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
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.**

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, there have been numerous studies examining methods that 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 paper, 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. 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 [13] provides a comprehensive overview of the results on different types of threshold policies. Of these models, [13] is the most relevant to our model, and we discuss it further in Section III-B.

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 Sections III and IV, we analyze these two problems. 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.

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 steps. Slot t refers to the slot defined by the interval $[t, t+1)$.

In general, switching on and off is also an energy consuming process. Therefore ideally we do not want to put 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 also 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.

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 notation.

T The length in slots of the time horizon.

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^-.

S_t Slots remaining until the node awakes; also observed at time t^-. $S_t = 0$ indicates node is awake at time t.

X_t $:= [B_t, S_t]^T$, the information state at time t.

Y_t The observation available to the node at time t.

A_t The number of random arrivals during the t-th slot.

p The probability of an arrival in each time slot.

$\mathcal{U}$ $:= \{0, 1\} = \{$Sleep, Stay Awake$\}$, the action space.

U_t The control R.V. denoting sleep decision for 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 t.

$\boldsymbol{\pi}$ $:= (\pi_1, \pi_2, \ldots)$, a sleep policy, with $\pi_i \in U$.

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 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:

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

$$Y_t = X_t \ .$$

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. Thus, model (1) is a controlled Markov chain with a time-invariant matrix of transition probabilities.

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.

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:

$$J^{\pi} := \limsup_{T\to\infty} \frac{1}{T} \cdot E^{\pi} \left\{ \sum_{t=0}^{T-1} D \cdot U_t + \sum_{t=1}^{T} c \cdot B_t \mid \mathcal{F}_0 \right\} . \quad (2)$$

In Problem (**P2**), the cost function for minimization is J_T^{π}, defined as:

$$J_T^{\pi} := E^{\pi} \left\{ \sum_{t=0}^{T-1} D \cdot U_t + \sum_{t=1}^{T} c \cdot B_t \mid \mathcal{F}_0 \right\} . \quad (3)$$

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 (2).

A. Characterization of Optimal Policy

Due to the assumption of an infinite buffer size, the controlled Markov chain in Problem (**P1**) has a countably infinite state space. For such systems, an average cost optimal policy is not guaranteed to exist. See [14, pp. 128–132] for such counterexamples. However, [14] also presents sufficient conditions for the existence of an average cost optimal stationary policy.

Lemma 1: Problem (**P1**) satisfies the (BOR) assumptions of Theorem 7.5.6 of [14], and therefore, there exists an optimal stationary Markov policy $\boldsymbol{\pi}^*$ that minimizes (2).

We now present the main result of this section.

Theorem 1: In Problem (**P2**), the optimal control at state $X = [B, 0]^T$, for $B > 0$, is $U^* = 1$. At the boundary state $[0, 0]^T$, the optimal control, U^*, is given by:

$$\left(\frac{p}{1-p}\right)\left(\frac{N-1}{2}\right) \underset{U^*=1}{\overset{U^*=0}{\lessgtr}} \frac{D}{c} . \quad (4)$$

The first half of the theorem follows from the fact that the node must eventually serve packets, and can thus be proved using an interchange argument. Since we know there exists an optimal stationary Markov policy, the second half follows by comparing the average expected cost under the two possible controls. 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 $\boldsymbol{\pi_1}$ [13].

B. Related Work and Possible Extensions

The arguments we use to prove Theorem 1 are quite similar to those applied to the embedded Markov chain model of [13]. 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 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. We have not yet explored relaxing our assumptions in a similar manner. 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 (4), 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 (3). 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 from classical stochastic control theory (see for example [15, pp. 78–79]) that there exists an optimal deterministic Markov policy. While, in principle, we can compute this optimal policy through a dynamic program, 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 simple form.

A. Optimal Policy Away from the Boundary

We now identify the optimal policy in a piecewise manner, beginning with the slots at the end of the time horizon.

Lemma 2: 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.

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 we approach the end of the time horizon.

We now proceed to the case when the node is awake, the queue is non-empty, and the process is sufficiently far from the end of the time horizon.

Lemma 3: If $0 \leq t < T - \frac{D}{c}$ and $X_t = \begin{bmatrix} B_t \\ 0 \end{bmatrix}$ for some $B_t > 0$, the optimal control at slot t to minimize $J_t^{\boldsymbol{\pi}}$ is $U_t^* = 1$; i.e., serve a job in slot $[t, t+1)$.

B. Optimal Policy at the Boundary State

We know from Lemma 2 that the optimal control at $X_t = \left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]$ is to sleep when $t \geq T - \frac{D}{c}$. We now examine the optimal control at this state when $z^* - N < t < T - \frac{D}{c}$.

Lemma 4: If $t = z^* := \left\lfloor T - \frac{D}{c} \right\rfloor$ and $X_t = \left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]$, the optimal control policy to minimize $J_t^{\boldsymbol{\pi}}$ is to sleep for the duration of the time horizon.

Lemma 5: If $z^* - N < t < z^*$ and $X_t = \left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]$, the optimal control at slot t to minimize $J_t^{\boldsymbol{\pi}}$ is described by the threshold decision rule:

$$c \cdot \sum_{j=1}^{z^*-t} \left\{ p^j \left(T - t - j\right) \right\} - D \cdot \sum_{j=0}^{z^*-t} p^j \underset{U_t^*=1}{\overset{U_t^*=0}{\lesseqgtr}} 0 \ . \tag{5}$$

The LHS of (5) is non-increasing in t, and thus, Lemmas 2, 4, and 5 imply that the optimal policy at $X_t = \left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]$ is non-increasing over time, from slot $z^* - N + 1$ until the end of the time horizon. The natural follow-up question is whether the optimal policy at $X_t = \left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]$ is necessarily monotonic over the entire duration of the time horizon. Intuitively, this might make sense if we extend the logic behind Lemma 2 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, 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 $X_t = \left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]$, computed by dynamic program, is displayed in Fig. 1. Clearly, this policy is not monotonic in time.

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:

$$\left(\frac{p}{1-p}\right) \cdot \left(\frac{N-1}{2}\right) \geq \frac{D}{c} \ , \tag{6}$$

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.

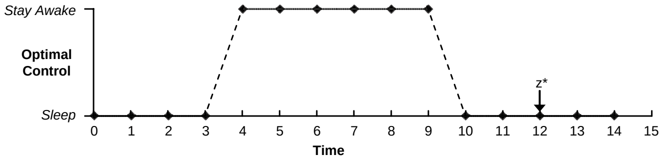

Fig. 1. Optimal control policy at $X_t = \left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]$, when $T = 15$, $N = 3$, $c = 10$, $D = 21$, and $p = \frac{2}{3}$

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 (6). 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:

$$\left(\frac{p}{1-p}\right) \cdot \left(\frac{N-1}{2}\right) < \frac{D}{c} \ , \tag{7}$$

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 $X_{t+1} = \left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]$.

C. Discussion

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

If Conjectures 1 and 2 turn out to be true, they imply, in combination with Lemmas 2-5, that the optimal control policy at $X_t = \left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]$ is of the form:

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

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 Fig. 2.

Moreover, Conjecture 1 states that form (b) is not possible if condition (6) holds. Our numerical results not only support these conclusions, but also show the following:

Observation 1: If the time horizon is sufficiently long, the optimal control is of the form (a) if condition (6) holds, but of the form (b) or (c) if the negation, (7), 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-A, and assume the node is at state $\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]$ at some time t. 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 $t+N-1$ 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 $(t, t+N-1)$. If there are M arrivals during this period, the queue length at time $t+N$

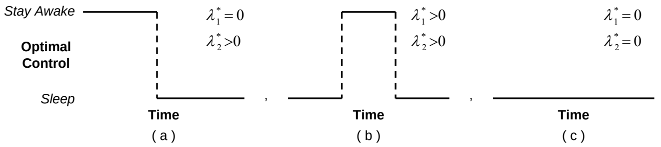

Fig. 2. Possible structural forms for the optimal control policy at $X_t = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

under $\boldsymbol{\pi_1}$ is M more than the queue length under $\boldsymbol{\pi_0}$. With each non-arrival after time $t+N$, $\boldsymbol{\pi_1}$ "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 precisely the comparison (4), which describes the optimal stationary policy at the boundary state in the infinite horizon case.

Returning to the finite horizon problem, we see that (6) and (7) together are equivalent to (4). 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 $t \rightarrow 0$. 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 of the time horizon.

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 Fig. 2 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 vice versa. In particular, 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.

Finally, if the conjectures turn out to be true, we can directly calculate the thresholds λ_1^* and λ_2^* through an index representing the expected difference in cost between the two available policies. Doing so enables a complete characterization of the optimal policy in a manner computationally simpler than computing the entire policy via a dynamic program.

V. Conclusion

In this report we studied the problem of optimal sleep scheduling for a wireless sensor network node, and considered two separate optimization problems. For the infinite horizon average expected cost problem, we completely characterized the optimal control at each state in the state space. For the finite horizon expected cost problem, we described the optimal policy for all states except the boundary state. The most 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."

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] D. Estrin J. Heidemann F. Silva C. Intanagonwiwat, R. Govindan, "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. 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.

[14] L. I. Sennott, *Stochastic Dynamic Programming and the Control of Queueing Systems*, John Wiley and Sons, 1999.

[15] P. R. Kumar and P. Varaiya, *Stochastic Systems: Estimation, Identification, and Adaptive Control*, Prentice-Hall, 1986.