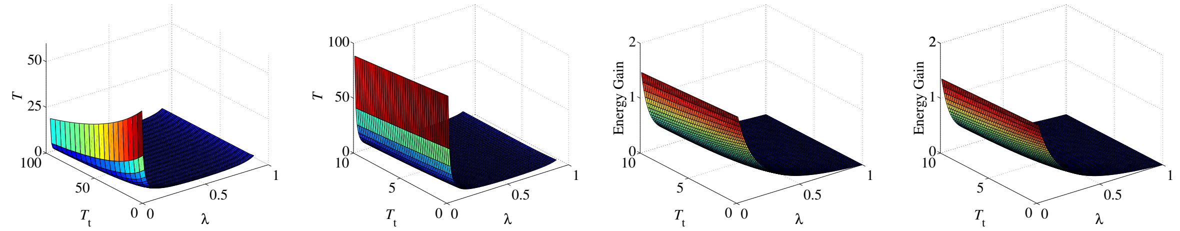
Figures 5(a) and 5(c) respectively depict the expected sojourn time $T$ and the energy gain $G$ against the traffic input rate $\lambda$ and the multiplicative factor $a$ when sleep windows are deterministic. The results obtained when the sleep windows are exponentially distributed are displayed in Figs. 5(b) and 5(d).

Interestingly enough, the multiplicative factor $a$ does not impact the gain $G$. It impacts greatly $T$ but only at very low input rates. Observe that $T$ increases exponentially with an increasing $a$ for small $\lambda$ which is reflected in Figs. 5(a) and 5(b).



(a) sojourn time, deterministic sleep windows (b) sojourn time, exponential sleep windows (c) energy gain, deterministic sleep windows (d) energy gain, exponential sleep windows

Figure 6: Impact of $l$ on $T$ and $G$ with either deterministic or exponential $\{S_i\}_i$.



(a) sojourn time, deterministic sleep windows (b) sojourn time, exponential sleep windows (c) energy gain, deterministic sleep windows (d) energy gain, exponential sleep windows

Figure 7: Impact of $T_t$ on $T$ and $G$ with either deterministic or exponential $\{S_i\}_i$.

### 6.1.3 Impact of the exponent $l$

The third parameter used in type I like power saving classes (scenarios D-I and D-II) is the exponent $l$. In order to assess the impact of the maximum sleep window size on the performance of the system, we perform a numerical analysis in which the multiplicative factor is $a = 2$, the initial window size is $T_{\min} = 2$, the vacation trigger time $T_t = 0$, and the exponent $l$ is varied from 0 to 10. We evaluate the expected sojourn time $T$ and the energy gain $G$ both for deterministic (scenario D-I) and exponentially distributed (scenario E-I) sleep windows. We report the results in Fig. 6.

Figures 6(a) and 6(c) respectively depict the expected sojourn time $T$ and the energy gain $G$ against the traffic input rate $\lambda$ and the exponent $l$ when sleep windows are deterministic. The results obtained when the sleep windows are exponentially distributed are displayed in Figs. 6(b) and 6(d).

Alike the multiplicative factor, the exponent $l$ has a large impact on $T$ only for a very low traffic input rate, and has no impact on $G$ whatever the rate $\lambda$.

Observe in Fig. 6(a) that $T$ becomes almost insensitive to $l$ beyond $l = 7$ (for small $\lambda$). Here the initial vacation window $T_{\min}$ is 2. We have computed $T$ considering larger values of $T_{\min}$, and have observed that $T$ saturates faster with $l$ when the initial sleep window is larger. A similar behavior is observed in the exponential case for higher $T$; cf. Fig. 6(b).

### 6.1.4 Impact of Vacation Trigger Time $T_t$

The fourth and the last parameter used in type I like power saving classes is the vacation trigger time $T_t$. In order to assess the impact of the vacation trigger time on the performance of the system, we perform a numerical analysis in which the multiplicative factor is $a = 2$, the initial window size is $T_{\min} = 2$ the exponent $l = 9$ and the vacation trigger time is varied from 0 to 10. We evaluate the expected sojourn time $T$ and the energy gain $G$ both for deterministic (scenario D-I) and exponentially distributed (scenario E-I) sleep windows. We report the results in Fig. 7.

Figures 7(a) and 7(c) respectively depict the expected sojourn time $T$ and the energy gain $G$ against the traffic input rate $\lambda$ and the vacation trigger time $T_t$ when sleep windows are deterministic. The results obtained when the sleep windows are exponentially distributed are displayed in Figs. 7(d) and 7(b).

As expected, decreasing vacation trigger time enhances the probability of the system to go on vacation resulting larger response time $T$ and larger gain $G$ as well for any $\lambda$.

### 6.1.5 The expected sojourn time $T$

The numerical results of the expected sojourn time $T$ are reported in Figs. 3–6, parts (a) and (b). As already mentioned, $T$ is fairly insensitive to parameters $l$ and $a$ except for very small values of $\lambda$. However, $T$ increases linearly as $T_{\min}$ increases.

In scenarios D-I, E-I and E-II, as $\lambda$ increases, $T$ first decreases rapidly then becomes fairly insensitive to $\lambda$ up to a certain point beyond which $T$ increases abruptly. This can easily be explained. The sojourn time is essentially composed of two main components: the delay incurred by the vacations of the server and the queueing delay once the server is active. As the input rate increases, the first component decreases while the second one increases. For moderate values of $\lambda$, both components balance each other yielding a fairly insensitive sojourn time. The large value of $T$ at small $\lambda$ is mainly due to the ratio $\mathbb{E}[I_a]/\mathbb{E}[I]$ (recall (29)), whereas the abrupt increase in $T$ at large $\lambda$ is due to the term $\frac{\lambda \mathbb{E}[\sigma^2]}{2(1-\rho)}$, which is the waiting time in the $M/G/1$ queue without vacation.

The situation in scenario D-II is different in that $T$ is not large at small input rates $\lambda$. Recall that in this scenario, all sleep window are equal to a constant $T_{\min}$. As a consequence, the delay incurred by the vacations of the server is not as large as in the other scenarios. The balance between the two main components of the sojourn time stretches down to small values of $\lambda$.

### 6.1.6 The expected energy gain $G$

The numerical results of the expected energy gain $G$ are reported in Figs. 3–6, parts (c) and (d). As already mentioned, $G$ is insensitive to parameters $l$ and $a$ for any $\lambda$, and sensitive to $T_{\min}$ up to a certain initial sleep window size.

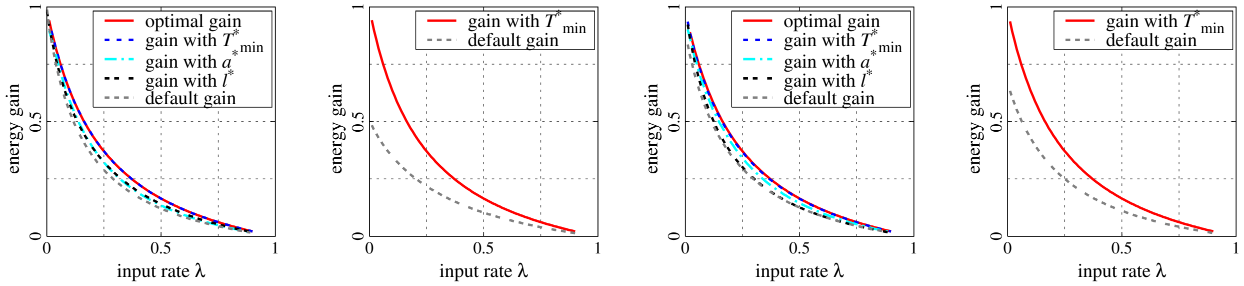
The expected energy gain $G$ decreases monotonically as $\lambda$ increases which can be explained as follows. The larger the input traffic rate $\lambda$, the shorter we expect the idle time to be and hence the smaller the gain.

## 6.2 Constrained Optimization Problem

We have solved the constrained optimization program introduced in Sect. 5.1 as follows

- $\mathcal{P}_1$ for $T^*_{\min}$ when $a = 2$ and $l = 9$ (default values) with $T_{\text{QoS}} = 50$ for scenario D-I and $T_{\text{QoS}} = 100$ for scenario E-I, and when $a = 1$ or $l = 0$ with $T_{\text{QoS}} = 50$ for scenario D-II and $T_{\text{QoS}} = 100$ for scenario E-II;
- $\mathcal{P}_2$ for $a^*$ with $T_{\min} = 2$ and $l = 9$ (default values) with $T_{\text{QoS}} = 50$ for scenario D-I and $T_{\text{QoS}} = 100$ for scenario E-I;
- $\mathcal{P}_3$ for $l^*$ when $T_{\min} = 2$ and $a = 2$ (default values) with $T_{\text{QoS}} = 50$ for scenario D-I and $T_{\text{QoS}} = 100$ for scenario E-I;
- $\mathcal{P}_4$ for $(T_{\min}, a, l)^*$ with $T_{\text{QoS}} = 50$ for deterministic sleep windows and $T_{\text{QoS}} = 100$ for exponential sleep windows.

The optimal gain achieved by the four programs $\mathcal{P}_1$–$\mathcal{P}_4$ and the gain obtained when using the default values are illustrated in Fig. 8 against the input rate $\lambda$, for deterministic (Figs. 8(a) and 8(b)) and exponential (Figs. 8(c) and 8(d)) sleep windows. The right-hand-side graphs depict the optimal gain (returned by program $\mathcal{P}_1$ when $a = 1$) and the gain achieved under the default protocol parameter ($T_{\min} = 2$).



(a) scenario D-I, $T_{\text{QoS}} = 50$

(b) scenario D-II, $T_{\text{QoS}} = 50$

(c) scenario E-I, $T_{\text{QoS}} = 100$

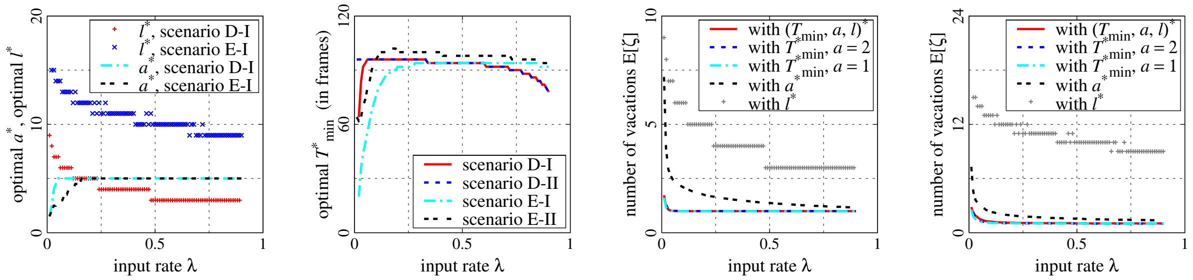
(d) scenario E-II, $T_{\text{QoS}} = 100$

Figure 8: Maximized/default gain versus the input rate $\lambda$.

The most relevant observation to be made on each of Figs. 8(a) and 8(c) is the match between the curve labeled "optimal gain" (result of program $\mathcal{P}_4$) and the curve labeled "gain with $T^*_{\min}$" (result of program $\mathcal{P}_1$). The interest of this observation comes from the fact that $\mathcal{P}_4$ involves a multivariate optimization whereas $\mathcal{P}_1$ is a much simpler single variate program. The explanation for this match is as follows. The program $\mathcal{P}_1$ is being solved for the optimal $T_{\min}$. It thus quickly reduces the number of vacations $\mathbb{E}[\zeta]$ to 1 (refer to Fig. 9) and thereby makes the role of both $a$ and $l$ insignificant. Hence, the energy gain maximized by $\mathcal{P}_1$ tends to the optimal gain returned by $\mathcal{P}_4$.

When maximizing the gain by optimizing $T_{\min}$ (program $\mathcal{P}_1$; see Fig. 9(b)), we observe in all scenarios but scenario D-II that, optimally, $T_{\min}$ should first increase with the input rate $\lambda$ then decrease with increasing $\lambda$ for large values of $\lambda$. This observation is rather counter-intuitive and we do not have an explanation for it at the moment. Our intuition that $T_{\min}$ should decrease as $\lambda$ increases is confirmed only in scenario D-II.

To maximize the energy gain, one could minimize the factor multiplying $C_{\text{sleep}}$, in other words minimize $\mathbb{E}[\zeta]$. As a consequence, if $T_{\min}$ is optimally selected, then the initial sleep window will be set large enough so that the server will rarely go for a second vacation period, thereby eliminating the unnecessary energy consumption incurred by potential subsequent listen windows. As a consequence, the multiplicative factor $a$ and the exponent $l$ will have a negligible effect on the performance of the system.



(a) optimal $a^*$ and $l^*$ versus $\lambda$

(b) optimal $T^*_{\min}$ versus $\lambda$

(c) deterministic sleep windows

(d) exponential sleep windows

Figure 9: Optimal parameters obtained from $\mathcal{P}_1$–$\mathcal{P}_3$ and set to obtain $\mathbb{E}[\zeta]$ versus $\lambda$.

# 7 Conclusion and Perspectives

In this paper, we have analyzed the $M/G/1$ queue with repeated inhomogeneous vacations. In all prior work, repeated vacations are assumed to be i.i.d., whereas in our model the duration of a repeated vacation can come from an entirely different distribution. Using transform-based analysis, we have derived various performance measures of interest such as the expected system response time and the gain from idling the server. We have applied the model to study the problem of power saving for mobile devices. The impact of the power saving strategy on the network performance is easily studied using our analysis. We have formulated various constrained optimization problems aimed at determining optimal parameter settings. We have performed an extensive numerical analysis to illustrate our results, considering four different strategies of power saving having either deterministic or exponentially distributed sleep durations. We have found that the parameter that most impacts the performance is the initial sleep window size. Hence, optimizing this parameter solely is enough to achieve quasi-optimal energy gain.

Other important research directions are considered. Namely,

**Other traffic profiles.** It is interesting to consider more bursty real time traffic as well as TCP traffic. We expect that much of this work may have to be performed through simulations as the queueing analysis may become intractable. It is important to examine how our optimized parameters perform when a new type of traffic is introduced, and whether our robust design for the worst case Poisson traffic maintains its robustness beyond the Poisson arrival processes.

**Extensions of the protocol.** We intend to examine rendering $T_{\min}$ dynamic, by choosing its value at the $n$th idle time as a function of the $V_\zeta$ (or of its expectation) in the $(n-1)$-st idle time.

# References

[1] "IEEE std 802.16e-2005 and IEEE std 802.16-2004/cor 1-2005 (amendment and corrigendum to IEEE Std 802.16-2004)," 2006.

[2] J. B. Seo, S. Q. Lee, N. H. Park, H. W. Lee, and C. H. Cho, "Performance analysis of sleep mode operation in IEEE 802.16e," in *Proc. of IEEE VTC 2004-Fall*, vol. 2, (Los Angeles, California, USA), pp. 1169–1173, September 2004.

[3] Y. Xiao, "Energy saving mechanism in the 802.16e wireless MAN," *IEEE Communication letters*, vol. 9, pp. 595–597, July 2005.

[4] Y. Xiao, "Performance analysis of an energy saving mechanism in the IEEE 802.16e wireless MAN," in *Proc. of IEEE CCNC 2006*, vol. 1, pp. 406–410, January 2006.

[5] K. Han and S. Choi, "Performance analysis of sleep mode operation in IEEE 802.16e mobile broadband wireless access systems," in *Proc. of IEEE VTC 2006-Spring*, vol. 3, (Melbourne, Australia), pp. 1141–1145, May 2006.

[6] B. T. Doshi, "Queueing systems with vacations - a survey," *Queueing Systems - Theory and Applications*, vol. 1, no. 1, pp. 29–66, 1986.

[7] S. W. Fuhrmann and R. B. Cooper, "Stochastic decompositions in the M/G/1 queue with generalized vacations," *Operations Research*, pp. 1117–1129, Sep.-Oct. 1985.

[8] J. Gaver, D. P., "A waiting line with interrupted service, including priorities," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 24, no. 1, pp. 73–90, 1962.

[9] R. B. Cooper, "Queues served in cyclic order : Waiting times," *Bell Syst. Tech. J.*, no. 49, pp. 399–413, 1970.

[10] Y. Lévy and U. Yechiali, "Utilization of idle time in an m/g/1 queueing system," *Mgmt. Sci.*, no. 22, pp. 202–211, 1975.

[11] J. G. Shanthikumar, "Some analyses of the control of queues using level crossing of regenerative processes," *J. Appl. Prob.*, no. 17, pp. 814–821, 1980.

[12] M. Scholl and L. Kleinrock, "On the m/g/1 queue with rest periods and certain service-independent queueing disciplines," *Opns. Res.*, no. 17, pp. 705–719, 1983.

[13] J. Teghem, “Control of service processes in a queuing system,” *Eur. J. Opnl. Res.*, vol. 23, pp. 141–158, 1986.

[14] J. G. Shanthikumar, “On stochastic decomposition in m/g/1 type queues with generalized server vacations,” *Operation Research*, vol. 36, pp. 566–569, July- August 1988.

[15] A. P. Azad, S. Alouf, and E. Altman, “Sleep Mode Analysis via Workload Decomposition,” research report, INRIA, 2010. https://hal.inria.fr/hal-01352876.

[16] S. Alouf, E. Altman, and A. P. Azad, “Analysis of an m/g/1 queue with repeated inhomogeneous vacations with application to ieee 802.16e power saving mechanism,” in *Proceedings of QEST*, (Saint-Malo, France), pp. 27–36, September 2008.

[17] R. W. Wolff, “Poisson arrivals see time averages,” *Operations Research*, vol. 30, no. 2, pp. 223–231, 1982.

[18] L. Kleinrock, *Queueing Systems: Theory*, vol. 1. John Wiley and Sons, 1975.

[19] R. B. Cooper, *Introduction to Queueing theory*, vol. 2. North-Holland(Elsevier), NewYork, 1981.

[20] H. Takagi, *Queueing Analysis: Vacation and Priority Systems Vol. 1*. Elsevier, North Holland, 1991.