Optimal control of sleep periods for wireless terminals
Amar Prakash Azad, Sara Alouf, Eitan Altman, V. S. Borkar, Georgios Stavrou Paschos

To cite this version:

HAL Id: inria-00591761
https://inria.hal.science/inria-00591761
Submitted on 10 May 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Optimal Control of Sleep periods for Wireless terminals

Amar Prakash Azad*, Sara Alouf*, Eitan Altman*, Vivek Borkar† and Georgios S. Paschos‡

* joint LAB of Alcatel-Lucent and INRIA, BP95, Sophia-Antipolis Cedex, France, email:{altman.salouf}@sophia.inria.fr, amar.azad@ieee.org
† School of Technology, TIFR, Mumbai, email: borkar@tifr.res.in
‡ ECE, University of Thessaly, Volos, Greece, email: gpasxos@uth.gr

Abstract—We consider a mobile connected to a base station, saving energy by shutting off its transceiver, and we try to answer the fundamental question: what is the optimal sleep policy? Firstly, we study the model from optimal control perspective. We consider off-times (periods of inactivity) of unknown duration. We study the question of scheduling “waking up” instants in which the mobile communicates with the base station and checks whether the inactivity period is over. There is a cost proportional to the delay from the moment the off-time ends until the mobile discovers it, a (small) running cost while the mobile is sleeping and also a cost for waking up.

We show that constant sleep periods are optimal for Poisson arrivals and derive the optimal period. We show that this structure does not hold for other off-time distributions but manage to obtain suboptimal solutions which perform strictly better than the constant one. We finally obtain structural properties for optimal policies for the case of arbitrary distribution of off-times.

Technological restrictions often permit a limited set of policies to be implemented. Motivated by this, we investigate classes of policies with specific constraints. What is the optimal policy within each class and what are the optimal parameters for it? To answer these questions, we adopt the parametric optimization approach which entails cost minimization for a given parameterized policy and selection of the best policy among a class. We provide the optimal solution for each class which can be used in closed form or evaluated numerically depending on the case. Our framework allows us to compare the performance of obtained optimal policies, proposed suboptimal policies as well as that of standard policies like IEEE 802.16e.

Index Terms—Dynamic programming, optimization, performance evaluation, WiMAX.

I. INTRODUCTION

Mobile terminals using contemporary radios consume a significant amount of energy, while being idle, by using unnecessary compartments of their electrical circuits. Therefore, one way to reduce the consumed energy and increase battery life, is to shut off the transceiver totally, i.e. put the mobile terminal into sleep mode. This approach has been recently supported by novel protocols like IEEE 802.16e (see [1]) and 3GPP LTE (see [2]) where a general framework for sleep is defined. Since these standards allow for vendor design of the sleep policy, optimal design of such policies is an open issue of great interest.

Nevertheless, shutting off the transceiver whenever there is no scheduled activity has side effects on the responsiveness of the terminal. If the attention of the mobile is suddenly required, the transceiver might be shut off and thus the mobile unavailable. The longer the sleep periods, the longer the expected response delay. Therefore, one can identify an inherent tradeoff of energy management: increase sleep period length to improve energy saving or decrease sleep period length to reduce delays. Careful scheduling of sleep periods is then needed in order to minimize energy consumption while keeping the delays small.

A. Related Work

Since the initial announcement of IEEE 802.16e Standards for mobility [1], there has been an important volume of performance studies on the subject. The first approach chronologically is found in [3]. In an effort to relax some assumptions, [4]–[6] study the impact of outgoing traffic, [7], [8] study the effect of setup time while [9]–[11] deal with queueing implications in the analysis. [12] deals with the multiclass version of sleep mode in IEEE 802.16e. The impact of bidirectional traffic on sleep mode is studied in [13] and correlated traffic is studied in [14] which is applied to IEEE 802.16m.

The above models assume a Poisson process for the packet arrivals. The Poisson modeling is rationalized by the fact that the activity requests are generated by a potentially very large population of sources. In [15], the authors are using hyper-Erlang distribution for the packet interarrival period. In [16], [17], hyper-exponential arrivals are proposed. In any of the above cases, an exogenous arrival process that does not depend on the energy management scheme is considered. Moreover, the delay metric taken is the average packet delay in the system.

Regarding the process of arriving packets, there are other works that provide evidence of heavy-tailed off-time distributions on the Internet and on the World Wide Web (e.g., a Pareto type distribution). In [18] the operator’s idle periods are found to be heavy-tailed. As heavy-tailed distributed random variables can be well approximated by hyper-exponential distributions [16], [19], this gives motivation to study off-times with hyper-exponential distributions. In [20] we modeled the arrival process as an hyper-exponential process to investigate sleep mode.

Recent works [7], [21], [22] focus on heuristic adaptive algorithms whose goal is to control the sleep period length according to the incoming arrival process. The work [23] derives an optimal sleep policy using average cost structure...
for a given number of consecutive sleep durations. In [24],
we derive the optimal sleep policy in a given class of policies.

Our work departs from the existing models in the following aspect: rather than assuming an exogenous independent arrival
process, we have in mind elastic arrival processes in which (i) the
duration of the activity period does not depend on the
response delay—defined as the duration between the instant a
request is issued and the instant at which the service actually
begins—and (ii) the off-time begins when the activity of the
mobile ends. Both assumptions are appropriate to interactive
applications such as web browsing, see [25] for a coupling
between traffic models and actual applications. As a result,
the measure for delay is taken to be the mobile’s response
delay to the oldest activity request taking place while in sleep
mode.

B. Contribution

In this section we clarify our contribution. Our objective is
to obtain the optimal sleep policy; the policy which minimizes
the energy consumption and the system response delay simultaneuously, while maintaining the desired balance between the
two (the balance is expressed via a weight). For the off-time
period distribution we will consider a. Exponential distribution;
b. Hyper-exponential distribution; c. General distribution.

Off-times of exponentially distributed length correspond to
the Poisson arrivals case. Also, we shall consider the case
when the parameter of the exponential distribution is unknown
but we have a known prior distribution on that parameter.
This is equivalent to using a hyper-exponential distribution
for the off-time. Modeling the hyper-exponential distribution
for off-times will offer insight to the case of heavy-tailed
distributions as well. Lastly, we will provide structural results
for the optimal policy in case of general distributions.

In the above framework we will seek optimal sleep policies.
Also, technological restrictions often allow only certain types
of policies to be implemented. Motivated from such scenarios,
we constrain the set of policies to those fulfilling certain
conditions and seek the optimal policies within this set.
Furthermore, we study the performance of the IEEE 802.16e
standard mechanism. In particular, these standards provide
degrees of freedom for the vendor which we utilize to optimize
the performance. We show, that in many cases, the proposed
standards are suboptimal in the sense that even if the best
parameters are selected, the optimal performance cannot be
achieved.

Our contributions are the following:
1. We formulate the problem of cost minimization where the
cost is a weighted function of energy consumption and
response delay measured from the first activity request
when inactive. We show that this cost indeed depends
on the off-time distribution as well as the selected sleep
policy.
2. We use dynamic programming (DP) and show that:
   - For exponential off-times, the constant sleep period is
     optimal and given in closed form.
   - For hyper-exponential off-times, interesting structural
     properties exist. In particular, we show that the optimal
     policy has sleep periods of bounded length. Asymptoti-
     cally, the optimal policy converges to the constant policy
     corresponding to the smallest rate phase, irrespective of
     the initial state. This optimal policy can be computed
     numerically using value iteration.
   - For any general off-time distribution, the optimal policy
     has sleep periods of bounded length.
3. We propose suboptimal policies using policy iteration
   which perform strictly better than optimal “homoge-
   neous” policies and are simpler to compute. We show
   numerically the performance of such suboptimal solutions
   using one stage and two stage policy iteration.
4. We use parametric optimization to identify optimal pa-
   rameters for the following family of sleep policies: (i) Random exponential periods; they can be derived in
   closed form for any off-time distribution. (ii) Constant
   periods. (iii) Scaled and General random periods. (iv)
   Semi-constant periods. (v) Multiplicative periods; similar
to those used in WiMAX. (vi) General deterministic
   periods;
5. We compare the proposed policies with that of the IEEE
   802.16e standard [1] under various statistical assump-
   tions.

C. Structure

The rest of the paper is structured as follows: Section II
outlines our system model, introduces the cost function and
presents some preliminaries calculations for the rest of the
paper. Section III introduces DP and derives the optimal sleep
control and relevant characteristics for hyper-exponential off-
times. Section III-D tackles the problem of finding the optimal
policy under the worst case process of arrivals. In IV, the
parametric optimization preliminaries are laid out. Section
IV-A investigates policies with identically distributed sleep
periods while those with non-identically distributed ones are
 tackled in Section IV-B. Numerical results and a comparative
study of the different optimal policies and of the IEEE 802.16e
standards are reported in Section V. We last conclude the paper
in Section VI.

II. SYSTEM MODEL

We consider a mobile terminal connected to a base station.
The terminal goes through consecutive active and inactive
periods. Focusing on periods of inactivity, the terminal goes
through a sequence of sleep and listen periods until an incom-
ing activity request is detected, see Fig. 1. In particular, in the
beginning of each sleep period, the device chooses the sleep
period length (also called sleep mode window in the IEEE
802.16e standards) while the listen period length window is
considered

 negligible. At the end of the sleep period, the terminal
communicates with the base station to check for activity
requests. In case there is no incoming activity waiting at
the base station, a fixed energy cost is incurred for checking
the system state. In case at least one activity has arrived,
the inactivity period is finished and a delay cost is incurred
depending on the waiting time of the first activity request (the
An equivalent modeling of the system is one that considers a server that goes on repeated vacations, see [10]. The incoming traffic load is replaced by customers waiting to be served and the service period is a fixed parameter, which makes the energy consumed during sleep periods instead.

In this paper we will use the same notation as [10] but refer to sleep periods instead.

Let $X$ denote the number of sleep periods in an inactivity period. $X$ is a discrete random variable (rv) taking values in $\mathbb{N}^*$. The duration of the $k$th sleep period is a rv denoted $B_k$, for $k \in \mathbb{N}^*$. For analytical tractability, we consider periods $\{B_k\}_{k \in \mathbb{N}^*}$ that are mutually independent rvs. The time at the end of the $k$th sleep period is a rv denoted $T_k$, for $k \in \mathbb{N}^*$. We denote $T_0$ as the time at the beginning of the first sleep period; by convention $T_0 = 0$. We naturally have $T_k = T_{k-1} + B_k = \sum_{i=1}^{k} B_i$. Observe that a generic inactivity period ends at time $T_X$.

We will be using the following notation $\forall s := \mathbb{E}\left[\exp(-s Y)\right]$ to denote the Laplace-Stieltjes transform of a generic rv $Y$ evaluated at $s$. Hence, we can readily write $T_k(s) = \prod_{i=1}^{k} B_i(s)$.

Let $\tau$ denote the time length between the start of the first sleep period and the arrival of an activity request; this time is referred to as the “off-time”. Since a generic inactivity period ends at time $T_X$, the service of the first activity request to arrive during the inactivity period will be delayed for $T_X - \tau$ units of time. $\tau$ is a rv whose probability density function is $f_\tau(t)$, $t \geq 0$. In most of the cases, and unless otherwise stated, we will be assuming that $\tau$ is hyper-exponentially distributed with $n$ phases and parameters $\lambda = (\lambda_1, \ldots, \lambda_n)$ and $\eta = (q_1, \ldots, q_n)$. In other words, we have

$$f_\tau(t) = \sum_{i=1}^{n} q_i \lambda_i \exp(-\lambda_i t), \quad \sum_{i=1}^{n} q_i = 1. \tag{1}$$

Given its definition, the off-time $\tau$ is also the conditional residual inter-arrival time. Observe that when $n = 1$, $\tau$ will be exponentially distributed with parameter $\lambda = \lambda_1$, which, thanks to the memoryless property of this distribution, is equivalent to having a Poisson arrival process with rate $\lambda$.

The power required by a mobile to keep the transceiver on while listening to the channel and checking for requests (packets or flows) is denoted $P_L$. We assume that the listen period is a fixed parameter, which makes the energy consumed during listen periods a fixed quantity, name it $E_L$. This is actually a penalty paid at the end of each sleep period. Instead, while sleeping (the device is turned off), the power consumed is denoted by $P_S$, with $P_S < P_L$. The energy consumed by a mobile during sleep period $B_k$ is then equal to $E_L + P_SB_k$, and that consumed during a generic inactivity period is equal to $E_LX + P_S T_X$. The device is then eager to use longer sleep periods in order to save energy and extend battery lifetime.

In the spirit of achieving a Quality of Service (QoS) tradeoff, we are interested in finding the optimal policy that minimizes energy consumption and delay. Formally, we are interested in minimizing the cost of the power save mode, which is seen as a weighted sum of the energy consumed during the power save mode and the extra delay incurred on the traffic by a sleeping mobile. Let $V$ be this cost; it is written as follows

$$V := \mathbb{E}\left[\epsilon (T_X - \tau) + \epsilon (E_LX + P_SB_L)\right] = -\epsilon \mathbb{E}[\tau] + \epsilon E_L \mathbb{E}[X] + \eta \mathbb{E}[T_X] \tag{2}$$

where $\epsilon$ is a normalized weight that takes value between 0 and 1; $\bar{\epsilon} = 1 - \epsilon$; $\eta := \bar{\epsilon} + \epsilon P_S$; and the expectation is taken over the random off-time $\tau$ as well as the randomness of selected sleep periods (if they are selected at random).

The derivation of the elements of (3) when $\tau$ is hyper-exponentially distributed is straightforward. We derive

$$P(X = k) = P(\tau > T_{k-1}) - P(\tau > T_k) = \sum_{i=k}^{n} q_i T_{k-1}(\lambda_i) (1 - B_k(\lambda_i)),$$

$$\mathbb{E}[\tau] = \sum_{i=1}^{n} q_i / \lambda_i,$$

$$\mathbb{E}[X] = \sum_{k=0}^{n} \sum_{i=1}^{n} q_i T_k(\lambda_i),$$

$$\mathbb{E}[T_X] = \sum_{k=0}^{n} \sum_{i=1}^{n} q_i T_k(\lambda_i) \mathbb{E}[B_{k+1}]. \tag{4}$$

Using (3)-(5), the cost can be rewritten

$$V = -\bar{\epsilon} \mathbb{E}[\tau] + \sum_{k=0}^{n} \sum_{i=1}^{n} q_i T_k(\lambda_i) (\epsilon E_L + \eta \mathbb{E}[B_{k+1}]). \tag{6}$$

Equation (6) is very interesting in that the off-time $\tau$ appears only through its expectation. Given that the cost depends only on $\mathbb{E}[\tau]$, so will the minimal cost and the optimal control. This is true for any distribution of the sleep periods.

For convenience, we have grouped the major notation used in the paper in Table II.

III. DYNAMIC PROGRAMMING

Dynamic programming (DP) is a well-known tool which allows to compute the optimal decision policy to be taken at each intermediate observation point, taking into account the whole lifetime of the system. Considering our system model, we want to identify the optimal sleep policy where decisions are taken at each intermediate wake-up instance. Hence, a DP

1Given that our objective is to study energy saving, the value $X = 0$ is not allowed.
approach is a natural candidate for determining the optimal policy.

The observation points are at the end of the sleep periods, i.e., at time \(t_k\). The conditional residual off-time at a time \(t\) is denoted \(\tau_t\). We introduce the following DP:

\[
V^*_k(t_k) = \min_{b_{k+1} \geq 0} \left\{ \mathbb{E}[c(t_k, b_{k+1})] + P(\tau_{t_k} > b_{k+1})V^*_k(t_{k+1}) \right\}.
\]

Here, \(V^*_k(t_k)\) represents the optimal cost at time \(t_k\) where the argument \(t_k\) denotes the state of the system at time \(t_k\). The terms \(P(\tau_{t_k} > b_{k+1})\) and \(c(t_k, b_{k+1})\) respectively represent the transition probability and the stage cost at \(t_k\) when the control is \(b_{k+1}\). In generic notation, the per stage cost is

\[
c(t, b) = \bar{c} \mathbb{E}[(t - \tau(t))] + \epsilon (E_L + P S).
\]

We can see that each stage is characterized by the distribution of the residual off-time \(\tau_t\). The state of the system in sleep mode can then be described by the distribution of \(\tau_t\).

In the rest of this section, three cases will be considered following the distribution of the off-time. We start with the DP solution for exponential off-times, then derive some structural properties of the DP solution for hyper-exponential off-times. Last, the case of general off-times is considered: structural properties of the optimal policy are found and then suboptimal solutions through DP are discussed.

### A. Exponential Off-Time

By setting \(n = 1\) in (1) we study the particular case of Poisson arrivals with rate \(\lambda\) (in the next subsection we will study the case of arbitrary \(n\)). In such a case, both the off-time \(\tau\) and the conditional residual off-time \(\tau_t\) will be exponentially distributed with parameter \(\lambda\), for all \(t\) (i.e., for any stage).

The distribution of \(\tau_t\) is characterized solely by the rate \(\lambda\). In other words, as time goes on, the state of the system is always represented by the parameter \(\lambda\). Henceforth, the DP involves a single state, denoted \(\lambda\).

We are faced with a Markov Decision Process (MDP), a single state \(\lambda\), a Borel action space (the positive real numbers) and discrete time. Note that the sleep durations are not discrete. However, decisions are taken at discrete embedded times: the \(k\)th decision is taken at the end of the \((k-1)\)st sleep period. Therefore, we are dealing with a discrete time MDP. This is called “negative” dynamic programming [26]. It follows from [27] that we can restrict to stationary policies (that depend only on the state) and that do not require randomization. Since there is only one state (at which decisions are taken) this implies that one can restrict to sleep periods that have fixed size and that are the same each time a decision has to be taken. In other words, the optimal sleep policy is the constant one. Hence the optimal value is given by the minimization of the following MDP:

\[
V^*(\lambda) = \min_{b \geq 0} \left\{ \epsilon \mathbb{E}[(b - \tau(\lambda))] + \epsilon (E_L + P S) + P(\tau(\lambda) > b)V^*(\lambda) \right\}.
\]

#### 1) Distribution of the Conditional Residual off-time \(\tau_t\):

The tail of \(\tau_t\) can be computed as follows

\[
P(\tau_t > a) = \frac{P(\tau > t + a \mid \tau > t)}{P(\tau > t)} = \frac{\sum_{j=1}^{n} q_i \exp(-\lambda_j t)}{\sum_{j=1}^{n} q_j \exp(-\lambda_j t)}
\]

where

\[
g_i(q, t) := \frac{q_i \exp(-\lambda_i t)}{\sum_{j=1}^{n} q_j \exp(-\lambda_j t)}, \quad i = 1, \ldots, n.
\]

The Lambert \(W\) function, satisfies \(W(x) \exp(W(x)) = x\). As \(y \exp(y) = x\) has an infinite number of solutions \(y\) for each (non-zero) value of \(x\), the function \(W(x)\) has an infinite number of branches.

### Table I

<table>
<thead>
<tr>
<th>(X)</th>
<th>Number of sleep periods</th>
</tr>
</thead>
<tbody>
<tr>
<td>(B_k)</td>
<td>Duration of (k)th sleep period</td>
</tr>
<tr>
<td>(T_k)</td>
<td>Elapsed time until (k)th sleep period, (T_k = \sum_{i=1}^{k} B_i)</td>
</tr>
<tr>
<td>(T_0)</td>
<td>Starting time of power save mode, (T_0 = 0)</td>
</tr>
<tr>
<td>(E_L)</td>
<td>Energy consumed in the listen period</td>
</tr>
<tr>
<td>(P_S)</td>
<td>Power consumed by a mobile in a sleep state</td>
</tr>
<tr>
<td>(c, \bar{c})</td>
<td>Normalized energy/delay weight, (0 &lt; \bar{c} \leq 1), (\bar{c} = 1 - \epsilon)</td>
</tr>
<tr>
<td>(V)</td>
<td>Cost function</td>
</tr>
<tr>
<td>(c(t, b))</td>
<td>Cost incurred by sleep period of size (b) having started at time (t)</td>
</tr>
<tr>
<td>(W_{-1})</td>
<td>Branch of the Lambert (W) function that is real-valued on the interval ([\exp(-1), 0]) and always below (-1)</td>
</tr>
<tr>
<td>(\lambda, q)</td>
<td>Rate/probability vector in the (n)-phase hyper-exponential distribution, (\lambda = (\lambda_1, \ldots, \lambda_n)), (q = (q_1, \ldots, q_n))</td>
</tr>
<tr>
<td>(\tau)</td>
<td>Off-time (i.e., arrival time of first activity request)</td>
</tr>
<tr>
<td>(\tau_t)</td>
<td>Conditional residual off-time (i.e., from (t) to the arrival of first activity request)</td>
</tr>
<tr>
<td>(B)</td>
<td>{(B_k)}_{k \geq 0}, generic sleep policy</td>
</tr>
<tr>
<td>(b)</td>
<td>Parameter of the policy (B)</td>
</tr>
<tr>
<td>(\alpha)</td>
<td>Parameter of the distribution of (B_k) for Scaled policy</td>
</tr>
<tr>
<td>(\mathcal{P})</td>
<td>Distribution of (B_k) for Scaled and General Discrete policies</td>
</tr>
<tr>
<td>(\gamma)</td>
<td>Laplace-Stieltjes transform of a random variable (Y)</td>
</tr>
<tr>
<td>(\eta)</td>
<td>(= \bar{c} + \epsilon P_S), (0 &lt; \eta \leq 1 + \epsilon P_S)</td>
</tr>
<tr>
<td>(\zeta)</td>
<td>(= 1 + \frac{\lambda \bar{E}_L}{\eta}, \quad i = 1, \ldots, n), (\zeta &gt; 1)</td>
</tr>
</tbody>
</table>

#### Proofs

**Proof:** The proof is given in Appendix A.

**Proposition III.1.** The optimal sleep period length for exponential off-time and the minimal cost are given by

\[
b^* = -1 \frac{1}{\lambda} \left( \zeta + W_{-1}(-e^{-\zeta}) \right); \quad V^*(\lambda) = -1 \frac{1}{\lambda} \left( \bar{c} + \epsilon W_{-1}(-e^{-\zeta}) \right),
\]

with \(\zeta := 1 + \lambda \epsilon \bar{E}_L / \eta\), and where \(W_{-1}\) denotes the branch of the Lambert \(W\) function\(^2\) that is real-valued on the interval \([-\exp(-1), 0]\) and always below \(-1\).

---

\(^2\)The Lambert \(W\) function, satisfies \(W(x) \exp(W(x)) = x\). As \(y \exp(y) = x\) has an infinite number of solutions \(y\) for each (non-zero) value of \(x\), the function \(W(x)\) has an infinite number of branches.
We denote $g_i(q, t)$ as the $n$-tuple of functions $g_i(q, t)$, $i = 1, \ldots, n$. Observe that $g(q, 0) = q$. The operator $g$ transforms the distribution $q$ into another distribution $q'$ such that $\sum_{j=1}^{n} q'_j = 1$ and $q'_j > 0$.

Equation (11) is nothing but the tail of a hyper-exponential rv having $n$ phases and parameters $\lambda$ and $g(q, t)$. Except for the probabilities of the $n$ phases, the off-time $\tau$ and its residual time $\tau_i$ have the same distribution and same parameter $\lambda_i$. As time goes on, the residual time keeps its distribution but updates its phases’ probabilities, through the operator $g$. It can be shown that

$$g_i(q, b_1 + b_2) = g_i(g_i(q, b_1), b_2).$$

In other words, the operator $g$ is such that the result of the transformation $b_1 + b_2$ units of time is the same as that of a first transformation after $b_1$ units of time, followed by a second transformation after $b_2$ units of time.

To simplify the notation, we will drop the subscript of the residual off-time $\tau_i$, and instead, we will add as argument the current probability distribution (which is transformed over time through the operator $g$). For instance, if at some point in time, the residual off-time has the probability distribution $q'$, then we will use the notation $\tau(q')$.

The results above can be extended to account for a random passed time $T$. We have

$$P(\tau > T + a \mid \tau > T) = \sum_{i=1}^{n} g_i(q, T) \exp(-\lambda_i a)$$

where

$$g_i(q, T) := \frac{g_i(T(\lambda_i))}{\sum_{j=1}^{n} q_j T(\lambda_j)} = \frac{g_i(T(\lambda_i))}{P(\tau > T)}.$$ (14)

There is an abuse of notation in the definition of $g_i(q, T)$, as this function depends on the distribution of $T$ and not on the rv $T$ itself. The function $g_i(q, T)$ is not a rv. Observe that (12), where time is deterministic, is a particular case of (14).

Asymptotic properties of $g$ are provided next.

Define the composition $g^n(q, b) = g(g^{n-1}(q, b), b) = g(q, mb)$, where $g^i(q, b)$ is the vector whose $i$th element is $1$ and all other elements are zero.

**Lemma III.1.** Fix $q$ and let $I(q)$ be the smallest $j$ for which $q_j > 0$. Then $\lim_{m \rightarrow \infty} g^n(q, b) = e(I(q))$.

**Proof:** The proof is given in Appendix B.

**Lemma III.2.** For any $q$ we have $\lim_{q' \rightarrow q} V(q') = V(q)$.

**Lemma III.3.** For all $q$, $V(q) \leq \pi$.

**Lemma III.4.** Without loss of optimality, one may restrict to policies that take only actions within $[0, \bar{b}]$ where $\bar{b} = (1/\epsilon)(\pi + 1/(\min \lambda_i))$.

**Proof:** Let $u$ be an $\epsilon$-optimal Markov policy that does not use randomization, where $\epsilon \in (0, 1)$. If $u_i > \bar{b}$ for some $i$ then the expected immediate cost at step $i$ is itself larger than 1 plus the total expected cost that would be incurred under the policy $w$: $E[(b - c(q)) I_{c(q) < b}] > \pi + 1$. Thus, by switching from time $i$ onwards to $w$, the expected cost strictly decreases by at least 1 unit; thus $u$ cannot be $\epsilon$-optimal.

We conclude that the MDP can be viewed as one with a countable state space, compact action space, discrete time, and non-negative costs (known as “negative dynamic programming”). Using [26] we then conclude:

(i) The optimal value (minimal cost) is given by the mini-
mal solution of the following DP:
\[
V(q) = \min_{b \geq 0} \left\{ \epsilon E \left[ (b - \tau(q)) \mathbb{I}_{\tau(q) \leq b} \right] + \epsilon (E_L + bP_S) + P(\tau(q) > b)V(g(q, b)) \right\}.
\]  
(15)

(ii) Let \( B(q) \) denote the set of all \( b \)'s that minimize the right hand side of (15) for a given \( q \). Then any policy that chooses at state \( q \) some \( b \in B(q) \) is optimal.

The value iteration can be used as an iterative method to compute \( V(q) \). Starting with \( V_0 = 0 \) we write
\[
V_{k+1}(q) = \min_{b \geq 0} \left\{ \epsilon E \left[ (b - \tau(q)) \mathbb{I}_{\tau(q) \leq b} \right] + \epsilon (E_L + bP_S) + P(\tau(q) > b)V_k(g(q, b)) \right\}.
\]

Then \( V(q) = \lim_{k \to \infty} V_k(q) \), see [28]. The iteration is to be performed for every possible state \( q \). Lemma III.1 implies that the moving state, \( g(q, b) \), converges asymptotically to \( \epsilon(I(q)) \).

To complete the value iteration, we compute, for a fixed \( b \),
\[
E \left[ (b - \tau(q)) \mathbb{I}_{\tau(q) \leq b} \right] = b - \sum_{i=1}^{n} \left( \frac{1 - \exp(-\lambda_i b)}{\lambda_i} \right)
\]

C. General Distribution of Off-Time

In this section, off-times have a general distribution. As a consequence, one can no longer expect that the residual off-time will keep the same distribution over time, updating only its parameters. Therefore, the system state is the instant \( t \) at which a sleep period is to start. We use again \( \tau > t \) to denote the conditional residual value of \( \tau \) at time \( t \) (i.e., \( \tau - t \) given that \( \tau > t \).

As a state space, we consider the set of non-negative real numbers. An action \( b \) is the duration of the next sleep period. We shall assume that \( b \) can take value in a finite set. The set of \( t \) reachable (with positive probability) by some policy is countable. We can thus assume without loss of generality that the state space is discrete. Then the following holds:

Proposition III.2.

(i) There exists an optimal deterministic stationary policy.

(ii) Let \( V^0 = 0 \), \( V^{k+1} := \mathcal{L}V^k \), where
\[
\mathcal{L}V(t) := \min_b \{ c(t, b) + P(\tau_t > b)V(t + b) \}
\]
where \( c(t, b) \) has been defined in (7). Then \( V^k \) converges monotonically to the optimal value \( V^* \).

(iii) \( V^* \) is the smallest nonnegative solution of \( V^* = \mathcal{L}V^* \). A stationary policy that chooses at state \( t \) an action that achieves the minimum of \( \mathcal{L}V^* \) is optimal.

Proof: (i) follows from [26, Thm 7.3.6], and (ii) from [26, Thm 7.3.10]. Part (iii) is due to [26, Thm 7.3.3].

Observe that \( V^k \) expresses the optimal cost for an action of minimizing the total cost over a horizon of \( k \) steps.

Proposition III.3. Assume that \( \tau_t \) converges in distribution to some limit \( \tau \). Define \( v(b) := \mathcal{L}(b) / [1 - P(\tau > b)] \). Then
\[
\lim_{t \to \infty} V^*(t) = \min_b v(b).
\]

(ii) Assume that there is a unique \( b \) that achieves the minimum of \( v(b) \) and denote it by \( \hat{b} \). Then there is some stationary optimal policy \( b(t) \) such that for all \( t \) large enough, \( b(t) \) equals \( \hat{b} \).

Proof: The proof is given in Appendix C.

To recapitulate, we have shown, that for a general off-time, it is enough to consider deterministic policies to achieve optimal performance. Also, if the residual off-time distribution converges in time then the optimal policy converges to the constant policy and in fact becomes constant after finite time (under the appropriate conditions). This can be shown to be the case with the hyper-exponential distribution. Indeed, its residual time converges in distribution to an exponential distribution, having as parameter the smallest among the rates of the hyper-exponential distribution.

1) Suboptimal policies through Dynamic Programming: In this section, we propose a suboptimal solution approach using policy iteration for a few stages. For the rest of the stages, we consider a static control that is computed through parametric optimization, which is done next.

Consider a class of policies in which all sleep periods are i.i.d. exponentially distributed rvs with parameter \( b \). We will refer to this class as the “Exponential sleep policy.” With this policy, the cost, denoted \( V_e \), depends only on \( E[\tau] \), as detailed hereafter. Conditioning on a given off-time \( \tau \), the number of sleep periods decremented by one is a Poisson variable with rate \( \tau / b \). It is straightforward to write
\[
\]

Equation (3) can be rewritten (recall that \( \eta = \epsilon + \epsilon P_S \))
\[
V_e = \epsilon (P_S + E_L / b)E[\tau] + \epsilon E_L + \eta b.
\]  
(16)

Observe that (16) stands for any distribution of \( \tau \). We next find the optimal total cost under the Exponential policy.

Proposition III.4. The cost \( V_e \) is a convex function having a minimum at
\[
b_e^* = \sqrt{\frac{E_LE[\tau]}{\epsilon + \epsilon P_S}}.
\]  
(17)

The minimal total cost is
\[
V_e^* = \epsilon (P_S E[\tau] + E_L) + 2 \sqrt{\epsilon (\epsilon + \epsilon P_S) E_L E[\tau]}.
\]  
(18)

Proof: Let us compute the first and second derivative of the cost w.r.t. \( b \). We find
\[
V_e' = \eta - \frac{\epsilon E_L E[\tau]}{b^2}; \quad V_e'' = \frac{2 \epsilon E_L E[\tau]}{b^3}.
\]

Clearly, \( V_e'' \geq 0 \) for any \( b > 0 \), hence \( V_e \) is a convex function. The derivative \( V_e' \) has a root at \( b_e^* \) as given in (17), which yields a minimum in the cost \( V_e \) at \( b_e^* \). Substituting the optimal \( b_e^* \) in (16) we obtain the minimal cost (18).

The optimal control is \( b_e^* \). Proposition III.4 is really interesting in that it says that with i.i.d. exponential sleep periods, only the expected off-time defines the optimal control. The off-time \( \tau \) can be generally distributed. Therefore, Proposition III.4 stands valid for any user application.
Now that we have computed the static control for all stages, we proceed with one stage policy iteration. With this iteration, the sleep periods have the form $(b_1, B, B, \ldots)$ where $B$ is an exponentially distributed rv with mean $b$. We can use DP to compute the optimal policy within this class. The problem is exponential distributed rv with mean $\tau$. The optimal control identified through DP is $b^*_1$ and $b^*$.

When $\tau$ is hyper-exponentially distributed, the system state is the distribution $q$ which is transformed after each stage through the operator $g$.

If we add the constraint that the first sleep period should be exponentially distributed with the same distribution as $B$, then we will be back to the problem of finding an optimal exponentially distributed sleep period with state-independent mean. Since we do not impose this restriction, the policy obtained after one stage iteration will do strictly better than the exponential sleep policy.

This suboptimal method for one stage policy iteration can be extended to more stages. Instances of the two stage policy iteration are provided in Sect. V. As the number of stages of the policy iteration increases, the suboptimal solution converges to the optimal solution (obtained from (15) if $\tau$ is hyper-exponentially distributed).

D. Worst Case Performance

We consider in this section the case where the off-time is exponentially distributed with an unknown parameter. When the distribution of the parameter is known (Bayesian framework) the problem reduces to the study of the hyper-exponentially distributed off-time. In practice there could be many situations when the statistical distribution of the off-time is unknown or hard to estimate. In such non-Bayesian frameworks, we can conduct a worst-case analysis: optimize the performance under the worst case choice of the unknown parameter. We assume that this parameter lies within the interval $[\lambda_a, \lambda_b]$. The worst case is identified as follows $\lambda_w := \arg \max_{\lambda \in [\lambda_a, \lambda_b]} \min (E_b, k \cdot b)$. Given that $\tau$ is assumed to be exponentially distributed, it is enough to analyze the case of the Constant sleep policy, which has been found to be the optimal in Sect. III-A. The minimal cost under this policy is given in (10). We have studied (10) using the mathematics software tool, Maple 11. We found the following: $V^*(\lambda)$ is a monotonic function, decreasing with $\lambda$; $\lim_{\lambda \to +\infty} V^*(\lambda) = \epsilon E_L$; and $\lim_{\lambda \to 0} V^*(\lambda) = +\infty$. Thus, the optimal control under worst case is the one corresponding to the smallest rate in the interval considered, i.e., $\lambda_w = \lambda_a$.

IV. Parametric Optimization

In this section we provide computations of the total cost of equation (6) for the case of some interesting parametric policies. We repeat here the equation for ease of reading.

$$V(b) = -\epsilon E[\tau] + \sum_{i=1}^{\infty} \sum_{b_i=0}^{n} \frac{q_i}{1 - \exp(-\lambda_i b)}$$

(20)

where $b$ above is the vector selected sleep period lengths. Then we calculate the optimal parameters for these policies which can be used for comparison. In particular, for the Exponential and Constant policies we give closed form results while the rest of the policies are derived up to a point where they are evaluated numerically.

A. Identically Distributed Sleep Periods

This section deals with identically distributed sleep periods, in other words, the control is static. Let $B$ be a generic rv having the same distribution as any of the sleep periods. Then, (20) can be rewritten as

$$V(b) = -\epsilon E[\tau] + (\epsilon E_L + \eta E[B]) \sum_{i=1}^{\infty} \frac{q_i}{1 - \exp(-\lambda_i b)}.$$  

(21)

We now propose different policies and derive the optimal control in each case. The policies that are considered are: (i) “Exponential” policy: $B$ is exponentially distributed; one can control $b$, the expectation of $B$; (ii) “Constant” policy: $B$ is deterministic; one can control the constant sleep period length $b$; (iii) “Scaled” policy: $B$ is a scaled version of a known random variable $S$; one can control the scale $a$; (iv) “General discrete” policy: $B$ has a discrete distribution with known possible values; one can control the distribution $p$.

1) The Exponential Policy: In this policy, sleep periods are i.i.d. exponential random variables with mean $E[B] = b$. The variable $T_k$ is then Erlang distributed with shape $k$ and rate $1/b$; $E[T_k] = kb$. The off-time $\tau$ can have any distribution (we do not need $\tau$ to be hyper-exponentially distributed). With this policy, the cost, denoted $V_e(b)$, depends only on $E[\tau]$, as detailed hereafter. Conditioning on a given off-time $\tau$, the number of sleep periods decremented by one is a Poisson variable with rate $\tau/b$. It is straightforward to write

$$E[X] = E[\tau]/b + 1; \quad E[T_X] = bE[X] = E[\tau] + b.$$  

Equation (2) can be rewritten (recall that $\eta = \epsilon + \epsilon P_S$)

$$V_e(b) = \epsilon (P_S + E_L/b)E[\tau] + (\epsilon E_L + \eta b).$$  

(22)

Remark IV.1. Equation (22) stands for any distribution of $\tau$. We naturally obtain the same expression if we substitute $B(s)$ for $1/(1 + bs)$ in (21). The same result is obtained via DP in section III-C1.

2) The Constant Policy: In this policy, all sleep periods are equally sized. In other words, $B = b$. The performance is optimized by controlling the size of $b$. Substituting $B(s)$ for $\exp(-sb)$ in (21) yields the following simplified expression for the cost (the subscript stands for “constant”)

$$V_c(b) = -\epsilon E[\tau] + (\epsilon E_L + \eta b) \sum_{i=1}^{n} \frac{q_i}{1 - \exp(-\lambda_i b)}.$$  

(23)
Proposition III.1 yields the optimal values and the minimizer for the case of \( n = 1 \).

**Proposition IV.1.** The cost \( V_c(b) \) is a convex function having a minimum in \( [0, \infty] \).

**Proof:** The proof is given in Appendix D.

Proposition IV.1 proves the existence of a global minimum. Unfortunately, we are not able to derive the optimal \( b^*_c \) analytically and use numerical methods to find \( b^*_c \). The dimensionality of the problem can be showcased by the following result.

**Proposition IV.2.** When \( n > 1 \), no optimal constant policy (deterministic with constant sleep period) can be independent of \( q = (q_1, \ldots, q_n) \).

**Proof:** We develop a proof by contradiction. We assume that the optimal \( b^*_c \) does not depend on \( q \). Hence, the coefficients of the \( q_i \)'s in (32) must be null, namely, \( 1 - \exp(-\lambda_i b^*_c)(\zeta_i + \lambda_i b^*_c) = 0 \) for \( i \in \{1, \ldots, n\} \). For a given \( i \), the solution is \( b^*_c = -\frac{1}{\lambda_i}(\zeta_i + W_1(-e^{-\zeta_i})) \) (cf. proof of Proposition III.1).

Since \( b^*_c \) is a constant, the left-hand-side of the above equality must be a constant whatever \( i \) is. This is not the case (the left-hand-side depends on \( \lambda_i \)). We have thereby shown that the optimal control \( b^*_c \) must depend on \( q = (q_1, \ldots, q_n) \) when \( n > 1 \).

3) **The Scaled Policy:** In this third policy, we consider the random sleep period \( B \) to be a factor \( \alpha \) of a random variable \( S \) with a general distribution, i.e. \( B = \alpha S \). For a given distribution of \( S \), the scaling factor \( \alpha \) is controlled to optimize the performance. The cost \( V_s(\alpha) \) (the subscript stands for “scaled”) follows readily from (21) using \( B(s) = \alpha S \) and \( \mathbb{E}[B] = \alpha \mathbb{E}[S] \).

We consider now that \( S \) is a discrete random variable taking values in a finite set \( \{a_j\}_{j=1,\ldots,J} \) with a probability distribution \( \mathbf{p} = (p_1, \ldots, p_J) \), i.e., \( \mathbb{P}(S = a_j) = p_j \) and \( \sum_{j=1}^J p_j = 1 \). Hence, \( \mathbb{E}(S) = \sum_{j=1}^J p_j \exp(-\alpha a_j) \), and \( \mathbb{E}[S] = \sum_{j=1}^J p_j a_j \).

This policy advocates to have each sleep period follow a discrete general distribution, taking values in \( \{\alpha a_j\}_{j=1,\ldots,J} \). The probability distribution \( \mathbf{p} \) is assumed fixed whereas the set of possible values can be scaled for minimal cost.

The optimization problem can be stated as

\[
V^*_s = \min_{\alpha > 0} V_s(\alpha); \quad \alpha^* = \arg \max V^*_s. \tag{24}
\]

It is intractable to solve analytically (24), we will therefore resort to a numerical resolution (cf. Section V).

4) **The General Discrete Policy:** The fourth policy resembles the third one in that it equally considers a discrete general sleep period for the variable \( B \). However, the set of possible values is now fixed (i.e., \( \alpha = 1 \)) whereas the probability distribution \( \mathbf{p} \) can be optimized for minimal cost. We denote the cost as \( V_g(\mathbf{p}) \), where the subscript stands for “general”, and write

\[
V_g(\mathbf{p}) = -\mathbb{E}[r] + \sum_{i=1}^n q_i \left( \epsilon E_L + \eta \sum_{j=1}^J p_j a_j \right) \frac{1}{1 - \sum_{j=1}^J p_j \exp(-\lambda_i a_j)}.
\]

Our objective is to find \( \mathbf{p}^* = \arg \min_{\mathbf{p}} V_g(\mathbf{p}) \) such that \( 0 \leq p_j \leq 1 \) for \( j = 1, \ldots, J \) and \( \sum_{j=1}^J p_j = 1 \). This optimization problem can only be solved numerically.

**B. Non-identically Distributed Sleep Periods**

If we relax the constraint of identically distributed sleep periods, the mobile is free to choose any sleep period distribution at each waking up instant, a fact that complexes the problem immensely. We will narrow the problem by considering only deterministic sleep periods.

The \( k \)-th sleep period is now of fixed size \( b_k \), the instants \( \{T_k\}_{k \in \mathbb{N}} \) are now deterministic, and we let \( t_k = T_k \) for any \( k \) to reflect this. We have \( t_0 = 0 \) and \( t_k = \sum_{j=1}^k b_j \).

The policies that are considered in this section are: (i) “Semi-Constant” policy: Most sleep periods are equal; (ii) “Multiplicative” policy: Sleep periods increase with time; (iii) “General Deterministic” policy: Sleep periods can last for any positive time.

1) **The Semi-Constant Policy:** As the name indicates, only a few sleep periods are allowed to be of different size. This is expected to bring some improvement with respect to the Constant policy. For the sake of illustration, let the first sleep period be of size \( b_0 \) while all the subsequent ones are of size \( b \).

This particular policy will be referred to as “one-stage” policy. The cost is

\[
V_{sc}(b_0, b) = -\mathbb{E}[r] + (\epsilon E_L + \eta b) \sum_{i=1}^n q_i \exp(-\lambda_i b_0) \frac{1}{1 - \exp(-\lambda_i b)}. \tag{25}
\]

**Proposition IV.3.** For \( n = 1 \), all sleep periods are equal at optimum, i.e., \( b_0 = b^* = b^*_c \) (recall (9)).

Notice that the \( m \)-stage Semi-Constant policy approaches the absolute optimal policy when \( m \to \infty \). However, it adds immense computational complexity as \( m \) grows. Hence, for practical purposes, one can optimize until a few stages.

Also, note that the LTE standards propose an \( m \)-stage Semi-Constant policy where the first \( m \) stages have equal sleep period length and then the system switches to another value (usually larger).

2) **The Multiplicative Policy:** This policy is inspired by the power save mode of the IEEE 802.16e [1], and more precisely, by type I power saving classes. There, the length of a sleep period is doubled over time until a maximum permissible sleep window, denoted \( b_{\text{max}} \), is reached. The size of the \( k \)-th sleep period is then \( b_k = b_1 2^{\min(k-1,l)} \), \( k \in \mathbb{N}^* \) where \( l := \log_2(b_{\text{max}}/b_1) \). We also have

\[
t_k = b_1 \left( 2^{\min(k,l)} - 1 + 2^l (k-l) \mathbb{1}_{k \leq l} \right), \quad k \in \mathbb{N}^*.
\]
The cost of the power save mode of the IEEE 802.16e Standard can be derived from (20), yielding

\[ V_{\text{Std}} = -\varepsilon E[\tau] + \sum_{k=0}^{\infty} \sum_{i=1}^{n} q_i e^{-\lambda_i t_k} \left[ \varepsilon E_L + \eta b_i 2^{\min\{k,l\}} \right]. \]

(26)

Instead of doubling the sleep periods over time, the multiplicative policy increases the sleep periods by a multiplicative factor \( f \) (in the Standard policy, \( f = 2 \)). The performance is then optimized by controlling the factor \( f \). In this policy, we have

\[ b_k = b_1 f^{\min\{k-1,l\}}, \quad k \in \mathbb{N}^* \]

\[ t_k = b_1 \left( \frac{f^{\min\{k,l\}} - 1}{f - 1} + f^{l-k-l} 1_{k>l} \right), \quad k \in \mathbb{N}^* \]

\[ V_m(f) = -\varepsilon E[\tau] + \sum_{k=0}^{\infty} \sum_{i=1}^{n} q_i e^{-\lambda_i t_k} \left[ \varepsilon E_L + \eta b_{k+1} \right] \quad (27) \]

\[ f^* = \arg \min_{f>1} V_m(f). \quad (28) \]

The optimal \( f^* \) and the minimal cost \( V_m^* = V_m(f^*) \) (the subscript stands for “multiplicative”) will be computed numerically.

3) The General Deterministic Policy: In this section, no particular pattern is imposed on the sleep periods. This policy is a generalization of the Semi-Constant policy as \( m \to \infty \). We denote the cost as \( V_d(\mu) \) where the subscript stands for “deterministic” and \( \mu := (b_1, b_2, \ldots) \) is the deterministic policy. The cost has the same expression as (27). A necessary condition for the existence of an optimal control sequence \( \mu^* = (b_1^*, b_2^*, \ldots) \) is that \( \text{grad} \ V_d(\mu^*) = 0 \). Our next step is then to compute the partial derivatives. We have, for \( j \in \mathbb{N}^* \),

\[ \frac{\partial V_d(\mu)}{\partial \mu_j} = \sum_{i=1}^{\infty} \eta q_i \left[ e^{-\lambda_j t_{j-1}} - \sum_{k=j}^{\infty} \lambda_i e^{-\lambda_i t_k} \left( b_{k+1} + \frac{\varepsilon E_L}{\eta} \right) \right]. \]

(29)

**Proposition IV.4.** When \( n > 1 \), no optimal deterministic policy (with arbitrary period length) can be independent of \( q = (q_1, \ldots, q_n) \).

**Proof**: The proof is given in Appendix E.

The contradiction between (33) and (34) arises because of \( n > 1 \). For \( n = 1 \) more results are expected. Indeed, in section III, it shown that the optimal policy is given in closed form and it is constant in time and a function of the arrival rate \( \lambda_1 \).

V. NUMERICAL INVESTIGATION

In this section we present numerical results when the off-time \( \tau \) is either exponentially or hyper-exponentially distributed. In each case, the best control and the corresponding cost are computed.

Also, we utilize the results of section IV to compare several classes of policies. The cost \( V \), captures the main performance measures: energy consumed during the sleep duration and the extra delay incurred due to the sleep mode. The cost \( V \) is a weighted sum of both metrics. From (2), it comes that a large value of \( \epsilon \) makes \( V \) more sensitive to the energy consumption than to the extra delay, whereas a small \( \epsilon \) gives more weight to the delay.

The various policies are compared through: (i) the optimal expected sleep duration, (ii) the minimal cost achieved, and (iii) the relative improvement with respect to the IEEE 802.16e protocol. The improvement ratio, denoted \( I \), is defined as follows: \( I := (V_{\text{Std}} - V^*)/V_{\text{Std}} \), where the cost \( V_{\text{Std}} \) of the standard’s policy is calculated using (26). The parameters of the Standard policy are \( b_1 = 2 \) and \( l = 10 \). The physical parameters are set to the following values: \( E_L = 10, P_S = 1 \).

**A. Exponential Off-Time**

In this case, the optimal is to fix all sleep periods to the value found in (9).

We evaluate three sleep policies from section IV (cf. Table II) and compare them. The performance of each policy de-
TABLE II

<table>
<thead>
<tr>
<th>Policy</th>
<th>Optimal cost</th>
<th>Control and its optimal value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exponential</td>
<td>(18)</td>
<td>expected sleep duration, (17)</td>
</tr>
<tr>
<td>Constant</td>
<td>(10)</td>
<td>size of fixed sleep duration, (9)</td>
</tr>
<tr>
<td>Standard</td>
<td>(26), n = 1</td>
<td>–</td>
</tr>
</tbody>
</table>

pends on the arrival rate \( \lambda \) and on the normalized weight \( \epsilon \). In the following evaluation, we will alternatively vary one of the parameters and fix the other.

We first vary \( \lambda \) and fix \( \epsilon \) to 0.1 and 0.9. The weight \( \epsilon \) equal to 0.1 mimics the situation when energy consumption is given lower priority over delay, while \( \epsilon \) equal to 0.9 mimics the opposite situation. Looking at Fig. 2, one can observe the impact of the arrival rate \( \lambda \) on (i) the optimal expected sleep duration (cf. Fig. 2(a)), (ii) the minimal cost (cf. Fig. 2(b)), and (iii) the cost improvement (cf. Fig. 2(c)). We naturally find that the expected sleep duration decreases as \( \lambda \) increases, as foreseen in (9). The physical explanation is that, a large arrival rate forces the mobile to be available after shorter breaks, otherwise the cost is too high.

Of more interest are the curves reported in Fig. 2(a), where the optimal cost achieved by the Constant policy always outperforms the costs of the two other policies. This is in agreement with the discussion in Section III, namely, that the constant policy should be the optimal among all possible policies. The Exponential policy outperforms the Standard policy for a large range of values of \( \lambda \) as seen more clearly in Fig. 2(c) where \( \epsilon = 0.1 \).

Observe in Fig. 2(b) how the cost decreases asymptotically to \( \epsilon E_L \) (1 for \( \epsilon = 0.1 \) and 9 for \( \epsilon = 0.9 \)) as the rate \( \lambda \) increases. The same trend is observed for the cost of the Standard policy. As \( \lambda \) decreases, the increase in \( V_{\epsilon}^* \) and \( V_{\epsilon}^* \) is due to the increase of the optimal expected sleep duration, while for \( V_{\text{Std}} \) the increase is due to the extra (useless and costly) listening.

We next vary \( \epsilon \) and fix \( \lambda \) to 0.1 (low traffic) and 5 (high traffic). The results are depicted in Fig. 3. As \( \epsilon \) gets smaller, the extra delay gets more penalizing, enforcing then smaller optimal sleep durations. This is observed in Fig. 3(a). As mentioned earlier, smaller optimal sleep durations yield smaller optimal costs. Thus, the optimal costs increase as \( \epsilon \) increase, as can be observed in Fig. 3(b). For \( \epsilon < 0.1 \), the cost of the Standard policy is fairly insensitive to \( \epsilon \). This is due to its compromising nature; the first small periods guarantee responsiveness if the off-time is short, and the large periods guarantee a good energy performance if the off-time is large. Evidently, this results in a total cost always above the optimal policies. Also, we can observe that the Standard has been designed to favor energy over delay: it performs quite close to the optimal policy when \( \epsilon \to 1 \), i.e. when the cost is indifferent to delay.

Looking at Fig. 3(c), we find again that the Constant policy is the best and that the Exponential policy outperforms the Standard policy in most cases: the Exponential policy yields a substantial improvement over a large range of values of \( \lambda \) and \( \epsilon \).
We can conclude that the optimal cost is highly sensitive to the Multiplicative factor at very low traffic. This is not surprising as sleep periods increase exponentially in the Multiplicative policy, and this is much likely to happen when traffic is very low.

VI. CONCLUSION

The control of sleep periods for mobile terminals is studied. In particular, an optimization problem is formulated where the goal is to minimize energy consumption in wireless networks taking into account the incurred response delays. Previous models studied in the literature have considered an exogenous arrival process, whereas we considered an on-off model in which the off-time begins when the terminal goes to sleep mode and where the duration of the on-time does not depend on the delay imposed by the sleeping during the inactivity period. We derived the optimal policy in case of a Poisson arrival process and found many structural properties of the optimal policy for hyper-exponential and general off-times. Suboptimal policies have been derived in this case using one and two stage policy iteration. Also, we considered several constrained classes of policies with certain optimization parameters and degrees of freedom. Among them, we considered classes that depict the functionality of IEEE 802.16e standards. We showed how these classes are optimized and compared them with each other yielding insightful conclusions. When the off-time is hyper-exponentially distributed, we showed that the IEEE 802.16e standard can be improved substantially if the multiplicative factor is optimized. Also, if one gives small weight to the mobile’s response delay and favors the minimization of energy use, then both the exponential and scaled policies are candidate to substitute the standard. The optimal control for the Exponential policy is found in closed form for a general off-time.

Furthermore, our model is general enough to capture other standards for sleep mode with small alterations, like the DRX of 3GPP LTE [2] and the WMM of IEEE 802.11 (WiFi) [29], a work left for the future. In terms of the model itself, a possible extension is to utilize renewal theory to capture the effect of residual interarrival off-time, thus changing the scope from interactive applications to applications where the service time biases the arrival process. Note, that the case \( n = 1 \), analyzed in this paper, covers both cases, but for \( n > 1 \) we have only covered interactive applications. Finally, in this paper, we have focused on policies that keep no memory of the past. Alternatively, one can study adaptive policies which can exploit the history of the inactivity period.

VII. ACKNOWLEDGEMENTS

The authors of INRIA were partly supported by the ANR ECOSCELL. The work of V Borkar and E Altman was partly supported by Indo-French Centre for the Promotion of Advanced Research (IFCPAR), project 4000-IT. The work of G Paschos is partly supported by European Commission IST projects FP7-ICT-257740 (TRENDO) and STREP-FP7-INFO-ICT-224218 (OPNEX).
Proof: From (8) we can express
\[ V(\lambda) = \frac{\epsilon E\left[(b - \tau(\lambda))\mathbb{1}_{\tau(\lambda) \leq b}\right] + \epsilon (E_L + bP_S)}{1 - P(\tau(\lambda) > b)} \] (30)

Substituting
\[ E\left[(b - \tau(\lambda))\mathbb{1}_{\tau(\lambda) \leq b}\right] = \lambda b - 1 + \exp(-\lambda b) \]
and \( P(\tau(\lambda) > b) = \exp(-\lambda b) \) in (30) and differentiating w.r.t. \( b \) we obtain
\[ V'(\lambda) = \frac{\eta \left( \frac{1 - \exp(-\lambda b)(\zeta + \lambda b)}{(1 - \exp(-\lambda b))^2} \right)}{1 - \exp(-\lambda b)}. \] (31)

At the extremum of \( V(\lambda) \), denoted \( b^* \), we must have
\[ 1 - \exp(-\lambda b^*)(\zeta + \lambda b^*) = 0 \] \[ \Leftrightarrow \exp(-\zeta - \lambda b^*) (\zeta - \lambda b^*) = -\exp(-\zeta). \]

The last expression is of the form \( y \exp(y) = x \) with \( y = -\zeta - \lambda b^* \) and \( x = -\exp(-\zeta) \). The solution \( y \) is the Lambert \( W \) function [30], denoted \( W \), at the point \( x \). Hence, \(-\zeta - \lambda b^* = W(-\exp(-\zeta)) \). Since \( \zeta \geq 1 \), we have \(-\exp(-1) \leq -\exp(-\zeta) < 0 \). Therefore, we need \( W(-\exp(-\zeta)) \) to be real-valued in \([-\exp(-1), 0]\). Also, given that \( \zeta + \lambda b^* \geq 1 \), we need \( W(-\exp(-\zeta)) \) to be always negative and smaller than -1. Both conditions are satisfied by the branch numbered -1. Hence, \(-\zeta - \lambda b^* = W_{-1}(-\exp(-\zeta)) \) and (9) is readily found.

Replacing (9) in (30) and using the relation \( \exp(y) = x/y \), one can derive (10).

Similarly we proceed to the second order conditions to determine if \( b^* \) yields minimum cost. The second derivative function of the cost is
\[ V''(\lambda) = \frac{\eta \lambda e^{-\lambda b}}{(1 - e^{-\lambda b})^3} \left\{ (1 + e^{-\lambda b})(1 + \zeta + \lambda b) - 4 \right\}. \]

The sign of \( V''(\lambda) \) depends on the value of \( z_1(b) := (1 + \exp(-\lambda b))(1 + \zeta + \lambda b) \). The following can be derived
\[ z'_1(b) = \lambda e^{-\lambda b}(1 - \exp(-\lambda b)(\zeta + \lambda b)) \]
\[ \lim_{b \to 0} z'_1(b) = -\lambda_1(1 - \zeta) < 0 \]
\[ \lim_{b \to \infty} z'_1(b) = \lambda_1 > 0 \]

The derivative \( z'_1(b) \) is null for \( b = b^* > 0 \), negative for \( b < b^* \) and positive for \( b > b^* \). Hence, \( z_1(b) \) decreases from \( \lim_{b \to 0} z_1(b) = 2(1 + \zeta_1) > 4 \) to its minimum \( z_1(b^*) = \frac{-(W_{-1}(e^{-\zeta_1} - 1))^2}{W_{-1}(e^{-\zeta_1})} > 4 \) and then increases asymptotically to \(+\infty\). We have shown that \( z_1(b) > 4 \) for any positive \( b \). Therefore, \( V''(\lambda) > 0 \) for any positive \( b \). \( V(\lambda) \) is then a convex function in \( b \) and the extremum \( b^* \) is a global minimum, which concludes the proof. 

Proof: Let \( \alpha(i) := \frac{\exp(-\lambda b)}{\exp(-\lambda c_i)} \). Then (12) can be rewritten
\[ g_i(q, b) = \frac{q_i \alpha_i}{\sum_{j=I(q)} q_j \alpha_j}. \]

In particular,
\[ g^m_{I(q)}(q, b) = \frac{q_{I(q)}}{q_{I(q)} + \sum_{j > I(q)} q_j \alpha^m_j}. \]

Since \( \lambda_i \leq \lambda_j \) for \( I(q) < i < j \), then \( \alpha_j < \alpha_i \leq \alpha_{I(q) + 1} \leq \alpha_{I(q)} = 1 \). Hence
\[ g^m_{I(q)}(q, b) \geq \frac{q_{I(q)}}{q_{I(q)} + \alpha^m_{I(q) + 1} \sum_{j > I(q)} q_j \alpha^m_j} \]
\[ \quad \Leftrightarrow \sum_{j > I(q)} g^m_j(q, b) \leq \frac{\alpha^m_{I(q) + 1} \left( 1 - q_{I(q)} \right)}{q_{I(q)} + \alpha^m_{I(q) + 1} \left( 1 - q_{I(q)} \right)} \]
\[ \leq \alpha^m_{I(q) + 1} - \frac{q_{I(q)}}{q_{I(q)}}. \]

We then have that \( \lim_{m \to \infty} \sum_{j > I(q)} g^m_j(q, b) = 0 \), which implies the lemma.

Proof: By the bounded convergence theorem,
\[ \lim_{t \to \infty} c(t, b) = \epsilon E[(b - \tilde{\tau}) \mathbb{1}_{\tilde{\tau} \leq b}] + \epsilon (E_L + bP_S) = \bar{c}(b). \]

Let \( V^0 := 0 \). Then \( \hat{V}^1 := \lim_{t \to \infty} (\tilde{\mathcal{L}} V^0)(t) = \min_b \bar{c}(b) \)

which is a constant. Assume that \( \hat{V}^k := \lim_{t \to \infty} V^k(t) \) exists for some \( k \). Then
\[ \hat{V}^{k+1} := \lim_{t \to \infty} (\hat{\mathcal{L}} V^k)(t) \]
\[ = \lim_{t \to \infty} \min_b \left\{ c(t, b) + P(\tau_t > b) V^k(t + b) \right\} \]
\[ = \min_b \left\{ \bar{c}(b) + P(\tilde{\tau} > b) \hat{V}^k \right\} \]

which is a constant. Hence by the monotone convergence of \( V^k \) to \( V^* \), the limit \( \hat{V} := \lim_{t \to \infty} V^*(t) \) exists and satisfies the limit dynamic programming (DP) \( \hat{V} = \min_b \bar{c}(b) \). This DP corresponds to an MDP that has a single state and thus there exists an optimal constant deterministic policy that takes always the same \( b \), which we denote \( \hat{b} \). This gives \( \hat{V} = \bar{c}(\hat{b}) + P(\tilde{\tau} > \hat{b}) \hat{V} \) so that
\[ \hat{V} = \frac{\bar{c}(\hat{b})}{1 - P(\tilde{\tau} > \hat{b})} = v(\hat{b}) = \min_b v(b). \]

Any other stationary (constant) deterministic policy \( b \) for the limit DP gives a larger value
\[ \frac{\bar{c}(b)}{1 - P(\tilde{\tau} > b)} \geq \hat{V} \]
This establishes (i).

In part (ii), the last inequality is strict for all \( b \neq \hat{b} \). Since the limit DP is obtained from the original one by considering large \( t \), it follows that for all \( t \) large enough, \( \hat{b} \) will give a strictly lower value of \( c(t, b) + P(\tau_1 > b) V(t + b) \) than any other value of \( b \). Thus by part (iii) of the previous theorem, \( \hat{b} \) is optimal at all \( t \) large enough.

APPENDIX D

PROOF OF PROPOSITION IV.1

Proof: For convenience, we introduce \( \xi_i = \frac{\lambda_i e E_k}{\eta} + 1 \) for \( i = 1, \ldots, n \). The first and second derivative functions of \( V_c(b) \) are, respectively

\[
V_c'(b) = \sum_{i=1}^{n} q_i \left( \frac{1 - \exp(-\lambda_i b)(\xi_i + \lambda_i b)}{(1 - \exp(-\lambda_i b))^2} \right)
\]

\[
V_c''(b) = \sum_{i=1}^{n} \eta_i \frac{\lambda_i e^{\xi_i}}{(1 - e^{-\lambda_i b})^3} \left( (1 + e^{-\lambda_i b})(1 + \xi_i + \lambda_i b) - 4 \right).
\]

To study the sign of \( V_c''(b) \), we need to evaluate the functions

\[ z_i(b) := \frac{1 - \exp(-\lambda_i b)}{(1 + \xi_i + \lambda_i b)} \text{ for } i = 1, \ldots, n. \]

The function \( z_i(b) \) has a minimum at \( b_i := -\frac{1}{\lambda_i}(\xi_i + W^{-1}(-e^{-\xi_i})) \), and \( z_i(b) \) is always above 4 for any \( i \in \{1, \ldots, n\} \). Hence, \( V_c''(b) \geq 0 \) for any positive \( b \), implying that \( V_c(b) \) is a convex function (the derivative \( V_c'(b) \) increases with \( b \)). We have \( \lim_{b \to \infty} V_c(b) = \eta = \bar{\epsilon} + eE_S > 0 \), \( \lim_{b \to 0} V_c(b) = -\infty \) which implies that there exists some \( b^*_i > 0 \) such that \( V_c(b^*_i) = 0 \). Therefore, \( V_c(b) \) has a global (strictly positive) minimum at \( b^*_i \).

APPENDIX E

PROOF OF PROPOSITION IV.4

Proof: We prove Proposition IV.4 by contradiction. We assume that there exists an optimal control \( \mu^* = (b_1^*, b_2^*, \ldots) \) which does not depend on \( q = (q_1, \ldots, q_n) \). Therefore, the coefficients of the probabilities \( \{q_i\}_{i=1}^{n} \) in (29) must all be null when \( \mu = \mu^* \). In other words, we must have for \( i = 1, \ldots, n \) and \( j \in \mathbb{N}^+ \)

\[ \exp(-\lambda_i t_{j-1}^*) = \sum_{k=j}^{\infty} \lambda_i \exp(-\lambda_i t_k^*) \left( b_{k+1}^* + \frac{eE_L}{\eta} \right). \]

Subtracting, for a given \( i \), the expression for \( j \) from that for \( j - 1 \), we get after simplification

\[ b_{j+1}^* = \frac{\exp(\lambda_i b_j^*) - 1}{\lambda_i} - \frac{eE_L}{\eta} \]

which must hold for \( j \in \mathbb{N}^+ \) and for any \( i \in \{1, \ldots, n\} \). Since \( b_{j+1}^* \) must be constant, it is imperative that

\[ \frac{\exp(\lambda_i b_j^*) - 1}{x} = \ldots = \frac{\exp(\lambda_i b_j^*) - 1}{x} \]

for \( j \in \mathbb{N}^+ \). For a given \( j \), the above equality holds only when \( b_j^* = 0 \), i.e., if there were no \( j \)th sleep period. Given that (34) must hold for \( j \in \mathbb{N}^+ \), then all sleep periods need to be null. However, on the other hand, replacing \( b_j^* = 0 \) in (33) yields \( b_{j+1}^* = -eE_L/\eta < 0 \), which is absurd. Equations (33) and (34) contradict each other. Therefore, the starting hypothesis is wrong and the optimal control must depend on \( q = (q_1, \ldots, q_n) \).}

REFERENCES


