Drowsy Transmission: Physical Layer Energy Optimization for Transmitting Random Packet Traffic

Husheng Li†, Lin Zhong ‡ and Kun Zheng†
†Dept. of Electrical Engineering and Computer Science, The University of Tennessee, Knoxville, TN 37996
‡Dept. of Electrical and Computer Engineering, Rice University, Houston, TX 77005

***Abstract*—Energy efficiency has become increasingly important to mobile systems on which wireless interfaces are among the largest power consumers. While existing physical layer power optimization mostly focuses on improving the transmission efficiency, our recent work has showed that wireless interfaces can spend most of its time and energy in very short idle periods between transmitting two packets [9]. In this work, we present a physical layer optimization method, *drowsy transmission*, which explicitly considers the power cost of such idle periods in physical layer power optimization through joint power control/rate selection and power management. We provide a control theoretical formulation of the optimization problem and present a dynamic programming based solution and its approximation that is close form and practical. We further offer an on-line learning technique to cope with unknown channel and traffic. Using a power model from a commercial wireless network interface card, we demonstrate that drowsy transmission can reduce the energy per bit by 70% and 40% in comparison to power control/rate selection-based optimization and optimization with disjoint power control/rate selection and power management, respectively . Moreover, the achieved energy per bit is very close to the theoretical lower bound. Our evaluation shows that the proposed on-line learning technique can assess the channel and approach the performance under pre-known channel in as short as 200ms. We also show that our optimization introduces negligible packet delays.**

I. INTRODUCTION

Limited battery capacity and heat dissipation capability have made energy efficiency a critical concern to the design of modern mobile systems. Wireless interfaces are among the largest power consumers on mobile systems [14]. In this work, we aim at improving their energy efficiency by jointly optimizing the energy consumption in transmission time and in idle time through a combination of hardware power management and conventional power-saving mechanisms at the physical layer, namely power control and rate selection.

Numerous techniques have been investigated to improve the energy efficiency of various layers of wireless communication systems. Most physical (PHY) layer solutions focus on reducing the energy consumption for transmission by using transmit power control, rate selection or both [12]. These solutions either completely ignore the energy cost in idle time or assume fixed power consumption, e.g. see [5], [6], [12], [17]. As we have showed recently [9], wireless interfaces can spend a very high percentage of time and energy in short idle periods even during active transmission time. Such short idle periods, as below tens of milliseconds, are out of the reach of conventional power-saving mechanisms provided by higher layers of the protocol stack, such as IEEE 802.11 MAC Power-Saving Mode (PSM), other proposed power-saving protocol, e.g. [8], and application-specific solutions, e.g. [4], [16].

Physical layer decisions, such as power control and rate selection, essentially determine the pattern of such idle periods, given the traffic from the upper layers. Consequently, they significantly impact the energy-saving opportunities in the idle periods through power management. In this work, we seek to jointly optimize idle time power management and transmission time power control/rate selection, subject to traffic and channel dynamics. The optimization essentially produces optimized idle period patterns and selects the best sleep mode for each idle period between two packet transmissions. We call it *drowsy transmission* because the wireless interface enters sleep modes even during active transmission. We formulate the optimization as a control problem and provide a dynamic programming based approximating solution. Using power models from a commercial wireless interface, we show that the joint optimization can significantly reduce the energy per bit transmission up to 40% with idle time considered in comparison to separate optimization of power control/rate selection and idle time power management, depending on the data rate. More importantly, we show that the energy per bit achieved by the proposed optimization is very close to the theoretical lower bound set by optimization in which traffic and channel dynamics are assumptively pre-known. We further provide on-line learning and adaptive solutions to accommodate unknown traffic and channels.

Our work represents a radically different approach to the energy optimization of wireless communication systems by applying power management to physical layer energy optimization. Our work highlights that energy consumptions in transmission time and idle time are highly correlated and disjoint optimization is suboptimal. The proposed problem formulation and approximate solutions can be readily applied to the physical layer implementation of existing wireless technologies, such as 802.11 and 3GPP.

The rest of the paper is organized as follows. In Section II, we provide background information and address related work. In Section III, we present the problem formulation of drowsy transmission along with the wireless power model used and the assumptions made. We offer a dynamic programming based approximate solution to the problem in in Section IV and an on-line learning technique for coping with unknown traffic and channels in Section V. We provide the experimental evaluation of the proposed techniques in Section VI, discuss the limitations of our work and conclude in Section VII.

II. Background and Related Work

A. Power Profile of Wireless Interfaces

Wireless transmission on mobile systems is drastically more power-hungry than reception, often requiring multiple times more energy per bit. As a result, much of the wireless energy optimization focuses on transmission power, mainly through physical layer mechanisms such as power control and rate selection, which set the transmission signal power and choose the modulation method, respectively.

Many modern wireless interfaces are designed for very high peak data rates, which are rarely achieved in practice due to bottle-necks in the rest of the system and network. This leads to abundant brief idle periods even during busy time of the wireless interfaces, as highlighted in [9]. Power control and rate selection have an essential impact on the pattern of such idle periods. For example, higher rates and transmission power are likely to lead to longer idle periods because the wireless interface can send out the data faster. While existing physical layer energy optimization often ignores the energy consumption of such idle periods, our prior work showed that it can account for over 80% of total energy consumption of wireless interfaces during busy time [9].

Meanwhile, modern wireless interfaces support power-saving modes in which an interface is no longer functional but consumes dramatically less power. Many power-saving modes only take multiple microseconds to multiple milliseconds to wake up, thus making it possible to put the interface into a power-saving mode for as short as several microseconds yet still save energy [9]. Usually, a deeper power-saving mode, i.e. more power reduction and longer wakeup latency, requires a longer idle period to be profitable. On the other hand, it may save more energy for a longer idle period.

Transmission energy optimization and idle period energy optimization can be at odd with each other: slower transmission through lower transmit power or rate may lead to lower energy per bit in transmission; but it may also lead to shorter idle periods in which less energy can be saved because deeper power-saving modes cannot be applied. Drowsy transmission is about making the best tradeoff between them through a joint optimization of transmission power control/rate selection and idle period power management.

B. Physical Layer Energy Optimization

Physical layer energy optimization of wireless communication systems traditionally focuses on energy consumption of transmission, while ignoring the energy consumption in the idle periods between two transmissions or assuming a constant power consumption [5], [6], [12], [17].

Transmit power control and rate selection are two most widely studied mechanisms for reducing transmission energy cost. The authors of [12] provided an excellent survey of energy optimization work using power control and rate selection. They also investigated joint optimization of power control and rate selection based on an energy model. The authors of [13] investigated packet scheduling with dynamic modulation scaling, an incarnation of rate selection.

In [9], we showed that idle periods between transmissions during busy time of wireless interfaces can consume a large share of overall energy and more importantly, we showed that many of these idle periods are long enough to benefit from power-saving modes supported by modern wireless interfaces and proposed a MAC layer solution, μPM. However, μPM does not employ any physical layer mechanisms, such as power control and rate selection. As a result, it does not change the inter-frame idle periods or reduce transmission energy cost. In contrast, drowsy transmission presented here is based on joint optimization that leverages power control and rate selection to change such idle periods in order to achieve better tradeoffs between energy costs for transmission and the idle periods.

Our methodology also echoes the joint optimization of dynamic voltage scaling (DVS) and power management of processors [11], [15]. However, the joint optimization for processors may not be applied to wireless interfaces because of channel dynamics. That is, when a task is executed by a processor, it is assumed to be finished deterministically. In contrast, when a packet is transmitted by a wireless interface, it has a certain chance of loss due to channel uncertainty. As a result, the joint optimization for wireless interfaces, as addressed in this work, is considerably more complicated.

III. Problem Formulation

A. Assumptions

For simplicity of analysis, we make the following assumptions for the problem formulation of drowsy transmission.

- The system is time slotted. The duration of each time slot is denoted by D_s. At the beginning of a slot, the wireless interface makes control decisions regarding its state.
- The length of packets is constant. We denote by L the number of information bits within each packet. Different packets are transmitted separately and are not aggregated into one. The random process of packet arrivals is stationary; and we denote the expected number of arriving packets within one time slot by λ. We assume that $\frac{\lambda L}{D_s}$, i.e. the average number of arriving bits per second, is smaller than average channel capacity such that the data traffic can be stabilized. Denoting by $A_{n:m}$ ($n \leq m$) the number of arriving packets during time slots n and m (both included), we have $E\left[A_{n:m}\right] = \lambda(m-n+1)$ (note that $A_{n:m}$ is a random variable). We consider elastic data traffic, i.e. there are no deadlines for the deliveries of data packets. Therefore, the wireless interface considers only energy efficiency. We also assume that buffer size is limited and denote by $X_{\max}$ the maximal number of packets that the buffer can contain.
- We assume that current channel condition, denoted by C_n at time slot n, is known (maybe partially) at the wireless interface, as well as the statistics of the random process of channel condition. We denote by $C_{n:m}$ channel conditions from time slot n to m (both included). In practical systems, this assumption can be achieved by allowing feedback of channel state information in a

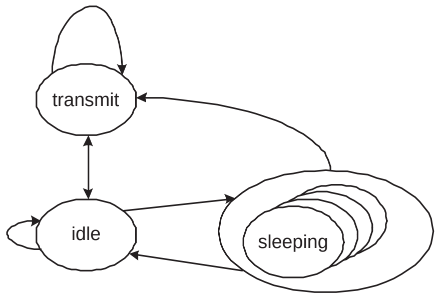

Fig. 1: State diagram of wireless interface.

channel quality index (CQI) channel, as in many systems like WiMAX, WCDMA and cdma2000.

- We assume that transmit power and transmission rate do not change during the transmission of one packet. This assumption is reasonable since the wireless interface cannot receive channel condition information during transmission due to half-duplex constraint.
- We do not consider the energy consumption by the computing of control decisions. This is reasonable because modern baseband processors are much more efficient in comparison with the wireless interface. They are also fast enough such that the time overhead is negligible.
- We assume that there is no incoming packet because our focus is on power optimization of transmission. However, we can also include the cost of missing incoming packets into the cost function and compute the optimal control policy via dynamic programming, which is beyond the scope of this paper.

Note that these assumptions capture the essence of energy optimization. The framework proposed in this paper can be however extended to more general cases. For example, we can add corresponding penalty to cost function, which will be discussed in following sections, to address the latency of real-time data traffic. When considering packet reception procedure, we can incorporate the prediction of incoming packets from another wireless interface.

B. Power Model of Wireless Transceiver

We assume that the wireless interface can be in one of three operational modes: transmit, idle, and sleep, denoted by $\mathcal{T}$, $\mathcal{I}$ and $\left\{\mathcal{S}^k\right\}_k$, as is typically for modern commercial wireless transceivers. The sleep mode has K sub modes, which are implemented with the power-saving modes supported in hardware [1], [10]. When the interface is not transmitting, it immediately enters the idle mode in which it can capture incoming data. While in the idle mode, the interface can determine whether to enter a sleep mode and which one to enter. Transitions between two modes engender fixed latencies and fixed energy overheads. For the transition from sleep mode to transmit mode, the more power saving a sleep mode, the higher latency and energy overhead. The mode transitions are illustrated in Fig. 1.

Idle mode consumes more power, denoted by $P_{\mathcal{I}}$, than sleep mode, denoted by $P_{\mathcal{S}^k}$, $k = 1, ..., K$. For transmit mode, the transmit power, denoted by P_T can be set under the constraint of $P_T < P_{\max}$. The transmission rate is denoted by R, measured in bits/second. Besides the transmit power, the device also consumes an overhead power P_O. In summary, the power consumption is given by $P_{\mathcal{T}} = P_T + P_O$ in transmit mode, $P_{\mathcal{S}^k}$ in sleep mode k, $P_{\mathcal{I}}$ in the idle mode. $E_{M_1 \to M_2}$ denotes the energy overhead when switching from mode M_1 to mode M_2. We also denote by $T_{M_1 \to M_2}$ the number of time slots needed for switching from mode M_1 to mode M_2.

C. Problem Formulation

Intuitively, our problem is how to control the wireless interface to achieve an energy efficient transmission under the constraint of a buffer of fixed, limited length for outgoing packets. For such a control problem, we first need to clarify its control and state spaces.

1) Control Space: The wireless interface can take the following possible control actions.

- Transmit power and rate: at time slot n, we denote transmit power by $P_T(n)$ and transmission rate by R_n, respectively. We also denote by $\zeta(P_T, R, C)$ the binary random variable that equals 1 when succeeding in transmitting a packet and equals 0 otherwise, when using transmitting power P_T and transmission rate R, provided current channel condition C. Note that the randomness of $\zeta(P, R, C)$ lies in future channel conditions.
- Mode selection: when there are packets in the buffer, the interface should always stay in or transit to transmit mode; when the buffer is empty and the mode is transmit, it enters the idle mode immediately and then decides whether to enter a sleep mode and which one. For time slot n, we denote by N_n the decision on the mode in the next decision time slot.

Based on the above notations, we denote by μ_n the 3-tuple of control actions $(P_T(n), R_n, N_n)$ at time slot n and denote the whole control space by Ω.

2) State Space: The state of the wireless interface depends on the number of packets in the buffer, denoted by X_n at time slot n. In the transmit mode, the evolution of state X_n is given by

$$\begin{aligned} X_{n+d} &= \min\left\{X_n - \zeta\left(P_T(n), R_n, C_{n:n+d-1}\right)\right. \\ &+ \left. A_{n:n+d-1}, X_{\max}\right\}, \qquad \text{when } X_n > 0, \end{aligned} \quad (1)$$

where $d = \left\lceil \frac{L}{R_n D_s} \right\rceil$ is the number of time slots needed to convey a packet and

$$X_{n+1} = \min\left\{A\left(n:n\right), X_{\max}\right\}, \qquad \text{when } X_n = 0. \quad (2)$$

The evolution of X_n in other modes can be derived similarly.

Other factors of the system are the wireless channel condition C_n and the current operational mode M_n. Therefore, we denote by a 3-tuple $S_n = (M_n, X_n, C_n)$ the state of the wireless interface at time slot n. Note that, in practical systems, the random processes of packet arrivals and channel conditions may not be Markovian, i.e. the future may be also dependent on the history besides the current state. However, it is difficult to incorporate all history into the state due to limited memory. Therefore, we can approximate the practical case by considering the current buffer situation and channel condition as the system state.

3) Cost Function: Essentially, the control policy targets at maximizing the energy efficiency, i.e. using the minimal average energy to convey data packets. Therefore, we define the cost function as

$$J = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} E\left[E_t + \kappa I(X_t = X_{\max})\right], \tag{3}$$

where E_t is the energy consumed during time slot t, $I(X_t = X_{\max})$ is the characteristic function of the event that the buffer cap is reached and κ is the corresponding penalty. The purpose of the term $\kappa I(X_t = X_{\max})$ is to provide an incentive for the wireless interface to transmit packets for avoiding the penalty. Otherwise, the optimal control policy is to stay in a sleep mode that minimizes the total energy consumption and does not transmit at all.

Based on the cost function definition, we minimize the energy consumption by solving the following optimization problem:

$$\mu^* = \arg\min_{\mu\in\Omega} J(\mu), \tag{4}$$

where $\mu \triangleq (\mu_1, \mu_2, ...)$ means the whole control policy and J is obviously a functional of μ.

IV. Dynamic Programming Based Optimization

In this section, we present a dynamic programming-based solution to optimize the control policy.

A. Brief Introduction of Dynamic Programming

Essentially, dynamic programming finds the optimal state path in the trellis derived from the state diagram in Fig.1. In dynamic programming, we define a cost function for the whole system and one cost-to-go function for each time slot. The cost-to-go function is the expected cost from current time slot to final time slot. We can compute the cost-to-go functions in an iterative manner and then obtain the optimal control policy by selecting the next state that has the minimal cost-to-go function. The details of dynamic programming can be found in [3].

B. Modified Cost Function

We apply dynamic programming to solve the optimization problem in (3) and thus obtain the control policy. Note that, based on our definition of state, dynamic programming may not achieve optimal performance when the packet arrival and channel condition processes are non-Markovian. However, as will be seen in Section VI, the results obtained from dynamic programming are close to optimal performance with non-causal knowledge of data traffic and channel condition that is infeasible in practical systems.

One difficulty of applying dynamic programming to the optimization problem in (3) is that the summation in (3) diverges to infinity as $T \to \infty$. Therefore, we modify the cost function slightly by adding a weighting factor to each term, which is given by

$$J \triangleq (1-\alpha) \lim_{T\to\infty} \sum_{t=1}^{T} \alpha^{t-1} E\left[E_t + \kappa I(X_t = X_{\max})\right], \tag{5}$$

where $0 < \alpha < 1$ is a weighting factor. Thus, the summation in (5) is upper bounded. It is easy to verify that the modified cost function in (5) converges that in (3) as $\alpha \to 1$. Therefore, we usually set α close to 1.

C. Cost-to-go Functions

The fundamental machinery in dynamic programming is *cost-to-go functions* from which the control policy is derived. First, we study the finite horizon case in which the total number of time slots, T, is bounded. Then, we extend to infinite horizon case, i.e. $T \to \infty$.

1) Finite Horizon Case: When T is bounded, we define the cost-to-go functions as

$$J_n(S_n) \triangleq \frac{1-\alpha}{\alpha^{n-1}} \min_{\mu} \sum_{t=n}^{T} \alpha^{t-1} E\left[E_t + \kappa I(X_t = X_{\max})\right]. \tag{6}$$

Intuitively, $J_n(S_n)$ means the minimal average cost from time slot n to time slot T, provided that the state at time slot n is S_n. It is easy to verify that the modified cost function in (5) is equal to $E\left[J_1(\mathcal{T}, 0, C_1)\right]$, where the expectation is over all possible realizations of channel condition C_1, provided that the interface begins from transmit mode and empty buffer.

Due to Bellman's optimality principle [2], the optimal cost-to-go functions can be computed recursively:

$$\begin{aligned} J_n(S_n) &= \min_{\mu} \{(1-\alpha)\left(E_{n:m} + E\left[\kappa I(X_m = X_{\max})\right]\right) \\ &+ \alpha E\left[J_m(S_m)\right]\}, \end{aligned} \tag{7}$$

where m is the next decision time slot which is a function of the control policy and $E_{n:m}$ is the energy consumption during time slots n and m, and the final condition is given by

$$J_T(S_T) = (1-\alpha)\left(E_T + \kappa I(X_T = X_{\max})\right). \tag{8}$$

By computing the optimal cost-to-go functions, the corresponding control policy is also obtained implicitly during the procedure of minimization.

2) Infinite Horizon Case: In practical wireless interfaces, the operation time can be considered as infinite. Therefore, the assumption that T is bounded in the finite horizon case is unreasonable. Moreover, the number of cost-to-go functions is proportional to T, thus requiring prohibitively large memory to store corresponding control policies. Therefore, we need to extend the finite horizon case to infinite horizon case, i.e. $T \to \infty$, which is justified by the following proposition, whose detailed proof is ignored due to limited space.

Proposition 1: For all n and all S, $J_n(S)$ converges as $T \to \infty$.

Based on Prop. 1, the cost-to-go function $J_n(S)$ is no longer dependent on time slot index n, as $T \to \infty$, thus can be written as $J(S)$. Based on (6) and (7), $J(S)$ is given by

$$\begin{aligned} J(S) &= (1-\alpha) \min_{\mu} \sum_{t=1}^{\infty} \alpha^{t-1} E\left[E_t + \kappa I(X_t = X_{\max})\right] \\ &= \min_{\mu} E\left[(1-\alpha)(\tilde{E} + \kappa I(\tilde{X} = X_{\max})) + \alpha J(\tilde{S})\right] \end{aligned} \tag{9}$$

where $\tilde{X}$ is the number of packets in the buffer at the next decision time slot and $\tilde{E}$ is the energy consumption between the two decision time slots. The number of cost-to-go functions is equal to the number of states, thus substantially less than that of the finite horizon case.

Now, we discuss the computation of cost-to-go functions in the follows six situations. Without loss of generality, we assume that the current time slot index is 1 since the control policy is time-shift-invariant in the infinite horizon case. Therefore, the current state is denoted by (M, X, C_1).

Transmit Mode with Non-empty Buffer: when the current mode is transmit and the buffer is non-empty, the control action is to choose optimal transmit power and transmission rate. When using transmit power P_T and transmission rate R, it needs $\tau = \left\lceil \frac{L}{RD_s} \right\rceil$ time slots to complete the transmission. The corresponding energy consumption is $\tilde{E} = \tau D_s(P_T + P_O)$.

Recall that $\zeta\left(P_T, R, C_{1:\tau}\right)$ is the random variable representing the success of packet transmission and $A_{1:\tau}$ is the number of newly arriving packets. Therefore, the number of packets in the buffer at the next decision time slot is a random variable, which is given by

$$\tilde{X} = \min\left\{X + A_{1:\tau} - \zeta\left(P, R, C_{1:\tau}\right), X_{\max}\right\}. \qquad (10)$$

Correspondingly, the state at the next decision time slot is given by $\tilde{S} = \left(\mathcal{T}, \tilde{X}, C_{\tau+1}\right)$. Therefore, the corresponding cost-to-go function is given by

$$\begin{aligned} J\left(\mathcal{T}, X, C_1\right) &= \min_{P_T, R} E\left[(1-\alpha)(\tilde{E} + \kappa I(\tilde{X} = X_{\max}))\right. \\ &+ \left.\alpha J(\tilde{S})\right]. \end{aligned} \qquad (11)$$

Transmit Mode with Empty Buffer: when the current mode is transmit and the buffer is empty, the wireless interface should transit from transmit mode to idle mode. It requires $T_{\mathcal{T}\to\mathcal{I}}$ time slots and energy $E_{\mathcal{T}\to\mathcal{I}}$. The number of packets in the buffer at the next decision time will be $\tilde{X} = \min\left\{A_{1:T_{\mathcal{T}\to\mathcal{I}}}, X_{\max}\right\}$. Therefore, the cost-to-go function is given by

$$\begin{aligned} & J\left(\mathcal{T}, 0, C_1\right) \\ =\ & (1-\alpha)\left(E_{\mathcal{T}\to\mathcal{I}} + \kappa E\left[I(\tilde{X} = X_{\max})\right]\right) \\ & + \alpha E\left[J\left(\mathcal{I}, \tilde{X}, C_{T_{\mathcal{T}\to\mathcal{I}}+1}\right)\right]. \end{aligned} \qquad (12)$$

Idle Mode with Non-empty Buffer: when the current mode is idle and the buffer is non-empty, the wireless interface should return to transmit mode. Recall that the number of time slots needed for transiting from idle mode to transmit mode is $T_{\mathcal{I}\to\mathcal{T}}$ and the corresponding energy consumption is $E_{\mathcal{I}\to\mathcal{T}}$. The number of packets in the buffer at the next decision time slot is given by $\tilde{X} = \min\left\{X + A_{1:T_{\mathcal{I}\to\mathcal{T}}}, X_{\max}\right\}$. Therefore, the corresponding cost-to-go function is given by

$$\begin{aligned} J\left(\mathcal{I}, X, C_1\right) &= (1-\alpha)\left(E_{\mathcal{I}\to\mathcal{T}} + \kappa E\left[I(\tilde{X} = X_{\max})\right]\right) \\ &+ \alpha E\left[J\left(\mathcal{T}, \tilde{X}, C_{T_{\mathcal{I}\to\mathcal{T}}+1}\right)\right]. \end{aligned} \qquad (13)$$

Idle Mode with Empty Buffer: when the current mode is idle and the buffer is empty, the wireless interface needs to decide whether to remain in idle mode or transit to sleep mode. If the decision is sleep mode, it should choose the most suitable sleep mode since there are K sleep modes. If the wireless interface chooses to remain in idle mode, then the next decision will be made at time slot 2. The corresponding number of packets will be $\tilde{X}_I = \min\{A_{1:1}, X_{\max}\}$ and the energy consumption is $P_I D_s$. If the wireless interface decides to transit to sleep mode k, then the next decision will be made at time slot $T_{\mathcal{I}\to\mathcal{S}^k}+1$. The corresponding number of packets in buffer will be $\tilde{X}_S^k = \min\left\{A_{1:T_{\mathcal{I}\to\mathcal{S}^k}}, X_{\max}\right\}$ and the energy consumption will be $E_{\mathcal{I}\to\mathcal{S}^k}$. Then, the cost-to-go function is

$$\begin{aligned} & J\left(\mathcal{I}, 0, C_1\right) \\ =\ & \min\left\{(1-\alpha)\left(P_I D_s + \kappa E\left[I(\tilde{X}_I = X_{\max})\right]\right)\right. \\ +\ & \alpha E\left[J\left(\mathcal{I}, \tilde{X}_I, C_2\right)\right], \\ & \min_k\left\{(1-\alpha)\left(E_{\mathcal{I}\to\mathcal{S}^k} + \kappa E\left[I(\tilde{X}_S^k = X_{\max})\right]\right)\right. \\ +\ & \left.\left.\alpha E\left[J\left(\mathcal{S}^k, \tilde{X}_S^k, C_{T_{\mathcal{I}\to\mathcal{S}^k}+1}\right)\right]\right\}\right\}. \end{aligned} \qquad (14)$$

Sleep Mode with Non-empty Buffer: when the buffer is non-empty, the wireless interface should return to the transmit mode. Suppose the wireless interface is in the k-th sleep mode. The transition needs $T_{\mathcal{S}^k\to\mathcal{T}}$ time slots and the energy consumption is $E_{\mathcal{S}^k\to\mathcal{T}}$. The number of packets in the buffer at the next decision time slot is $\tilde{X} = \min\left\{A_{1:T_{\mathcal{S}^k\to\mathcal{T}}}, X_{\max}\right\}$. Thus, the cost-to-go function is given by

$$\begin{aligned} & J\left(\mathcal{S}^k, X, C_1\right) \\ =\ & (1-\alpha)\left(E_{\mathcal{S}^k\to\mathcal{T}} + \kappa E\left[I(\tilde{X} = X_{\max})\right]\right) \\ +\ & \alpha E\left[J\left(\mathcal{T}, \tilde{X}, C_{T_{\mathcal{S}^k\to\mathcal{T}}+1}\right)\right]. \end{aligned} \qquad (15)$$

Sleep Mode with Empty Buffer: when there is no packet in the buffer, the wireless interface can choose to remain in sleep mode or transit to transmit mode. Without loss of generality, we suppose that the current sleep mode is k. If remaining in sleep mode, the next decision will be made at time slot 2, the energy consumption is $P_{S^k} D_s$ and the number of packets in the next decision time slot is $\tilde{X}_S = \min\{A_{1:1}, X_{\max}\}$. If transiting to transmit mode, the next decision will be made at time slot $T_{\mathcal{S}^k\to\mathcal{T}}$. The energy consumption is $E_{\mathcal{S}^k\to\mathcal{T}}$. The number of packets in the next decision time slot is $\tilde{X}_T = \min\left\{A_{1:T_{\mathcal{S}^k\to\mathcal{T}}}, X_{\max}\right\}$. Hence, the corresponding cost-to-go function is given by

$$\begin{aligned} & J\left(\mathcal{S}^k, 0, C_1\right) \\ =\ & \min\left\{(1-\alpha)\left(P_{S^k} D_s + \kappa E\left[I(\tilde{X}_S = X_{\max})\right]\right)\right. \\ +\ & \alpha E\left[J\left(\mathcal{S}^k, \tilde{X}_S, C_2\right)\right], \\ & (1-\alpha)\left(E_{\mathcal{S}^k\to\mathcal{T}} + \kappa E\left[I(\tilde{X}_T = X_{\max})\right]\right) \\ +\ & \left.\alpha E\left[J\left(\mathcal{T}, \tilde{X}_T, C_{T_{\mathcal{S}^k\to\mathcal{T}}+1}\right)\right]\right\}. \end{aligned} \qquad (16)$$

D. Computation of Cost-to-go Functions

For general cases, it is impossible to obtain an explicit expression for the cost-to-go functions. Therefore, we can only compute them using the iterative equations (11) – (16) numerically, i.e. substituting cost-to-go functions obtained in the previous iteration to the right hand side of the equations, computing new cost-to-go functions and stopping when a certain rule is satisfied. It is well known that the iterations converge for any initialization of the cost-to-go functions [3].

E. Extension to Real-time Traffic

As indicated in Section III, we assume that the data traffic is elastic, thus not considering the transmission delay in the cost function. However, transmission delay is a key issue in real-time data traffic (e.g. wireless video streaming). Our framework of dynamic programming can also be extended to meet the requirement of real-time data traffic. First, we need to extend the state space. Suppose that there are X packets in the buffer. We denote by d_1, ..., d_X their times-to-go, i.e. the differences between the deadlines and current time (measured in time slots). For packet k, the evolution of d_k is $d_k(m) = d_k(n) - (m - n)$, where $m > n$ are indices of time slots, when $d_k(n) \geq (m - n)$. If $d_k = 0$, we remove this packet from the buffer since it has missed its deadline. By adding a penalty for the event of missing deadline, we revise the cost function in (5) to

$$\begin{aligned} J &\triangleq (1-\alpha) \lim_{T\to\infty} \sum_{t=1}^{T} \alpha^{t-1} \\ & E\left[E_t + \kappa I(X_t = X_{\max}) + \gamma \sum_{k=1}^{X_t} I(d_k(t) = 0)\right], \end{aligned} \tag{17}$$

where γ is a constant representing the penalty of missing deadline. Due to limited space, detailed discussion and experimental results are beyond the scope of this paper.

V. Self-learning and Approximate Dynamic Programming

In Section IV, the computation of cost-to-go functions is based on the assumption that the distribution of packet arrival time and channel condition are known. However, under realistic situations, both the channel and the data traffic property can be time-varying. Therefore, the wireless interface should adapt its controlling policy based on the channel and data traffic. Therefore, we employ approximate dynamic programming to reduce the requirement of estimation and computation.

A. Approximate dynamic programming

We apply the Certainty Equivalence Control (CEC) [3] to simplify the computation of cost-to-go functions. The basic idea of CEC is to replace random variables in the computation of cost-to-go functions with their expectations, thus removing the computation of numerical integrals and converting the stochastic control into a deterministic one. Then, the computation of cost-to-go functions in (7) is changed to

$$J(S) = \min_{\mu} \left\{ (1-\alpha)(\tilde{E} + \kappa I(E[\tilde{X}] = X_{\max})) + \alpha J(E[\tilde{S}]) \right\}, \tag{18}$$

where $E[\tilde{S}] = (M, E[\tilde{X}], E[\tilde{C}])$ is the expected state (note that the next mode is not a random variable since it is determined by the control policy), $E[\tilde{X}]$ and $E[\tilde{C}]$ are the expectations of the number of packets in the buffer and channel condition at the next control time slot.

The CEC simplification is an approximation of the optimal dynamic programming in Section IV and thus incurs performance loss. However, we only need to estimate the expectations for the unknown channel and data traffic, e.g. the average number of arriving data packets and the expectation of channel condition given the current channel condition. The computation of the expectations requires much less time and incurs considerably less overhead. Moreover, the computation of cost-to-go functions and control policies is even more efficient because the numerical integrals in optimal dynamic programming case are avoided.

We further assume that the channel condition process is a martingale, i.e.

$$E\left[C_{t+1}|C_t\right] = C_t. \tag{19}$$

Then, we can use current channel condition to predict future channel conditions. This assumption is reasonable for slow fading channels. Then, the only parameter we need to estimate is λ, whose estimate is denoted by $\hat{\lambda}$, thus substantially reducing the amount of parameters to be estimated. Due to limited space, the detailed expressions of the cost-to-go functions are ignored.

VI. Experimental Evaluation

A. Policies for Comparison

We evaluate drowsy transmission based on the proposed joint optimization, or *Joint case*, against four other alternative cases. In the first case, the wireless interface is optimized in power control and rate selection, given the traffic statistics and channel conditions; it, however, does not enter any sleep mode during idle time between consecutive transmissions. This is the case for existing physical layer power optimization techniques, which optimize transmission power without considering power management opportunities in idle periods during busy time. We use this as the baseline for comparison and call it the *Baseline case*.

In the second case, the interface is jointly optimized in power control/rate selection and idle period power management, with the exact traffic and channel conditions pre-known. This case is idealistic and its power saving serves as the theoretical upper bound for evaluating the proposed joint optimization. We call it the *Perfect case 1*.

In the third case, the data traffic and channel conditions are also known *a priori*. However, the power/rate selection is optimized without awareness of idle and sleep modes. In contrast to the Baseline case, the wireless interface enters idle or sleep mode when there is no packet in the buffer. We call it the *Perfect case 2*.

In the fourth case, the interface is separately optimized in power control/rate selection and idle period power management, given the traffic statistics and channel condition. That is, the optimization is carried out in two disjoint stages: the power control and rate selection are first optimized for power using dynamic programming; then the best sleep mode is selected based on observing the pattern of the idle periods. This case is a simplistic combination of optimizations of transmission energy and idle energy. We call it the *Disjoint case*.

As we will show below that drowsy transmission can reduce power consumption significantly in comparison to the Baseline and Disjoint cases, achieving efficiency very close to the Perfect cases.

B. Performance Metrics

We evaluate optimization solutions with energy per bit and packet delay. Energy per bit refers to the average energy consumption by the wireless interface for transmitting one bit, which includes not only the transmission energy cost but also the energy spent during idle periods between two consecutive transmissions. It is important to note that our work targets at active time of wireless interfaces; the idle periods are during active time of the interfaces, not those elongated idle intervals as targeted by 802.11 MAC Power-Saving Mode.

Because of the optimization of transmission power rate, the latency overhead of waking up from sleep modes, and particularly the use of a transmit packet buffer, our joint optimization can introduce latencies into outgoing packets. Therefore, we will also examine the cumulative distribution function (CDF) for packet delays.

C. Experimental Setup

1) Wireless Channel Model: We assume that the system uses a 4MHz frequency band (denoted by W), where noise power density (PSD, denoted by N_0) is -174dBm. The channel gain (including path loss and fast fading), generated from Jakes fading process [7], ranges from -134dB to -124dB. The transmit signal power has four levels: 10mW, 40mW, 70mW, and 100mW; and the aggregated PA and antenna efficiency is 50%. Therefore, the corresponding power consumption for RF transmission, P_T, is 20mW, 80mW, 140mW and 200mW, respectively. There are five options for the transmission rate: 1Mbps, 2Mbps, 4Mbps, 6Mbps and 8Mbps. Assuming a complex channel, we can calculate the maximal transmission rate as $C = E\left[W \log_2\left(1 + \frac{P_T g}{2N_0 W G}\right)\right]$, where g is channel power gain, G is the gap to Shannon capacity (we assume $G = 5$dB) and the expectation is over the channel realization during the packet transmission. We assume that the transmission succeeds when $C \geq R$ and fails when $C < R$. Since it is impossible to compute the cost-to-go functions with respect to continuous channel conditions, we discretize the channel conditions into 15 levels, which is reasonable since the feedback of channel conditions can only be represented by limited number of bits in practical systems.

2) Power Model: We implement the power model of the wireless interface card based on data from a commercial 802.11b interface [1]. The wireless interface consumes 297mW in the idle mode. We assume that its idle mode consumption is close to the power consumption by the part of the wireless interface that is independent from the transmission signal power when it is in the transmit mode, or $P_O = 297$mW.

The wireless interface supports four sleep modes whose (power consumption, wake latency) are (190mW, 1us), (70mW, 25us), (60mW, 2ms), and (30mW, 5ms), respectively. We used this particular wireless interface for our power model only because its power and latency data are available for its power-saving modes. Newer wireless transceivers, and therefore interfaces, usually consume much lower power and take much shorter time to wake up from sleep modes. Yet it still holds that the more power-saving in a sleep mode, the longer the wake-up latency. Therefore, we expect that the relative power savings reported in our evaluation are still true for wireless interfaces of the state of the art.

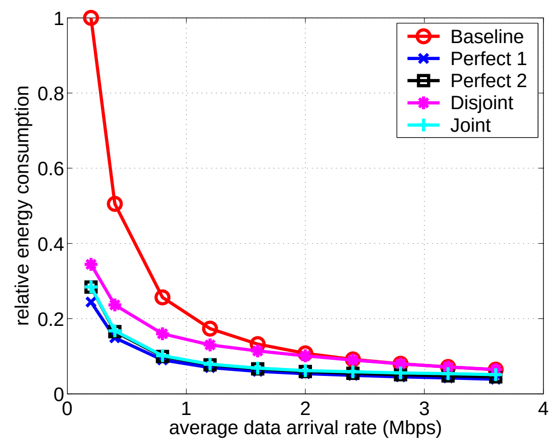

Fig. 2: Energy per bit, normalized by that of Baseline case when the data rate is 0.2Mbps ($\lambda = 0.005$), versus different packet arrival rates and control policies

3) Traffic Model: For simplicity, we assume Poisson process of packet arrival times. We test different average values of λ (recall λ means the average number of packets arriving within one time slot). We assume that $L = 4$kb (recall that L is the number of bits within a data packet). We also assume that the buffer can contain at most 20 data packets, i.e. $X_{\max} = 20$. Therefore, there are totally $20 \times 15 \times 6 = 1800$ states.

4) Self-learning and Approximate Dynamic Programming: We assume that the packet arrival process is stationary while the statistics are unknown. The control policy is initialized as Disjoint optimization which does not need the knowledge of λ. The wireless interface updates its estimate of λ every 1000 time slots (thus 0.1s) and then updates the cost-to-go functions and control policy using three iterations. The performance after each update is tested using 100000 time slots (duration of 10s).

D. Experiment Results

1) Results with Optimal Dynamic Programming: In the simulation, the cost-to-go functions are obtained by 100 iterations and the energy consumption is evaluated with 1 million time slots, which is equal to 100 seconds since we assume $D_s = 0.1$ms. The weighting factor α is set to 0.99.

The energy efficiency measured in energy per bit is shown in Figure 2. Note that the energy efficiencies are normalized by that of the Baseline case when $\lambda = 0.005$ (0.2Mbps). We observe that the energy efficiency is improved with lower energy per bit when the traffic becomes busy. This is intuitive since the wireless interface will spend less percentage of time and energy in idle periods, as λ is increased. We also observe that the joint optimization proposed in this paper achieves much higher energy efficiency than the Baseline scheme and is very close to the ideal case (Perfect case 1), particularly when λ is small. The detailed number will be given in Fig. 3.

The energy efficiency is also shown in Figure 3, in which the energy consumption per bit is normalized by that of the Baseline case with the same λ. We observe that, compared with the Baseline case, drowsy transmission based on the proposed joint optimization scheme consumes much less energy per bit.

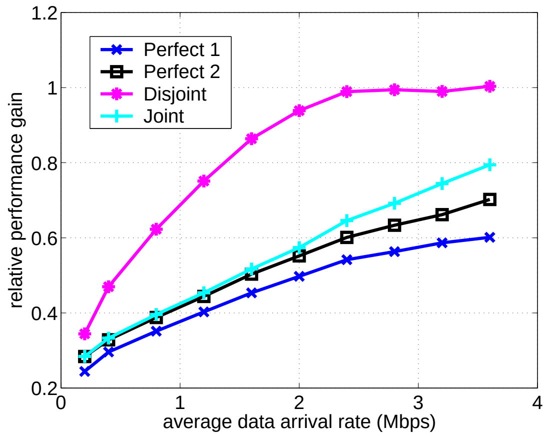

Fig. 3: Energy per bit, normalized by that of Baseline case with the same λ, versus different packet arrival rates and control policies

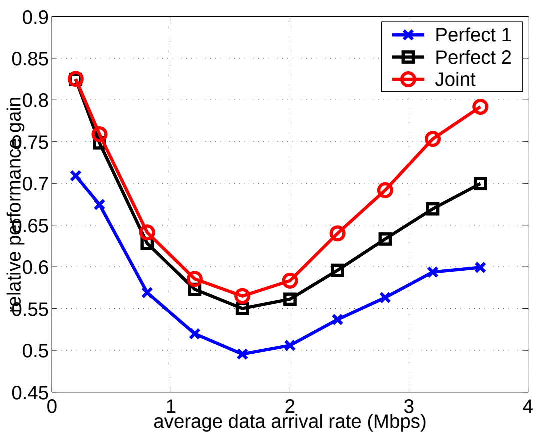

Fig. 4: Energy per bit, normalized by that of Disjoint case with the same λ, versus different packet arrival rates and control policies

For example, the joint optimization case consumes only around 30% and 60% energy compared with the Baseline one when $\lambda = 0.005$ (0.2Mbps) and $\lambda = 0.05$ (2Mbps), respectively. However, this performance gain is decreased as λ increases.

Figure 4 shows the energy consumption per bit of different schemes, normalized by that of the Disjoint case with the same λ. Using this figure, we can compare the optimal performances between jointly optimized and disjointly optimized systems with causal knowledge of data traffic and channel condition. We observe that, in the best case ($\lambda = 0.04$, i.e. 1.6Mbps), jointly optimizing the power/rate selection and power management can reduce almost half energy consumption per bit. This performance gain is reduced for larger or smaller λ since the wireless interface will be dominated by transmit mode when λ is large and by sleep mode when when λ is small.

Figure 5 shows the percentage of time in different modes (including transition between two modes) with different data arrival rates and optimization schemes. We observe that, for all three data arrival rates, drowsy transmission always results in more percentage of time in sleep mode, thus substantially

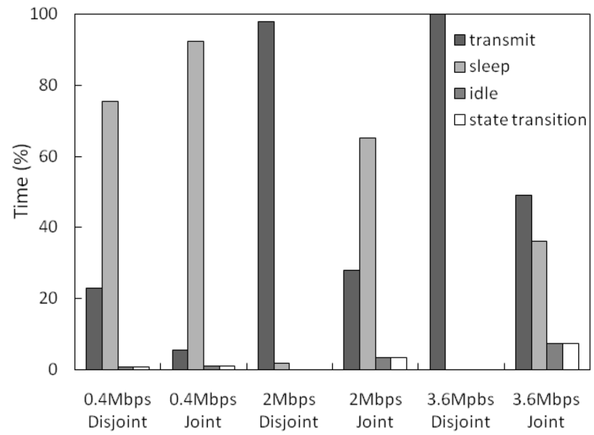

Fig. 5: Percentage of time in different modes and transitions between modes

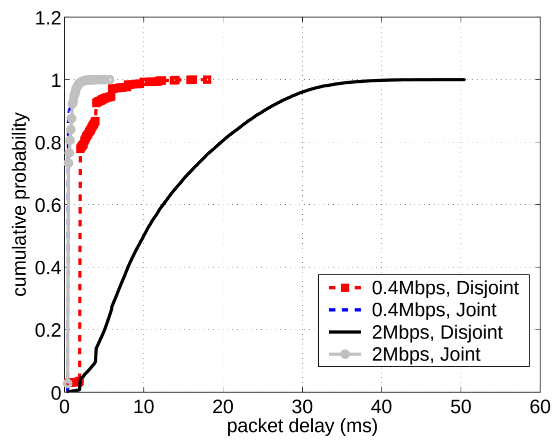

Fig. 6: Cumulative probability function of packet delays

saving energy.

Figure 6 shows the cumulative probability functions (CDF) of packet delays when data rates are 0.4Mbps (λ =0.01) and 2Mbps(λ =0.05), respectively. We observe that the joint optimization statistically yields much shorter delays, which benefits real-time data transmission.

2) Results with Self-learning and Approximate Dynamic Programming: Figure 7 shows the energy per bit, normalized by that of the joint optimization scheme with the channel condition and traffic statistics given, versus the time used for self learning (1 update of the estimation of λ and 3 iterations for the approximate dynamic programming every 0.1s) when $\lambda = 0.01, 0.02, 0.05, 0.09$ (0.4Mbps, 0.8Mbps, 2Mbps, 3.6Mbps). We observe that the performance approaches that of the optimal joint optimization very quickly (e.g. the wireless interface converges to near-optimal performance in only 0.2 seconds), which implies the practical feasibility of the dynamic programming based joint optimization.

VII. Conclusions and Discussions

In this work, we present the first study on how to incorporate power management of short inter-packet idle intervals into

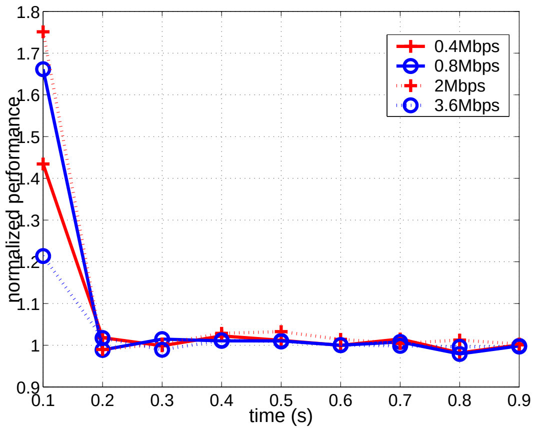

Fig. 7: Performance of self-learning and approximate dynamic programming

energy optimization at the physical layer. We leverage power control and rate selection for transmission time, and power-saving modes of wireless interfaces for idle time. The key is to jointly optimize them to achieve lower energy consumption for bit transmission with the idle time energy cost considered, which we call drowsy transmission. We provide a control theoretical formulation for drowsy transmission and its dynamic-programming based solution. We also offer an approximate but practical implementation of the solution. We present an adaptive algorithm based on on-line learning to cope with the dynamic nature of the wireless channel and the traffic. We evaluate our solutions with communication system models of industrial strength and power models from commercially available wireless interfaces. We show that drowsy transmission can considerably reduce the power consumption of an active wireless interfaces across a wide range of data rates. It also achieve energy efficiency very close to the theoretical upper bound. We further demonstrate the strength of joint optimization, showing it achieves better energy efficiency than two separate power optimizations of transmission time and idle time.

In this work, we focus on the transmission part of wireless communication. Yet extending our solutions to including receiving is possible. In communication systems that are full-duplex or have predictable receiving activities, e.g. devoted receiving time slots or predictable incoming traffic, the extension is straightforward. In such systems, the wireless interface can be considered as made of a transmitter and a receiver. The proposed solution can be readily applied to a transmitter with the time slots for receiving taken consideration. In communication systems that are half-duplex and do not predictable receiving activities, e.g. IEEE 802.11, our solution can be combined with techniques that specifically deal with unpredictable incoming traffic. For example, in [9], we leverage the traffic statistics to set the allowable unreachable time for a wireless interface. In the unreachable time, the wireless interface can ignore incoming packets but replies on the retransmission mechanism to guarantee reliable communication. Therefore, the joint optimization solution can be adapted for the joint power optimization of transmission and the unreachable time.

Acknowledgements

The work was supported in part by NSF awards CNS/CSR-EHS 0720825, CNS/NeTS-WN 0721894, CNS/CRI 0751173, and support from the TI Leadership University Program. The authors would like to our reviewers, whose comments helped improve this paper.

References

[1] C. Andren, Bozych, B. T., Rood, and D. Schultz, “Intersil: PRISM power management modes,” *Application Note AN9665*, February 1997.

[2] R. Bellman, *Dynamic Programming*. Princeton University Press., 1957.

[3] D. P. Bertsekas, *Dynamic Programming: Deterministic and Stochastic Models*. Prentice-Hall Inc., 1987.

[4] S. Chandra and A. Vahdat, “Application-specific network management for energy-aware streaming of popular multimedia formats,” in *Proc. USENIX Annual Technical Conference*, 2002, pp. 329–342.

[5] S. Cui, A. Goldsmith, and A. Bahai, “Energy-efficiency of mimo and cooperative mimo techniques in sensor networks,” *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 6, pp. 1089–1098, August 2004.

[6] ——, “Energy-constrained modulation optimization,” *IEEE Transactions on Wireless Communications*, vol. 4, no. 5, pp. 2349–2360, September 2005.

[7] W. C. Jakes, *Microwave Mobile Communications*. John Wiley and Sons, 1974.

[8] R. Kravets and P. Krishnan, “Power management techniques for mobile communication,” in *Proc. Int. Conf. Mobile Computing and Networking (MobiCom)*, 1998, pp. 157–168.

[9] J. Liu and L. Zhong, “Micro power management of active 802.11 interfaces,” in *Proc. ACM/USENIX Int. Conf. Mobile Systems, Applications and Services (MobiSys)*, 2008, pp. 146–159.

[10] *Data sheet for MAX2829 Single-/Dual-Band 802.11a/b/g World-Band Transceiver ICs*, Maxim Integrated Product.

[11] A. Miyoshi, C. Lefurgy, E. V. Hensbergen, R. Rajamony, and R. Rajkumar, “Critical power slope: understanding the runtime effects of frequency scaling,” in *Proc. Int. Conf. Supercomputing*, 2002, pp. 35–44.

[12] D. Qiao, S. Choi, A. Jain, and K. G. Shin, “MiSer: an optimal low-energy transmission strategy for ieee 802.11a/h,” in *Proc. Int. Conf. Mobile Computing and Networking (MobiCom)*, 2003, pp. 161–175.

[13] V. Raghunathan, S. Ganeriwal, M. Srivastava, and C. Schurgers, “Energy efficient wireless packet scheduling and fair queuing,” *ACM Trans. Embedded Computing Systems*, vol. 3, no. 1, pp. 3–23, 2004.

[14] A. Rahmati and L. Zhong, “Context-for-wireless: context-sensitive energy-efficient wireless data transfer,” in *Proc. ACM/USENIX Int. Conf. Mobile Systems, Applications and Services (MobiSys)*, 2007, pp. 165–178.

[15] T. Simunic, L. Benini, A. Acquaviva, P. Glynn, and G. De Micheli, “Dynamic voltage scaling and power management for portable systems,” in *Proc. Design Automation Conf.*, 2001, pp. 524–529.

[16] T. Simunic, L. Benini, P. Glynn, and G. De Micheli, “Dynamic power management for portable systems,” in *Proc. Int. Conf. Mobile Computing and Networking (MobiCom)*, 2000, pp. 11–19.

[17] E. Uysal-Biyikoglu and A. El Gamal, “On adaptive transmission for energy efficiency in wireless data networks,” *IEEE Transactions on Information Theory*, vol. 50, no. 12, pp. 3081–3094, Dec. 2004.