Optimal Sleep Scheduling for a Wireless Sensor Network Node

David Shuman and Mingyan Liu
Electrical Engineering and Computer Science Department
University of Michigan
Ann Arbor, MI 48109-2122
{dishuman,mingyan}@umich.edu

Abstract

We consider the problem of conserving energy in a single node in a wireless sensor network by turning off the node's radio for periods of a fixed time length. While packets may continue to arrive at the node's buffer during the sleep periods, the node cannot transmit them until it wakes up. The objective is to design sleep control laws that minimize the expected value of a cost function representing both energy consumption costs and holding costs for backlogged packets. We consider a discrete time system with a Bernoulli arrival process. In this setting, we characterize optimal control laws under the finite horizon expected cost and infinite horizon expected average cost criteria.

Index Terms

Wireless Sensor Networks, Power Management, Resource Scheduling, Markov Decision Processes, Vacation Models, Monotone Policies

I. Introduction

Wireless sensor networks have recently been utilized in an expanding array of applications, including environmental and structural monitoring, surveillance, medical diagnostics, and manufacturing process flow. In many of these applications, sensor networks are intended to operate for long periods of time without manual intervention, despite relying on batteries or energy harvesting for energy resources. Conservation of energy is therefore well-recognized as a key issue in the design of wireless sensor networks [1].

Motivated by this issue, there have been numerous studies on methods to effectively manage energy consumption while minimizing adverse effects on other quality of service requirements such as connectivity, coverage, and packet delay. For example, [2], [3], and [4] adjust routes and power rates over time to reduce overall transmission power and balance energy consumption amongst the network nodes. Reference [5] aggregates data to reduce unnecessary traffic and conserve energy by reducing the total workload in the system. Reference [6] makes the observation that when operating in ad hoc mode, a node consumes nearly as much energy when idle as it does when transmitting or receiving, because it must still maintain the routing structure. Accordingly, many studies have examined the possibility of conserving energy by turning nodes on and off periodically, a technique commonly referred to as duty-cycling. Of particular note, GAF [7] makes use of geographic location information provided for example by GPS; ASCENT [8] programs the nodes to self-configure to establish a routing backbone; Span [9] is a distributed algorithm featuring local coordinators; and PEAS [10] is specifically intended for nodes with constrained computing resources that operate in harsh or hostile environments. While the salient features of

these studies are quite different, the analytical approach is similar. For the most part, they discuss the qualitative features of the algorithm, and then perform numerical experiments to arrive at an energy savings percentage over some baseline system.

In this report, we also consider a wireless sensor network whose nodes sleep periodically; however, rather than evaluating the system with a given sleep control policy, we impose a cost structure and search for an optimal policy amongst a class of policies. In order to approach the problem in this manner, we need to consider a far simpler system than those used in the aforementioned studies. Thus, we consider only a single sensor node and focus on the tradeoffs between energy consumption and packet delay. As such, we do not consider other quality of service measures such as connectivity or coverage. The single node under consideration in our model has the option of turning its transmitter and receiver off for fixed durations of time in order to conserve energy. Doing so obviously results in additional packet delay. We attempt to identify the manner in which the optimal (to be defined in the following section) sleep schedule varies with the length of the sleep period, the statistics of arriving packets, and the charges assessed for packet delay and energy consumption.

The only other works we are aware of that take a similar approach are by Sarkar and Cruz, [11] and [12]. Under a similar set of assumptions to our model, with the notable exceptions that a fixed cost is incurred for switching sleep modes and the duration of the sleep periods is flexible, these papers formulate an optimization problem and proceed to numerically solve the optimal duration and timing of sleep periods through a dynamic program.

Our model of the duty-cycling node falls into the general class of vacation models. Applicable to a wide range of problems from machine maintenance to polling systems, vacation models date back to the late 1950s. Many important results on vacation models in discrete time can be found in [13] and [14]. Reference [15] was the first study to analyze the steady-state distribution of the queue length and unfinished work of the Geo/D/1 queue, which is the uncontrolled analog to the controlled queue in our system. Reference [16] extends these results to the Geo/D/1 queue with priorities.

Within the class of vacation models, we are particularly interested in systems resulting from threshold policies; i.e., control policies that force the queue to empty out and then resume work after a vacation when either the queue length or the combined service time of jobs in queue (learned upon arrival of jobs to the system) reaches a critical threshold. The introduction of [17] provides a comprehensive overview of the results on different types of threshold policies. Of these models, [17] is the most relevant to our model, and we discuss it further in Section III-D. The relevant discrete time infinite horizon optimization results are covered in [18] and [19], and are discussed further in Section III-A. Finally, for more on the equivalence of continuous and discrete time Markov decision processes, see [20].

The rest of this report is organized as follows. In the next section, we describe the general system model and formulate the finite horizon expected cost and infinite horizon average expected cost optimization problems. In Section III, we provide a brief review of some key results in average cost optimization theory for countable state spaces, and then characterize completely the optimal sleep policy for the infinite horizon problem. In Section IV, we partially characterize the optimal sleep policy for the finite horizon problem, and present two conjectures concerning the optimal control at the one state for which we have not yet specified the optimal policy. Section V concludes the report.

II. Problem Description

In this section we present an abstraction of the sleep scheduling problem outlined in the previous section that captures the essential features of the network model described above. We formulate the optimization problem along with a summary of assumptions and notation.

A. System Model

We consider a single node in a wireless sensor network. The node is modeled as a single-server queue that accepts packet arrivals and transmits them over a reliable channel. In order to conserve energy, the node goes to sleep (turns off its transmitter) from time to time. While asleep, the node is unable to transmit packets; however, packets continue to arrive at the node. This essentially results in a queueing system with vacations.

We consider time evolution in discrete time steps indexed by $t = 0, 1, \ldots, T$, with each increment representing a slot length. Slot t refers to the slot defined by the interval $[t, t+1)$. We assume that packets arrive randomly to the node according to a Bernoulli process, and that they are of equal length such that one packet transmission time occupies one time slot.

In general, switching on and off is also an energy consuming process. Therefore, we want to avoid putting the node to sleep very frequently. There are different ways to model this. One is to charge a switching cost whenever we turn on the node. In this study we adopt a different model. Instead of charging the node for switching, we require that the sleep period of the node has to be an integer multiple of some constant N in time slots. By adjusting the value of N we can prevent the node from switching too frequently.

We assume that even while asleep, the node accurately learns its current queue size at each time t. A node makes the sleeping decision (i.e., whether to remain awake or go to sleep) based on the current backlog information, as well as the current time slot. We assume that the sleep decision for the t-th slot is made at time t, while packet arrivals during the t-th slot start at t^+. Therefore packets arriving in a given slot are not eligible for transmission until the next slot.

There are two objectives in determining a good sleep policy. One is to minimize the packet queueing delay and the other is to conserve energy in order to continue operating for an extended amount of time. Accordingly, our model assesses costs to backlogged packets and energy consumed during the slots in which the node remains awake. The goal of this study is to characterize the control laws that minimize these costs over a finite or infinite time horizon.

B. Notation

Before proceeding, we present the following definitions and notation.

T	The length in slots of the time horizon under consideration.
N	The fixed number of slots for which the node must stay asleep once it goes to sleep.
B_t	The node's queue length at the beginning of the t-th slot. This quantity is observed at t^-. Note that this is also the queue length at the end of the $(t-1)$-th slot. B_0 is the initial queue length.
S_t	The number of slots remaining until the node awakes, including the t-th slot. This quantity is also observed at time t^-. $S_t = 0$ indicates the node is awake at time t.
X_t	$:= \begin{bmatrix} B_t \\ S_t \end{bmatrix}$, the information state at time t.
$\mathcal{X}$	The state space.
Y_t	The output/observation available to the node at time t.

A_t	The number of random arrivals during the t-th time slot. As mentioned earlier, arrivals are assumed to occur within $(t, t+1)$.
p	The probability of an arrival in each time slot.
$\mathcal{U}$	$:= \{0, 1\} = \{$Sleep, Stay Awake$\}$, the space of control actions.
U_t	The control random variable denoting the sleep decision for time slot t.
c	The per packet holding cost assessed at the end of each time slot.
D	The cost incurred in each time slot during which the node is awake.
$\mathcal{F}_t$	The σ-field induced by all information through time t.
$\boldsymbol{\pi}$	$:= (\pi_1, \pi_2, \ldots)$, a sleep policy. When a distinction between policies must be made, we write $\hat{\boldsymbol{\pi}}$ and $\tilde{\boldsymbol{\pi}}$.
$[x]^+$	$:= \begin{cases} x, & x \geq 0 \\ 0, & \text{otherwise} \end{cases}$

C. Assumptions

Below we summarize the important assumptions adopted in this study. These assumptions apply to both problems described in the next subsection.

1) We consider a node, which upon going to sleep, must remain asleep for a fixed number, *N*, slots. The node is allowed to take multiple vacations of length N in a row.
2) We assume a Bernoulli arrival process with known arrival rate, *p*, strictly between 0 and 1. Furthermore, we assume that the arrivals are independent of both the queue size and the allocation policy.
3) We assume that the A_t packets arriving in time slot t arrive within $(t, t+1)$, and cannot be transmitted by the node until the next time slot, i.e., the $(t+1)$-st slot, $[t+1, t+2)$.
4) We assume attempted transmission of a queued packet is successful $w.p.1$. Only one packet may be transmitted in a slot, and the transmission time of one packet is assumed to be one slot.
5) We assume the node has an initial queue size of B_0, a random variable taking on finite values $w.p.1$.
6) We assume the node has an infinite buffer size. Without this assumption we would need to introduce a penalty for packet dropping/blocking.
7) We assume that in addition to perfect recall, the node has perfect knowledge of its queue length at the beginning of each time slot, immediately before making its control decision for the t-th slot exactly at time t.

D. Problem Formulation

We consider two distinct problems. The first, Problem (**P1**), is the infinite horizon average expected cost problem. The second, Problem (**P2**), is the finite horizon expected cost problem. The two problems feature the same information state, action space, system dynamics, and cost structure, but different optimization criteria.

For both problems, the system dynamics are given by:

$$X_{t+1} = \begin{cases} \begin{bmatrix} B_t + A_t \\ S_t - 1 \end{bmatrix}, & \text{if } S_t > 0 \\ \begin{bmatrix} B_t + A_t \\ N - 1 \end{bmatrix}, & \text{if } S_t = 0 \text{ and } U_t = 0 \\ \begin{bmatrix} [B_t - 1]^+ + A_t \\ 0 \end{bmatrix}, & \text{if } S_t = 0 \text{ and } U_t = 1 \end{cases} \tag{1}$$

$$Y_t \;=\; X_t \;.$$

The information state, X_t, tracks both the current queue length and the current sleep status. Given the current state, X_t, the probability of transition to the next state, X_{t+1}, depends only on the random arrival, A_t, and the sleep decision, U_t. Note that when the node is asleep $(S_t > 0)$, the only available action is to sleep $(U_t = 0)$; however, when the node is awake $(S_t = 0)$, both control actions are available. Model (1) is a controlled Markov chain with time-invariant matrix of transition probabilities, $P_{ij}(u)$, given by the following (here state i is given by $i = \begin{bmatrix} i_b \\ i_s \end{bmatrix}$, and state j is given by $j = \begin{bmatrix} j_b \\ j_s \end{bmatrix}$):

$$P_{ij}(0) = \begin{cases} p, & j = \begin{bmatrix} i_b + 1 \\ i_s - 1 \end{bmatrix} \text{ and } i_s > 0 \\ 1-p, & j = \begin{bmatrix} i_b \\ i_s - 1 \end{bmatrix} \text{ and } i_s > 0 \\ p, & j = \begin{bmatrix} i_b + 1 \\ N - 1 \end{bmatrix} \text{ and } i_s = 0 \\ 1-p, & j = \begin{bmatrix} i_b \\ N - 1 \end{bmatrix} \text{ and } i_s = 0 \\ 0, & \text{otherwise} \end{cases}$$

and

$$P_{ij}(1) = \begin{cases} p, & j = \begin{bmatrix} i_b \\ 0 \end{bmatrix}, \; i_b > 0, \text{ and } i_s = 0 \\ 1-p, & j = \begin{bmatrix} i_b - 1 \\ 0 \end{bmatrix}, \; i_b > 0, \text{ and } i_s = 0 \\ p, & j = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{ and } i = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ 1-p, & j = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \text{ and } i = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ 0, & \text{otherwise} \end{cases} , \tag{2}$$

where i_s is the sleep status component of the state vector i, and i_b is the queue length of i.

Finally, we present the optimization criterion for each problem. For Problem (**P1**), we wish to find a sleep control policy $\boldsymbol{\pi}$ that minimizes J^π, defined as:

$$J^\pi := \limsup_{T\to\infty} \frac{1}{T} \cdot E^\pi \left\{ \sum_{t=0}^{T-1} D \cdot U_t + \sum_{t=1}^{T} c \cdot B_t \mid \mathcal{F}_0 \right\} . \tag{3}$$

In Problem (**P2**), the cost function for minimization is J_0^π, where the expected cost-to-go at time k, J_k^π, is defined as:

$$J_k^\pi := E^\pi \left\{ \sum_{t=k}^{T-1} D \cdot U_t + \sum_{t=k+1}^{T} c \cdot B_t \mid \mathcal{F}_k \right\} . \tag{4}$$

In both cases, we allow the sleep policy $\boldsymbol{\pi}$ to be chosen from the set of all randomized and deterministic control laws, $\boldsymbol{\Pi}$, such that $U_t = \pi_t(Y^t, U^{t-1})$, $\forall t$, where $Y^t := (Y_0, Y_1, \ldots, Y_t)$ and $U^{t-1} := (U_0, U_1, \ldots, U_{t-1})$.

In the next two sections, we study the infinite horizon (**P1**) and finite horizon (**P2**) problems, respectively.

III. Analysis of the Infinite Horizon Average Expected Cost Problem

In this section, we characterize the optimal sleep control policy $\boldsymbol{\pi}^*$ that minimizes (3). We begin by showing the existence of an optimal stationary Markov policy. We then show that the optimal policy is a threshold policy of the form: stay awake if and only if $S = 0$ and $B \geq \lambda^*$, where $\lambda^* = 0$ (never sleep) or $\lambda^* = 1$ (sleep only when the system empties out), depending on the parameters N, p, c, and D. As a matter of notation, we refer to the threshold policy with $\lambda^* = 0$, often called the "0-policy," as $\boldsymbol{\pi_0}$, and the threshold policy with $\lambda^* = 1$, often called the "1-policy," as $\boldsymbol{\pi_1}$ [17].

A. Existence of an Optimal Stationary Markov Policy

Due to the assumption of an infinite buffer size, the controlled Markov chain in Problem (**P1**) has a countably infinite state space. Recall that for such systems, an average cost optimal stationary policy is not guaranteed to exist. See [18, pp. 128–132] for such counterexamples. However, [18] also presents sufficient conditions for the existence of an average cost optimal stationary policy. We recall these conditions below and then show that the (BOR) set of assumptions is satisfied by Problem (**P1**).

Theorem 1 (Sennott): Assume that the following set (BOR) of assumptions holds (notations are explained following the theorem):
(BOR1). There exists a z-standard policy $\mathbf{g}$ with positive recurrent class R^g.
(BOR2). There exists $\epsilon > 0$ such that $G = \{i | C(i,u) \leq J^g + \epsilon$ for some $u\}$ is a finite set.
(BOR3). Given $i \in \{G - R^g\}$, there exists a policy $\theta_i \in \Re^*(z,i)$.

Then there exists a finite constant J and a finite function h, bounded below in i such that:

$$J + h(i) = \min_{u\in\mathcal{U}} \left\{ C(i,u) + \sum_j P_{ij}(u) \cdot h(j) \right\} , \ \forall i \in \mathcal{X}. \tag{5}$$

Moreover, a stationary policy $\mathbf{e}$ satisfying:

$$C(i,e(i)) + \sum_j P_{ij}(e(i)) \cdot h(j) = \min_{u\in\mathcal{U}} \left\{ C(i,u) + \sum_j P_{ij}(u) \cdot h(j) \right\} = J + h(i) , \ \forall i \in \mathcal{X} \tag{6}$$

is average cost optimal.

Remarks on Theorem 1: A Markov chain is said to be *z-standard* if there exists a distinguished state z such that the expected first passage time and expected first passage cost from state i to state z are finite for all $i \in \mathcal{X}$. A (randomized or stationary) policy $\mathbf{g}$ is said to be a *z-standard policy* if it induces a z-standard Markov chain. $C(i,u)$ is the one slot cost incurred at state i under control action u. J^g is the average cost per unit time under policy g. $\Re^*(z,i)$, where z refers to the distinguished state mentioned above, is the class of policies θ such that:

(i) $P_\theta(X_t = i, \text{ for some } t \geq 1 | X_0 = z) = 1$.
(ii) The expected time of first passage from z to i is finite.
(iii) The expected cost of first passage from z to i is finite.

The constant J represents the minimum average cost per unit time. Note that under the (BOR) assumptions, the minimum average cost is constant and therefore independent of the initial state. This is not true in general, even when an optimal policy exists. References [19] and [21] interpret the function h as a rough measure of how much we would pay to stop the process, but continue to incur a cost of J per slot thereafter. In this manner, h can be viewed as a cost potential function.

We now show that the hypotheses of Theorem 1 are met by Problem (**P1**).

Lemma 1: Problem (**P1**) satisfies the (BOR) assumptions of Theorem 1, and therefore, there exists an optimal stationary policy $\boldsymbol{\pi}^*$ that minimizes (3).

Proof: Let the distinguished state z be $\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]$ (the node is awake and the queue is empty). Consider the policy $\boldsymbol{\pi_0}$ of never sleeping (the case of a $\lambda^* = 0$ threshold). Given a fixed but arbitrary initial state $\left[\begin{smallmatrix}b_0\\0\end{smallmatrix}\right]$, the policy $\boldsymbol{\pi_0}$ induces a finite state Markov chain with a single positive recurrent class. In particular, the finite set of transient states is

$$T^{\boldsymbol{\pi_0}} = \left\{ \begin{bmatrix} b_0 \\ 0 \end{bmatrix}, \begin{bmatrix} b_0 - 1 \\ 0 \end{bmatrix}, \cdots, \begin{bmatrix} 2 \\ 0 \end{bmatrix} \right\},$$

the set of recurrent states is

$$R^{\boldsymbol{\pi_0}} = \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right\},$$

and the transition diagram is shown in Figure 1.

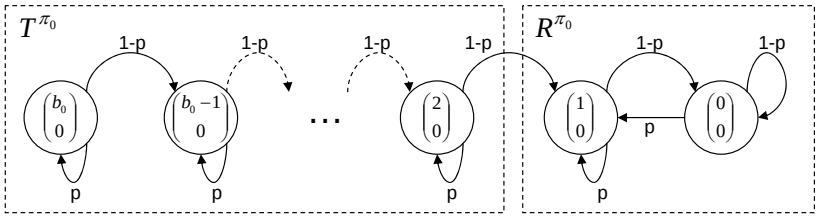

Fig. 1. Transition diagram induced by $\boldsymbol{\pi_0}$

For finite state Markov chains with a single positve recurrent class, the following three basic facts are true (see for example [22], [23]):

(i) The process enters the positive recurrent class (exits the transient states) in finite time

with probability 1, and subsequently reaches each state in the recurrent class in finite time with probability 1.

(ii) There exists a unique stationary distribution, $\bar{\mathbf{s}}^{\mathbf{g}}$, with

$$\bar{\mathbf{s}}^{\mathbf{g}} = \bar{\mathbf{s}}^{\mathbf{g}} \cdot P^{\mathbf{g}} \text{ and } \sum_{i \in \mathcal{X}} \bar{s}^{\mathbf{g}}(i) = 1.$$

(iii) The long run average cost $J^{\mathbf{g}}$ is equal to $\bar{\mathbf{s}}^{\mathbf{g}} \cdot \bar{\mathbf{c}}^{\mathbf{g}T}$, where $\bar{c}^{\mathbf{g}}(i) = C(i, g(i))$, the one slot cost at state i under action $g(i)$.

Thus, the first passage time from any state in the Markov chain induced by policy $\boldsymbol{\pi_0}$ to state $\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right] \in R^{\boldsymbol{\pi}_0}$ is finite w.p.1 by (i) above. A finite sum of bounded one slot costs is finite, and it therefore follows that the expected first passage cost from any state to $\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]$ is also finite under $\boldsymbol{\pi_0}$. We conclude $\boldsymbol{\pi_0}$ is a z-standard policy with positive recurrent class $R^{\boldsymbol{\pi}_0}$, and (BOR1) is satisfied.

Next, we calculate the average cost per unit time under $\boldsymbol{\pi_0}$ and examine the set $G^{\boldsymbol{\pi}_0}$. The unique stationary distribution under this policy is given by:

$$\bar{s}^{\boldsymbol{\pi}_0}(i) = \begin{cases} 1-p, & i = \left[\begin{smallmatrix}0\\0\end{smallmatrix}\right] \\ p, & i = \left[\begin{smallmatrix}1\\0\end{smallmatrix}\right] \\ 0\ , & \text{otherwise} \end{cases} . \tag{7}$$

In general for our model, $C(i,u) = c \cdot i_b + D \cdot u$. Under the 0-policy, u is always equal to 1, so we have:

$$C^{\boldsymbol{\pi}_0}(i,u) = C\left(i, \pi_0(i)\right) = D + c \cdot i_b\ . \tag{8}$$

Combining (7), (8), and property (iii) above, we get the average cost per unit time:

$$\begin{aligned} J^{\boldsymbol{\pi}_0} &= \sum_{i \in \mathcal{X}} s^{\boldsymbol{\pi}_0}(i) \cdot C\left(i, \pi_0(i)\right) \\ &= (1-p) \cdot D + p \cdot (D + c) \\ &= D + pc\ . \end{aligned} \tag{9}$$

Taking $\epsilon = \frac{1}{2}$ and setting $u = 0$, we have:

$$\begin{aligned} G^{\boldsymbol{\pi}_0} &= \{i \mid C(i,u) \leq J^{\boldsymbol{\pi}_0} + \epsilon \text{ for some } u\} \\ &= \left\{ i \in \mathcal{X} \mid c \cdot i_b \ \leq \ D + pc + \frac{1}{2} \right\} \\ &= \left\{ i \in \mathcal{X} \mid i_b \ \leq \ \frac{D}{c} + p + \frac{1}{2c} \right\}. \end{aligned}$$

Therefore, $G^{\boldsymbol{\pi}_0}$ is a finite set, and (BOR2) is satisfied.

Finally, let $j \in G^{\boldsymbol{\pi}_0}$ be arbitrary. Consider the policy θ_j of sleeping at state $x \in \mathcal{X}$ if $x_b < j_b$ or if $x_s > 0$, and serving if $x_s = 0$ and $x_b \geq j_b$. Then, $\theta_{\mathbf{j}} \in \Re^*\left(\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right], j\right)$, as the induced chain visits state j w.p.1, and the expected first passage time and cost from $\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]$ to j under policy θ_j are both finite. Thus, (BOR3) is also satisfied by Problem (**P1**), and we conclude that there exists an optimal stationary policy $\boldsymbol{\pi}^*$ that minimizes (3). ■

B. Optimal Policy When Queue Is Non-Empty

We now begin to identify the optimal stationary policy at each state in the state space.

Lemma 2: The optimal control at state $i = \begin{bmatrix} n \\ 0 \end{bmatrix}$ is $U^* = 1$, for all $n \in \mathbb{N}$, $n \geq 1$.

Proof: Let $n \in \mathbb{N}$, $n \geq 1$ be arbitrary. Assume the state at time k is $X_k = \left[\begin{smallmatrix} n \\ 0 \end{smallmatrix}\right]$. Consider the following three policies:

- $\hat{\boldsymbol{\pi}}$: stay awake for the $[k, k+1)$ slot, and behave optimally thereafter.
- $\bar{\boldsymbol{\pi}}$: go to sleep for N slots, and behave optimally thereafter.
- $\tilde{\boldsymbol{\pi}}$: stay awake for the $[k, k+1)$ slot, and then sleep; if $\bar{U}_{l_0} = 1$ (i.e. the server stays awake under $\bar{\boldsymbol{\pi}}$) at any time $l_0 \geq k+N$, then let $\tilde{U}_l = \bar{U}_l, \ \forall l > l_0$; otherwise, continue to sleep.

It is clear that $\hat{\boldsymbol{\pi}}$ is superior to $\tilde{\boldsymbol{\pi}}$ by construction, so we need to show that $\tilde{\boldsymbol{\pi}}$ is superior to $\bar{\boldsymbol{\pi}}$. If the node continues to sleep forever under $\bar{\boldsymbol{\pi}}$, the queue length grows *ad infinitum* since $p > 0$. This results in $J^{\bar{\boldsymbol{\pi}}} = \infty$, due to the linear holding cost structure. Yet, we have already shown there exists at least one policy, $\boldsymbol{\pi_0}$, with a finite average cost. Therefore, the policy of sleeping for all slots after time $k+N$ is suboptimal, and cannot occur under $\bar{\boldsymbol{\pi}}$. So eventually the node will awake under $\bar{\boldsymbol{\pi}}$.

Let τ denote the number of slots from time k until the first time the node awakes under policy $\bar{\boldsymbol{\pi}}$. We now compare the evolution of the Markov chain under $\bar{\boldsymbol{\pi}}$ and $\tilde{\boldsymbol{\pi}}$. For all realizations, a single packet is served τ slots later under $\bar{\boldsymbol{\pi}}$, and all other packets are served at the same time under both policies. Thus, the total cost from time k under $\bar{\boldsymbol{\pi}}$ is almost surely $\tau \cdot c$ greater than the total cost from time k under $\tilde{\boldsymbol{\pi}}$, and we conclude $\tilde{\boldsymbol{\pi}}$ is superior to $\bar{\boldsymbol{\pi}}$. By transitivity, $\hat{\boldsymbol{\pi}}$ is superior to $\bar{\boldsymbol{\pi}}$. Therefore, it is optimal to stay awake and serve at $\left[\begin{smallmatrix} n \\ 0 \end{smallmatrix}\right]$, for all $n \in \mathbb{N}$, $n \geq 1$. ■

C. Complete Characterization of the Optimal Policy

We now present the main result of this section.

Theorem 2: In Problem (**P2**), the optimal control at state $X = \begin{bmatrix} B \\ 0 \end{bmatrix}$ such that $B > 0$, is $U^* = 1$. At the boundary state $\left[\begin{smallmatrix} 0 \\ 0 \end{smallmatrix}\right]$, the optimal control, U^*, is given by:

$$\left(\frac{p}{1-p}\right)\left(\frac{N-1}{2}\right) \underset{U^*=1}{\overset{U^*=0}{\lessgtr}} \frac{D}{c} \ . \tag{10}$$

Proof: The first statement follows directly from Lemma 2. We showed in the proof of Lemma 1 that the average cost per unit time under the 0-policy (never sleep) is $D + pc$. We now know from Lemma 2 that the 0-policy is optimal at every awake state, except possibly the boundary $\left[\begin{smallmatrix} 0 \\ 0 \end{smallmatrix}\right]$. To determine the optimal policy at this state, we must compare the average cost per unit time of the 0-policy with that of the 1-policy (serve if the queue is non-empty, and sleep otherwise). The transition diagram under $\boldsymbol{\pi_1}$ is shown in Figure 2, with $T^{\boldsymbol{\pi_1}}$ denoting the set of transient states, and $R^{\boldsymbol{\pi_1}}$ denoting the single positive recurrent class.

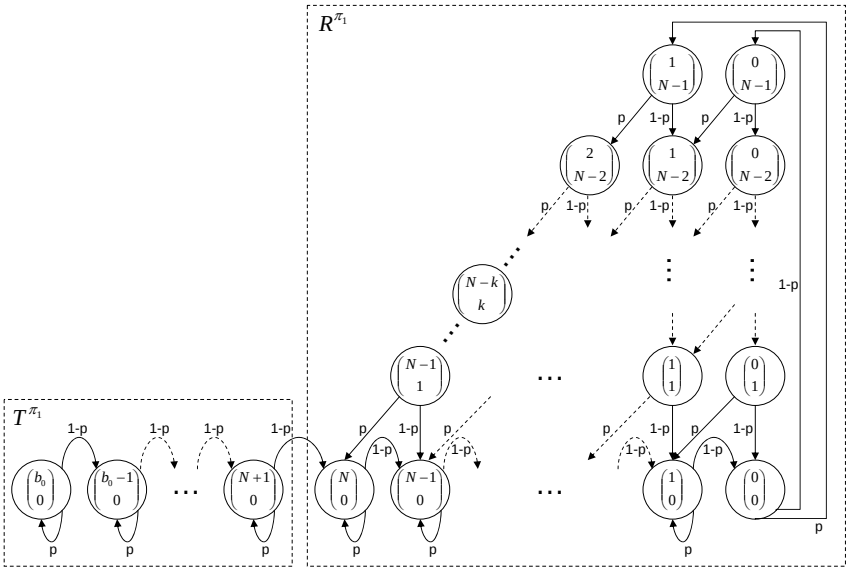

Fig. 2. Transition diagram induced by $\boldsymbol{\pi_1}$

Once again, this Markov chain has a unique stationary distribution, and it is straightforward to verify that the balance equations hold for the following stationary distribution:

$$\bar{s}^{\boldsymbol{\pi_1}}(i) \quad = \begin{cases} \frac{1-p}{N}, & i = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ \frac{1}{N} \cdot \sum_{m=0}^{N-j} \left\{ \binom{N}{m} \cdot (1-p)^m \cdot p^{N-m} \right\}, & i = \begin{bmatrix} j \\ 0 \end{bmatrix},\ j = 1, 2, \ldots, N \\ \frac{1-p}{N} \cdot \binom{k}{l} \cdot p^l \cdot (1-p)^{k-l}, & i = \begin{bmatrix} l \\ N-k \end{bmatrix}, \quad \begin{matrix} 1 \le k \le N-1 \\ 0 \le l \le k \end{matrix} \\ 0\ , & \text{otherwise} \end{cases} \tag{11}$$

From (11), we compute the average cost per unit time:

$$
\begin{aligned}
J^{\boldsymbol{\pi}_1} &= \sum_{i\in\mathcal{X}} s^{\boldsymbol{\pi}_1}(i)\cdot C(i,\pi_1(i)) \\
&= 0\cdot\frac{1-p}{N}+\sum_{j=1}^{N}\left\{(D+jc)\cdot\frac{1}{N}\cdot\sum_{m=0}^{N-j}\left\{\binom{N}{m}\cdot(1-p)^m\cdot p^{N-m}\right\}\right\} \\
&\quad+\sum_{k=1}^{N-1}\sum_{l=0}^{k}\left\{(lc)\cdot\frac{1-p}{N}\cdot\binom{k}{l}\cdot p^l\cdot(1-p)^{k-l}\right\} \\
&= \frac{D}{N}\cdot\sum_{j=1}^{N}\sum_{m=0}^{N-j}\left\{\binom{N}{m}\cdot(1-p)^m\cdot p^{N-m}\right\} \\
&\quad+\frac{c}{N}\cdot\sum_{j=1}^{N} j\cdot\sum_{m=0}^{N-j}\left\{\binom{N}{m}\cdot(1-p)^m\cdot p^{N-m}\right\} \\
&\quad+\frac{c(1-p)}{N}\cdot\sum_{k=1}^{N-1}\sum_{l=0}^{k}\left\{l\cdot\binom{k}{l}\cdot p^l\cdot(1-p)^{k-l}\right\} \\
&= \frac{D}{N}\cdot pN+\frac{c}{N}\cdot\left(\frac{p^2N(N-1)}{2}+pN\right)+\frac{c(1-p)}{N}\cdot\sum_{k=1}^{N-1}pk \\
&= pD+\frac{p^2c(N-1)}{2}+pc+\frac{c(1-p)}{N}\cdot\frac{pN(N-1)}{2} \\
&= pD+pc+\frac{pc(N-1)}{2}\cdot(p+(1-p)) \\
&= pD+\frac{pc(N+1)}{2}\quad . \qquad (12)
\end{aligned}
$$

Finally, combining (9) and (12), we compare the average costs for the two policies to determine the optimal policy at the boundary state $\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]$:

$$
J^{\boldsymbol{\pi}_1} = pD+\frac{pc(N+1)}{2} \underset{U^*=1}{\overset{U^*=0}{\lessgtr}} D+pc = J^{\boldsymbol{\pi}_0}\ . \qquad (13)
$$

Rearranging (13) gives (10). ■

D. Related Work and Possible Extensions

The arguments presented above are quite similar to those applied to the embedded Markov chain model of [17]. In that paper, Federgruen and So consider an analogous problem in continuous time with compound Poisson arrivals. By formulating the problem as a semi-Markov decision process embedded at certain decision epochs, they show that either a no vacation policy or a threshold policy is optimal under a much weaker set of assumptions. Specifically, they allow general non-decreasing holding costs, multiple arrivals, fixed costs for switching between service and vacation modes, and general i.i.d. service and vacation times. It is quite possible that we could similarly relax our assumptions, and still retain the structural result that either a threshold policy or a no vacation policy is optimal. We have not yet explored this extension. By imposing

the extra assumptions, however, we have arrived at the more specific conclusion that if the optimal policy is an N-threshold policy, it is indeed a 1-policy; additionally, we have identified condition (10), distinguishing the parameter sets on which the 0-policy is optimal from those on which the 1-policy is optimal.

IV. Analysis of the Finite Horizon Expected Cost Problem

In this section, we analyze the finite horizon problem, (**P2**), and attempt to characterize the optimal sleep control policy $\boldsymbol{\pi}^*$ that minimizes $J_0^{\boldsymbol{\pi}}$. Due to the finite time horizon and the assumption of a finite initial queue size, this problem features a finite state space (at most $[B_0 + T] \cdot N$ states). Additionally, we have a finite number of available control actions at each time slot. For such systems, we know the following from classical stochastic control theory (see for example [24, pp. 78–79]):

(i) There exists an optimal control policy; i.e., a policy $\boldsymbol{\pi}^*$ such that

$$J_0^{\boldsymbol{\pi}^*} = \inf_{\boldsymbol{\pi}} J_0^{\boldsymbol{\pi}} \ , \tag{14}$$

where the infimum in (14) is taken over all randomized and deterministic history-dependent policies.

(ii) Furthermore, there exists an optimal deterministic Markov policy (a policy that depends only on the current state X_k, not the past states $X_{k-1}, X_{k-2}, \ldots$).

(iii) Define recursively the functions

$$\begin{aligned} V_T(i) &:= c \cdot i_b \\ V_k(i) &:= \min_{u \in \{0,1\}} \left\{ c \cdot i_b + u \cdot D + \sum_{j \in \mathcal{X}} \left[P_{ij}(u) \cdot V_{k+1}(j) \right] \right\} \ \forall k \in \{0, 1, \ldots, T-1\}. \end{aligned} \tag{15}$$

A deterministic Markov policy $\boldsymbol{\pi}$ is optimal if and only if the minimum in (15) is achieved by $\pi_k(i)$, for each state i at each time k.

Next, we define the "cost-to-go" random variable associated with policy $\boldsymbol{\pi}$ over the time interval $[k, T]$:

$$C_k^{\boldsymbol{\pi}} := \sum_{t=k+1}^{T} c \cdot B_t + \sum_{t=k}^{T} D \cdot U_t \ ,$$

and the expected cost-to-go given all information through time k:

$$J_k^{\boldsymbol{\pi}} := E\left[C_k^{\boldsymbol{\pi}} \mid \mathcal{F}_k \right] \ .$$

The interpretation of (iii) is that

$$J_k^{\boldsymbol{\pi}^*} = \inf_{\boldsymbol{\pi}} J_k^{\boldsymbol{\pi}} \ \ \forall k \in \{0, 1, \ldots, T-1\}.$$

While, in principle, we can compute the optimal policy through the dynamic program (15), we are more interested in deriving structural results on the optimal policy, e.g., by showing that the optimal policy satisfies certain properties or is of a certain simple form. In order to accomplish this, we use the above results throughout the section to identify the optimal control

at each slot by comparing the expected cost-to-go under different deterministic Markov policies. Before proceeding, we note that for the remainder of this section, when we refer to the time k, we implicity assume $k \in \{0, 1, \ldots, T-1\}$.

A. Optimal Policy at the End of the Time Horizon

As with the infinite horizon problem, we identify the optimal policy in a piecewise manner, this time beginning with the slots at the end of the time horizon.

Lemma 3: If $T - \frac{D}{c} \leq k < T$, the optimal policy to minimize $J_k^{\boldsymbol{\pi}}$ is $U_t^* = 0 \quad \forall t \in \{k, k+1, \ldots, T-1\}$; i.e. sleep for the duration of the time horizon.

Proof: We proceed by backward induction on k. Let $l = T - k$. To prove this lemma, we essentially need to prove that the following hypothesis $h(l)$ is true for all l:

$$h(l) := \quad \text{Assuming } \tfrac{D}{c} \geq l, \text{ the policy } U_t^* = 0, \ \forall t \in \{T-l, T-l+1, \ldots, T-1\} \text{ minimizes } J_{T-l}^{\boldsymbol{\pi}}. \tag{16}$$

(i) *Induction Basis*: $l = 1$ $(k = T-1)$.

This is the case when we are choosing a control for the final slot, $[T-1, T)$. If there are no jobs queued, staying awake costs D and provides no reward. If there is a job queued, the net reward for staying awake is $c - D$; however, $l \cdot c = c \leq D$ by assumption of $h(1)$, and thus it is optimal to sleep. We conclude hypothesis $h(1)$ is true.

(ii) *Induction Step*: Let $l \in \left\{1, 2, \ldots, \left\lfloor \frac{D}{c} \right\rfloor - 1\right\}$ be arbitrary (corresponds to $k \in \left\{T-1, T-2, \ldots, \left\lceil T - \frac{D}{c} \right\rceil + 1\right\}$). Assume $h(l)$ is true, and show $h(l+1)$ is true.

We are now choosing the control for the slot $[T-l-1, T-l)$. By the assumption of $h(l+1)$, we know $\frac{D}{c} \geq l+1$, which implies $\frac{D}{c} \geq l$. Thus, by the induction hypothesis, we know that the node will go to sleep at time l, and remain asleep for the remainder of the time horizon. As with the base case, if the queue length at time l is zero, staying awake costs D with no reward. If $X_{T-l-1} = \begin{bmatrix} B_{T-l-1} \\ 0 \end{bmatrix}$ for some $B_{T-l-1} > 0$, the net reward from staying awake is $(l+1) \cdot c - D$. Yet, by $h(l+1)$, $(l+1) \cdot c \leq D$, and thus the net reward is non-positive. We conclude the optimal control at time $T-l-1$ is $U_{T-l-1}^* = 0$. Combined with the knowledge from the induction hypothesis that the node will sleep for the duration of the time horizon beginning at time l, this completes the induction step and the proof of the lemma. ■

The simple intuition behind the above lemma is that the incremental cost of staying awake for an extra slot remains constant at D throughout the time horizon; however, the benefit of doing so, as compared to sleeping for the duration of the horizon, diminishes as t approaches T.

B. Optimal Policy When Queue Is Non-Empty Before the End of the Time Horizon

The following lemma characterizes the optimal sleep policy when the node is awake, the queue is non-empty, and the process is sufficiently far from the end of the time horizon.

Lemma 4: If $0 \leq k < T - \frac{D}{c}$ and $X_k = \begin{bmatrix} B_k \\ 0 \end{bmatrix}$ for some $B_k > 0$, the optimal control at slot k to minimize $J_k^{\boldsymbol{\pi}}$ is $U_k^* = 1$; i.e., serve a job in slot $[k, k+1)$.

Proof: We consider two separate cases.

Case 1: $k \geq T - N$.

Consider the following three policies:

- $\hat{\boldsymbol{\pi}}$: stay awake for the $[k, k+1)$ slot, and behave optimally thereafter.
- $\bar{\boldsymbol{\pi}}$: go to sleep (and remain asleep for duration of time horizon).
- $\tilde{\boldsymbol{\pi}}$: stay awake for the $[k, k+1)$ slot, and then sleep (for duration of time horizon).

Define the rewards for following $\hat{\boldsymbol{\pi}}$ over $\bar{\boldsymbol{\pi}}$, $\tilde{\boldsymbol{\pi}}$ over $\bar{\boldsymbol{\pi}}$, and $\hat{\boldsymbol{\pi}}$ over $\tilde{\boldsymbol{\pi}}$:

$$\begin{aligned} R_k &:= C_k^{\bar{\boldsymbol{\pi}}} - C_k^{\hat{\boldsymbol{\pi}}} \ , \\ R_k^1 &:= C_k^{\bar{\boldsymbol{\pi}}} - C_k^{\tilde{\boldsymbol{\pi}}} \ , \text{ and} \\ R_k^2 &:= C_k^{\tilde{\boldsymbol{\pi}}} - C_k^{\hat{\boldsymbol{\pi}}} \ , \text{ respectively.} \end{aligned}$$

To show that $\hat{\boldsymbol{\pi}}$ is optimal in this case, it suffices to show:

$$E\left[R_k \mid \mathcal{F}_k\right] = E\left[R_k^1 \mid \mathcal{F}_k\right] + E\left[R_k^2 \mid \mathcal{F}_k\right] \geq 0 \quad w.p.1 \ . \tag{17}$$

This is fairly straightforward as we have:

$$E\left[R_k^1 \mid \mathcal{F}_k\right] = c \cdot (T-k) - D \quad w.p.\ 1, \quad \text{and} \tag{18}$$

$$E\left[R_k^2 \mid \mathcal{F}_k\right] \geq 0 \quad w.p.1, \text{ by construction.} \tag{19}$$

Combining these,

$$k < T - \tfrac{D}{c}, \ (18), \text{ and } (19) \ \Rightarrow \ (17) \ , \tag{20}$$

concluding the proof under Case 1.

Case 2: $k < T - N$.

Redefine the policies $\hat{\boldsymbol{\pi}}$, $\bar{\boldsymbol{\pi}}$, and $\tilde{\boldsymbol{\pi}}$ as follows:

- $\hat{\boldsymbol{\pi}}$: stay awake for the $[k, k+1)$ slot, and behave optimally thereafter.
- $\bar{\boldsymbol{\pi}}$: go to sleep for N slots, and behave optimally thereafter.
- $\tilde{\boldsymbol{\pi}}$: stay awake for the $[k, k+1)$ slot, and then sleep for N slots; if $\bar{U}_{l_0} = 1$ (i.e. the server stays awake under $\bar{\boldsymbol{\pi}}$) at any time $l_0 \geq k+N$, then let $\tilde{U}_l = \bar{U}_l \ \ \forall l > l_0$; otherwise, continue to sleep for the duration of the time horizon.

Let R_k, R_k^1, and R_k^2 be as in Case 1. To show that $\hat{\boldsymbol{\pi}}$ is optimal in this case, it once again suffices to show:

$$E\left[R_k \mid \mathcal{F}_k\right] = E\left[R_k^1 \mid \mathcal{F}_k\right] + E\left[R_k^2 \mid \mathcal{F}_k\right] \geq 0 \quad w.p.1 \ . \tag{21}$$

In the case that $\bar{\boldsymbol{\pi}}$ results in the node sleeping for the duration of the time horizon, we have:

$$E\left[R_k^1 \mid \mathcal{F}_k\right] = c \cdot (T-k) - D \geq 0 \quad w.p.1 \ , \tag{22}$$

where the last inequality follows from the assumption $k \leq T - \frac{D}{c}$. In the case that $\bar{\boldsymbol{\pi}}$ results in the node eventually staying awake for a slot, we have:

$$E\left[R_k^1 \mid \mathcal{F}_k\right] \geq c \cdot N \quad w.p.1 \ , \tag{23}$$

because the best case scenario for $\bar{\boldsymbol{\pi}}$ is that the service occurs in the $k + N^{th}$ slot, N slots after the same job is served under policy $\tilde{\boldsymbol{\pi}}$. Under all realizations, all other jobs are served at the same time by $\bar{\boldsymbol{\pi}}$ and $\tilde{\boldsymbol{\pi}}$. From (22) and (23),we conclude:

$$E\left[R_k^1 \mid \mathcal{F}_k\right] \geq 0 \quad w.p.1 \ . \tag{24}$$

Note also that, once again by construction:

$$E\left[R_k^2 \mid \mathcal{F}_k\right] \geq 0 \quad w.p.1 \ . \tag{25}$$

From (24) and (25), we conclude (21) holds, completing the proof of the lemma. ■

C. Optimal Policy When Node Is Awake and Queue Is Empty (Boundary State)

We know from Lemma 3 that the optimal control at $X_k = \left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]$ is to sleep when $k \geq T - \frac{D}{c}$. We now examine the optimal control at this state when $k < T - \frac{D}{c}$.

Lemma 5: If $k = z^* := \left\lfloor T - \frac{D}{c} \right\rfloor$ and $X_k = \left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]$, the optimal control policy to minimize $J_k^{\boldsymbol{\pi}}$ is to sleep for the duration of the time horizon.

Proof: This is trivial as, due to Lemma 3, the optimal policy entails sleeping for the duration of the time horizon, beginning at the following time slot, $z^* + 1$. Therefore, staying awake in the z^* time slot costs D and does not provide any benefit, because the node will not serve any jobs for the remainder of the time horizon. ■

Lemma 6: If $z^* - N < k < z^*$ and $X_k = \left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]$, the optimal control at slot k to minimize $J_k^{\boldsymbol{\pi}}$ is described by the threshold decision rule:

$$c \cdot \sum_{j=1}^{z^*-k} \left\{p^j \left(T - k - j\right)\right\} - D \cdot \sum_{j=0}^{z^*-k} p^j \underset{U^*=1}{\overset{U^*=0}{\lessgtr}} 0 \ . \tag{26}$$

Proof: We once again proceed by backward induction on k. Let $l = z^* - k$, and note that in order to prove this lemma, we need to prove the following hypothesis $h(l)$ for all l:

$$\begin{aligned} h(l) := \quad & \text{If } X_{z^*-l} = \begin{bmatrix}0\\0\end{bmatrix}, \text{ the optimal control at time } z^* - l \text{ to minimize } J_{z^*-l}^{\boldsymbol{\pi}} \\ & \text{is described by the threshold decision rule:} \\ & c \cdot \sum_{j=1}^{l} \left\{p^j \left(T - z^* + l - j\right)\right\} - D \cdot \sum_{j=0}^{l} p^j \underset{U^*=1}{\overset{U^*=0}{\lessgtr}} 0 \ . \end{aligned} \tag{27}$$

Redefine the policies $\hat{\boldsymbol{\pi}}$, $\bar{\boldsymbol{\pi}}$, and $\tilde{\boldsymbol{\pi}}$ once more as follows:

- $\hat{\boldsymbol{\pi}}$: stay awake for the $[k, k+1)$ slot, and behave optimally thereafter.
- $\bar{\boldsymbol{\pi}}$: go to sleep for N slots, and behave optimally thereafter.
- $\tilde{\boldsymbol{\pi}}$: stay awake for the $[k, k+1)$ slot. At each time $k+1, k+2, \ldots, z^*$ if there is a job in the queue, serve it; otherwise, go to sleep.

Let R_k, R_k^1, and R_k^2 be as in Lemma 4.

(i) *Induction Basis*: $l = 1$ $(k = z^* - 1)$.

The goal is to determine the optimal control for slot $[z^* - 1, z^*)$. From Lemmas 4 and 5, we know that if there is a job in the queue at z^*, the optimal policy is to serve, but if there is not, the optimal policy is to sleep. Furthermore, we know from Lemma 3 that the optimal policy is to sleep for the duration of the time horizon beginning at time $z^* + 1$, regardless of the queue size. This knowledge allows us to directly calculate the expected reward from following $\hat{\boldsymbol{\pi}}$ over $\bar{\boldsymbol{\pi}}$:

$$E\left[R_{z^*-1} \mid \mathcal{F}_{z^*-1}\right] = E\left[C_{z^*-1}^{\bar{\boldsymbol{\pi}}} - C_{z^*-1}^{\hat{\boldsymbol{\pi}}} \mid \mathcal{F}_{z^*-1}\right] = -D + p \cdot \left[c \cdot (T - z^*) - D\right] \quad w.p.1 \ . \tag{28}$$

Note that for $l = 1$, the LHS of (27) is equal to the RHS of (28). Therefore, if the LHS of (27) is greater than 0, we have $E\left[R_{z^*-1} \mid \mathcal{F}_{z^*-1}\right] > 0 \ w.p.1$, and the optimal policy is $u^*_{z^*-1} = 1$. Alternatively, if the LHS of (27) is less than 0, we have $E\left[R_{z^*-1} \mid \mathcal{F}_{z^*-1}\right] < 0 \ w.p.1$, and the optimal policy is $U^*_{z^*-1} = 0$. Thus, $h(1)$ is true, and the base case holds.

(ii) *Induction Step*: Let $l \in \{1, 2, \ldots, N-2\}$ be arbitrary (corresponds to $k \in \{z^* - 1, z^* - 2, \ldots, z^* - N + 2\}$). Assume $h(1), h(2), \ldots, h(l)$ are true, and show $h(l+1)$ is true.

We now define the index

$$w(k) := c \cdot \sum_{j=1}^{z^*-k} \left\{p^j \left(T - k - j\right)\right\} - D \cdot \sum_{j=0}^{z^*-k} p^j \ . \tag{29}$$

The following calculation demonstrates that $w(\circ)$ is a non-increasing function in k:

$$\begin{aligned} w(k-1) - w(k) &= \left[c \cdot \sum_{j=1}^{z^*-k+1} \left\{p^j \left(T - k + 1 - j\right)\right\} - D \cdot \sum_{j=0}^{z^*-k+1} p^j\right] \\ &\quad - \left[c \cdot \sum_{j=1}^{z^*-k} \left\{p^j \left(T - k - j\right)\right\} - D \cdot \sum_{j=0}^{z^*-k} p^j\right] \\ &= p^{z^*-k+1} \cdot \left[-D + c \cdot \left(T - z^*\right)\right] + c \cdot \sum_{j=1}^{z^*-k} p^j \\ &= p^{z^*-k+1} \cdot \left(c \cdot \left\lceil \frac{D}{c} \right\rceil - D\right) + c \cdot \sum_{j=1}^{z^*-k} p^j \\ &\geq 0 \quad \forall k \in \{z^* - N + 1, z^* - N + 2, \ldots, z^* - 1\} \ . \end{aligned} \tag{30}$$

From (30), it follows that

$$\begin{aligned} &w(K) \leq 0 \text{ for some } K \in \{z^* - N + 1, z^* - N + 2, \ldots, z^* - 1\} \\ &\quad \Rightarrow w(k) \leq 0 \ \ \forall k \in \{K, K+1, \ldots, z^* - 1\} \ . \end{aligned} \tag{31}$$

To demonstrate the validity of $h(l+1)$, we now consider two exhaustive cases:

Case 1: $w(z^* - l - 1) \leq 0$.

By (31) and $w(z^*-l-1) \leq 0$, we know that $w(t) \leq 0$, for all $t \in \{z^* - l, z^* - l + 1, \ldots, z^* - 1\}$; thus, if $X_t = \left[\begin{smallmatrix} 0 \\ 0 \end{smallmatrix}\right]$ for any $t \in \{z^* - l, z^* - l + 1, \ldots, z^* - 1\}$, the node will sleep for the duration of the time horizon under the optimal policy. Equipped with this full characterization of the optimal policy in all subsequent slots, we can directly calculate the expected reward for following $\hat{\boldsymbol{\pi}}$ over $\bar{\boldsymbol{\pi}}$:

$$E\left[R_{z^*-l-1} \mid \mathcal{F}_{z^*-l-1}\right] = E\left[C^{\bar{\boldsymbol{\pi}}}_{z^*-l-1} - C^{\hat{\boldsymbol{\pi}}}_{z^*-l-1} \mid \mathcal{F}_{z^*-l-1}\right] = w\left(z^* - l - 1\right) \leq 0 \ \ w.p.1. \tag{32}$$

From (32), we conclude that for case 1, the optimal policy is $u^*_{z^*-l-1} = 0$ and $h(l+1)$ holds.

Case 2: $w(z^* - l - 1) > 0$.

In this case, we once again make use of an interchange argument through the policy $\tilde{\boldsymbol{\pi}}$. By construction,

$$E\left[R^2_{z^*-l-1} \mid \mathcal{F}_{z^*-l-1}\right] \geq 0 \ \ w.p.1 \ ,$$

so to show that

$$E\left[R_{z^*-l-1} \mid \mathcal{F}_{z^*-l-1}\right] \geq 0 \ w.p.1 \ . \tag{33}$$

it suffices to show that

$$E\left[R^1_{z^*-l-1} \mid \mathcal{F}_{z^*-l-1}\right] \geq 0 \ w.p.1 \ .$$

Because we know the the control policy corresponding to every realization under both policies, we can directly calculate the expected reward for following $\tilde{\boldsymbol{\pi}}$ over $\bar{\boldsymbol{\pi}}$:

$$E\left[R^1_{z^*-l-1} \mid \mathcal{F}_{z^*-l-1}\right] = E\left[C^{\tilde{\boldsymbol{\pi}}}_{z^*-l-1} - C^{\bar{\boldsymbol{\pi}}}_{z^*-l-1} \mid \mathcal{F}_{z^*-l-1}\right] = w\left(z^* - l - 1\right) > 0 \quad w.p.1. \tag{34}$$

(34) implies (33), which in turn implies $u^*_{z^*-l-1} = 1$ and $h(l+1)$ holds. This concludes the induction step under case 2, and the proof of the lemma. ■

We note that Lemma 6 and its proof tell us that from slot $z^* - N + 1$ until slot $z^* - 1$, the optimal policy when the node is awake and the queue is empty is non-increasing over time. We also know from Lemmas 3 and 5 that the optimal control is $U^*_k = 0$, for all $k \geq z^*$. Combining these, we know the optimal policy at $X_k = \left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]$ is non-increasing over time, from slot $z^* - N + 1$ until the end of the time horizon. The natural follow-up question to ask is whether or not the optimal policy at $X_k = \left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]$ is necessarily monotonic over the entire duration of the time horizon. Intuitively, this might make sense if we extend the logic behind Lemma 3 to conclude that the marginal reward for serving a packet continues to increase as we move away from the end of the time horizon. However, as we explain further in Section IV-D, this intuition is not quite correct, as the following counterexample demonstrates.

Counterexample 1: Consider Problem (**P2**) with the parameters $T = 15$, $N = 3$, $c = 10$, $D = 21$, and $p = \frac{2}{3}$. The optimal sleep control policy at the boundary state $X_k = \left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]$, computed through the dynamic program (15), is displayed in Figure 3. Clearly, this policy is not monotonic in time.

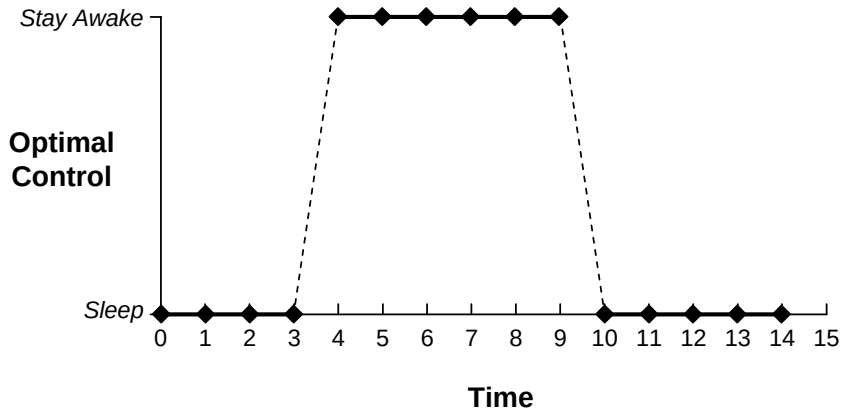

Fig. 3. Optimal control policy at $X_k = \left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]$ when $T = 15$, $N = 3$, $c = 10$, $D = 21$, and $p = \frac{2}{3}$

With such counterexamples in mind, we seek sufficient conditions for the optimal policy at the boundary state to be non-increasing over the entire time horizon. Based on the extensive numerical experiments we conducted, we believe the following conjecture is true, but have not yet been able to prove it.

Conjecture 1: If the parameters of problem **(P2)** satisfy the following condition:

$$\left(\frac{p}{1-p}\right)\cdot\left(\frac{N-1}{2}\right)\geq\frac{D}{c}\ , \tag{35}$$

the optimal policy when the node is awake and the queue is empty is non-increasing in time; i.e., if the expected cost-to-go $V_r\left(\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]\right)$ is minimized by sleeping, then for all $t>r$, the expected cost-to-go $V_t\left(\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]\right)$ is minimized by sleeping.

We have been able to show that if the expected cost-to-go function satisfies the following "supermodularity" condition:

$$V_t\left(\left[\begin{smallmatrix}1\\0\end{smallmatrix}\right]\right)-V_t\left(\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]\right)\leq V_{t+1}\left(\left[\begin{smallmatrix}1\\0\end{smallmatrix}\right]\right)-V_{t+1}\left(\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]\right)\quad\forall t\in\{0,1,\ldots,z^*-N\}\ , \tag{36}$$

then the optimal policy when the node is awake and the queue is empty is monotonically non-increasing in time. Thus, one possible method to complete the proof of Conjecture 1 would be to show that (35) $\Rightarrow$ (36), and then invoke the above fact; however, we have not yet been able to show this relationship.

Assuming the previous conjecture turns out to be true, we would also like to characterize the optimal policy at the boundary state when the parameters of Problem **(P2)** do not satisfy condition (35). One might think that the periodic nature of sleeping would lead to a periodic optimal policy at the boundary; however, based on numerical results, we believe the optimal policy at the boundary is still relatively "smooth," and can be characterized by the following conjecture.

Conjecture 2: If the parameters of problem **(P2)** satisfy the following condition:

$$\left(\frac{p}{1-p}\right)\cdot\left(\frac{N-1}{2}\right)<\frac{D}{c}\ , \tag{37}$$

and if for some k, the optimal control at state $X_k=\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]$ is $U_k^*=0$ and the optimal control at state $X_{k+1}=\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]$ is $U_{k+1}^*=1$, then for all $0\leq t<k$, the optimal control at state $X_t=\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]$ is $U_t^*=0$.

Conjecture 2 essentially says that there can be at most one jump up in the optimal control from $U_t^*=0$ at $X_t=\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]$ to $U_{t+1}^*=1$ at $X_{t+1}=\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]$.

D. Discussion

In this section, we discuss the numerical results supporting our belief in Conjectures 1 and 2, the intuition behind the conjectures, their implications if they turn out to be true, and the challenges we face in proving them.

If Conjectures 1 and 2 turn out to be true, then they imply, in combination with Lemmas 3-6, that the optimal control policy at $X_k=\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]$ is of the form:

$$U_k^*=\begin{cases}1\ (\text{serve}), & \text{if } \lambda_1^*\leq k<\lambda_2^*\\ 0\ (\text{sleep}), & \text{otherwise}\end{cases}\ ,$$

for some $\lambda_1^*,\lambda_2^*\in\{0,1,\ldots,z^*\}$, with $\lambda_1^*\leq\lambda_2^*$. Specifically, only three structural forms of the optimal control policy at $\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]$ are possible. These are shown in Figure 4.

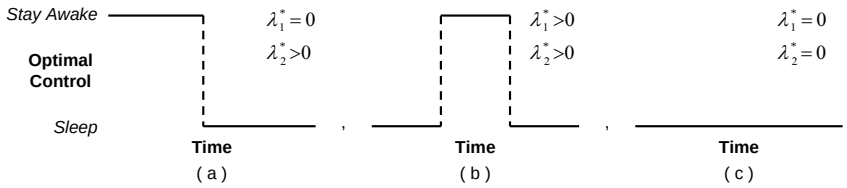

Fig. 4. Possible structural forms for the optimal control policy at $X_k = \left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]$

Moreover, Conjecture 1 states that form (b) is not possible if condition (35) holds. Our numerical results not only support these conclusions, but also show the following:

Observation 1: If the time horizon is sufficiently long, then in fact the optimal control is of the form (a) if condition (35) holds, but of the form (b) or (c) if the negation, (37), holds.

We now attempt to provide some intuition as to why the optimal policy at the boundary state could be of form (b). The underlying tradeoff at the state $\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]$ is between staying awake to reduce backlog costs and sleeping to avoid unutilized slots. In the infinite horizon problem, consider the two policies $\boldsymbol{\pi_0}$ (always awake) and $\boldsymbol{\pi_1}$ (sleep only at boundary state) described in Section III, and assume the node is at state $\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]$ at some time k. In our model, the order in which packets are served is of no importance (e.g. FIFO, LIFO). Therefore, let us assume that for every sample path, the packets arriving from time $k+N-1$ onward are served at exactly the same time under the two policies (by appropriate reordering of packets). Then the extra backlog charges incurred under $\boldsymbol{\pi_1}$ are entirely due to the packets arriving during $(k, k+N-1)$. If there are M arrivals during this period, the queue length at time $k+N$ under $\boldsymbol{\pi_1}$ is M more than the queue length under $\boldsymbol{\pi_0}$. With each non-arrival after time $k+N-1$, $\boldsymbol{\pi_1}$ "catches up" to $\boldsymbol{\pi_0}$ by one packet. Eventually, after M non-arrivals, the two policies will have served the same number of jobs and both will end up back at the state $\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]$. If we compare the expected energy charges incurred by $\boldsymbol{\pi_1}$ during the N unutilized slots of one such cycle to the expected extra backlog costs incurred by $\boldsymbol{\pi_0}$, we get (10), which describes the optimal stationary policy at the boundary state in the infinite horizon case.

Returning to the finite horizon problem, we see that (35) and (37) together are equivalent to (10). Let us now reconsider the two policies from the previous paragraph in the finite horizon context. The probability that the sleep policy catches up to the always awake policy before z^*+1, the time at which the node goes to sleep for good, increases as $t \to 0$. So Observation 1 makes intuitive sense as it just states that the optimal control at the boundary state in the finite horizon problem converges to the optimal control at the boundary state in the infinite horizon problem as we move farther and farther back from the end.

As we move closer to the end of the horizon, there is a higher probability of reaching time z^*+1 before the two policies reach the same state again. Any "extra" packets at z^*+1 will be charged for the rest of the time horizon, which has length $\left\lfloor \frac{D}{c} \right\rfloor$. This extra risk of going to sleep is likely the reason why form (b) is a possible form of the optimal policy. The middle bump in the policy plays the role of a "buffer zone" that incorporates the risk of unserved packets incurring charges throughout the shutdown zone at the end of the horizon.

Observation 2: The structural forms in Figure 4 lie on a spectrum in the sense that changing one parameter at a time leads to a shift in the form of the optimal policy from either form (a) to form (b) to form (c), or from form (c) to form (b) to form (a). In particular, holding all other parameters constant, the form of the optimal policy shifts from (c) to (b) to (a) as we individually

(or collectively) increase p, N, or c, but shifts from (a) to (b) to (c) as D increases. Analogous statements can also be made concerning the movements of the two individual thresholds with variations in the parameters.

We want to mention one last implication of the conjectures regarding the actual computation of the optimal policy. If Conjecture 1 turns out to be true, then we can use an index to calculate the threshold λ_2^*, which completes the specification of the optimal sleep policy when (35) is true. This is done in the following manner:

(i) For every $k \in \{0, 1, \ldots, z^* - 1\}$, write $k = z^* - j \cdot N - l$, where $j \in \mathbb{N}$, and $l \in \{0, 1, \ldots, N-1\}$.

(ii) Derive the general form of the indices $w^{(j)}(l)$, the expected reward for staying awake at state $X_{z^*-j\cdot N-l} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ and then acting optimally, as compared to sleeping for N slots and then acting optimally. $w^{(0)}$, the index for $z^* - N < k < z^*$, is given in (29), and we show $w^{(1)}$, the index for $z^* - 2N < k \leq z^* - N$, in the appendix. We have not yet generalized these indices to $w^{(j)}$.

(iii) Given a parameter set $\{T, p, N, c, D\}$, use the enumeration of k from (i) and the general form of the index w from (ii) to compute $w(k)$, for every $k \in \{0, 1, \ldots, z^* - 1\}$.

(iv) If $w(k) \leq 0$ for all k, let $\lambda_2^* = 0$; otherwise, let $\lambda_2^* = \max\{k : w(k) > 0\}$.

Then the optimal policy at state $X_k = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ is to stay awake if and only if $k \leq \lambda_2^*$. If Conjecture 2 also turns out to be true, then λ_1^* can be calculated similarly by creating a second index that is a function of k and λ_2^*. This methodology is computationally much simpler than computing the entire optimal policy through the dynamic program (15).

We now discuss briefly the challenges we have faced in proving Conjectures 1 and 2. In stochastic control problems, it is often the case that we can infer structural properties of the optimal control from certain properties of the value function, such as monotonicity, convexity, and supermodularity (see for example [25] and [26] for description of such techniques). In particular, supermodularity and submodularity are used throughout the queuing theory literature (for one such example, see [27]) to prove the optimal control policy has a threshold form. However, the threshold in these cases has usually been a threshold in queue size (one control action is optimal if the queue length is above a critical number and another is optimal if it is below the critical number), as opposed to a threshold in time. In our model, such a result is true, but fairly trivial. We can see from Lemmas 3-4 that not only is the optimal control monotonic in queue length at each time k, but the threshold is always 0 (always serve), 1 (serve only if queue is non-empty), or ∞ (never serve). We are looking to strengthen this result by finding a sufficient condition for the optimal control to be monotonic in time (i.e., have those critical queue length numbers at each slot be non-decreasing over the entire time horizon). We have not found any previous works in which modularity properties are used to show the optimal control policy is monotonic in time.

Unfortunately, in our case, neither the value function nor its components display the nice properties we desire, even when we restrict the parameter sets to those satisfying (35). For instance, based on Lemmas 3-6, we can reduce part of the dynamic program (15) to the following form:

$$V_t \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \min\{\alpha_t, \beta_t\} \quad , \tag{38}$$

where α_t is the expected cost-to-go under $U_t = 1$, and β_t is the expected cost-to-go under $U_t = 0$. One way to show that the optimal control at the boundary state is of the form (a) or (c) (i.e.,

monotonic in time) when condition (35) is satisfied would be to show:

$$(35) \Rightarrow \beta_t - \beta_{t+1} < \alpha_t - \alpha_{t+1} \ \forall t \leq z^* - N \ . \tag{39}$$

Note that (39) would imply:

$$\beta_t < \alpha_t \Rightarrow \beta_{t+1} < \alpha_{t+1} \ , \tag{40}$$

which guarantees the optimal policy at $\begin{bmatrix} 0 \\ 0 \end{bmatrix}$ is non-increasing in time. However, as we see in Figure 5, (39) is not necessarily true. We have tried numerous other approaches to prove Conjecture 1, to no avail.

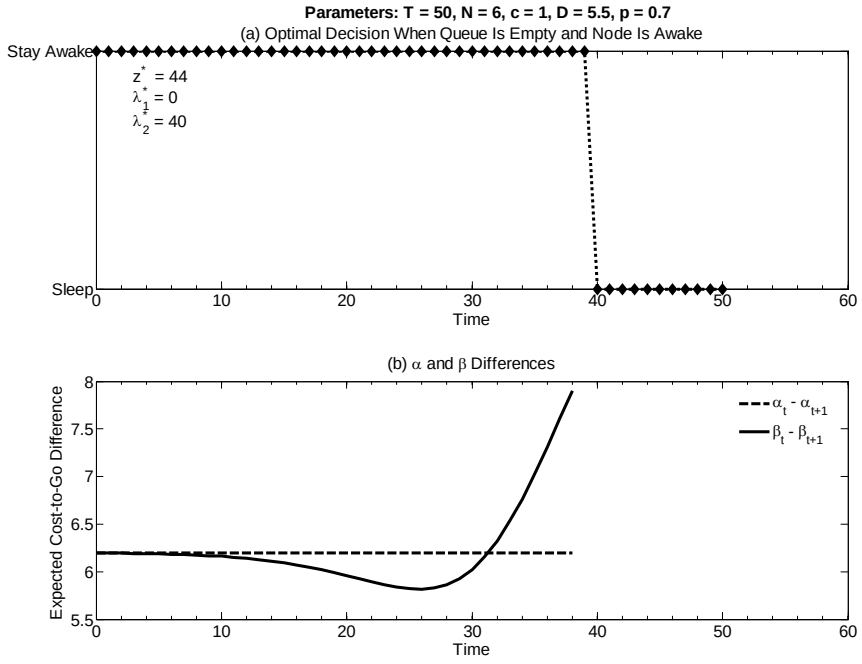

Fig. 5. Expected cost-to-go differences under the two available controls

V. Conclusion

In this report we studied the problem of optimal sleep scheduling for a wireless sensor network node, and considered two separate discrete time optimization problems. For the infinite horizon average expected cost problem, we demonstrated the existence of an optimal stationary Markov policy, and completely characterized the optimal control at each state in the state space. For the finite horizon expected cost problem, we completely characterized the optimal policy for all states except the boundary state where the node is awake and the queue is empty. One significant difference from the infinite horizon was the existence of a "shutdown" period at the end of the time horizon in which the queue stops serving packets, regardless of the queue size. We hypothesized a sufficient condition to guarantee an optimal control that is non-increasing over time when the queue is empty and the node is awake. Based on extensive numerical experiments, we also conjectured that even when this sufficient condition does not hold, there is at most one jump in the optimal control, providing a single "buffer zone."

We now mention a few possible extensions to this work. First, as discussed in Section III-D, it may be possible to relax a number of the assumptions (e.g., general non-decreasing holding costs in place of linear holding costs) and add fixed switching costs to the model, while still retaining the optimality of a threshold policy. However, we believe similar generalizations in the finite case may not be nearly as straightforward due to the inability to collapse the problem to certain decision epochs.

Second, an interesting alternative formulation of the problem is to frame it is a constrained optimization problem. Under this approach, rather than associate arbitrary costs with packet delay and energy consumption, one could directly minimize packet delay subject to a constraint that the node must be asleep for a certain portion of the time horizon. The obvious benefit of this methodology is the replacement of arbitrary costs with a user-friendly constraint which has a clear physical interpretation. We have not yet considered this model, but believe analysis on this front may be tractable.

Finally, one might consider optimal sleep scheduling for multiple nodes in a wireless sensor network. This extension is not at all straightforward, but there may be some hope to leverage the structural results from the single node case in a team-theoretic setting. Any attempt to incorporate additional quality of service objectives concerning coverage, connectivity, etc., may also drastically change the nature of the problem.

VI. Appendix

We present here the general form of $w^{(1)}$, referred to in Observation 2 in Section IV-D. The following index applies to $z^* - 2N < k = z^* - N - l \leq z^* - N$, where $l \in \{0, 1, \ldots, N-1\}$.

Define

$$
\begin{aligned}
w^{(1)}(k) &:= E\left[R_k \mid \mathcal{F}_k\right] \\
&= E\left[R_{z^*-N-l} \mid \mathcal{F}_{z^*-N-l}\right] \\
&= Pr\left(T - z^* + N + l \text{ arrivals in a row}\right) \cdot E\left[R_k \mid T - z^* + N + l \text{ arrivals}\right] \\
&\quad + \sum_{m=l+1}^{T-z^*+N+l-1} \left\{Pr\left(m \text{ arrivals before 1st non-arrival}\right) \cdot E\left[R_k \mid m, \mathcal{F}_k\right]\right\} \\
&\quad + \sum_{m=0}^{l} \Big\{Pr\left(m \text{ arrivals before 1st non-arrival}\right) \cdot \\
&\qquad \sum_{w=m}^{l+1} \Big\{Pr\left(\bar{\boldsymbol{\pi}} \text{ sleeps for good at } z^* - l + w \mid m, \mathcal{F}_k\right) \cdot \\
&\qquad\qquad E\left[R_k \mid \bar{\boldsymbol{\pi}} \text{ sleeps for good at } z^* - l + w, m, \mathcal{F}_k\right]\Big\}\Big\} \\
&= -D + p^{l+1}\left[c(l+1)(N+1)\right] + \sum_{j=2}^{N} p^{j+l}\left[-D + c\left(T - z^* + N - j\right)\right] \\
&\quad + \sum_{m=0}^{l} \left\{p^m(1-p) \sum_{w=m}^{l+1} \left[\Psi_{l,w,m} \cdot \Gamma_{l,w,m}\right]\right\} \quad w.p.1 \quad ,
\end{aligned}
$$

where

$$\Psi_{l,w,m} := Pr\left(\bar{\boldsymbol{\pi}} \text{ sleeps for good at } z^* - l + w \mid m \text{ arrivals before 1st non-arrival, } \mathcal{F}_{z^*-N-l}\right)$$

$$= \begin{cases} (1-p)^{N-1} p^{w-m} \left[\binom{N-1}{w-m} + \sum_{j=1}^{w-m-1} \left\{ \binom{N-1}{j} \sum_{i=1}^{w-m-j} \binom{j}{i} \right\} \right], \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad w \in \{m, m+1, \ldots, l\} \\ 1 - \sum_{w=m}^{l} \Psi_{l,w,m} \ , \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad w = l+1 \\ 0 \ , \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{otherwise} \end{cases}$$

and

$$\Gamma_{l,w,m} := D + E\left[R_{z^*-N-l} \mid m \text{ arrivals before 1st non-arrival, } \bar{\boldsymbol{\pi}} \text{ goes to sleep for good at } z^* - l + w, \ \mathcal{F}_{z^*-N-l}\right]$$

$$= \begin{cases} mc(N-1) - c(w-m) + \sum_{j=1}^{l-w} \left\{ p^j \left[-D + c\left(T - z^* + l - w - j\right) \right] \right\}, \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad w \in \{m, m+1, \ldots, l\} \\ mc(N-1) - c(l-m) + D - c\left(T - z^*\right) \ , \qquad\qquad\qquad\qquad w = l+1 \\ 0 \ , \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{otherwise} \end{cases}$$

REFERENCES

[1] D. Culler, D. Estrin, and M. Srivastava, "Guest editors' introduction: overview of sensor networks," *Computer*, vol. 37, no. 8, pp. 41–49, August 2004.

[2] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the Hawaii International Conference on Systems Sciences*, Maui, Hawaii, January 2000.

[3] J. Chang and L. Tassiulas, "Energy conserving routing in wireless ad hoc networks," in *Proceedings of IEEE INFOCOM 2000*, Tel Aviv, Israel, March 2000.

[4] J. Sheu, C. Lai, and C. Chao, "Power-aware routing for energy conserving and balance in ad hoc networks," in *Proceedings of the 2004 IEEE Conference on Networking, Sensing, and Control*, Taipei, Taiwan, March 2004, pp. 468–473.

[5] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed diffusion for wireless sensor networking," *IEEE/ACM Transactions on Networking*, vol. 11, no. 1, pp. 2–16, February 2003.

[6] L. M. Feeney and M. Nilsson, "Investigating the energy consumption of a wireless network interface in an ad hoc networking environment," in *Proceedings of IEEE INFOCOM 2001*, Anchorage, Alaska, April 2001.

[7] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing," in *Proceedings of the Seventh Annual ACM/IEEE International Conference on Mobile Computing and Networking*, Rome, Italy, July 2001.

[8] A. Cerpa and D. Estrin, "ASCENT: Adaptive Self-Configuring sEnsor Networks Topologies," *IEEE Transactions on Mobile Computing*, vol. 3, no. 3, pp. 272–285, July 2004.

[9] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," in *Proceedings of the Seventh Annual ACM/IEEE International Conference on Mobile Computing and Networking*, Rome, Italy, July 2001.

[10] F. Ye, G. Zhong, S. Lu, and L. Zhang, "PEAS: A robust energy conserving protocol for long-lived sensor networks," in *Proceedings of the Tenth Annual IEEE International Conference on Network Protocols*, Paris, France, November 2002.

[11] M. Sarkar and R. L. Cruz, "Analysis of power management for energy and delay trade-off in a WLAN," in *Proceedings of the Conference on Information Sciences and Systems*, Princeton, New Jersey, March 2004.

[12] M. Sarkar and R. L. Cruz, "An adaptive sleep algorithm for efficient power management in WLANs," in *Proceedings of the Vehicular Technology Conference*, Stockholm, Sweden, May 2005.

[13] A. S. Alfa, "Vacation models in discrete time," *Queueing Systems*, vol. 44, pp. 5–30, 2003.
[14] H. Takagi, *Queueing Analysis, Vol 3: Discrete-Time Systems*, North-Holland, 1993.
[15] A. Gravey, J. Louvion, and P. Boyer, "On the Geo/D/1 and Geo/D/1/n queues," *Performance Evaluation*, vol. 11, pp. 117–125, 1990.
[16] J. A. Schormans, E. M. Scharf, and J. M. Pitts, "Prioritised Geo/D/1 telecommunications switch model," *Electronic Letters*, vol. 28, no. 6, pp. 597–598, March 1992.
[17] A. Federgruen and K. C. So, "Optimality of threshold policies in single-server queueing systems with server vacations," *Adv. Appl. Prob.*, vol. 23, no. 2, pp. 388–405, June 1991.
[18] L. I. Sennott, *Stochastic Dynamic Programming and the Control of Queueing Systems*, John Wiley and Sons, 1999.
[19] A. Arapostathis, V. S. Borkar, E. Fernandez-Gaucherand, M. K. Ghosh, and S. I. Marcus, "Discrete-time controlled Markov processes with average cost criterion: a survey," *SIAM J. Control and Optimization*, vol. 31, no. 2, pp. 282–344, March 1993.
[20] R. F. Serfozo, "An equivalence between continuous and discrete time Markov decision processes," *Operations Research*, vol. 27, no. 3, pp. 616–620, 1979.
[21] D. R. Robinson, "Markov decision chains with unbounded costs and applications to the control of queues," *Adv. Appl. Prob.*, vol. 8, pp. 159–176, 1976.
[22] P. Brémaud, *Markov Chains: Gibbs Fields, Monte Carlo Simulation, and Queues*, Springer-Verlag, 1991.
[23] P. G. Hoel, S. C. Port, and C. J. Stone, *Introduction to Stochastic Processes*, Waveland Press, 1987.
[24] P. R. Kumar and P. Varaiya, *Stochastic Systems: Estimation, Identification, and Adaptive Control*, Prentice-Hall, 1986.
[25] K. F. Hinderer, "On the structure of solutions of stochastic dynamic programs," in *Proceedings of the Seventh Conference on Probability Theory*, M. Iosifescu, Ed., Bucharest, 1984, pp. 173–182, Editura Academiei Republicii Socialiste România.
[26] J. E. Smith and K. F. McCardle, "Structural properties of stochastic dynamic programs," *Operations Research*, vol. 50, no. 5, pp. 796–809, 2002.
[27] E. Altman and S. Stidham Jr., "Optimality of monotonic policies for two-action Markovian decision processes with applications to control of queues with delayed information," *QUESTA*, vol. 21, pp. 267–291, 1995.