Maximizing WiFi Power-Save Mode Efficiency by Dynamically Selecting Mobile Wake-up Times

Haleh Tabrizi
Department of Electrical Engineering, Stanford University
CS229a Project Report

***Abstract*—The increasing demand for mobile computing has led to a large increase in the deployment of WiFi access points (APs) and the ever-increasing usage of mobile devices with limited energy capacity. Wireless cards have shown to account for about 50% of the total energy consumption in current handheld devices. For this reason, the 802.11 standard defines a Power-Save Mode (PSM) that aims at reducing the energy consumption of mobile devices. This mode allows energy conservation by permitting the mobile device to turn off its wireless card while the AP buffers its incoming packets. According to this scheme, the stations have to wake-up at certain time intervals to receive their buffered packets. To further increase energy conservation, a decision-making algorithm based on reinforcement learning is proposed that determines dynamic wake-up times based on the number of packets buffered at the AP. Depending on the packet arrival rate, simulation results show that energy conservation of up to 40% compared to the legacy PSM can be achieved with the proposed method.**

I. Introduction

The desire for broadband applications and high speed access to information has led to the development of sophisticated communication devices. People's lives our now contingent on personal computers, laptops, mobile phones and other devices that are connected to the global Internet. Furthermore, the general desire is to be able to interact with information anywhere and at any time. Hence there is an ever-growing demand for portable communication devices. These devices with finite battery capacity are limited by their run-time. Therefore, energy conservation is a critical requirement to elongate the portable device run-time. One of the main sources of mobile device Internet access is IEEE 802.11 wireless local area network (WLAN) that utilizes the license-free spectrum. IEEE 802.11 WLAN interfaces are increasingly incorporated in multimedia portable electronic devices to provide Internet access capabilities wherever WLAN APs are located.

According to the 802.11 standard, all stations in PSM (STA) that are associated with an AP are synchronized to wake up at the same time. At this time, the AP sends out a beacon (management) frame, that indicates what packets it has buffered. The station with a buffered packet will send out a PS-Poll frame to the AP and the AP sends the corresponding buffered packets to the station. Otherwise, the station will power off and wait till the next beacon time. One main issue with this protocol is that the station has to wake up at every beacon time even when there are no buffered packets . The energy consumed by the mobile device to power on from an off state is about 64% of the energy consumed when receiving a packet in power-on state. Hence if the number of wake-up times can be reduced while the delay introduced in the arrival of packets does not affect the target application, the energy consumption can be reduced further. There has been much work done in the literature that look at reducing mobile station power consumption in IEEE 802.11 [1]-[3].

With the knowledge of packet arrival rate statistics, this project explores how dynamic selection of STA wake-up times can increase its run-time. The selected STA wake-up times are integer multiples of beacon arrival times, i.e, the existing AP beacon transmission method does not need to be modified. The problem is modeled as a Markov Decision Process (MDP) and a cost function associated with each state-action pair is defined. Value iteration is then used to find an optimal policy for selecting STA wake-up times.

II. System Model

The communication system consists of a single AP and s associated stations as shown in figure 1. The WiFi AP is modeled as a buffer with finite capacity of q packets. Packets aimed at each associated station arrive at the AP with Poisson distribution of rate λ. The average IP data packet size is B bytes. Time is slotted and the times at the decision epochs are represented by t_1, t_2, t_3, ... At each time epoch t_k the STAs receive a beacon containing two PSM related piece of information: 1) the number of associated packets buffered at the AP and 2) the next wake-up time or the number of beacon intervals that the STA can sleep. Furthermore, at every t_k, a cost equivalent to the average power required to empty the AP buffer in addition to the cost proportional to the probability that packets get dropped until the next STA wake-up time are assigned to the AP. The goal is to determine STA wake-up times in such a manner that minimizes the long term cost. The wake-up decision at each decision epoch t_k is based on the current system state, x_k, and the future states or learned statistics of packet arrivals. It is assumed that the AP is capable of learning the network and application statistics.

The STA can switch between three different modes of operation: 1) sleep mode, during which the STA cannot receive or transmit packets and consumes the least amount of energy, 2) active mode, during which the STA can transmit and receive packets, and 3) transition mode, during which the STA transitions from sleep mode to active mode. The amount of power consumed when transitioning from active mode to sleep mode is negligible and hence omitted in this analysis. The

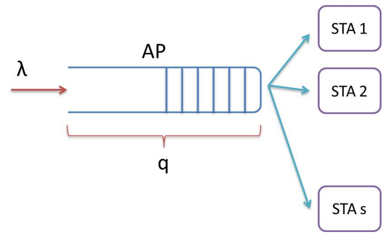

Fig. 1. Access point and stations.

amount of power consumed at each state is denoted by P_{sleep}, P_{active}, and P_{trans}, respectively. Let T_{STA}, T_b, and T_{trans} be the time required by the STA to receive a single packet of size D, receive a beacon frame, and the time required for the STA to switch from sleep mode to the active mode, respectively. As is apparent from their definitions, T_{STA} and T_b depend on the downlink data rate of the AP. Let b denote the time between two consecutive beacon frames. Since IEEE 802.11 utilizes the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) MAC protocol, there are generally both downlink and uplink traffic in one beacon interval. For simplicity, we assume the ratio between uplink and downlink packets is fixed and is equal to $d : u$. The maximum number of downlink packets that can be transmitted from the AP to the STA in one beacon interval b is denoted by N_d and is defined as:

$$N_d = \left\lfloor \frac{d}{d+u} \cdot \frac{b - T_b - T_{trans}}{T_{STA}} \right\rfloor \tag{1}$$

where $\lfloor m \rfloor$ indicates the largest integer smaller than m. The number of downlink packets that can be transmitted in one beacon frame without a sleep to active transition is defined similarly without the transition term: $M_d = \left\lfloor \frac{d}{d+u} \cdot \frac{b-T_b}{T_{STA}} \right\rfloor$.

The amount of energy consumed by a STA when there are x packets in the AP buffer with action a is denoted by $E_a(x)$. After the STA receives all the buffered packets x from the AP it goes back to sleep until the next scheduled wake-time. It is assumed that the packets that arrive at the AP during the transmission of the x packets are not transmitted instantaneously, but are buffered and constitute the next buffer state. The amount of energy consumed in one decision epoch is equal to the sum of four different sources of energy consumption: 1) the energy consumed for one sleep to active transition, 2) energy required for receiving beacon frames for the duration of time the STA is awake, 3) energy required for transmitting the maximum number of packets buffered in the AP that can be transmitted in a beacon intervals, and 4) energy consumed when the STA is sleep, i.e, for the remaining duration of time. This is represented by equation 2 below.

$$\begin{aligned} E_a(x) = T_{trans}P_{trans} + (\lfloor (x - N_d)/M_d \rfloor + 1)T_b P_{active} \\ +min\left\{x, N_d + (a-1)M_d\right\} T_{STA} P_{active} \\ +(ab - T_{awake})P_{sleep}, \end{aligned} \tag{2}$$

where T_{awake} is the total duration of time in one decision epoch that the STA is either transmitting and receiving packets or switching from sleep to active mode. In other words, it is the duration of time that the STA is awake and is defined by equation 3 below:

$$\begin{aligned} T_{awake} = T_{trans} + (\lfloor (x - N_d)/M_d \rfloor + 1)T_b \\ +min\left\{x, N_d + (a-1)M_d\right\} T_{STA}. \end{aligned} \tag{3}$$

III. Reinforcement Learning Problem Formulation

The problem of dynamically selecting STA wake-up times can be modeled as a discounted Markov Decision Process (MDP). The number of packets stored at the AP at the time of decision determines the state x of the system and hence $x \in \{0, 1, 2, ..., q\}$. The number of possible states at each decision epoch is equal to $q + 1$, and the state space X is,

$$X = \{0, 1, 2, ..., q\}\,. \tag{4}$$

The control action is denoted by a and determines the next beacon time that the STA should wake up at. The AP computes $a(x)$ based on the state of the system and communicates the information to the STA through the current beacon. The STA listens to the beacon, and by sending PS-Poll frame to the AP, it obtains as many of its buffered packets as possible. When finished receiving all the buffered packets, the STA goes to sleep and wakes up the next a beacon intervals. Let A denote the set of all possible actions:

$$A = \{a | a \in \{1, 2, ..., delay_{max}\}\}, \tag{5}$$

where $delay_{max}$ determines the maximum number of beacon intervals that a packet can be delayed in reaching its destination, and it is defined based on the underlying application delay requirements.

The next instant that the STA wakes up and listens to a beacon is the instant that was indicated by the previous received beacon. Consequently, the next instant that a wake-up decision can be communicated to the STA is its next wake-up time that is a beacon intervals from the previous decision time. Hence in this scenario the decision epochs are not fixed, but they are determined by the actions. For example, figure 2 shows beacons that are transmitted periodically at equal distances apart. The parameter b represents a beacon interval in seconds and it is the time between two consecutive beacons. In this figure, at time t_1 the first beacon carrying a decision $a_1 = 2$ is transmitted. This decision indicates that the STA can sleep during the next beacon and wake up in two beacon intervals. Furthermore, this decision indicates that the next decision epoch t_2 is two beacon intervals away from the first time epoch t_1. At time t_2 the decision is $a_2 = 3$ and indicates that the STA should wake up in three beacon intervals and the third decision a_3 will be communicated at that time.

Figure 3 represents this decision process as an MDP. The figure corresponds to the same actions taken in figure 2. This process begins at state $x_1 \in X$, and according to this state, some action $a_1 \in A$ is selected. As a result of this action, the

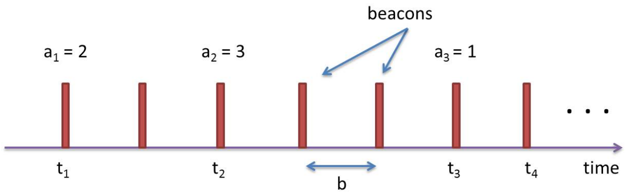

Fig. 2. Beacon intervals and time epochs.

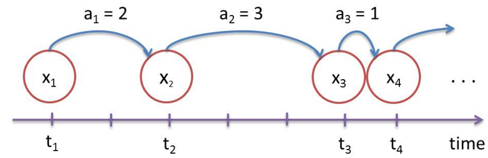

Fig. 3. Markov decision process.

state of the MDP randomly transitions to some successor state x_2 drawn according to $x_2 \sim P_{x_1,a_1}$. Then at state x_2, another action a_2 is selected and the state transitions again according to $x_3 \sim P_{x_2,a_2}$ and so forth. Let x_k be the state of the system at time t_k. If the system is in state x_k, the next state x_{k+1} is determined by the packet arrival rate λ, and the number of packets removed from the AP buffer in the time interval $[t_k, t_{k+1})$, where t_{k+1} is determined by the action a_k. Hence, given state $x_k \in X$ and a control action $a_k \in A$, the next state x_{k+1} is given by a stochastic function $f : X \times A \rightarrow X$ such that $x_{k+1} = f(x_k, a_k)$ and will be defined shortly.

Associated with each state x_k and action a_k for $k = 1, 2, ...$ is a cost function $C(x_k, a_k)$ that is defined as follows:

$$C(x_k, a_k) = \beta P(x_k, a_k) + (1 - \beta) D(x_k, a_k) \qquad (6)$$

$P(x_k, a_k)$ is the average power consumed by the STA when the AP has x_k buffered packets and the STA has to wake up in a_k beacon intervals. $D(x_k, a_k)$ is the cost of dropped packets, and $\beta \in [0, 1]$ is the trade-off factor between the two functions. Since the time is not divided equally between decision epochs, the cost of energy consumption is measured in terms of average power consumption:

$$P(x_k, a_k) = \frac{1}{a_k b} E_{a_k}(x_k), \qquad (7)$$

where $E_{a_k}(x_k)$ is the energy consumption function as described in the previous section, and $a_k b$ is the duration of time until the next decision epoch. If the AP buffer is full and cannot accommodate more packets, the packets that arrive until the STA's next wake-up time will be dropped. Assume at state x_k, action a_k has been selected. In $a_k b$ time duration, as much packets as possible, up to x_k packets will be transmitted. The remaining packets r_k, in addition to the packets that arrive in $a_k b$ time duration, h_k will fill up the buffer until the next decision epoch t_{k+1} and constitute state x_{k+1}. If $h_k + r_k$ is greater than the buffer size q, the excess amount $h_k + r_k - q$ will be dropped. and a cost proportional to the probability of dropped packets will be incurred.

$$D(x_k, a_k) = c \cdot \mathcal{P}(h_k > q - r_k), \qquad (8)$$

where $\mathcal{P}$ is the probability function and c is a constant determining the degree of importance of a dropped packet.

In order to solve the reinforcement learning problem, the cost-to-go function at state x given by $J(x)$ is written in the form of Bellman equation:

TABLE I
SYSTEM PARAMETERS

Parameter	Value	Parameter	Value
b	100	q	200
α	0.98	β	0.5
c	1000	$delay_{max}$	10
bit-rate	10Mb/s	B	512bytes
T_{STA}	2.3	P_{active}	1.4
T_b	1.33	P_{sleep}	0.045
T_{trans}	1	P_{trans}	2.3

$$J(x) = \min_{a \in A}[C(x, a) + \gamma \sum_{x'} P_{xa}(x') J(x'))] \qquad (9)$$

where γ is the discount factor, and $P_{xa}(x')$ is the probability of transitioning to state x' when action a is taken at state x. Equation 9 can be simplified and rewritten as follows:

$$J(x_k) = \min_{a \in A}[C(x_k, a_k) + \gamma \mathcal{P}(h_k \leq q - r_k) \cdot J(h_k + r_k) \qquad (10)$$
$$+ \gamma \mathcal{P}(h_k > q - r_k) \cdot J(q)]$$

If the number of packet arrivals h_k does not overload the buffer ($h_k \leq q - r_k$), the next state is equal to the number of arrivals plus the remainder of packets from the previous state: $x_{k+1} = r_k + h_k$. If the number of arrivals does overload the buffer, the excess packets will be dropped and the next state will correspond to a full buffer: $x_{k+1} = q$.

IV. SIMULATION RESULTS

There are various algorithms for solving MDPs. For an infinite horizon MDP with finite state and action spaces, value iteration is an efficient algorithm and it is the method employed here to obtain an optimal decision policy. Table I contains the values of the parameters used in obtaining a policy and performing simulations. All time parameters are in milliseconds and all power parameters are in Watts. Simulations have been performed for different values of λ. Figure 4 shows the simulation results for 670 decision epochs for the case that the packet arrival rate λ is 5 packets/sec. In this simulation scenario the packet drop rate is 0 and the power efficiency obtained compared to the legacy PSM is 36.63%. The two lower figures illustrates how actions are selected for different states. The top figure compares the power consumption when utilizing legacy PSM and the proposed PSM.

Power consumption comparison for 5 different packet arrival rates of $5, 10, 20, 50,$ and 100 packets/sec have been performed through simulations and the results are shown in

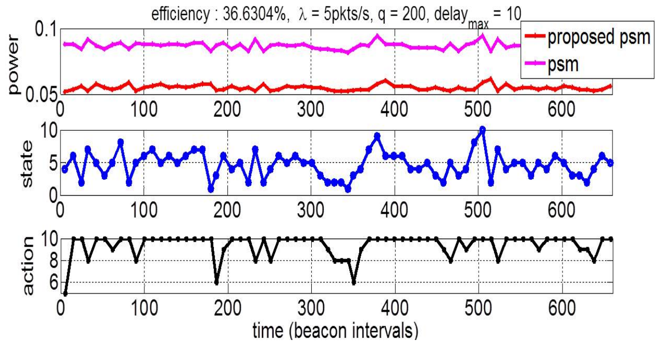

Fig. 4. Sample simulation results.

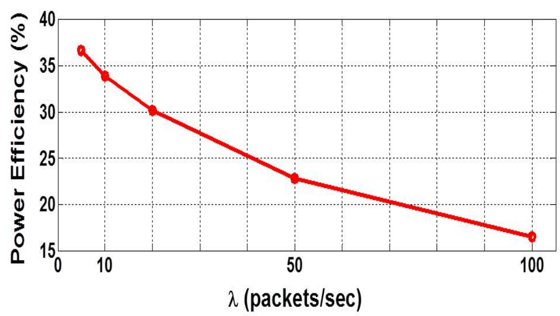

Fig. 5. Comparison of proposed PSM and legacy PSM.

Figure 5. The x-axis represents the arrival rate λ and the y-axis shows the power consumption ratio of proposed PSM to legacy PSM. As expected, the power efficiency decreases as λ increases. With increasing λ, the STA following the proposed PSM policy can sleep less often to prevent the AP buffer from overflowing and resulting in dropped packets. Hence as λ increases, the proposed PSM policy converges to the legacy PSM method.

V. Summary

This project examines dynamic wake-up time selection for stations in PSM mode as opposed to fixed wake-up times in IEEE 802.11 PSM. The objective is to minimize power consumption at the mobile station by reducing the number of times it has to wake up and receive a beacon frame. Furthermore, the delay in receiving the packets from the AP can't exceed the target application delay requirements and the number of packets buffered at the AP for the station at sleep can't overflow and result in packet drops. Based on these constraints and the packet arrival rate, an optimal wake-up time policy is obtained through reinforcement learning. It is found that a power efficiency of 40% can be obtained for low enough packet arrivals. Investigating the existance of a closed form policy for this problem is the next step for this project. Such closed form policy should determine the STA wake-up time based on the buffer size and packet arrival rate.

References

[1] P. Agrawal, A. Kumar, J. Kuri, M.K. Panda, V. Navda, and R. Ramjee, *Opsm - opportunistic power save mode for infrastructure ieee 802.11 wlan*, 2010, pp. 1–6.

[2] D. Cho J. Lee, *An energy-efficient downlink multiple access control considering congestion in wireless lans*, vol. 10, 2006, pp. 405–407.

[3] X. Ma J. Li C. Wang Y. He, R. Yuan, *Scheduled psm for minimizing energy in wireless lans*, 2007, pp. 154–163.