SOFA: A Sleep-Optimal Fair-Attention scheduler for the Power-Saving Mode of WLANs

Zheng Zeng, Yan Gao, and P. R. Kumar
Department of Computer Science,
University of Illinois at Urbana-Champaign, USA
Email: {zzeng2,yangao3, prkumar}@illinois.edu

Abstract—**Mobile devices adopt the IEEE 802.11 PSM (Power-Saving Mode) scheme and its enhancements to reduce their energy consumption when using Wi-Fi interfaces. However, the capability of PSM to save energy is limited when the WLANs are highly congested by other Wi-Fi clients.**

In this paper, instead of further pursuing the trade-off between power saving and the incurred delay on the client side, we take a different approach and explore the energy saving potential by considering the scheduling policy on the Access Point (AP) side. We find that the traditional packet-level first-come-first-serve policy is not sleep optimal since it keeps the PSM clients awake unnecessarily. We propose SOFA, an AP-centric scheme, which helps PSM clients save energy by minimizing the time they are forced to stay awake while downlink traffic is being transmitted to other clients. SOFA delivers downlink packets to the PSM clients in an optimal sequence, such that several objective are simultaneously achieved: (i) system-sleep optimality, (ii) energy-fairness, (iii) attention fairness, and (iv) no unnecessary deferral of packets beyond a beacon period. First, it determines an attention quota for each client at the beginning of each beacon period, without requiring any knowledge of available wireless capacity. Then it takes the attention "quota" and attention request as inputs to decide the downlink packet scheduling.

We prove the stability and optimality of SOFA. Simulation results shows SOFA dramatically decreases the energy consumption of PSM clients in a crowded WLAN, especially for those clients with small attention requests.

I. Introduction

There is increasing use of battery-powered portable devices, such as smart phones and PDAs, to access the Internet through Wi-Fi. Unfortunately Wi-Fi communication consumes a significant amount of energy. For instance, the Lucent IEEE 802.11 WaveLAN card [1][2] consumes 1.65 W, 1.4 W, and 1.15 W in the transmitting, receiving, and idle modes, respectively. Another example is the Motorola Droid phone, for which power consumption measurements that we have conducted show that the base energy consumption is around 200mW with the backlight off, and close to 400mW with the backlight on. In comparison, the phone's energy consumption soars to over 800mW when the Wi-Fi radio is active. It is obvious that reducing Wi-Fi power consumption is crucial for extending the battery life of portable devices.

The IEEE 802.11 Power Saving Mode (PSM) has been proposed to reduce such power drain. Unlike the original Wi-Fi Constantly Awake Mode (CAM) which draws power over 1W all the time, a mobile device running PSM can go to sleep which incurs power consumption of only around 50mW [2]. However, when a PSM device is sleeping, it cannot transmit or receive any packets; hence, PSM clients conserve energy at the cost of larger packet delivery delay.

We now provide a brief introduction to how standard PSM works. In an infrastructure WLAN, the access point (AP) sends beacon packets periodically. Each beacon packet indicates the beginning of a new beacon period, and informs the PSM clients about the presence of buffered packets via the Traffic Indication Map (TIM) field in the beacon packet. The PSM clients wake up periodically, slightly prior to the beginning of each beacon period, to listen to the TIM. If the bitmap for the client is not set to 1 in the TIM, then the client goes back to sleep immediately. Otherwise, the client has to remain awake till the last packet scheduled for it in the current beacon period is delivered. The client knows whether a packet it has received is the last one in the beacon interval by inspecting the MORE DATA bit field in the packet header. If it is set to 0, then the packet is the last one, hence, after receiving this packet from the AP, the corresponding client can go back into and stays in the sleep mode till the beginning of the next beacon period, if the client has no packet to transmit.

However, as pointed out by several researchers [3][4][5], the power consumption of a PSM client depends not only on the traffic load destined for it, but also on the traffic loads destined for other clients. The reason is that when the AP serves more than one client or there is background traffic, the packet transmission from the AP to the PSM client may be susceptible to interruption by data transmission from other hosts or from the AP to other hosts. This results in the prolongation of the time for the PSM client to receive the buffered data client, and thus more power consumption.

In order to better understand this phenomenon as well as the desired objective, consider the simple example illustrated in Figures 1-3, where there is a WLAN with one AP serving three clients, 1, 2 and 3, who have downlink traffic only. At the beginning of the first beacon period BP_1, there is no downlink packet buffered at the AP, and all the clients go back to sleep. During BP_1, four packets to Client 1,

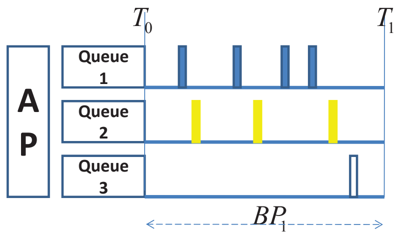

Figure 1. Packets arrive and are buffered at the AP in Beacon Period 1, denoted BP_1.

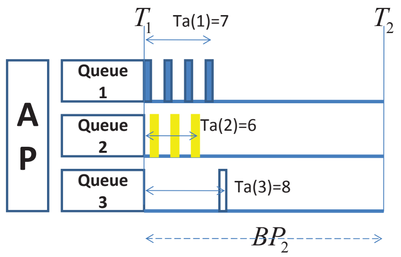

Figure 2. The AP delivers the buffered packets to clients employing the FCFS policy in BP_2.

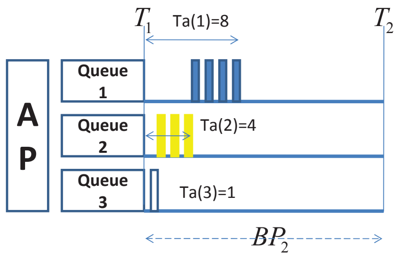

Figure 3. The alternative: The AP delivers the buffered packets to clients employing a policy that optimizes the total system's sleep time in BP_2.

three packets to Client 2, and one packet to Client 3, arrive at the AP from the Internet, as shown in Figure 1. The x-axis is the time line. Since all clients are sleeping, the AP buffers all the packets. When the second beacon period BP_2 commences, the clients wake up, and are ready to retrieve their buffered data. Figure 2 shows the time line of what happens if the packets are delivered by the AP in the normal first-come-first-serve (FCFS) order. Figure 3 shows the more desirable alternative time line of packet deliveries if they are delivered in the order that the sum of the three clients' awake time is minimized, or, equivalently, if the total sleep time over all nodes in the system, which we call system sleep time, is maximized. In both figures, $Ta(i)$ represents the duration that Client i keeps awake, in units of packet transmission time. If we consider the sum of clients' awake time as a metric representing the system energy consumption, then the second scheduling policy also achieves *system sleep optimality* by minimizing the system's total energy consumption. This example reveals that although we usually consider FCFS policy to be a fair scheduling policy that gives all clients fair service and attention, it is in fact not optimal vis-a-vis energy savings. Another point worth noting is that the FCFS policy makes Client 3 wait eight units of time to retrieve a single packet. Intuitively, users expect that the more service/attention a user gets, the more energy consumption it can afford, and thus should bear. Thus we may regard Client 3 as not being treated fairly in terms of energy consumption. In other words, FCFS does not provide energy-fairness in this example. Even the alternative scheduling in Figure 3 is not perfect. Suppose the wireless channel is lossy in BP_2 and only the first four packets are delivered before BP_2 ends. Then Client 1 does not receive its *fair attention*. Moreover, all its packets are *deferred* to the next beacon period although one of its packets would have been the first to transmit in BP_2 under the FCFS principle.

Motivated by this simple example, we address the following design problem: *What is the optimal strategy to deliver the downlink packets so that the PSM clients waste the least time staying awake and overhearing downlink traffic being transmitted to other clients?* Clearly, the absolutely sleep-optimal solution is for the AP to wake up one client, send all and only its packets to it, put it back to sleep, then wake up the next client, and repeat the process. However, this solution is not practical since the AP cannot wake up a client anytime it wants to, since the client is not capable of receiving any information from the AP while it is sleeping.

To provide a solution to this basic design problem, in this paper we present SOFA, a downlink traffic scheduler for the AP that achieves *system sleep-optimality*, and *energy fairness*, while meeting an *attention fairness constraint*, and *non-beyond beacon period deferral constraint*. Further, SOFA is analytically proved to be stable. The terminologies in italics will be formally introduced and explained in Section III, but the readers should have got a flavor of them from the above example.

The remainder of the paper is organized as follows. We discuss related work in Section II. Section III describes the problem formulation, lists the notation and terminology, and describes the objectives. Following that, in Section IV, we present the design of SOFA. Section V analytically proves the optimality and fairness of SOFA. Section VI summarizes simulation results. Finally, concluding remarks are made in Section VII.

II. Related work and Motivation

We classify the related work in the area of energy savings for Wi-Fi radios into three categories. The first category comprises of dynamic sleep/wakeup control strategies on the client side. The second category comprises of strategies that do scheduling on the AP side to reduce the energy consumption. Our work falls in this category. The third category comprises of strategies that use support from upper layers, such as network layer and application layer, to help PSM users save power.

A. The Client Side

A lot of work has been done concerning tuning the adaptive PSM to achieve a tradeoff between delay and energy saving. Krashinsky and Balakrishnan [6] have carried out a simulation analysis of PSM focusing on web-browsing traffic, and presented a Bounded Slowdown Protocol (BSD). Similarly, smart-PSM [7] provides bounded delay while minimizing energy consumption, by estimating the response-time distribution, and determines the PSM client's action sequence accordingly. Other related work includes XEM [8], which switches the wireless interface into off-mode instead of sleep-mode during long idle durations, and Forced Idling [9] which puts the radio in a low-power idling state, to avoid wasting energy due to overhearing background communications.

OPSM [10] targets a similar problem as us: the authors observe that the performance of standard PSM degrades due to congestion and background traffic. However, they choose to work on the client side.

B. *The Access Point side*

There is a lot of work on the AP side that realizes that the PSM performs poorly under heavy background traffic. The authors of [11] point out that for a selected PSM client, the existence of background traffic from the other PSM clients can result in extra energy drain. Their following work, Scheduled PSM [12], employs a TDMA scheme to handle this problem. Although it saves energy, Scheduled PSM requires modifications on both the AP side and the client side, and changes the IEEE 802.11 standard, which makes it hard to deploy.

LPTSPT and DEEM [13] use heuristics to approximate the globally optimal energy consumption of PSM clients. However, they do not consider fairness issues. Napman [3] leverages AP virtualization and a new energy-aware fair scheduling algorithm to make different PSM clients wake up at different times, so that the AP can serve them separately. This solution is beautiful; however, due to hardware limitation the AP running Napman can serve at most four PSM clients.

There is much work that embraces the idea of staggering the awake times of different PSM clients to save power, such as Centralized PSM [14] and LAWS [15]. Reference [16] studies alternative protocols that use multiple bits of the TIM to indicate the order of transmissions. All these approaches require both client and AP modifications. Our solution is more readily deployed than theirs, although in theory their solution can help the PSM clients save more energy than ours.

C. *Upper layer support*

Some researchers try to help PSM users save energy through obtaining reports from or modifying upper layers, such as network layer and application layer. PSM-throttling [17] reshapes the TCP traffic into periodic bursts, so that the client can turn on/off its wireless network interface accordingly. In [18], a proxy is used to batch packets from various streaming applications, so that the device can sleep between the receptions of batched packets.

III. Problem formulation and System Model

In this paper we consider a WLAN system consisting of one access point, m PSM clients which are numbered $\{1, 2, ..., m\}$, and n Constant Active Mode (CAM) clients which are always awake and numbered $\{m+1, m+2, ..., m+n\}$. We assume that the AP has full control of the downlink traffic and can use any scheduling policy as long as it does not violate the IEEE 802.11 standard. which is true in many enterprise WLANs nowadays. At the beginning of each beacon period BP, the AP notifies the PSM clients of the presence of buffered packets for them, through the TIM field in the beacon packet. We assume that every PSM client wakes up for beacons at the beginning of every beacon period. In the future we can relax this assumption and allow each client to choose a different DTIM window to trade delay for power conservation, but in this paper we stick to the basic model. If the client's corresponding TIM field in the beacon packet is set, it stays awake and receives its buffered packets; otherwise it goes back to sleep immediately. A MORE bit in the data packet from the AP indicates whether more packets are buffered at the AP. When there are no more packets for it buffered at the AP, the client goes back to sleep after sending a NULL packet to the AP.

A. *Notation*

Let $BP(k)$ denote the kth beacon period, and $AVEATT_i(k)$ denote the average *attention* per beacon period that Client i gets by the end of $BP(k)$. Throughout this paper the *attention* of Client i is defined as the time that the AP spends to transmit downlink packets to Client i, since in this paper our goal is to enforce *airtime fairness* [19], which has been widely regarded as a better fairness goal than throughput fairness, and has in fact been enforced by some enterprise WLAN AP manufacturers such as Meru Networks [20]. However, others may adopt other attention metrics of interest to them, such as throughput, as they wish. Let $AWAKE_i(k)$ denote the time that Client i stays awake in the k-th beacon period, $BP(k)$, till the AP allows the client to go back to sleep by setting the MORE bit to zero in the last data packet it sends to Client i in the beacon interval.

In our scheduler design, the AP transmits downlink packets to PSM clients before it sends packets to any CAM clients. In order to achieve attention fairness between PSM clients and CAM clients, the AP must decide "attention quotas" for the PSM clients. We define $CAP_{PSM}(k)$ as the maximum (or "cap" on the) time for the AP to serve all PSM clients during $BP(k)$, which is determined at the beginning of $BP(k)$, and $ACTUAL_{PSM}(k)$ (or $ACTUAL_{CAM}(k)$ respectively) as the actual time that the AP spends to serve PSM (CAM) clients during $BP(k)$, which is measured by the AP at the end of $BP(k)$. $ATT_i(k)$ denotes the attention, i.e., service time, that Client i gets during $BP(k)$.

B. *Desired objectives*

Here we describe the targets and constraints that we have in mind concerning the design the scheduler at the AP. Our goal is to achieve *system-sleep optimality* and *energy-fairness* by allowing the AP to schedule downlink traffic properly. As mentioned in Section I, we are mostly interested in reducing the energy that the PSM clients waste overhearing the transmissions to other clients. Therefore, in the remainder of this paper, "energy" refers to the specific energy consumption of the PSM clients. *System-Sleep Optimality* is then defined as follows.

Definition 1. *System-Sleep Optimality: In any $BP(k)$, suppose that the set of dowlink packets for the AP to deliver to the clients during a beacon period is given, or, equivalently say all the $ATT_i(k)$ are given. We call a WLAN scheduling policy as system-sleep optimal if the total awake time of the PSM clients, $\sum_{i=1}^{m} AWAKE_i(k)$, is minimized.*

We say the scheduling policy is *energy-fair* if it attains the min-max *Energy per Unit Attention*, where the latter is a new metric that we propose:

Definition 2. *Energy per Unit Attention (E_{ua}): Recalling that Client i must keep awake for $AWAKE_i(k)$ during $BP(k)$ in order to get service $ATT_i(k)$ from the AP, Client i's Energy per Unit Attention during the kth beacon period is defined as*

$$EUA_i(k) := \frac{AWAKE_i(k)}{ATT_i(k)}. \tag{1}$$

We now define *energy-fairness* as follows

Definition 3. *Energy fairness: Suppose that for a beacon period $BP(k)$, all the service attentions, i.e., the $ATT_i(k)$s, are specified. Then, a WLAN scheduler is said to be energy-fair if it achieves*

$$\min \max_{i=1}^{m} EUA_i(k) \tag{2}$$

over all scheduling schemes.

The logic behind this definition is quite intuitive and can be explained by a simple example. Suppose there are two clients, A and B. If A requires more attention than B does, but waits less time to get attention than B does, then we regard B as having been treated unfairly, and the goal of our scheduler design is to prevent such events from happening.

Besides the goal of saving energy, the design should meet three additional constraints described below. We call the first constraint *attention fairness* among all PSM clients and CAM clients. The motivation for this goal is that although PSM clients are given the privilege of being served before the CAM clients in order to save their energy, we do not want PSM clients to gain more attention then CAM clients. No client should take more than its fair share of the available wireless media capacity. Attention fairness is a long-term min-max fairness as defined below:

Definition 4. *Attention fairness: Suppose that in every $BP(k)$ with $k \geq 1$, the attention requested for downlink traffic of Client i is λ_i, while the total amount of attention (for downlink traffic) that the AP can serve is $TOTALATT$, with $\sum_{i=1}^{m+n} \lambda_i < TOTALATT$. We say that the system achieves attention fairness when there is a threshold such that the clients' attention allocation is as follows:*

$$\begin{aligned} AVEATT_i(k) &= \lambda_i \texttt{ if } \lambda_i < \xi, \\ AVEATT_i(k) &= \xi \texttt{ if } \lambda_i \geq \xi, \end{aligned} \tag{3}$$

and

$$\sum_{i=1}^{m+n} AVEATT_i(k) = TOTALATT. \tag{4}$$

We call ξ as the attention upper bound.

The second constraint is a *no unnecessary deferral-beyond-beacon-period constraint*, motivated by the design objective that the energy savings should not be attained at the expense of introducing excessively large media access delay to the PSM clients. It is obvious that the less frequently a client wakes up, the less energy the client consumes, but the price that is paid is the large delay that the client encounters. For example, if a client with downlink constant rate UDP traffic only requires 1/10 of the wireless airtime capacity, then the optimal power saving schedule for this client is for the AP to put this client to sleep for the first 9 beacon periods out of every 10 beacon periods, and then serve this client only during the 10th beacon period. In this way the client's attention request is met with least awake time. However such scheduling results in large communication delay and delay jitter exceeding a beacon period. Starving one client for one or several beacon periods can be catastrophic for some applications such as TCP connections and sometimes can make the client drop the WLAN connection. Therefore we do not allow such an approach in this paper. We enforce the constraint that if a client wakes up at the beginning of $BP(k)$ and has data packets buffered at the AP, then the AP cannot put it back to sleep without serving any of its already buffered traffic.

The last constraint is a *practical* constraint, requiring that all the designs and modifications should be done on the AP side only, and that they should conform to the IEEE 802.11 standard. This is for practical reasons, since a solution requiring modifications on the client side would be very difficult, if not impossible, to deploy. Similarly, designs not compatible with IEEE 802.11 will find it much more difficult to gain widespread acceptance over the short or medium term.

IV. SOFA DESIGN

A. Overview

The SOFA scheduler is designed to minimize the total energy consumption of all PSM clients and achieve energy fairness among PSM clients, while satisfying *attention fairness, no-deferral-beyond-beacon-period* and *practical* constraints. The basic idea is that in each beacon period the access point serves the PSM clients one by one. After it finishes serving one PSM client, which occurs when there are no more buffered packets at the AP for that client, or because the client has run out of its attention quota for this

BP, the AP sets the MORE DATA bit of the last packet to 0 so that the client knows that it can go back to sleep. After serving all PSM clients, the AP spends the rest of the beacon period serving CAM clients.

This AP scheduling procedure consists of three steps. First, at the beginning of the kth beacon period $BP(k)$, the AP determines $CAP_{PSM}(k)$. Second, it determines $CAP_i(k)$ for each Client i. More generally, $CAP_i(k)$ is defined as the maximum attention that PSM Client i can get during $BP(k)$, while throughout this paper $CAP_i(k)$ refers to the maximum time for the AP to deliver downlink packets to Client i during $BP(k)$. The AP then orders the PSM clients based on $CAP_i(k)$ and the *Predicted Attention Request* PA_i. By PA_i we mean the estimated transmission time to deliver all packets of Client i that are currently buffered at the AP. Last, the AP spends the remaining portion of the beacon period to serve the CAM clients. We note that the protocol is flexible with respect to how it treats the CAM clients: it can adopt any existing research [20][19] to achieve attention fairness among CAM clients. Next we elaborate on the first two steps of SOFA.

B. CAP determination

Although the AP knows the duration of its own beacon period (denoted by BP in the sequel), the total available attention/service time for downlink traffic, TOTALATT, is usually less than T, because of the existence of uncoordinated uplink traffic, other WLANs in the neighborhood, transmission failures caused by channel fading, collisions, and many other reasons. Even worse, since TOTALATT is also time-varying, it is very difficult to estimate $TOTALATT(k)$ at the beginning of kth beacon period $BP(k)$. Therefore $CAP_{PSM}(k)$ cannot be directly assigned. We propose an adaptive algorithm to assign $CAP_{PSM}(k)$ based on the AP's historical observation of the system's wireless medium, and the performance of the clients. The pseudo-code specification is given in Procedure 1. Recall that $ACTUAL_{PSM}(k-1)$ ($ACTUAL_{CAM}(k-1)$, respectively) is the actual time that the AP spends to serve PSM (CAM, respectively) clients during $BP(k)$. When the wireless medium is not fully utilized, $Idle(k)$ in Procedure 1 is the wireless media idle time experienced by the AP during $BP(k)$. The main idea of this Procedure 1 is that we divide the clients into two groups: the PSM group and the CAM group. When the wireless capacity is not large enough to support all clients' attention requests, then during the current beacon period, the AP gives more attention to the group which has historically received less attention. The fairness and optimality properties of the protocol are proved in section V.

C. CAP allocation and serving sequence

After $CAP_{PSM}(k)$ is determined, the AP starts scheduling downlink traffic for the PSM clients. The AP needs to make two decisions: (1) In the current beacon period, among all the PSM clients that have packets buffered at the AP, how much service should the AP assign to each client, and (2) In what sequence should the AP deliver data packets to those clients. We have already pointed out that although packet level first-come-first-served (FCFS) is a good fairness policy, scheduling packets in a FCFS manner is not sleep optimal. This raises the question of what policy to use. Theorem 1 helps in answering this question by determining the best order to serve PSM clients once their service times ATT_i for $1 \leq i \leq m$ are given.

Procedure 1 ADJUSTMENT of $CAP_{PSM}(k)$

```
Input: $CAP_{PSM}(k-1)$, $ACTUAL_{PSM}(k-1)$,
    $TOTAL_A(k-1)$, $ACTUAL_{CAM}(k-1)$, $Idle(k-1)$
Output: $CAP_{PSM}(k)$
 1: if $k = 1$ then
 2:     $CAP_{PSM}(k) \leftarrow T/4$
 3:     return
 4: end if
 5: if $Idle(k-1) > 0$ then
 6:     if $ACTUAL_{PSM}(k-1) < CAP_{PSM}(k-1)$ then
 7:         $CAP_{PSM}(k) = CAP_{PSM}(k-1)$
 8:     else
 9:         $CAP_{PSM}(k) = CAP_{PSM}(k-1)$
            $+ACTUAL_{CAM}(k-1) \times \theta$
10:     end if
11: else
12:     if $ACTUAL_{PSM}(k-1) = TOTAL_A(k-1)$ then
13:         $CAP_{PSM}(k) \leftarrow ACTUAL_{PSM}(k-1)$
14:     end if
15:
            $a = \max_{i=1\ldots m} AVEATT_i(k-1)$
            $b = \max_{i=m+1,\ldots,m+n} AVEATT_i(k-1)$
16:     if $ACTUAL_{PSM}(k-1) = CAP_{PSM}(k-1)$ then
17:         $CAP_{PSM}(k) += (b-a) \times \varphi$
18:     else
19:         $CAP_{PSM}(k) += \min\{0, (b-a) \times \varphi\}$
20:     end if
21: end if
```

Since all PSM clients wake up at the beginning of the beacon period, and Client i does not go to sleep until it has been served for ATT_i, $AWAKE_i$ consists of the time interval spent on the clients that are served prior to Client i, and the time interval spent on Client i (i.e., ATT_i).

Let us define a *noninterrupted schedule* as a schedule in which once the AP starts to serve a client, it continuously works on this client till the service assigned to this client is completed. In contrast, an *interleaved schedule* is a schedule in which some client's service is interrupted by service to others.

Lemma 1. *The system sleep optimum is achieved by a noninterrupted schedule.*

Proof: Suppose the system sleep optimum is achieved by an interleaved schedule. Then, the serving time of some clients is split into several subintervals τ. Denote the last subinterval of the client by τ_e. A client can go to sleep only after its τ_e has been served. Since the schedule is interleaved, there must be a client whose τ_e follows a subinterval of another client that is not the last subinterval. Swapping the serving order of the two subintervals yields another schedule. The new schedule reduces the total energy consumption because client i can go to sleep earlier. Therefore, the optimal scheduling must be a noninterrupted schedule. ■

Thus, we consider only noninterrupted schedules. That is, once the AP starts to serve client i, it continuously serves it for time interval ATT_i.

Theorem 1. *Given a set of service times $\{ATT_i\} : 1 \leq i \leq m$ for a beacon period, serving the corresponding clients in ascending order of ATT_i achieves the system sleep optimum.*

Proof: Consider a schedule σ that is not in the ascending order of ATT_i and denote its total energy consumption by $E(\sigma)$. Then, there must be some ATT_n and ATT_m such that $ATT_n \leq ATT_m$ and m is served prior to n. Another schedule σ' can be obtained by swapping the serving order of n and m. σ' reduces the total energy consumption since

$$E(\sigma') = E(\sigma) - (k+1)(ATT_m - ATT_n) < E(\sigma),$$

where k is the number of clients that are served between ATT_m and ATT_n. Thus, σ' is a strictly better schedule than σ, and establishing the result. ■

Theorem 2. *Given a set of service times $\{ATT_i\} : 1 \leq i \leq m$ for a beacon period, serving the corresponding clients in the ascending order of ATT_i achieves energy fairness.*

Proof: If a schedule σ is not in the ascending order of ATT_i, then there must be some ATT_n and ATT_m such that $ATT_n \leq ATT_m$ with m served prior to n. We will show that another schedule σ' that swaps only the serving orders of n and m will achieve smaller EUA than σ. By Definition 3, $EUA_n(\sigma) = \frac{T_{a_n}(\sigma)}{ATT_n}$ and $EUA_m(\sigma) = \frac{T_{a_m}(\sigma)}{ATT_m}$. Since m is served prior to n in σ, we have $T_{a_m}(\sigma) \leq T_{a_n}(\sigma)$. Thus, we have $EUA_m(\sigma) < EUA_n(\sigma)$.

σ' swaps n and m and preserves other clients in the same order as σ. Hence, only the clients between m and n change their EUA value. Let j be some client served between m and n. After swapping m and n, $T_{a_j}(\sigma') = T_{a_j}(\sigma) - ATT_m + ATT_n \leq T_{a_j}(\sigma)$. Thus,

$$EUA_j(\sigma') \leq EUA_j(\sigma). \tag{5}$$

Since $T_{a_n}(\sigma') < T_{a_n}(\sigma)$, we have

$$EUA_n(\sigma') < EUA_n(\sigma). \tag{6}$$

We know $T_{a_m}(\sigma') = T_{a_n}(\sigma)$ and $ATT_m \geq ATT_n$, hence

$$EUA_m(\sigma') \leq EUA_n(\sigma). \tag{7}$$

From (5), (6) and (7),

$$\max_i EUA_i(\sigma') \leq \max_i EUA_i(\sigma). \tag{8}$$

Therefore, we can conclude that the ascending order achieves the minimum of $\max_i EUA_i$ over all possible schedules. ■

To summarize, we have determined both an attention-fair and a sleep-optimal order for serving clients, *provided* we know their service times $\{ATT_i : 1 \leq 1 \leq m\}$. Therefore the next issue is to determine how to choose $\{ATT_i : 1 \leq 1 \leq m\}$ so as to yield attention fairness as well as satisfy the no-deferral constraints. In the following subsections, we start with three intuitive scheduling solutions that *do not* work well, in order to show that this is a non-trivial problem and how an examination of their failures leads us to the right solution: SOFA.

D. The Least Attention First (LAF) Scheduler

Procedure 2 provides the pseudo-code of the Least Attention First (LAF) scheduler, and Procedure 3 briefly describes how the AP handles the PSM client l which currently has the highest priority. At the beginning of the beacon period, the AP picks up the client that has had least historical attention and transmits to it all its buffered packets. Then it picks the client with second least attention to serve, and so on.

The Serve-with-Prioity procedure tries to make PSM client l receive all of its buffered packets before the other clients. First it employs the idea of "extra backoff", to give l a higher priority to retrieve its packets. Note that in each beacon period, the AP must receive one or multiple control messages from a PSM client (to prove the client is awake) before it sends out the data packets to it. For a *static* PSM client, every single data packet transmission from the AP requires a PS-POLL message from the client; and for a *dynamic* PSM client, the AP must first receive a NULL packet indicating the client is awake for the rest of beacon period, before transmitting the buffered packets for that client. The "extra backoff" makes the AP wait longer for those control packets. Yet when the extra backoff terminates, the AP can switch to serving other clients without wasting too much time waiting for the control packets when client l fails, malfunctions, or just does not wake up. For practical reasons, we cannot allow the AP waiting for l's control packets forever. Therefore, besides the steps showing in Procedure 2, the AP takes away client l's priority if the PS-POLL or NULL packet does not come with a grace period, say 5ms, to give the priority to other clients, and may return to l the priority if its PS-POLL or NULL packet does arrive later in this beacon period. Furthermore we make use of the Unscheduled Automatic Power Save Delivery (U-APSD) defined by the IEEE 802.11e standard. When a PS-POLL/NULL packet from l is received, instead of sending out one packet to l, the AP can send dequeue multiple packets of l and send them in a burst, using the 802.11e TXOP-like mechanism before its CAP is met. The Serve-with-Prioity procedure is also used in the other scheduling policies in this paper.

However, LAF cannot guarantee that it serves clients in the increasing order of service time ATT_l. For example, consider three clients A, B, and C, whose attentions are

Procedure 2 Least Attention First

```
Input: $CAP_{PSM}(k), AVEATT_i(k-1), i = 1, ..., m$
Output: $ATT_i(k), AVEATT_i(k), i = 1, ..., m$
1: $U = \{c_1, \cdots, c_m\}, CAPR(k) \leftarrow CAP_{PSM}(k)$
2: while $U \neq \Phi$ and $CAPR(k) > 0$ do
3:     $p \leftarrow \arg\min_l \{AVEATT_l(k-1) | c_l \in U\}$
4:     $ATT_p(k)$ = Serve-with-Priority($c_p, \infty$)
5:     UPDATE $AVEATT_p(k)$
6:     $U \leftarrow U - \{c_p\}$
7:     $CAPR(k) \leftarrow CAPR(k) - ATT_p(k)$
8: end while
```

$AVEATT_A = AVEATT_B > AVEATT_C$. Suppose however C has bursty traffic while the traffics of A and B are smooth. Then during the beacon period when C's burst arrives, the AP serves C first, although $ATT_A = ATT_B < ATT_C$. Moreover, the existence of bursty traffic also prevents LAF from satisfying the no-deferral constraint. Therefore we must set an upper bound on ATT_i, which leads to the next policy.

E. The History-Based Attention (HBA) Scheduler

As shown in Procedure 4, HBA calculates the attention capacity bound for each client at the beginning of each beacon period, and then serves the clients in the increasing order of estimated service time. The function $f(AVEATT_l(k-1), U)$ in the third line calculates the proportion of service time that client l can get, based on its own historical attention as well as the attention of all clients in the set U. As long as function $f(x, U)$ decreases with x, HBA assigns the AP's attention fairly among all PSM clients, because it always assigns more service "quota" to clients with less historical attention. Although HBA appears to meet our requirements at first glance, it has a fundamental problem: HBA may waste service time. When there exists a client i whose predicted attention is less than its "quota," the extra capacity assigned to client i is wasted and cannot be reused by other clients whose attention demands are larger than their "quotas." Therefore, even if the total $CAP_{PSM}(k)$ during $BP(k)$ is sufficient to meet all clients' attention requirements, the use of HBA can result in some clients suffering degraded performance. In order to eliminate such service time wastage, one can consider another scheduler design, *Online* HBA, as we next describe.

F. Online HBA Scheduler

Procedure 5 gives the pseudo-code of *Online* HBA scheduling. Instead of pinning down the service bound of every client at the beginning of the beacon period, *Online* HBA updates the service bounds of the remaining clients that have not been served yet. The AP does such updating every time it finishes serving one PSM client, say client i, and puts it back to sleep. If client i has used up its "quota," then the remaining service time is re-assigned for usage by the remaining clients. In this way *Online* HBA reduces the

Procedure 3 Serve-with-Priority(Client, Attention Bound)

```
Input: client $l$, $CAP_l$
Output: $ACTUAL_l$
1: if PS-POLL from $l$ has been received then
2:     GOTO 7
3: end if
4: $CAP \leftarrow CAP_l$
5: while (this $BP$ not finished) and (remaining $CAP > 0$) do
6:     isExtraBackoff ← false, mark PS-POLL from $l$ not received
7:     Do normal 802.11 standard backoff
8:     while No Event do
9:         Switch event:
10:        CASE (PS-POLL from $l$ is received):
11:        mark PS-POLL from $l$ received
12:        HeadofTXqueue ← Dequeue($l$)
13:        if isExtraBackoff then
14:            GOTO 35
15:        end if
16:        CONTINUE
17:        CASE (Awake Notification (a NULL packet) from $l$)
18:        HeadofTXqueue ← Dequeue($l$,CAP), mark $l$ AWAKE
19:        if isExtraBackoff then
20:            GOTO 38
21:        end if
22:        CONTINUE
23:        CASE (Backoff timeout)
24:        if PS-POLL from $l$ has been received then
25:            GOTO 35
26:        else
27:            if $l$ AWAKE then
28:                GOTO 38
29:            else
30:                isExtraBackoff ← true; DO Extrabackoff;
31:                CONTINUE
32:            end if
33:        end if
34:        CONTINUE
35:        CASE (Extra Backoff timeout)
36:        send one of the other packets with highest priority
37:        BREAK
38:    end while
39:    packet.MORE ← 0 if it is the last packet to $l$
40:    send the packet popped up from TXQueue (immediately
       without backoff), UPDATE CAP
41: end while
42: TREAT $l$ as a CAM client and start to transmit TXQueue
43: return $ACTUAL_l$ = the time taken to transmit $l$'s packets
```

Procedure 4 HBA

```
Input: $CAP_{PSM}(k), AVEATT_1(k-1), ..., AVEATT_m(k-1)$
Output: $ATT_i(k), AVEATT_i(k), i = 1, ..., m$
1: $U = \{c_1, \cdots, c_m\}, CAPR(k) \leftarrow CAP_{PSM}(k)$
2: for each $c_l \in U$ do
3:     $CAP_l = CAPR(k) \times f(AVEATT_l(k-1), U)$
4:     $ATT_l(k) = \min(CAP_l(k), PA_l(k))$
5: end for
6: while $U \neq \Phi$ and $CAPR(k) > 0$ do
7:     $p \leftarrow \arg\min_l \{ATT_l(k) | c_l \in U\}$
8:     Serve-with-Priority($c_p, ATT_l(k)$)
9:     UPDATE $AVEATT_p(k)$
10:    $U \leftarrow U - \{c_p\}$
11:    $CAPR(k) \leftarrow CAPR(k) - ATT_p(k)$
12: end while
```

Procedure 5 Online HBA

```
Input: $CAP_{PSM}(k), AVEATT_1(k-1), ..., AVEATT_m(k-1)$
Output: $ATT_i(k), AVEATT_i(k), i = 1, ..., m$
 1: $U = \{c_1, \cdots, c_m\}, CAPR(k) \leftarrow CAP_{PSM}(k)$
 2: for $j = 1, \cdots, m$ do
 3:     for each $c_l \in U$ do
 4:         $CAP_l^j = CAPR(k) \times f(AVEATT_l(k-1), U)$
 5:         $ATT_l(k) = \min(CAP_l^j(k), PA_l(k))$
 6:     end for
 7:     $p \leftarrow \arg\min_l \{AVEATT_l | c_l \in U\}$
 8:     Serve-with-Priority($c_p, ATT_p(k)$)
 9:     UPDATE $AVEATT_p(k)$
10:     $U \leftarrow U - \{c_p\}$
11:     $CAPR(k) \leftarrow CAPR(k) - ATT_p(k)$
12: end for
```

service time wastage in comparison to HBA. Unfortunately *Online* HBA cannot fully get rid of service time wastage.

Let us look into a simple example. Consider a WLAN with one AP and two PSM clients 1 and 2 associated with the AP. Let $CAP(k) = 50ms$, $PA_1(k) = 10ms$, and $PA_2(k) = 40ms$ for every beacon period k. Suppose that the service time bound assignment function f is defined as follows:

$$f(AVEATT_1, \{1,2\}) = \frac{AVEATT_2^2}{AVEATT_1^2 + AVEATT_1^2}, \quad (9)$$

$$f(AVEATT_2, \{1,2\}) = \frac{AVEATT_1^2}{AVEATT_1^2 + AVEATT_2^2} \quad (10)$$

In this WLAN, the attention that client B achieves is never above $20ms$. The reason for this is that the service bound assignment function f over compensates client A by giving it too much service quota. Suppose $AVEATT_2(k) > 20ms$ at $BP(k)$. Then we have $f(AVEATT_2, \{1,2\}) < 0.2$, and therefore

$$\begin{aligned} ATT_2(k) &= CAP_{PSM}(k) \times f(AVEATT_2, \{1,2\}) \\ &< 10ms = ATT_1(k). \end{aligned} \quad (11)$$

Note that the AP serves client 2 before client 1 (from lines 7-8 in Procedure 5), and so client 2 only gets $10ms$ service time, which results in $AVEATT_2$ falling below $20ms$ again. *Online* HBA cannot reach a stable scheduling sequence even when the network environment and traffic are stable. We have been unable to find any function f that always avoids causing the above situation in all general WLAN settings. A scheduling policy that is not *stable* is the last thing a system needs. We come up with the next design in order to solve the stability issue,

G. The Attention Fair (AF) Scheduler

After summarizing the pros and cons of the other attempts at designing scheduling policies, we now design the Attention Fair policy, AF, as given in Procedure 6. AF is derived from *Online* HBA, with three major modifications.

First, in each beacon period the AP running AF first serves those clients whose quota is larger than their predicted attention. In this way, no service time is wasted. Second, we

Procedure 6 AF

```
Input: $CAP_{PSM}(k), AVEATT_1(k-1), ..., AVEATT_m(k-1)$
Output: $ATT_i(k), AVEATT_i(k), i = 1, ..., m$
 1: $U \leftarrow \{c_1, \cdots, c_m\}, CAPR(k) \leftarrow CAP_{PSM}(k)$
 2: for $j \leftarrow 1, \cdots, m$ do
 3:     $V \leftarrow \Phi$
 4:     for each $c_l \in U$ do
 5:         $f(l, U) = \frac{\sum_{i \in U} AVEATT_i(k-1) - AVEATT_l(k-1)}{\sum_{i \in U} AVEATT_i(k-1) \times (m-j)}$
 6:         if (j=1) then
 7:             $CAP_l^j = CAPR(k) \times f(AVEATT_l(k-1))$
 8:         else
 9:             $CAP_l^j = CAP_l^{j-1} + R_{j-1} \times f(AVEATT_l(k-1))$
10:         end if
11:         $ATT_l(k) = \min(CAP_l^j(k), PA_l^j(k))$
12:         if ($CAP_l^j > PA_l^j(k)$) then
13:             $V = V \cup \{c_i\}$
14:         end if
15:     end for
16:     if $V \neq \Phi$ then
17:         $p \leftarrow \arg\min_l \{ATT_l | c_l \in V\}$
18:     else
19:         $p \leftarrow \arg\min_l \{ATT_l | c_l \in U\}$
20:     end if
21:     $ATT_p(k)$ = Serve-with-Priority($c_p, CAP_l^j$)
22:     $R_j = CAP_l^j - ATT_p(k)$
23:     UPDATE $AVEATT_p(k)$
24:     $U \leftarrow U - \{c_p\}$
25:     $CAPR(k) \leftarrow CAPR(k) - ATT_p(k)$
26: end for
```

carefully choose a specific quota assignment function f as specified in line 5. Third, as *Online* HBA does, AF adjusts the quotas of the remaining clients which are still awake each time the AP finishes serving any client. However, instead of re-assigning the quota, AF simply increases each client's quota if the client that is just going to sleep has not used up its quota. The details of the algorithm are shown in line 9. Note that with this new quota adjustment algorithm, any client's attention quota never decreases before it gets served within a beacon period. The last two modifications will be revisited and explained in the next section when we prove the stability of AF.

H. Attention update

The average attention $AVEATT_i$ for client i is updated as follows,

$$AVEATT_i(k+1) = \alpha \times AVEATT_i(k+1) + (1-\alpha) \times ATT_i(k), \quad (12)$$

at the end of each beacon period, whiere $\alpha \in (0,1)$ is a smoothing factor.

V. Protocol Analysis

In this section we prove that when both the wireless medium and the clients' attention requests are stable, then by running SOFA on the AP, the clients' service times $ATT_1, ..., ATT_m, ..., ATT_{m+n}$ converge to the fair point described in Definition 4, and furthermore, that the order that the AP serves all PSM clients within a beacon period also converges to the order that is both system sleep-optimal and energy-fair. We start by proving that if all clients of the AP are PSM nodes, then by running AF on the AP, $(ATT_1, ..., ATT_m)$ will converge to the fair point, and the order that the AP serves all PSM clients within a beacon period also converges as above.

A. The Stability of AF

First the following assumptions are made. Since we are analyzing the situation when the wireless medium and the clients' attention requests are stable, we assume that $\lambda_i(k) \equiv \lambda_i$ and $CAP_{PSM}(k) \equiv CAP$ for all k. Also, we assume $PA_i(k) \equiv \lambda_i$. The Stability of AF is trivial to prove when $\sum \lambda_i < CAP$; therefore we only study the case where $\sum \lambda_i \geq CAP$. Recall that according to Definition 4, ξ is the attention upper bound. Without loss of generality we assume, by renaming the clients if necessary, that $\lambda_i \leq \lambda_{i+1}$ and $PA_i \leq PA_{i+1}$, for $i = 1, ..., m-1$. In order to prove the stability of AF, we must show two things. First, we must show that there exists one (equilibrium) state which, once hit, the system will stay forever at. Second, we must show that the system will move to the equilibrium state eventually regardless of the initial or current state.

Lemma 2. *At $BP(k)$, if we have $ATT_i(k) = AVEATT_i(k-1) = \lambda_i$ (if $\lambda_i < \xi$); and $ATT_i(k) = AVEATT_i(k-1) = \xi$ (if $\lambda_i \geq \xi$), then at $BP(k)$, we must have $ATT_i(k+1) = AVEATT_i(k) = \lambda_i$ (if $\lambda_i < \xi$), and $ATT_i(k+1) = AVEATT_i(k) = \xi$ (if $\lambda_i \geq \xi$).*

Proof: We can check that Lemma 2 is true by analyzing AF at $BP(k+1)$ and simply calculating $ATT_i(k+1)$. ■

Lemma 2 claims that the equilibrium state satisfies the property that for any client i, its ATT_i and $AVEATT_i$ both equal its fair attention share.

Note that AF does not waste any service time, which means that $\Sigma_i ATT_i(k) = CAP$ for all $k \geq 1$. Therefore, when k is large enough we must have $\Sigma_i AVEATT_i(k) = CAP$ too, no matter what the initial setting of $AVEATT_i(0)$ is. Therefore, during the proof of system state convergence, we can assume that $\Sigma_i AVEATT_i(k) = CAP$ for all $k \geq 1$ without loss of generality.

Lemma 3. $\texttt{Min}_i AVEATT_i(k)$ *increases with k, as long as*

$$\min_{1\leq i\leq m} AVEATT_i(k) \leq \min\{\frac{1}{m}, PA_1, ..., PA_m\}. \quad (13)$$

Proof: At the beginning of $BP(k)$, the first round of attention quota assignment results in

$$\begin{aligned} CAP_i^1 &= \frac{\sum_j AVEATT_j(k-1) - AVEATT_i(k-1)}{(m-1)\sum_j AVEATT_j(k-1)} \times CAP \\ &= \frac{CAP - AVEATT_i(k-1)}{m-1} \\ &= \frac{\sum_{j\neq i} AVEATT_j(k-1)}{m-1}, \end{aligned} \quad (14)$$

where the second equality holds because

$$\Sigma_i AVEATT_i(k-1) = CAP$$

. Equation (14) shows that for any client i, its CAP_i^1 is the average attention of all other clients, which is always larger than $\min_i AVEATT_i(k-1)$. Assume $AVEATT_l(k-1) = \min_i AVEATT_i(k-1)$, then, clearly

$$CAP_i^1 \geq AVEATT_l(k-1), \texttt{ for all } i \neq l. \quad (15)$$

Because $AVEATT_l(k-1) \leq \frac{CAP}{m}$, we have

$$\begin{aligned} CAP_l^1 &= \frac{CAP - AVEATT_l(k-1)}{m-1} \\ &\geq \frac{CAP}{m} \geq AVEATT_l(k-1). \end{aligned} \quad (16)$$

Putting (15) and (16) together, we have

$$CAP_i^1 \geq AVEATT_l(k-1), \texttt{for } i = 1, ..., m. \quad (17)$$

Let CAP_i denote the *real* attention bound that is assigned to client i right before the AP serves it. Recall that the attention bound of each client never decreases within a beacon period (from lines 6-9 in Procedure 6). Therefore

$$CAP_i \geq AVEATT_l(k-1), \texttt{for } i = 1, ..., m. \quad (18)$$

Since the service never exceeds the demand, we have for all i,

$$PA_i \geq AVEATT_i(k-1) \geq AVEATT_l(k-1). \quad (19)$$

Note that client i's service time is

$$ATT_i(k) = \min\{CAP_i, PA_i\}, \texttt{for } i = 1, ..., m. \quad (20)$$

By substituting (18), (19) into (20), we get

$$ATT_i(k) \geq AVEATT_l(k-1), \texttt{for } i = 1, ..., m. \quad (21)$$

Because $AVEATT_i$ is updated by taking a linear combination of old $AVEATT_i$ and current service time ATT_i as given in (12), and (21) holds for all clients, it follows that we must have

$$\min_{1\leq i\leq m} AVEATT_i(k) \geq \min_{1\leq i\leq m} AVEATT_i(k-1). \quad (22)$$

The proof is complete.

■

Lemma 4. *It is impossible that*

$$\min_{1\leq i\leq m} AVEATT_i(k) < \min\{\frac{CAP}{m}, PA_1, ..., PA_m\}$$

for all k.

Proof: The proof is by contradiction. Since $\min_i AVEATT_i(k)$ is bounded and non-decreasing, $\lim_{k\to\infty} \min_i AVEATT_i(k)$ exists. Suppose the limit is $\underline{A}$, with $\underline{A} < \min\{\frac{CAP}{m}, PA_1, ..., PA_m\}$. Picking the next beacon period $BP(k)$ and taking its limit, we can study what is happening as we get close to the limit. We see that $\lim_{k\to\infty} \min_i AVEATT_i(k) = \underline{A}$, and let c_l denote the client with the smallest attention quota in the limit. Then, considering the next beacon period and taking the limit, we must have $\lim_{k\to\infty} \min_i AVEATT_i(k+1) = \underline{A}$ too.

Suppose the client with the smallest attention quota in the limit is c_j.

Next we will show that it is impossible that $j = l$, and also impossible that $j \neq l$. If $j = l$, then we must have $\lim_k ATT_l(k+1) = \underline{A}$, and this only happens if $\lim_k AVEATT_i(k) \equiv \frac{CAP}{m}$ for all i. Hence $\underline{A} = \frac{CAP}{m}$, which is a contradiction. On the other hand, if $j \neq l$, then we must have $\lim_k ATT_j(k+1) < \underline{A}$, which contradicts (21). ■

Now we can prove the stability of AF.

Theorem 3. *When the AP runs the proposed AF scheduling policy, the scheduling order and attention assignment converge to the equilibrium state in Lemma 2.*

Proof: According to Lemma 4, there must exist a beacon period $BP(k)$ such that

$$\begin{aligned}\min_i AVEATT_i(k) &= \min\{\frac{CAP}{m}, PA_1, ..., PA_m\} \\ &= \min\{\frac{CAP}{m}, PA_1\}. \qquad (23)\end{aligned}$$

The second equality holds because we have assumed that PA_1 is the smallest PA. If $\min_i AVEATT_i(k) = \frac{CAP}{m}$, then $AVEATT_i(k) = \frac{CAP}{m}$ for all i; hence we have reached the equilibrium state in Lemma 2. Otherwise, $\min_i AVEATT_i(k) = PA_1$. Then c_1 has both smallest attention request and highest attention quota. According to AF, c_1 must be the first client to be served in every beacon period from then on, which is how c_1 is supposed to be served in the equilibrium state. By recursion it follows that the behavior of the remaining $m-1$ clients also converges to the equilibrium state. ■

Theorem 4. *The equilibrium state is system sleep-optimal and achieves min-max unit attention energy fairness.*

Proof: Clearly, in the equilibrium state, the AP serves the clients in the ascending order of ATT_i. Then the statement is true according to Theorems 1 and 2. ■

In the next section we show that PSM and CAM clients share attention fairly.

B. Attention Fairness between PSM and CAM clients

Theorem 5. *If CAM clients also achieve attention fairness among themselves, then all clients achieve attention fairness.*

Proof: Suppose at $BP(k)$, the CAM clients share their attention quota fairly amongst themselves, and the PSM clients share their attention quota fairly among themselves too. Procedure 1 identifies the following four cases and adjusts $CAP_{PSM}(k+1)$ accordingly. we use ξ_1 to denote the attention upper bound of PSM clients, and ξ_2 for CAM clients.

- Case 1: Neither CAM nor PSM clients use up their quota. Then $CAP_{PSM}(k+1) = CAP_{PSM}(k)$ from line 7 in Procure 1.
- Case 2: Only PSM clients use up their quota. Then CAP_{PSM} increases in the next beacon period from line 9.
- Case 3: Only CAM clients use up their quota. Then from line 20, CAP_{PSM} decreases in the next beacon period only if $\xi_1 > \xi_2$, and stays the same otherwise.
- Case 4: Both PSM and CAM clients use up their quota. Then we compare ξ_1 and ξ_2. In the next beacon period, we give more quota to whichever is a smaller upper bound, from line 18.

From the manner in which Procedure 1 handles the four cases, we see that if the total wireless capacity is sufficient to serve all PSM and CAM clients, then the choice of CAP_{PSM} makes every client satisfied. Otherwise, the AP will keep giving more quota to the group with lower quota upper bound, till $\xi_1 = \xi_2$. Therefore the global attention fairness is achieved. ■

Table I
THE FLOW RATES OF 5 CLIENTS

Client No.	1	2	3	4	5
Flow rate (kbps)	100	300	700	1200	1200

VI. Performance Evaluation

In this section, we evaluate SOFA via the NS-2 simulator [21]. We first verify that SOFA assigns attentions to PSM and CAM clients fairly. Then we show that SOFA significantly reduces the system energy consumption by comparing its performance with three other scheduling policies. We also show that SOFA achieves better energy fairness than other scheduling policies.

A. Simulation setup

We use ns-2.33 with Power Management Extension [22] to conduct our simulation. For the energy model, we use the default settings of ns-2.33 Power Management Extension. The powers consumed by the wireless network interface in the transmit, receive, idle and sleep state are 660mW, 395mW, 35mW, and 1mW, respectively. The beacon period is 100ms. The default packet size is 800 bytes, and the default wireless link rate is 2Mbps if not stated otherwise. All wireless nodes are within each other's transmission range.

B. Attention Fairness verification

The network setup is as follows. There is one AP (denoted by node A) and there are five clients associated with it. Clients numbered 1-4 are PSM nodes, and client numbered 5 is a CAM node. Each client receives a CBR UPD flow from the AP. The flow rates are given in Table I. Another wireless node B is not associated with A, but it shares the wireless medium with A's WLAN, and broadcasts CBR packets. Node B plays the role of a wireless neighbor, whose existence makes it difficult for the AP to know the total

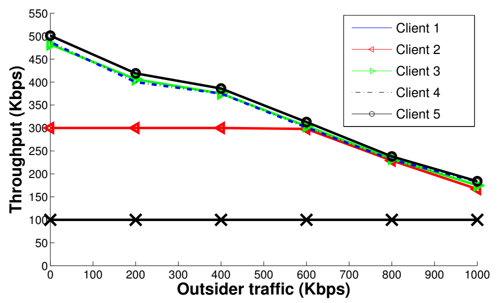

Figure 4. Per-client throughput for UDP downlink traffic with varying CBR background jamming traffic in a single rate WLAN.

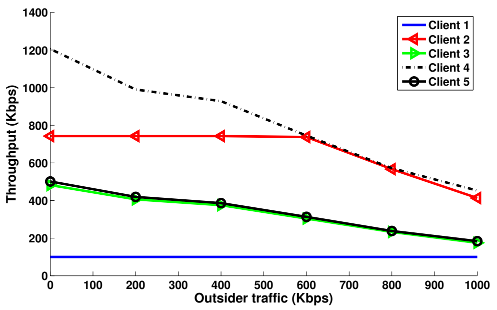

Figure 5. Per-lient throughput for UDP downlink traffic with varying CBR background jamming traffic in a multi-rate WLAN.

available service time that it can provide to its clients. We run several sets of simulations with node B's flow rate varying from 0 to 1Mbps. Figure 4 gives the throughput of every client when the link rate is 2Mbps for all clients. Since the link rates are all the same, attention/airtime fairness is equivalent to throughput fairness. In Figure 4, clients 3-5 which have saturated traffic share the attention fairly, while client 1 which has a lesser attention request gets well served. Client 2 is satisfied when the outsider, B's traffic is low. But when B's flow rate increases and takes more wireless medium, the available attention/service time of the WLAN increases. Therefore Client 2 cannot be fully satisfied, but it still gets a fair share of attention vis-a-vis client 4-6.

Figure 5 shows the throughput of every client when clients have different link rates. The network settings are the same as above, except that the link rates of Clients 2 and 4 increase to 5.5Mbps and the flow rate of Client 2 increases to 750Kbps. Figure 5 further verifies that SOFA achieves attention/airtime fairness among PSM and CAM clients in multi-rate wireless network. For example, when B's flow rate is 800Kbps, the throughput of Client 2 is 556Kbps and the throughput of Client 5 is 238Kbps, but that is because 2's bandwidth is 2.25 times of 5's. These two clients' airtime (the time that the AP sends data packet and receives ACK) are the same.

C. Power consumption evaluation

Now we compare the performance of four scheduling policies in a WLAN consisting of an AP, three PSM clients

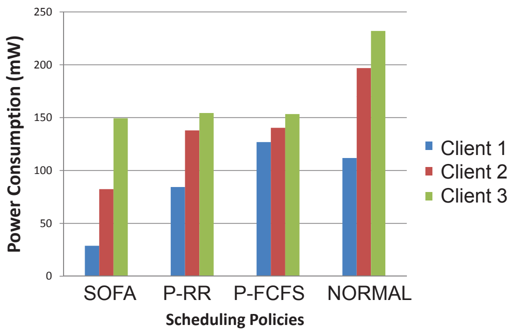

Figure 6. Comparing the power consumption of three clients under four scheduling policies.

numbered 1-3, and one CAM client numbered 4. The flow rates of the downlink CBR UDP traffic to Clients 1-4 are 150Kbps, 300Kbps, 400Kbps and 400Kbps respectively. The four scheduling policies are: SOFA, Priority-Round-Robin (P-RR), Priority-First-Come-First-Serve (P-FCFS), and Normal. In the case of normal scheduling, on receiving a PS-POLL packet or a NULL frame from a PSM client, the AP enqueues one or more PSM packets at the tail of the transmission queue. In the second and third scheduling policies, the buffered PSM packets are transmitted immediately by queueing the packets in a separate transmission queue with a higher priority. When there are multiple PSM clients, P-RR schedules their packet deliveries in Round-Robin manner, while P-RR schedules them in FCFS manner. Figure 6 compares the energy consumptions of three clients. Because there is no uplink traffic, and the energy the node spent in sleep state is almost negligible, therefore the client's energy consumption mostly comes from receiving packets, whether destined to it or not. Consequently, the energy saving comes mostly from putting the client to sleep more. Figure 7 compares the *energy per unit attention* of three clients, which is the energy for the client to receive 1MB downlink data. Here we replace airtime with throughput since they are equivalent in a single rate WLAN. Clearly SOFA outperforms the other three polices. SOFA especially helps client 1 which has the least attention request by reducing its EUA by over 2/3 compared with any other scheduling algorithm. This is because SOFA favors clients with smaller attention demand.

VII. Conclusion

In this paper we propose a downlink traffic scheduler on the AP of a WLAN, called SOFA, which helps its PSM clients save energy by allowing them to sleep more, hence increase battery lives. If a client has buffered packets at the AP in a beacon period, and that client decides to receive it, it has to remain awake from the beginning of the beacon period till the last packet scheduled for it in the beacon period is delivered. Therefore a large portion of energy wastage (for

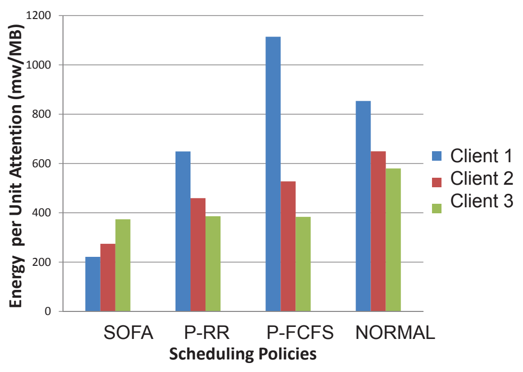

Figure 7. Comparing the Energy-per-Unit-Attention of three clients under four scheduling policies.

the client) comes from the AP transmitting other clients'a packets before it finishes transmitting the client's last packet to it. SOFA manages to reduce such energy wastage and maximizes the total sleep time of all clients. SOFA favors clients with smaller attention requests by allowing them to spend less energy to get one unit of attention, while still helping other clients with larger attention requests to sleep more compared with other popular scheduling policies like round-robin and FCFS. Furthermore, SOFA enforces attention fairness among clients and avoids unnecessary beyond beacon period deferral of packets. SOFA is practical, and its stability and performance have been rigorously proved.

VIII. ACKNOWLEDGMENTS

The authors are grateful for a variety of valuable comments from the anonymous reviewers. This material is based upon work partially supported by NSF under Contracts CNS-0905397, CNS-1035340, CNS-1035378, CCF-0939370, and CNS-1035340, USARO under Contract No. W911NF-08-1-0238, and AFOSR under Contract FA9550-09-0121.

REFERENCES

[1] M. Stemm, R. H. Katz, and Y. H. Katz, "Measuring and Reducing Energy Consumption of Network Interfaces in Hand-Held Devices," in *IEICE Transactions on Communications*, 1997.

[2] Lucent, "IEEE802.11 WaveLAN PC Card - Users Guide ."

[3] E. Rozner, V. Navda, R. Ramjee, and S. K. Rayanchu, "NAPman: network-assisted power management for wifi devices," in *MobiSys*, 2010.

[4] Y. He and R. Yuan, "A Novel Scheduled Power Saving Mechanism for 802.11 Wireless LANs," *IEEE Transactions on Mobile Computing*, 2009.

[5] P. Agrawal, A. Kumar, J. Kuri, M. K. Panda, V. Navda, R. Ramjee, and V. N. Padmanabhan, "Analytical models for energy consumption in infrastructure WLAN STAs carrying TCP traffic," in *COMSNETS*, 2010.

[6] R. Krashinsky and H. Balakrishnan, "Minimizing Energy for Wireless Web Access Using Bounded Slowdown," in *ACM MOBICOM*, 2002.

[7] D. Qiao and K. G. Shin, "Smart Power-Saving Mode for IEEE 802.11 Wireless LANs," in *IEEE INFOCOM*, 2005.

[8] G. A. A, M. C. B, E. G. B, and A. P. B, "802.11 Power-Saving Mode for Mobile Computing in Wi-Fi hotspots: Limitations, Enhancements and Open Issues," in *Wireless Netw*, 2008.

[9] S. Biswas and S. Datta, "Reducing Overhearing Energy in 802.11 Networks by Low-Power Interface Tuning," in *International Conference on Performance, Computing and Communications*, 2004.

[10] P. Agrawal, A. Kumar, J. Kuri, M. Panda, V. Navda, and R. Ramjee, "OPSM - Opportunistic Power Save Mode for Infrastructure IEEE 802.11 WLAN," in *Communications Workshops (ICC)*, 2010.

[11] Y. He, R. Yuan, X. Ma, and J. Li, "Analysis of the impact of background traffic on the performance of 802.11 power saving mechanism," *Comm. Letters.*, March 2009.

[12] Y. He, R. Yuan, X. Ma, J. Li, and C. Wang, "Scheduled PSM for Minimizing Energy in Wireless LANs," *Network Protocols, IEEE International Conference on*, 2007.

[13] J. Lee, C. Rosenberg, and E. K. P. Chong, "Energy efficient schedulers in wireless networks: design and optimization," *Mob. Netw. Appl.*, vol. 11, June 2006.

[14] Y. Xie, X. Luo, and R. K. C. Chang, "Centralized PSM: an AP-centric power saving mode for 802.11 infrastructure networks," in *Proceedings of the 32nd international conference on Sarnoff symposium*, 2009.

[15] H.-P. Lin, S.-C. Huang, and R.-H. Jan, "A power-saving scheduling for infrastructure-mode 802.11 wireless LANs," *Comput. Commun.*, vol. 29, November 2006.

[16] J. A. Stine and G. D. Veciana, "A comprehensive energy conservation solution for mobile ad hoc networks," in *IEEE International Communication Conference*, 2002.

[17] E. Tan, L. Guo, S. Chen, and X. Zhang, "PSM-throttling: Minimizing Energy Consumption for Bulk Data Communications in WLANs," *Network Protocols, IEEE International Conference on*, 2007.

[18] S. Chandra and A. Vahdat, "Application-specific Network Management for Energy-Aware Streaming of Popular Multimedia Formats," in *Proceedings of the General Track of the annual conference on USENIX Annual Technical Conference*, 2002.

[19] T. Joshi, A. Mukherjee, Y. Yoo, and D. P. Agrawal, "Airtime Fairness for IEEE 802.11 Multirate Networks," *IEEE Transactions on Mobile Computing*, vol. 7, pp. 513–527, 2008.

[20] "http://www.merunetworks.com/technology/rflayer/atf.php."

[21] "The network simulator - ns-2," in *http://www.isi.edu/nsnam/ns/*.

[22] "http://ns-2.blogspot.com/2008/09/wireless-lan-power-management-extension.html."