Analysis of an M/G/1 Queue with Repeated Inhomogeneous Vacations with Application to IEEE 802.16e Power Saving Mechanism

Sara Alouf, Eitan Altman, Amar Prakash Azad

Analysis of an M/G/1 Queue with Repeated Inhomogeneous Vacations with Application to IEEE 802.16e Power Saving Mechanism*

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

Abstract

The goal of this paper is to establish a general approach for analyzing queueing models with repeated inhomogeneous vacations. At the end of a vacation, the server goes on another vacation, possibly with a different *probability distribution, if during the previous vacation there have been no arrivals. In case there have been one or more arrivals during a vacation then a busy period starts after a warm-up time. In order to get an insight on the influence of parameters on the performance, we choose to study a simple $M/G/1$ queue (Poisson arrivals and general independent service times) 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 vacations of the server. 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 achieved in power save mode, with constraints as QoS conditions to be met. 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.*

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 on the other hand it forces a trade off in terms of delay for various QoS services e.g. voice and video traffic. The mobility extension of WiMAX [5] is one of the most recent technologies whose sleep mode operation is discussed in detail and is being standardized.

The IEEE 802.16e standard [5] 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. Sleep and listen windows are interleaved as in type I classes. However, unlike type I classes, a node may send or receive traffic during listen windows if the requests handling time is short enough. The related 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 [7], 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 [8, 9]. While [8] considers incoming traffic solely, both incoming and outgoing traffic are considered in [9]. In [4], the authors evaluate the performance of the type I power saving class in terms of packet delay and power consumption through the analysis of a semi-Markov chain.

In this paper, we propose a queueing-based modeling framework that is general enough to study many of the power save operations described in standards and in the lit-

*This work has been partially supported by the French national project ANR WINEM ANR-06-TCOM-05. The work of the third author was part of the INRIA associate group DAWN.

erature. 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 [5]. 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 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 very rich literature on queues with vacations, see e.g. the survey by Doshi [2]. 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 upon empty queue.

The rest of the report 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 concludes the report and outlines some perspectives.

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 empties. 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 λ. Let σ 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) 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 ζ vacation periods denoted $V_1, \ldots, V_\zeta$; (ii) a *warm-up* period; it is a fixed duration denoted T_w during which the server is warming up to start serving requests; (iii) 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):

- $\hat{V}_i$ refers to the end of the ith 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 $I = \hat{V}_\zeta = \sum_{i=1}^{\zeta} V_i$;
- T_Z refers to the beginning of the busy period B; we define $Z := X(T_Z)$ as the queue size at the beginning of a busy period;
- T_i refers to the first time the queue size *decreases* to the value i (i.e. $X(T_i) = i$) for $i = Z - 1, \ldots, 0$; observe that the cycle ends at T_0.

The times $\{T_i\}_{i=Z,Z-1,\ldots,0}$ delimit Z subperiods in B. We can write $B = \sum_{i=1}^{Z} B_i$ where $B_i = T_{i-1} - T_i$.

The random variable Z is in fact the number of arrivals from $t = 0$ until time T_Z, even though all of the arrivals occur between $\hat{V}_{\zeta-1}$ and T_Z. Introduce Z_I as the number of requests that have arrived up to time $\hat{V}_\zeta$ (i.e. during period I) and Z_w as the number of arrivals during the warm-up period T_w. Hence $Z = Z_I + Z_w$. Observe that $X(I) = Z_I$.

A possible trajectory of $X(t)$ during a regeneration cycle is depicted in Fig. 1 where we have shown the notation introduced so far. The introduction of the notation A, A_w and Q_Z is deferred until Sect. 3.5.

3. Analysis

This section is devoted to the analysis of the queueing system presented in Sect. 2. We will derive the distributions of ζ and Z, the expectations of ζ, I, Z, B and $X(t)$ and the second moments of I and Z, and last compute the system response time. The gain from idling the server is introduced when applying the model to study the power saving mechanism; see Sect. 4.

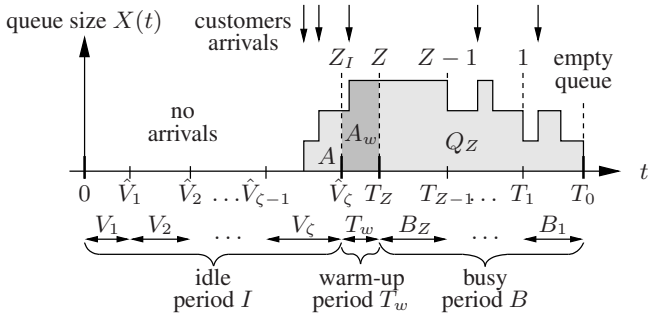


Figure 1. Sample trajectory of the queue size during a regeneration cycle.

3.1. The number of vacations

To compute the distribution of ζ, 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 $\hat{V}_{i-1} = \sum_{k=1}^{i-1} V_k$. 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 $L_k(s) = \mathrm{E}[\exp(-sV_k)]$ the Laplace Stieltjes transform (LST) of V_k, we can readily write

$$\begin{aligned} P(\zeta = 1) &= P(A_1^c) = \mathrm{E}[\mathbb{1}\{A_1^c\}] = \mathrm{E}[\mathrm{E}[\mathbb{1}\{A_1^c\}|V_1]] \\ &= \mathrm{E}\left[1 - \exp(-\lambda V_1)\right] = 1 - L_1(\lambda), \\ P(\zeta = i) &= P(A_i^c) \prod_{k=1}^{i-1} P(A_k) = (1 - L_i(\lambda)) \prod_{k=1}^{i-1} L_k(\lambda), \\ P(\zeta \geq i) &= \prod_{k=1}^{i-1} P(A_k) = \prod_{k=1}^{i-1} L_k(\lambda), \end{aligned} \tag{1}$$

for $i > 1$, where we have used the fact that arrivals are Poisson with rate λ. The product $\prod_{k=a}^{b} L_k(\lambda)$ is defined as equal to 1 for any $b < a$. Using (1), we can write

$$\mathrm{E}[\zeta] = \sum_{i=1}^{\infty} iP(\zeta = i) = \sum_{i=1}^{\infty} P(\zeta \geq i) = \sum_{i=1}^{\infty} \prod_{k=1}^{i-1} L_k(\lambda). \tag{2}$$

3.2. The idle period

The idle period $I = \sum_{i=1}^{\zeta} V_i$ can be rewritten as $I = \sum_{i=1}^{\infty} V_i \mathbb{1}\{\zeta \geq i\}$. Since V_i does not depend on the event of no arrivals during $\hat{V}_{i-1}$, we have for a Poisson arrival process, using (1),

$$\mathrm{E}[I] = \sum_{i=1}^{\infty} \mathrm{E}[V_i] \prod_{k=1}^{i-1} L_k(\lambda).$$

The second moment of I can be written as $\mathrm{E}[I^2] = \mathrm{E}[I_a] + 2\mathrm{E}[I_b]$, where

$$\begin{aligned} I_a &:= \sum_{i=1}^{\infty} V_i^2 \mathbb{1}\{\zeta \geq i\}, \\ I_b &:= \sum_{i=1}^{\infty} \sum_{j=1}^{i-1} V_i V_j \mathbb{1}\{\zeta \geq i\} = \sum_{i=1}^{\infty} \sum_{j=1}^{i-1} V_i V_j \prod_{k=1}^{i-1} \mathbb{1}\{A_k\}. \end{aligned}$$

Observe that in I_b, only $\mathbb{1}\{A_j\}$ and V_j depend on each other.

$$\begin{aligned} \mathrm{E}[V_j \mathbb{1}\{A_j\}] &= \mathrm{E}\Big[\mathrm{E}\left[V_j \mathbb{1}\{A_j\}|V_j\right]\Big] = \mathrm{E}\Big[V_j P(A_j|V_j)\Big] \\ &= \mathrm{E}\Big[V_j \exp(-\lambda V_j)\Big] = -\left.\frac{dL_j(s)}{ds}\right|_{s=\lambda}. \end{aligned} \tag{3}$$

Using (3) and the LST of V_i, we find after some calculus

$$\begin{aligned} \mathrm{E}[I_a] &= \sum_{i=1}^{\infty} \mathrm{E}\left[V_i^2\right] \prod_{k=1}^{i-1} L_k(\lambda), \\ \mathrm{E}[I_b] &= \sum_{i=1}^{\infty} \mathrm{E}[V_i] \prod_{k=1}^{i-1} L_k(\lambda) \sum_{j=1}^{i-1} \frac{1}{L_j(\lambda)} \left.\frac{-dL_j(s)}{ds}\right|_{s=\lambda}. \end{aligned}$$

3.3. The initial backlog in busy periods

The number of requests waiting in the queue at the beginning of a busy period is $Z = Z_I + Z_w$. Since the arrival process is Poisson, it is obvious that Z_w, the number of arrivals during a warm-up period T_w, is a Poisson variable with parameter λT_w. We then have

$$\mathrm{E}[Z_w] = \lambda T_w, \tag{4}$$

$$\mathrm{E}\left[Z_w^2\right] = \lambda T_w \left(\lambda T_w + 1\right). \tag{5}$$

Proposition 1 *The first and second moments of Z_I are*

$$\mathrm{E}[Z_I] = \lambda \mathrm{E}[I], \tag{6}$$

$$\mathrm{E}[Z_I^2] = \lambda^2 \mathrm{E}[I_a] + \lambda \mathrm{E}[I]. \tag{7}$$

The distribution of Z_I and the detailed derivation of Proposition 1 can be found in [1]. Since Z_I and Z_w are independent random variables, we have (using (4)-(7))

$$\mathrm{E}[Z] = \lambda(\mathrm{E}[I] + T_w), \tag{8}$$

$$\frac{\mathrm{E}[Z^2]}{\mathrm{E}[Z]} = \lambda \frac{\mathrm{E}[I_a] + \mathrm{E}[I] T_w}{\mathrm{E}[I] + T_w} + \lambda T_w + 1. \tag{9}$$

3.4. The busy period

Recall from Sect. 2 that a busy period is composed of Z subperiods. These periods are delimited by the times $\{T_i\}_{i=Z,Z-1,\ldots,0}$; see Fig. 1. The busy period can be expressed as $B = \sum_{i=1}^{Z} B_i$. Observe that B_1 is the busy period of a simple $M/G/1$ queue without vacations. The busy periods $\{B_i\}_i$ are i.i.d. and have the same distribution as the busy period of an $M/G/1$ queue. Therefore,

$$\mathrm{E}[B] = \mathrm{E}\left[\mathrm{E}[B|Z]\right] = \mathrm{E}\left[Z\mathrm{E}[B_1]\right] = \mathrm{E}[Z]\mathrm{E}[B_1]. \tag{10}$$

Considering the loss free $M/G/1$ queue, we know that the load $\rho := \lambda \mathrm{E}[\sigma]$ is equal to the server utilization $\mathrm{E}[B_1]/(\mathrm{E}[B_1] + 1/\lambda)$. Hence, $\mathrm{E}[B_1] = \frac{\mathrm{E}[\sigma]}{1-\rho}$ and using (8), (10) can be rewritten

$$\mathrm{E}[B] = \frac{\rho}{1-\rho}(\mathrm{E}[I] + T_w). \tag{11}$$

3.5. The queue size

In this section, we focus on deriving the expected queue size $\mathrm{E}[X(t)]$. For convenience, and without loss of generality, we have let $t = 0$ at the beginning of a regeneration cycle. The queue is empty until the first customer arrival in the vacation V_ζ, so $X(t) = 0$ for $0 \le t \le \hat{V}_{\zeta-1}$. After the first arrival, the queue may only increase up to the time T_Z, so $X(t)$ is a non-decreasing step function for $\hat{V}_{\zeta-1} < t \le T_Z$. After time T_Z, the queue may decrease or increase according to whether a service has ended or a customer has arrived to the queue. Define

$$A := \int_{\hat{V}_{\zeta-1}}^{\hat{V}_\zeta} X(t)dt,\ A_w := \int_{\hat{V}_\zeta}^{T_Z} X(t)dt,\ Q_Z := \int_{T_Z}^{T_0} X(t)dt,$$

as the total area under the curve $X(t)$ for the idle, warm-up and busy periods respectively, as can be seen in Fig. 1. The subscript Z in Q_Z expresses the fact that the initial queue size is Z. We can write

$$\mathrm{E}[X] = \frac{\mathrm{E}[A] + \mathrm{E}[A_w] + \mathrm{E}[Q_Z]}{\mathrm{E}[I] + T_w + \mathrm{E}[B]}.$$

Using (11), we can rewrite $\mathrm{E}[X]$ as follows

$$\mathrm{E}[X] = (1-\rho)\frac{\mathrm{E}[A] + \mathrm{E}[A_w] + \mathrm{E}[Q_Z]}{\mathrm{E}[I] + T_w}. \qquad (12)$$

Proposition 2 *The expectation of A is given by*

$$\mathrm{E}[A] = \frac{\lambda}{2}\sum_{i=1}^{\infty}\prod_{k=1}^{i-1} L_k(\lambda)\,(1 - L_i(\lambda))\sum_{j=0}^{\infty}\left.\frac{d^2 L_i(s)}{ds^2}\right|_{s=j\lambda}. \qquad (13)$$

Proposition 3 *The expectation of Q_Z is given by*

$$\mathrm{E}[Q_Z] = \frac{\mathrm{E}[Z]}{1-\rho}\left(\frac{\mathrm{E}[\sigma]}{2} + \frac{\lambda\mathrm{E}[\sigma^2]}{2(1-\rho)} + \frac{\mathrm{E}[\sigma]}{2}\frac{\mathrm{E}[Z^2]}{\mathrm{E}[Z]}\right). \qquad (14)$$

The derivation of Propositions 2 and 3 is detailed in [1]. Last, we compute the expectation of A_w. Recall that there are Z_I customers at the beginning of the warm-up period. We can readily write $A_w = Z_I T_w + \int_0^{T_w} N(t)dt$. Hence,

$$\mathrm{E}[A_w] = \mathrm{E}[Z_I]T_w + \int_0^{T_w}\lambda t dt = \lambda T_w(\mathrm{E}[I] + T_w/2) \qquad (15)$$

where (6) is used to get (15).

3.6. The expected sojourn time

Let T denote the expected system response time or, equivalently, the expected time a customer spends in the queue. It is straightforward to write T using Little's formula $T = \frac{\mathrm{E}[X]}{\lambda}$, where $\mathrm{E}[X]$ is given in (12). After the replacement of the elements of $\mathrm{E}[X]$ with their respective expressions, the expected sojourn time can be rewritten

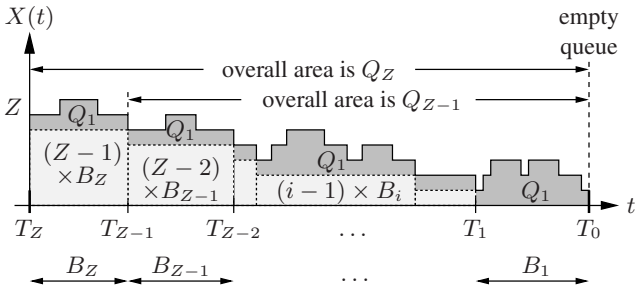


Figure 2. Structure of Q_Z.

$$\begin{aligned} T &= \frac{1/\lambda - \mathrm{E}[\sigma]}{\mathrm{E}[I] + T_w}\mathrm{E}[A] + T_w\frac{\mathrm{E}[I] + T_w/2}{\mathrm{E}[I] + T_w} + \frac{\rho\mathrm{E}[I_a]}{2(\mathrm{E}[I] + T_w)} \\ &+ \mathrm{E}[\sigma] + \frac{\lambda\mathrm{E}[\sigma^2]}{2(1-\rho)} \end{aligned} \qquad (16)$$

where we have used (8), (9), (14) and (15). Observe that the first three terms of (16) are the contribution of the vacation and warm-up periods to the expected sojourn time, whereas the last two terms are the expected sojourn time in the $M/G/1$ queue. 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. This system can be modeled as an $M/G/1$ queue. When the energy saving mechanism is enabled, the server will then go on repeated vacations 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. This will last for a time 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 (in practice $T_w = T_l$).

Let S_i be a generic random variable representing the time for which a node is sleeping during the ith vacation period. We then have $V_1 = S_1$ and $V_i = T_l + S_i$ for $i = 2, \ldots, \zeta$. In this report, 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 3 (resp. 4) maps the state of an $M/G/1$ queue with repeated vacations (resp. an $M/G/1$ queue) to the possible states of a mobile node.

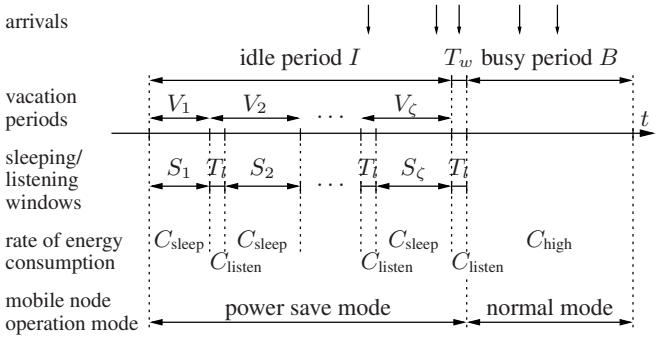


Figure 3. Mapping the $M/G/1$ queue with repeated vacations to the node states.

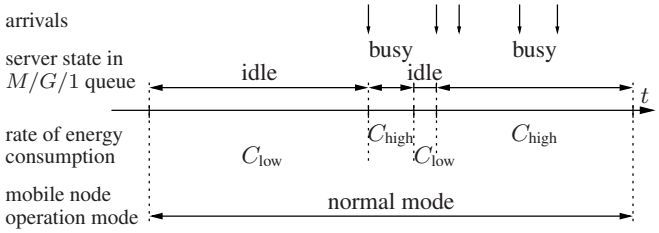


Figure 4. Mapping the $M/G/1$ queue to the normal mode of a mobile node.

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 under power saving.

Having in mind the possible node states, we can distinguish between four possible levels of energy consumption, that are, from highest to lowest: C_{high}, experienced during exchanges of packets; C_{listen}, experienced when checking for downlink packets; C_{low}, when the mobile node is inactive; and C_{sleep}, when the mobile node is in sleep state.

In power save mode, during busy periods (that are on average equal to $\mathrm{E}[B]$), the energy consumption per unit of time is C_{high}. During idle periods, the consumption is C_{listen} in listen windows (whose length is T_l) and is equal to C_{sleep} the rest of the idle period. Observe that there are on average $\mathrm{E}[\zeta]-1$ listen windows in each idle period; see Fig. 3. The energy consumption rate is

$$E_{\text{sleep}} := \frac{\mathrm{E}[B]}{\mathrm{E}[I]+T_w+\mathrm{E}[B]}C_{\text{high}} + \frac{T_l(\mathrm{E}[\zeta]-1)+T_w}{\mathrm{E}[I]+T_w+\mathrm{E}[B]}C_{\text{listen}} + \frac{\mathrm{E}[I]-T_l(\mathrm{E}[\zeta]-1)}{\mathrm{E}[I]+T_w+\mathrm{E}[B]}C_{\text{sleep}}.$$

Observe that $\mathrm{E}[B]/(\mathrm{E}[I]+T_w+\mathrm{E}[B]) = \rho = \lambda\mathrm{E}[\sigma]$ because we have assumed an unlimited queue.

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

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

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 relative economy, or the *energy gain* is defined as

$$G := \frac{E_{\text{no sleep}} - E_{\text{sleep}}}{E_{\text{no sleep}}}.$$

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

$$G = \frac{(1-\rho)\left(\frac{C_{\text{low}}}{C_{\text{high}}} - \frac{T_l\mathrm{E}[\zeta]}{\mathrm{E}[I]+T_l}\frac{C_{\text{listen}}}{C_{\text{high}}}\right)}{\rho+(1-\rho)\frac{C_{\text{low}}}{C_{\text{high}}}}. \qquad (17)$$

The energy consumption rate when the power save mechanism is activated is rewritten

$$E_{\text{sleep}} = C_{\text{high}}\left(\rho + \frac{(1-\rho)T_l\mathrm{E}[\zeta]}{\mathrm{E}[I]+T_l}\frac{C_{\text{listen}}}{C_{\text{high}}}\right).$$

All performance metrics found so far have been derived as functions of: (i) *network* parameters: such as the load ρ, the input rate λ, and moments of the service time ($\mathrm{E}[\sigma]$ and $\mathrm{E}[\sigma^2]$); (ii) *physical* parameters: such as the consumption rates C_{low}, C_{high} and C_{listen}, neglecting C_{sleep}; (iii) *combined* physical and network parameters: such as the listen window T_l and warm-up period T_w; (iv) the LSTs of the vacation periods and their first and second moments. In the following we specify the distribution of the sleep windows $\{S_i\}_i$ and explicitly derive $\{L_i(s)\}_i$, $\{\mathrm{E}[V_i]\}_i$ and $\{\mathrm{E}[V_i^2]\}_i$.

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

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

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

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. Doing so equates the length of all sleep windows. Note 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 μ_i, for $i = 1, 2, \ldots$. Similar to what was done in Sect. 4.2, we let

$$\mathrm{E}[S_i] = \frac{1}{\mu_i} = a^{\min\{i-1,l\}} T_{\min}, \quad i = 1, 2, \ldots \tag{18}$$

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

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

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

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

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. However, when $T_l > 0$, the expected area $\mathrm{E}[A]$ can only be computed numerically, because of the infinite series composed of the second derivatives of the LSTs; see (13).

Scenario E-II
The last case considered in this report 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 (18). Hence $\mu_i = 1/T_{\min}$ for any i. The LSTs of the $\{V_i\}_i$ and their first and second moments simplify to

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

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

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

5. Exploiting the analytical results

Beside performance evaluation, we will use our analytical model to solve a large range of optimization problems. In the following sections we propose a multiobjective formulation of the optimization problem, where the performance objectives are the energy gain and the response time. We formulate the multiobjective problem as a constrained optimization one: the energy gain will be optimized under a constraint on the expected sojourn time.

5.1. Direct optimization

Assume that the traffic parameters information (e.g. the arrival rate) are directly available, or they can be measured or estimated. The objective is to optimize the protocol parameters defined earlier, namely, $T_{\min}$, a, and l. We define the following generic non-linear program:

$$\begin{array}{ll} \text{maximize} & G \\ \text{subject to} & T \leq T_{\text{QoS}} \end{array} \tag{19}$$

where G is given in (17). The program (19) maximizes the energy gain, or equivalently, minimizes the expected energy consumption rate, conditioned on a maximum system response time T_{QoS}. The value of T_{QoS} is application-dependent; it needs to be small for interactive multimedia whereas larger values are acceptable for web traffic.

The decision variables in (19) will be one or more protocol parameters. For a given distribution of the sleep windows $\{S_i\}_i$, the expected number of vacations $\mathrm{E}[\zeta]$, the expected idle period $\mathrm{E}[I]$, and subsequently the gain G will depend on the protocol parameters $T_{\min}$, a and l and on the (fixed) physical parameters C_{low}, C_{high} and C_{listen}.

We propose four types of optimization program (19). 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. The second mathematical program, denoted $\mathcal{P}_2$, has as decision variable the multiplicative factor a whereas $T_{\min}$ and l are given. The decision variable of the third program, denoted $\mathcal{P}_3$, is the exponent l. The parameters $T_{\min}$ and a are given. 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) exponential sleep windows $\{S_i\}_i$.

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* economy of energy under power saving. We consider two different constraints on the expected sojourn time corresponding to the cases where the application is sensitive either to the worst case value (hard constraint) or the average value (soft constraint).

Hard constraints
Here, the application is 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 λ. The problem is to find the protocol parameter θ that achieves

$$\begin{array}{l} \max_\theta \sum_\lambda p(\lambda) G(\lambda, \theta) \\ \text{subject to } T(\lambda, \theta) \leq T_{\text{QoS}} \ \forall \lambda. \end{array} \tag{20}$$

Soft constraints
In this 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 θ that achieves

$$\begin{array}{l} \max_\theta \sum_\lambda p(\lambda) G(\lambda, \theta) \\ \text{subject to } \sum_\lambda p(\lambda) T(\lambda, \theta) \leq T_{\text{QoS}}. \end{array} \tag{21}$$

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 θ 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 λ. The problem then is to find θ that achieves

$$\begin{array}{l} \max_\theta \min_\lambda G(\lambda, \theta) \\ \text{subject to } T(\lambda, \theta) \leq T_{\text{QoS}} \ \forall \lambda. \end{array} \tag{22}$$

Observe that the worst possible gain is the one obtained when the traffic input rate tends to $\frac{1}{\mathrm{E}[\sigma]}$. Thus $\min_\lambda G(\lambda, \theta) \approx 0$. Therefore, the above problem is meaningful only for a restricted range of small values of λ for which the worst energy gain is far above 0.

Soft constraints
Similar to problem (21), assume that the application is sensitive to the expected delay, the problem is to find θ that achieves

$$\begin{array}{l} \max_\theta \min_\lambda G(\lambda, \theta) \\ \text{subject to } \sum_\lambda p(\lambda) T(\lambda, \theta) \leq T_{\text{QoS}}. \end{array} \tag{23}$$

Again, the problem is meaningful only when λ is small.

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 (16) and the expected energy gain G given in (17); cf. Sect. 6.1. In addition we have solved Problems $\mathcal{P}_1$–$\mathcal{P}_4$ for given values of the protocol parameters; cf. Sect. 6.2. Instances of the problems (20), (21), (22) and (23) are provided in Sect. 6.3.

Physical and network parameters have been selected as follows: $C_{\text{low}}/C_{\text{high}} = C_{\text{listen}}/C_{\text{high}} = 0.2, T_w = T_l = 1, \mathrm{E}[\sigma] = 1, \mathrm{E}[\sigma^2] = 2$, and $T_{\text{QoS}} = 50, 100$. Unless otherwise specified, the protocol parameters are set to the *default* values: $T_{\min} = 2$, $a = 2$ and $l = 9$ in scenarios D-I and E-I, and $T_{\min} = 2$, $a = 1$ and $l = 0$ in scenarios D-II and E-II. We have varied λ in the interval $(0, 1)$, $T_{\min}$ in $(1, 100)$, a in $(1, 10)$, and let l take integer values in $(0, 10)$.

6.1. Performance evaluation

Numerical evaluation for the expected sojourn time T and the expected energy gain have been depicted graphically in Figs. 5-8 in all four scenarios when sleep windows are deterministic and exponentially distributed.

To study the impact of $T_{\min}$, we set $a = 2$ and $l = 9$ in scenarios D-I and E-I. Figure 5 depicts T and G against λ and $T_{\min}$. The size of the initial sleep window has a large impact on T for any value of λ. More precisely, T increases linearly with an increasing $T_{\min}$ for any λ; see Figs. 5(a), 5(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. 5(c) and 5(d).

We set $a = 1$ and $l = 0$ in scenarios D-II and E-II. Figure 6 depicts T and G against λ and $T_{\min}$. 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. 6(c). Observe that a larger $T_{\min}$ yields a larger sleep time but it also reduces $\mathrm{E}[\zeta]$ which together explains why the impact on the energy gain is not significant.

To study the impact of the multiplication factor a, we set $T_{\min} = 2$ and $l = 9$ in scenarios D-I and E-I. Figure 7 depicts T and G against λ and a. 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 λ which is reflected in Figs. 7(a) and 7(b).

Finally, to study the impact of the exponent l, we set $T_{\min} = 2$ and $a = 2$ in scenarios D-I and E-I. Figure 8 depicts T and G against λ and l. 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 λ. Observe in Fig. 8(a) that T becomes almost insensitive to l beyond $l = 7$ (for small λ). 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 $T_{\min}$ is larger. A similar behavior is observed in the exponential case for higher T; cf. Fig. 8(b).

Notice that, in D-I, E-I and E-II, as λ increases, T first decreases rapidly then becomes fairly insensitive to λ up to a 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 λ increases, the first component decreases while the second one increases. For moderate values of λ, both components balance each other yielding a fairly insensitive sojourn time. The large value of T at small λ is mainly due to the ratio $\mathrm{E}[I_a]/\mathrm{E}[I]$ (recall (16)), whereas the abrupt increase in T at large λ is due to the term $\frac{\lambda \mathrm{E}[\sigma^2]}{2(1-\rho)}$, which is the waiting time in the $M/G/1$ queue without vacations.

The situation in scenario D-II is different in that T is not large at small input rates λ. Recall that in this scenario, all $\{S_i\}_i$ 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 λ. As already mentioned, G is insensitive to l and a for any λ, and sensitive to $T_{\min}$ up to a certain initial sleep window size. The expected energy gain G decreases monotonically as λ increases which can be explained as follows. The larger the input traffic rate λ, 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: (i) $\mathcal{P}_1$ for $T^*_{\min}$ when $a = 2$ and $l = 9$ (default values) with $T_{\mathrm{QoS}} = 50$ for D-I and $T_{\mathrm{QoS}} = 100$ for E-I, and when $a = 1$ or $l = 0$ with $T_{\mathrm{QoS}} = 50$ for D-II and $T_{\mathrm{QoS}} = 100$ for E-II; (ii) $\mathcal{P}_2$ for a^* with $T_{\min} = 2$ and $l = 9$ (default values) with $T_{\mathrm{QoS}} = 50$ for D-I and $T_{\mathrm{QoS}} = 100$ for E-I; (iii) $\mathcal{P}_3$ for l^* when $T_{\min} = 2$ and $a = 2$ (default values) with $T_{\mathrm{QoS}} = 50$ for D-I and $T_{\mathrm{QoS}} = 100$ for E-I; (iv) $\mathcal{P}_4$ for $(T_{\min}, a, l)^*$ with $T_{\mathrm{QoS}} = 50$ for deterministic $\{S_i\}_i$ and $T_{\mathrm{QoS}} = 100$ for exponential $\{S_i\}_i$.

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. 9 against the input rate λ, for deterministic (Figs. 9(a) and 9(b)) and exponential (Figs. 9(c) and 9(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 ($T_{\min} = 2$).

The most relevant observation to be made on each of Figs. 9(a) and 9(c) is the match between the curves labeled "optimal gain" (result of $\mathcal{P}_4$) and "gain with $T^*_{\min}$" (result of $\mathcal{P}_1$). 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. This match can be explained as follows. The program $\mathcal{P}_1$ is being solved for the optimal $T_{\min}$. It thus quickly reduces the number of vacations $\mathrm{E}[\zeta]$ to 1 (cf. Fig. 10) 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 achieved by $\mathcal{P}_4$.

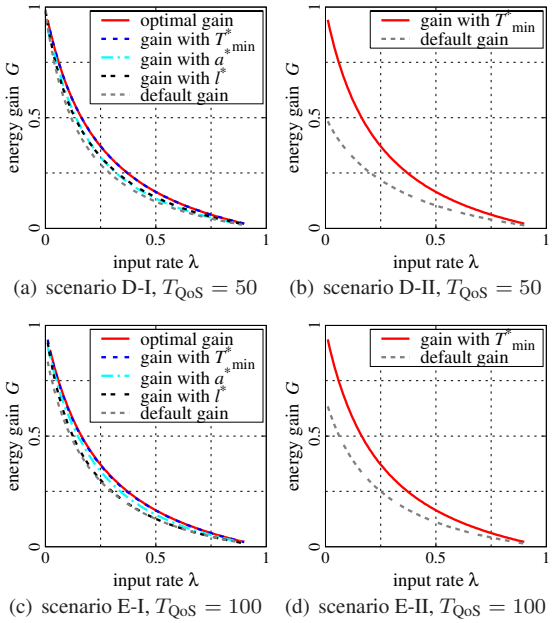


Figure 9. Gain versus the input rate λ.

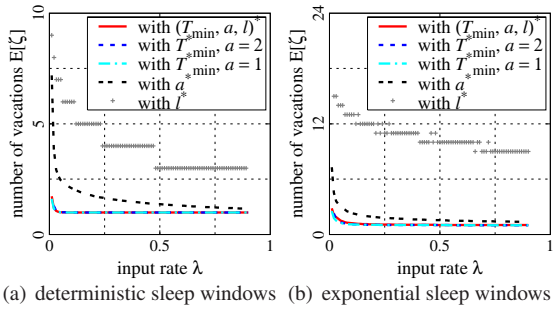


Figure 10. Expected number of vacations $\mathrm{E}[\zeta]$ versus λ when parameters are optimally set.

The values of the optimal protocol parameters returned by programs $\mathcal{P}_1$–$\mathcal{P}_4$ are given in Table 1. Comparing the optimal values of $T_{\min}$ as returned by programs $\mathcal{P}_1$ and $\mathcal{P}_4$ in the deterministic case (cf. columns 2 and 10 in Table 1), it appears that they are very close to each other, confirming our argument that the single variate $\mathcal{P}_1$ is a very good approximation of the multivariate optimization done in $\mathcal{P}_4$.

When maximizing the gain by optimizing $T_{\min}$ we observe in all scenarios but scenario D-II that, optimally, $T_{\min}$ should first increase with the input rate λ then decrease with increasing λ for large values of λ. 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 λ increases is confirmed only in D-II.

Looking at $\mathrm{E}[\zeta]$, should the optimal value $T^*_{\min}$ be used, it appears that $\mathrm{E}[\zeta]$ decreases asymptotically to 1 as λ increases; see Fig. 10. The reason behind this is the energy consumption during listen windows and warm-up periods. To maximize the energy gain, one could minimize the factor multiplying C_{sleep}, in other words minimize $\mathrm{E}[\zeta]$. As a

Table 1. Optimal values of the protocol parameters from programs $\mathcal{P}_1$–$\mathcal{P}_4$

	$T^*_{\min}$ from $\mathcal{P}_1$				a^* from $\mathcal{P}_2$		l^* from $\mathcal{P}_3$		$\mathcal{P}_4$, deterministic case			$\mathcal{P}_4$, exponential case		
λ	D-I	D-II	E-I	E-II	D-I	E-I	D-I	E-I	$T_{\min}$	a	l	$T_{\min}$	a	l
0.02	64	96	20	62	1.5	2.0	15	8	72	1.5	1	27	2.5	2
0.05	94	96	50	74	4.5	3.0	14	6	92	2.0	3	32	1.5	3
0.10	96	96	76	96	5.0	4.0	13	6	92	5.0	1	42	1.5	1
0.20	96	96	92	100	5.0	5.0	12	5	92	5.0	1	47	1.5	9
0.40	94	94	94	100	5.0	5.0	11	4	92	1.5	1	47	1.5	8
0.60	94	94	94	98	5.0	5.0	10	3	92	1.5	1	47	1.5	7
0.90	78	78	92	94	5.0	5.0	9	3	77	1.5	1	42	2.0	6

Table 2. Distribution of the input rate λ

λ	0.02	0.05	0.1	0.2	0.5
$p(\lambda)$	0.3125	0.3125	0.1875	0.1250	0.0625

Table 3. Optimal value of $T_{\min}$ (in frames)

	Expectation analysis		Worst-case analysis	
	hard constraint	soft constraint	hard constraint	soft constraint
D-I	65	92	64	64
D-II	96	97	94	94
E-I	22	50	21	21
E-II	69	79	62	62

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.

6.3. Expectation and worst case analysis

In this section, we report the results of an expectation and a worst case analysis, considering the expected energy gain as performance metric. We will solve the problems stated in (20), (21), (22) and (23). The decision variable is the initial sleep window size $T_{\min}$. Each problem is solved for each of the four scenarios defined in Sects. 4.2 and 4.3. We consider $a = 2$ and $l = 9$ in scenarios D-I and E-I. Recall that we necessarily have $a = 1$ and $l = 0$ in scenarios D-II and E-II. We consider that λ may take five different values. These values and the corresponding probabilities $p(\lambda)$ are given in Table 2. The values of the parameter $T_{\min}$ found for each of the problems are reported in Table 3.

7. Conclusion

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 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.

References

[1] S. Alouf, E. Altman, and A. P. Azad. Analysis of an M/G/1 queue with repeated inhomogeneous vacations – application to IEEE 802.16e power saving. Research Report RR-6488, INRIA, 2008. `http://hal.inria.fr/inria-00266552`.

[2] B. T. Doshi. Queueing systems with vacations - a survey. *Queueing Systems*, 1(1):29–66, 1986.

[3] S. W. Fuhrmann and R. B. Cooper. Stochastic decomposition in the M/G/1 queue with generalized vacation. *Operations Research*, 33(5):1117–1129, September-October 1985.

[4] 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*, volume 3, pages 1141–1145, Melbourne, Australia, May 2006.

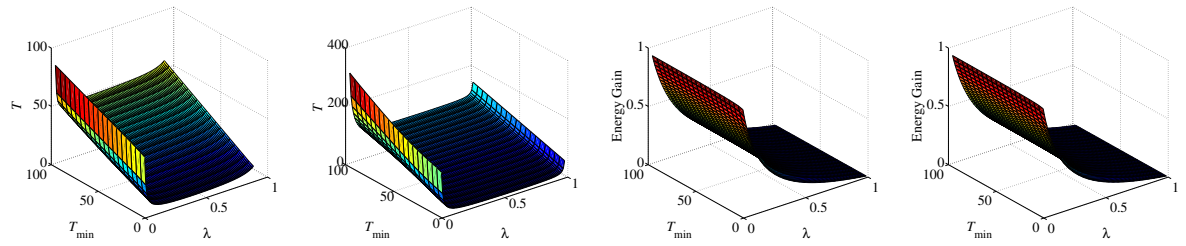
[5] IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems. *IEEE Std 802.16e-2005 and IEEE Std 802.16-2004/Cor 1-2005 (Amendment and Corrigendum to IEEE Std 802.16-2004)*, 2006.

[6] J. Keilson and L. D. Servi. A distribution form of Little's law. *Operations Research Letters*, 7(5):223–227, 1983.

[7] 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*, volume 2, pages 1169–1173, Los Angeles, California, USA, September 2004.

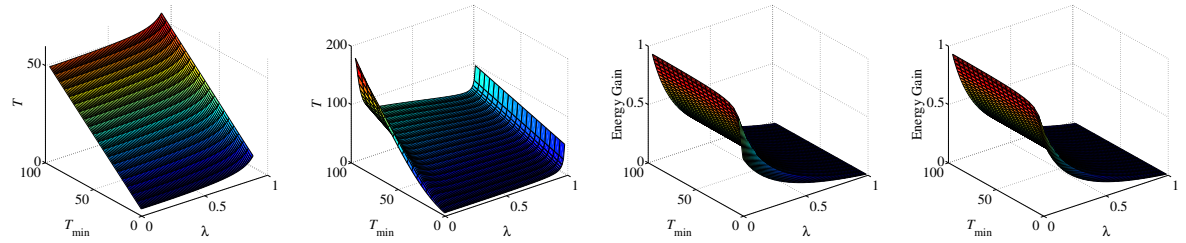
[8] Y. Xiao. Energy saving mechanism in the 802.16e wireless MAN. *IEEE Commun. Lett.*, 9(7):595–597, July 2005.

[9] Y. Xiao. Performance analysis of an energy saving mechanism in the IEEE 802.16e wireless MAN. In *Proc. of IEEE CCNC 2006*, volume 1, pages 406–410, January 2006.



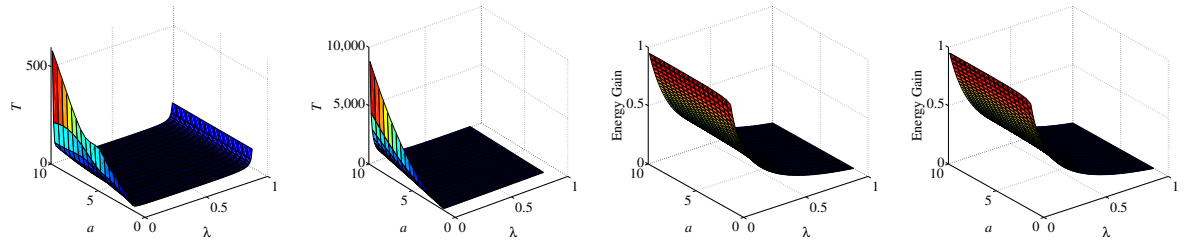
(a) sojourn time, deterministic S_i (b) sojourn time, exponential S_i (c) energy gain, deterministic S_i (d) energy gain, exponential S_i

Figure 5. Impact of $T_{\min}$ on T and G in type I like power saving classes.



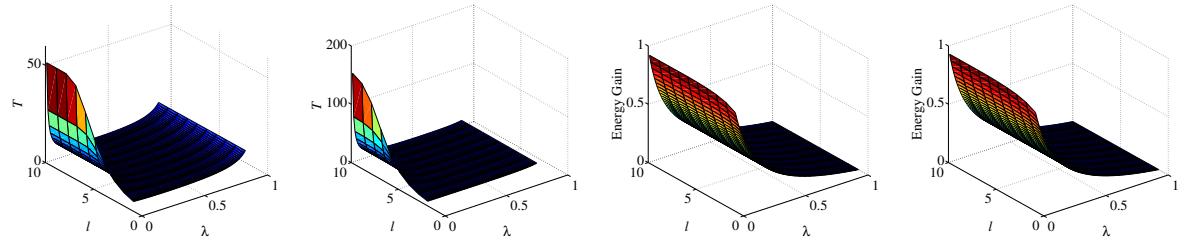
(a) sojourn time, deterministic S_i (b) sojourn time, exponential S_i (c) energy gain, deterministic S_i (d) energy gain, exponential S_i

Figure 6. Impact of $T_{\min}$ on T and G in type II like power saving classes.



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

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



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

Figure 8. Impact of l on T and G with either deterministic or exponential $\{S_i\}_i$.