Arbitrating Traffic Contention for Power Saving with Multiple PSM Clients

Dachuan Liu, Haining Wang, *Senior Member, IEEE,* Gang Zhou, *Senior Member, IEEE,* Weizhen Mao
and Boyang Li, *Student Member, IEEE,*

***Abstract*—Data transmission over WiFi quickly drains up mobile devices' battery. Although IEEE 802.11 standards provide Power Save Mode (PSM) to help mobile devices conserve energy, PSM fails to bring expected benefits in many real scenarios. In particular, when multiple PSM mobile devices associate to a single Access Point (AP), PSM does not work well under transmission contention. Optimizing power saving of multiple PSM clients is a challenging task, because each PSM client expects to complete data transmission early so that it can turn to low power mode. In this paper, we define an energy conserving model to describe the general PSM traffic contention problem. We prove that the optimization of energy saving for multiple PSM clients under constraint is an NP-Complete problem. Following this direction, we propose a solution called HPSM to address one specific case, in which multiple PSM clients associate to a single AP. In HPSM, we first use a basic sociological concept to define the richness of a PSM client based on the link resource it consumes. Then we separate these *poor* PSM clients from *rich* PSM clients in terms of link resource consumption, and favor the former to save power when they face PSM transmission contention. We implement prototypes of HPSM based on the open source projects Mad-wifi and NS-2. Our evaluations show that HPSM can help the *poor* PSM clients effectively save power while only slightly degrading the rich's performance in comparison to the existing PSM solutions.**

***Index Terms*—WiFi, PSM, AP, Traffic Contention, Scheduling.**

I. INTRODUCTION

As WiFi offers high bandwidth and free data transmission, it is a preferable choice for most mobile users to access Internet services. However, the data transmission over WiFi quickly drains up the battery of a mobile device [33], [7], [6]. IEEE 802.11 standards provide PSM for mobile devices to conserve energy. PSM allows mobile devices to retain in sleep status by turning off network components if there is no data to be transmitted, and mobile devices wake up periodically to check whether there are packets buffered on AP.

In an ideal scenario that one AP serves one PSM client, the client can finish receiving buffered packets as soon as possible. However, it is quite common that there are multiple PSM clients associated to a single AP in real environments. Each PSM client has to contend with the others for the data channel. Assisting all PSM clients to save power is a hard work, since each of them hopes to save as much power as possible. We create an energy conserving model to benefit all PSM clients under fairness constraint. We prove that optimizing the energy conserving under the fairness constraint is an NP-Complete problem. Motivated by the problem and the observations on a real device experiment, we put our focus on handling a specific PSM contention problem. When one client occupies the channel, the others have to wait under high power mode. If some clients have many data packets buffered at AP to be transmitted (e.g., users download large files to the device), they will dominate the data transmission during the whole beacon period, costing those clients that have only very few packets buffered at AP (e.g., users check e-mail or send short messages on the device) a long period of waiting time to win the contention for data transmission. Thus, the existing PSM favors the clients that consume more link resource with higher energy efficiency, but penalize the clients with less link resource consumption in power saving. This leads to an unfairness problem. And the accumulated power consumption of a poor, which is in less power efficiency, becomes unnecessarily high as time goes by.

We propose a harmonious power saving mechanism (HPSM) to address the unfairness problem in PSM. The key idea is to use the fundamental sociological concept [11] to handle PSM traffic contention. We define the link resource as public resource and mobile devices' battery life as private resource. If a PSM client consumes much public resource, it is *rich*; otherwise, it is *poor*. In real society, poor people could not afford more expensive public services like higher education, but they pay much lower tax rate. Similarly, in a system consisting of one AP and multiple PSM clients, the clients consuming a large portion of link resource should pay more for the service, but those consuming a small portion of resource should get the service with less cost. HPSM intentionally prioritizes the data transmission of poor clients over that of rich clients. The early completion of data transmission allows the poor clients to switch back to sleep status for more significant power saving. Meanwhile, as poor clients only occupy the data channel for a very short period of time, the tax in terms of latency and energy paid by these rich clients are minor.

We implement prototypes of HPSM based on Mad-wifi and NS-2. We first conduct an experimental evaluation using a testbed with four smartphones, then involve more clients in the simulation environments. Our evaluations show that HPSM

This work was partially supported by DHS grant D15PC00187 and NSF grant CNS-1253506.

Dachuan Liu is with the University of Delaware, the College of William and Mary, USA. E-mail: dliu@cs.wm.edu

Haining Wang is with the University of Delaware, USA. E-mail: hnw@udel.edu

Gang Zhou, Weizhen Mao and Boyang Li are with the College of William and Mary, USA. E-mail: {gzhou,wm,boyang}@cs.wm.edu

can help a poor PSM client save up to 25% energy comparing to existing PSM solutions on the real device. We also evaluate the effectiveness of HPSM with different ratios between the poor PSM clients over the rich, and how the fairness constraint impacts the performance of the clients. As HPSM is an AP side solution, it is easy to be incrementally deployed.

We summarize our major contributions as follows:

- We define a problem of power saving under contention from multiple PSM clients, and we prove that problem is NP-Complete.
- Motivated by the problem and the observations on real experiments, we propose to classify the PSM clients as poor and rich in terms of their link resource usage and provide an AP side solution to save power under contention.
- We implement the prototype and show that the solution can help the poor save power while only slightly degrading the performance of the rich through both real and simulation experiments.

The remainder of this paper are organized as follows. The related works are introduced in Section II. The PSM traffic contention problem and the energy conserving model are described in Section III. In Section IV, we analyze the specific contention problem based on the experimental results. Section V details the design of HPSM and describes its implementation. The evaluation is given in Section VI. Finally, we discuss and conclude this work in Sections VII and VIII, respectively.

II. RELATED WORK

Although PSM is proposed to help mobile devices conserve energy, it cannot bring all expected benefits in many cases. To date, a variety of schemes have been developed to enhance PSM from different aspects. We divide these previous efforts into three categories: extending sleep period, handling traffic contention, and miscellaneous. Our work falls in the second category.

A. Extending Sleep Period

In [17] and [26], adaptive techniques for WiFi management are firstly proposed to minimize WiFi energy consumption as well as to bound the network latency for a specific network activity. STPM [5] adapts PSM behaviors by considering both the current traffic and the application intent. The current traffic is monitored and the application intent is retrieved from available API. An adaptive U-APSD approach is also proposed in [8] for web browsing and file downloading. In [24], two power saving mechanisms are proposed. The first mechanism Sleeping allows the network subcomponents to stay at sleep status, when these components are idle. The second one Rate-Adaption modifies the network devices' frequency based on network traffic rate. M. Khan et al. [16] mainly addressed the rate adaption problem of 802.11 n. They proposed to adjust the rate for striking a balance between throughput and energy conservation.

Some works save power through fine-grained adaption. Micro Power [21] allows the WiFi interface to stay at low power mode in short intervals (several microseconds), which provides considerable opportunities for clients to save power. SAPSM [25] utilizes a classifier to identify delay-sensitive applications. A client switches between CAM and PSM according to application's network activities. Also, an algorithm is proposed in [9] to dynamically enable and disable a WiFi interface based on the instant multimedia buffer size.

Some other works save power by reshaping traffics. In [28], PSM-throttling reshapes the TCP packet chunks so that a client can receive packets continuously. Catnap [10] allows AP to buffer packets for clients' applications. The buffered packets are sent in burst so that the clients can take advantage of long idle periods for power saving. These works enhance power saving mechanisms for individual clients or specific application scenarios. They do not address the power saving issue under the contention from multiple PSM clients, which is the focus of our work. Note that our work could work in complement with some of them in a mixed application scenario. For example, HPSM may cooperate with Catnap to improve a client's performance by assigning higher priority to it when it faces the contention from other clients.

B. Handling Traffic Contention

In centralized PSM [30], different parameters are introduced, such as listen interval and beacon interval, to enable clients to wake up at different periods. LAWS [20] allows AP to reduce contention through scheduling polling requests from clients. In [18], the authors formulated the problem of saving power over all PSM clients and presented heuristic power saving solutions for multiple PSM clients. These works can alleviate the contention problem, but they do not consider the balance between power saving and network performance (latency) at a fine-granularity level. They need the modifications on both AP and client sides, which impedes them from being deployed in large scales. In [15], [14], the authors revealed that PSM background traffics could trigger PSM clients to drain battery. They proposed Scheduled PSM [13] to handle the contention based on a TDMA scheme. However, this work requires AP to take over the channel using RTS/CTS or Self-CTS at the beginning of each time slot. It is complex and requires the modification on IEEE 802.11 standards. Differently, HPSM is an AP side solution that strikes a balance between power saving and network performance.

Napman [27] is proposed to isolate PSM packets from CAM packets. PSM packets are sent through a high priority queue without impacting fairness. Napman leverages virtual AP technology to handle the PSM traffic contention. But the virtual AP technology can only handle the contention from limited number (4 by default) of PSM clients. In addition, it cannot handle the contention when clients associate to the same SSID. J. Manweiler et al. [23] investigated the network contention from multiple APs. They proposed Sleepwell to help clients save power by avoiding the rush hour of traffic. SOFA [31] aims to minimize the total energy consumption of the system. It calculates the 'quota' for all clients, and schedules packets based on the 'quota'. The authors did not consider the impacts on system performance incurred by interaction among PSM clients and AP. X. Lei et al. [19] proposed a solution to reduce the contention between PSM nodes in Wireless Sensor Networks (WSN). The PSM clients are divided into three groups corresponding to three priorities, then wake up for data transmission based on their priorities. Yet they did not consider the priority assignment.

As far as we know, our work is the first to classify the PSM clients into “rich” or “poor” in terms of their link resource usage, and offer the poor with higher priority over the rich for receiving packets. Meanwhile, we consider the loss of link resource caused by the channel reservation.

C. Miscellaneous

DozeAP [12] helps a smartphone save power when it enables tethering over WiFi. It allows the hotspot to negotiate with clients so that the packet loss rate during the hotspot’s sleep period is reduced. C. Zhang et al. [32] proposed a set of optimization mechanisms to lower the power usage by reducing the redundant codec operations, video tail traffics, and dynamically relocating the channel to maximize the transmission efficiency. In [29], 3G is the primary channel for data transmission and WiFi is utilized as the second radio for power saving. F. Lu et al. [22] improved the power saving for WiFi devices from a different perspective. By leveraging the sparsity in Direct Sequence Spread Spectrum of 802.11 b, a proposed transceiver can transmit and receive packets at low clock rate. Thus, it helps a client save power. These works resolve the different problems from our work. They improve a client’s power efficiency by utilizing the system’s resources. HPSM may also work in complement with some of them, for example, HPSM can be deployed with DozeAP so that the poor client’s performance will be improved and the hotspot can still save energy.

III. PSM TRAFFIC CONTENTION PROBLEM

In this section, we first briefly describe the PSM traffic contention problem. Then, we define a model to optimize energy conserving for multiple PSM clients. We show that the optimization is an NP-Complete problem.

A. The Contention Problem

When there is only one PSM client associating to an AP, the communication process between them is simple and clear. PSM can assist the client to conserve energy as expected. However, multiple PSM clients associate to an AP is quite common in real environment. In such case, PSM could fail to bring expected energy conservation. When AP communicates with multiple PSM clients in a beacon period. At the beginning of a beacon period, the AP checks its buffer and identifies packets which need be forwarded to the PSM clients. It broadcasts TIM to notify all PSM clients. After the clients receive the TIM message, all of them try to send PS_Poll to AP. But only one of them can win the contention and receive packets from AP each time. During this period, the other PSM clients have to wait at active status. The contention increases the waiting time of data transmission for all clients. Probably all PSM clients could stay actively in the whole beacon period, and hence have no chance to save power.

B. Energy Conserving Model Definition

Under PSM, AP passively responds to the PS_Poll, and the packets are delivered in random sequence for the contention. In a long term, we assume that each client can get the channel for receiving packet with equal probability. All clients will receive packets in turn; therefore, they could stay actively in a whole beacon period, which results in no energy conserving. If AP schedules the packets of some clients with higher priority so that they can finish data transmission earlier in a beacon period, these clients can switch to sleep status in the remainder of the beacon period for power saving. However, the schedule will bring unfairness to the other clients since it increases the delay of receiving packets to them. Hence, the loss of fairness should be bounded so that the negative consequence is tolerable for those clients. We define an energy conserving model to proactively schedule packets for achieving this objective. Given a sequence of packets, the model will schedule the packets to maximize the power saving for all clients. In the meantime, the unfairness incurred by the schedule is bounded. In the following, we present the details of the energy conserving model.

Assume we have a sequence of packets $Q = \{p_1, p_2 \dots p_m\}$ for n clients. For each packet $p_i \in Q$, p_i is identified by a tuple $(c, \#)$, where c denotes the client it belongs to, and the notation $\#$ denotes its position in Q. We introduce $p_i.c$ to denote the client which p_i belongs to and $p_i.\#$ to denote p_i’s position in Q $(p_i.\# = i)$. After the packets are re-scheduled (switched) by the model, a new sequence is generated as $Q' = \{p'_1, p'_2 \dots p'_{m'} \mid m' \leq m\}$, where $Q' \subseteq Q$ $(p'_\ell \in Q)$. For a packet p'_ℓ in Q', if $(p'_\ell.\# - \ell) > 0$, it means the packet has been moved forward and the packet switch incurs unfair delay to those packets which are in front of this packet before switch. We use $p'_\ell[val] = p'_\ell.\# - \ell$ to denote the loss of fairness if a packet has been moved forward; otherwise, $p'_\ell[val] = 0$. We add up the number of positions being moved forward for each $p'_\ell \in Q'$ to denote the total loss of fairness. $\mathcal{B}$ is defined as the fairness bound. We have the constraint: $\sum_{\ell=1}^{|Q'|} p'_\ell[val] \leq \mathcal{B}$. Then we measure the energy conserving after packets switch. During packet switching, we need to guarantee the packet sequence of a specific client should not be changed. For all $l \in [1, |Q'|]$, where $p'_\ell[c] = j$, we use the notation T_{S_j} to denote $p'_\ell[val]$ where p'_ℓ is the last packet of the client in sending queue. The larger this value, the earlier the client can finish data transmission. Similarly, the larger the sum of T_{S_j}, the more energy the clients can conserve. Note although the last packet’s delivery time determines the energy conserving, we could still move multiple packets forward to make a client finish packet receiving earlier. For example, the positions of a client’s last two packets are 6 and 7. If we attempt to move the last packet forward to position 5, both packets need to be moved. So, T_{S_j} represents the energy conserving and the sum of $p'_\ell[val]$ is for the loss of fairness, where $p'_\ell[c] = j$. The formal definition of the model is described in Figure 1.

C. NP-Complete Proof

In this section, we show that the energy conserving problem is NP-Complete. Seeking the optimal energy conserving for all

INPUT: $m, n \in \mathbb{Z}^+$, a fairness bound $\mathcal{B}$ and a input sequence of $Q = \{p_1, p_2 \dots p_m\}$.

CONSTRAINS: $\sum_{l=1}^{|Q'|} |p'_\ell[val]| \leq \mathcal{B}$.

QUESTION: Is there a new sequence Q', where $Q' \subseteq Q$, with $\text{Max}(\sum_{j \in [1,n]} T_{S_j})$?

Notations: Let p'_ℓ denote the ℓth element in Q'. Since $Q' \subseteq Q$, $p'_\ell \in Q$. For all $l \in [1, |Q'|]$ where $p'_\ell.c = j$, we use the notation T_{S_j} to denote the last $(p'_\ell[val])$, where $p'_\ell[c] = j$.

Fig. 1: The energy conserving model definition.

clients $(\sum_{j \in [1,n]} T_{S_j})$, we introduce a parameter k to represent the energy conserved under any schedule that is constrained by the fairness bound, such that $(\sum_{j \in [1,n]} T_{S_j}) \geq k$.

First, we show that this problem is in NP. The packets are chosen from Q and put in Q', then the verification process calculates $\sum_{l=1}^{|Q'|} |p'_\ell[val]|$ and $\sum_{j \in [1,n]} T_{S_j}$ for comparison. The whole process takes polynomial time in the size of input.

Second, we demonstrate that a known NP-Complete problem, the Partition problem, is reducible to our energy conserving model in polynomial time. The definition of the Partition problem is:

Given a finite set $A = \{a_1, a_2, \dots, a_n\}$, where a_i is a positive number, is there $A' \subset A$, such that $\sum_{a'_i \in A'} a'_i = 1/2 \sum_{a_i \in A} a_i$?

Our goal is to prove that there exists a reduction $F(\cdot)$ in polynomial time, which can bring yes instance $F(X)$ of the energy conserving problem iff X is yes instance of the Partition problem. Given the set A in the Partition problem, $F(\cdot)$ will convert it to the energy conserving problem. For each element $a_i \in A$, we create a packet p_i, where $p_i.c = i$ and $p_i[val] = a_i$. We also set $\mathcal{B} = 1/2 \sum_{i \in [1,n]} a_i$ and $k = 1/2 \sum_{i \in [1,n]} a_i$. The conversion is completed in polynomial time. Then we show that the energy conserving model returns yes with inputs Q, $\mathcal{B}$ and k iff A has a partition.

If A has a partition, there exists a set $A' \subset A$, where $\sum_{a'_i \in A'} a'_i = 1/2 \sum_{a_i \in A} a_i$. For each $a'_i \in A'$, we have a packet p'_i corresponding to a'_i. Then we create a new queue in which we put these packets in Q' such that $\sum_{p'_i \in Q'} p'_i[val] = 1/2 \sum_{a_i \in A} a_j = \mathcal{B}$ and $\sum_{j \in [1,n]} T_{s_j} = \sum_{p'_i \in Q'} p'_i[val] = 1/2 \sum_{a_i \in A} a_i = k$. On the contrary, if the energy conserving model returns yes when the inputs are Q, $\mathcal{B}$ and k, there exists a new queue in which we have the packets moved forward in Q', where $\sum_{p'_i \in Q'} p'_i[val] \leq \mathcal{B}$ and $\sum_{j \in [1,n]} T_{s_j} \geq k$. $T_{s_j} = p'_i[val]$, where $p'_i.c = j$, since each client only has one packet based on our construction. We have $\sum_{p'_i \in Q'} p'_i[val] = 1/2 \sum_{a_i \in A} a_i$, because $\mathcal{B} = k = 1/2 \sum_{a_i \in A} a_i$. For each $p'_i \in Q$âĂŹ, where $p'_i[val] > 0$, we have a corresponding a'_i in A' that is a partition of A.

D. Rich and Poor in the Contention Problem

Although saving power for all PSM clients under the constraint is an NP-Complete problem, it is possible to help some PSM clients save power. We can consider a WiFi system containing one AP and a group of PSM clients as a small society, in which the PSM clients present as the residents. We define the link bandwidth as public resource and the client's battery as private resource. The clients consuming a large portion of public resource (i.e., link bandwidth) are classified as the rich, and the clients consuming a small portion of link bandwidth are classified as the poor. To be fair, the rich should pay more private resource (i.e., battery consumption) for using public resource while the poor should pay less of its private resource. However, the contention problem causes unfair energy consumption to the poor clients. For example, two PSM clients associate to AP. Client 1 has two packets buffered on AP, and client 2 has 10 packets. If AP schedules all client 2's packets to deliver first, client 2 can finish receiving packets as early as possible. However, client 1 has to wait for a long period at high power status, which is unfair for client 1. Alternatively, if AP delivers all packets to client 1 first, client 1 can finish receiving packets and switch to sleep status soon. And client 2 just waits for a short period at high power status, which could be tolerable for client 2. As mentioned in the model definition, we should consider the impact of the induced delay for re-scheduling the packets. In the following, we first show how the traffic contention impacts the poor client's performance unfairly through the experiment with real devices in Section IV. Then, we present how we handle the problem in Section V.

IV. ANALYZE THE PSM CONTENTION PROBLEM

We demonstrate the negative impacts of PSM traffic contention upon a poor PSM client in terms of power consumption and performance by conducting real experiments.

A. Experimental Setup

Our testbed consists of one physical AP and four smartphones working under the PSM mode. We plug a D-Link DWA552 (AR5416 Chipset) in a Dell Precision 390 workstation, which works as the AP. The running OS is Opensuse with kernel 2.6.37.6. We choose the driver ath9k provided by the open source project Mad-wifi, as it provides stable performance and supports virtual AP technology. We use four Sprint HTC Hero smartphones equipped with the WiFi chipset TI 1251 as the PSM clients. The OS is Android ROM of Cyanogenmod 7 with customized kernel which supports standard PSM. We use Monsoon [4] to measure power consumption on a smart phone. AP creates an object corresponding to a client when the client associates to AP. We calculate the average latency and record packet retransmission on AP for each object when the events (packet delivery or retransmission) are detected. The WiFi system parameters are listed in Table I. The beacon interval is set to 100ms. The bit rate is set to 1Mbps so that we can easily observe the contention when the bandwidth is relatively small. One of the smart phone is set as poor. We send one packet to the poor every beacon period. The bit rate is about 44Kbps. We send about five packets to each rich client during a beacon period under bit rate of 220Kbps. All packets carry the payload in the same length. Each experiment is 180 seconds long and repeated three times. During the measurement, we turn off the

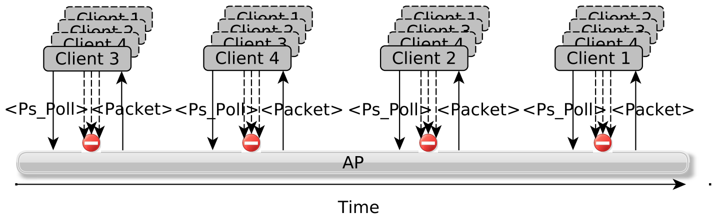


Fig. 2: Multiple PSM clients contend for data transmission.

smart-phones' screens and no other applications are running in the background.

TABLE I: System parameter setup

Parameters	Value
802.11 mode	b
Beacon interval	100ms
Bit rate	1 Mbps
Poor's bit rate	1 pkt/beacon(44kbps)
Rich's bit rate	5 pkts/beacon(220kbps)

B. Impact of PSM Traffic Contention

Based on our testbed, we conduct a series of experiments to investigate the impact of PSM traffic contention upon PSM clients. There are one poor PSM client and three rich PSM clients associating to a single AP. At the beginning, all of them are in sleep status. Then we start to send packets to them. AP broadcasts TIM information at the beginning of the beacon period. PSM clients wake up to check TIM, and find that they have packets buffered on AP. Then they send PS_Poll to AP for receiving packets. Each PSM client will contend for the channel with the others so that AP can receive its PS_Poll. The packet request and delivery will be in a random sequence, as shown in Figure 2.

1) Snapshot of A Beacon Period: The poor client switches to sleep status after data transmission. However, the rich clients will keep in active status for data transmission. Each rich client could send two continuous PS_Polls at the beginning of a beacon period: one is the reply to the beacon broadcast message; and the other is the response to the more bit in the last received packet. The consequence is that the poor spends a long period of time, during which it stays at high power mode, to win the contention.

Figure 3 depicts the poor's power measurement of a typical beacon period when it faces the contention from three rich clients. The poor almost waits 40ms for receiving a packet after it wakes up. On average, the poor client has to wait 33.76ms in a beacon period before it's PS_Poll is handled and the related power consumption is 44mJ. By contrast, in the scenario that only one client associate to AP, the client only need to wait 13.13ms and the corresponding power consumption is merely 18.63mJ. Thus, the contention cause 136% increase in power consumption for the poor client. The rich stays at active status during a whole beacon period, which makes it look like a CAM client. This is because the rich always has packets to receive in the beacon period and it has to wait while AP is serving the other two.

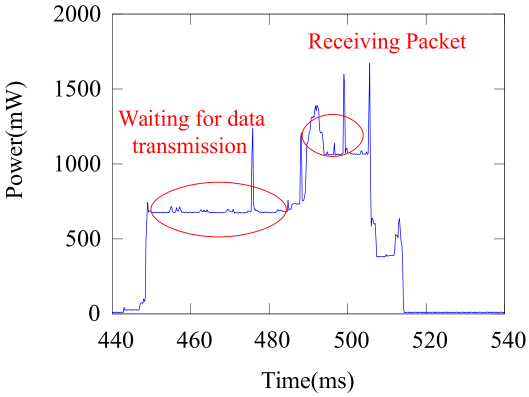


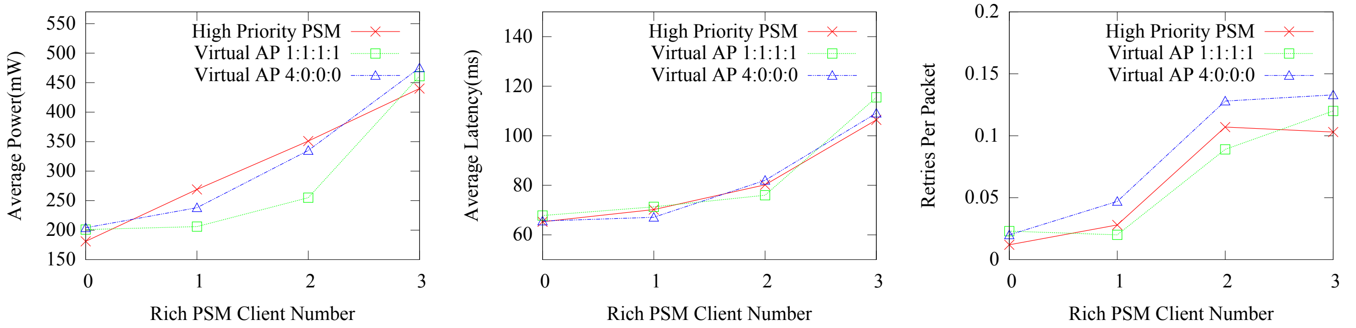
Fig. 3: Power measurement in a beacon period

2) Escalated Contention: To further demonstrate the PSM contention impact upon the poor's power consumption and performance, we vary the number of rich clients from 0 to 3 in the controlled experiments. We implement a high priority PSM scheduler on the AP, which puts CAM packets into normal sending queue but puts PSM packets into sending queue with high priority. In the escalated contention, we also compare the scenarios between with and without virtual AP technology.

Contention's impact on power consumption. Figure 4 (a) shows that the poor's power consumption linearly increases with the rich's number increasing from 0 to 3. There are three schemes being measured. For virtual AP 1:1:1:1, we divide a physic AP into four virtual APs (VAP_1, VAP_2, VAP_3, VAP_4). The virtual APs broadcast beacon messages at stagger times with the gap of 25ms. Each PSM client associates to a different virtual AP. The virtual AP technology can alleviate the contention to some extent. Every client wakes up at stagger time, so the contention is not as serious as the basic high priority PSM scheme. When the number of rich clients is less than 3, the poor client can complete data transmission in a short period. Since no client associates to VAP_4, the poor client experiences light contention. However, when there are three rich clients, the contention in VAP_1's beacon period is exacerbated due to the involvement of data transmission from VAP_4.

For virtual AP 4:0:0:0, we still divide a physic AP into 4 virtual APs as above. But all PSM clients associate to the same virtual AP. The heavy contention causes the poor client waste energy for receiving packets. As shown in Figure 4 (a), under virtual AP 4:0:0:0, the poor client's energy consumption is quite similar to that under high priority PSM. Thus, the virtual AP technology cannot handle the PSM traffic contention well when all PSM clients associate to a single AP. Unfortunately, it is quite common in real environments that multiple PSM clients associates to either a physical AP or a virtual one.

Contention's impact on latency. Besides power consumption, the contention also affects the poor client's network performance. Figure 4 (b) shows that the average latency of packet delivery increases with more PSM clients involved. We



(a) (b) (c)

Fig. 4: Power consumption of the poor client (a), average latency of the poor client (b), and retries per packet of the poor client (c).

expect to see that the average latency of virtual AP 1:1:1:1 is much lower than that of the high priority PSM. However, virtual AP technology's improvement on average latency is not as much as that of power consumption. The reason is that packets are buffered for some time at AP, which forms the major part of average latency.

Contention's impact on retransmission. Figure 4 (c) shows average packet retransmissions of the three schemes, from which we can see that the average retransmission increases with the contention is intensified. Under the heavy contention, the link will get to be saturated and the probability of packet loss will increase. Both the clients and AP could suffer packet losses under the contention. Consequently, they have to retransmit the dropped packets, which will impose extra workload on the system and incur more packet losses later on.

In summary, a poor client suffers high energy consumption and performance degradation under PSM traffic contention. Although PSM traffic contention is not the only factor impacting the poor client's performance, the degradation of performance is evident when traffic contention is intensified. So, we focus on handling traffic contention to improve the poor client's performance. As power is the client's private resource and link bandwidth is the public resource of the wireless system, it is unfair that the poor spends plenty of private resource for using a small portion of public resource. A client should pay proportional private resource for public resource it consumes. In the next section, we present the design of HPSM and explain how it addresses this unfairness problem under the PSM traffic contention.

V. DESIGN AND IMPLEMENTATION

We propose a new power save solution HPSM to help poor clients save power when they face the contention from rich clients. HPSM's basic idea is introduced in this section. We also implement prototypes of HPSM based on a real system and a simulation system.

A. HPSM Design

We schedule the downlink packets to favor the poor PSM clients. Although the clients are identified as poor mainly based on the public resource they consume, different strategies of scheduling packets will have different impacts on the system's performance and energy consumption. In the following, we first assume all PSM clients are poor and explain the strategy chosen for scheduling packets. Then, we refine the classification of poor clients based on the claimed link resource and the packet schedule strategy.

Serve the clients associating to the AP. In the preliminary experiments, we observed that a PSM client contends with the others for taking over the channel. Such a contention will make the clients wait at high power mode. If a client can receive packets from AP without interruption, no energy will be wasted because of the contention. So, uninterrupted delivery is the preferable choice. Furthermore, the sum of all clients' energy consumption is minimal when AP uninterruptedly serves the clients that are sorted in ascending order of time for receiving their buffered packets. For example, the service time needed by the four clients is $T_{C_1} = 15ms$, $T_{C_2} = 12ms$, $T_{C_3} = 20ms$ and $T_{C_4} = 8ms$, AP should serve the clients in the order of C_4, C_2, C_1 and C_3. However, if AP does not follow the shortest-job-first (SJF) order, e.g., the order is C_4, C_1, C_2 and C_3, the awake time of C_2 and C_1 in this case is longer than that in SJF. In other words, the energy consumption of the latter case is larger. It has been proved in [31].

The schedule also benefits adaptive PSM clients. When the buffered packets of an adaptive PSM client does not exceed a threshold, the client will work under the static PSM mode. In such a case, we consider PS_Poll packet in the transmission time. When the number of buffered packets exceeds the threshold, the client will not send PS_Poll to AP for receiving the next packet. Instead, AP will continuously send packets to the clients. Therefore, the transmission time is approximately the total length of buffered packets divided by the bitrate. We assume that a static PSM client has x_1 buffered packets with lengths $L_1, L_2, \ldots, L_{x_1}$ and an adaptive PSM client has x_2 packets with lengths $L'_1, L'_2, \ldots, L'_{x_2}$. The length of PS_Poll is L_{poll} and the data rate of link is B. AP sends the packet right after it receives the PS_POLL message. For the static PSM client, it spends $\frac{L_i + L_{poll}}{B} + \delta$ to receive one packet and request for the next one, where

δ represents the interval between receiving packet L_i and sending PS_POLL for the next packet L_{i+1}. δ varies with the hardware and software of the client. The approximate time which the static PSM client spend to receive all packets is $T_1 = (\frac{L_1+L_{poll}}{B}+\delta)+(\frac{L_2+L_{poll}}{B}+\delta)+\cdots+\frac{L_{x_1}}{B}$ and that spent by the adaptive PSM client is $T_2 = \frac{L'_1+L'_2+\cdots+L'_{x_2}}{B}$.

AP could serve two types of clients (the PSM clients and CAM clients) in a beacon period. We divided all clients into two groups. The first group only consists of poor PSM clients, and the second group contains rich PSM clients and CAM clients. AP uninterruptedly serves the poor PSM clients in order at first. Since the rich PSM clients in the second group will stay awake for the whole beacon period, AP serves them and CAM clients with the same priority. It means that AP serves the clients in the second group based on the packet's time stamp. Because AP can send packets to PSM clients via either high priority queue or normal queue, scheduling packets based on their time stamp can well balance power saving and fairness for PSM clients and CAM clients [27].

If AP first serves all the PSM clients in order and then the CAM clients (i.e., following the SJF order), the total energy consumption should be the minimal. However, this SJF schedule strategy cannot be applied directly in a real system because those clients, which need a long period for data transmission, could suffer starvation under the schedule. In other words, these clients could have a little chance to start packet receiving. The waiting time could be intolerable for some applications, and make the PSM clients waste much energy without receiving any packets. So, we introduce a delay constraint to mitigate the problem caused by packet scheduling.

Refine the classification of poor clients. If a PSM client fulfills the following two requirements, we classify it as poor. First, a poor PSM client should finish receiving packets in a beacon period. The logic behind this requirement is intuitive. If a client cannot finish receiving packets in a beacon period, it implies that the client has many packets to receive. Thus, the client will consume more link bandwidth and stay awake in that period. Such a client should not be classified as poor. Second, the delay incurred by the uninterrupted service should be tolerable for other clients. We explain this requirement by introducing the notation of Θ, which is the factor constraining the delay. In our case, it is calculated by $\Theta = T_{beacon} - T_{wait}$, where T_{wait} is the longest time a client can wait. Each client has a specific T_{wait}. The minimal one works for all the clients in the system.

As shown in Figure 5, we assume that AP has served m poor PSM clients in time T_p. The clients $m+1, m+2, \ldots, m+n$ have sent the requests and are waiting for receiving packets. The time of serving client $m+1$ is $T_{C_{m+1}}$. The time of sending one packet to the remainder clients is $T_w = t_{c_{m+2}} + t_{c_{m+3}} + \cdots + t_{c_{m+n}}$, where t_{c_i} represents the date transmission time during which client C_i receives the next packet from AP. If $T_p + T_{C_{m+1}} + T_w < T_{beacon} - \Theta$, client C_{m+1} fulfills the second requirement. Then it is identified as a poor PSM client and put in group one. Θ represents the constraint on delay in the system. It is set conservatively so that the clients can receive packets before the waiting time exceeds T_{wait}. Overall, a big number of Θ means that the system cannot tolerate long delay. When $\Theta = 0$, there is no constraint on delay in a beacon period and AP can identify the most clients as poor. When $\Theta = T_{beacon}$, no client will be identified as poor. And AP will serve all clients following the 802.11 standards. We have more analysis and discussion of Θ in Section VI-B based on the evaluation results.

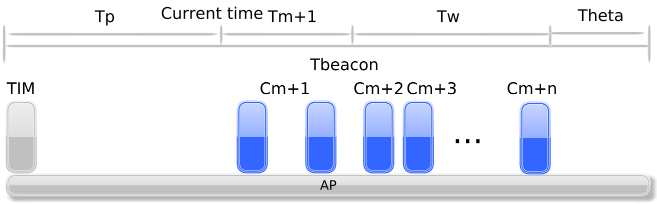


Fig. 5: AP identifies the poor client based on the time stamp in a beacon period.

We identify the poor PSM clients based on the buffered packets. In real cases, there could be unpredictable uplink and downlink packets when AP is serving a client. These packets will contend for the link resource. After AP serves the current poor client, the time left could not be long enough to serve the other clients. Therefore, AP needs to check whether the next poor PSM client in the group is still qualified for the uninterrupted service. In our solution, AP dynamically checks whether the poor client can still be classified as poor before AP starts to serve it.

Algorithm 1 shows how AP classifies the poor clients and serves them. Function `sort` puts the PSM clients in ascending order based on their service times. Function `getRemainderTime` returns the time left in the beacon period. When AP is going to serve a poor client, it holds the requests from the others. We create a scheduler to handle these requests, which are buffered in a pending queue. After a poor client is served, AP will check whether the next poor client has sent a request before. If the request is in the pending queue, AP starts to serve it immediately. Otherwise, AP will wait for the request of this client. During this period, the link is reserved for the poor client. It can start to receive packets in a short time. So, the reservation offers the largest benefit to the poor. AP should not handle any other client's request during this period. If AP tries to do so, the contention problem arises immediately because of the continuous communication between AP and the poor client. The whole process can be summarized as follows:

1) AP calculates the service time for each PSM client.
2) The PSM clients are sorted in ascending order of service time.
3) AP browses and identifies the poor PSM clients.
4) AP serves the poor PSM clients following the sorted order.
5) AP dynamically updates the poor client group after the current poor client is served.
6) The jobs in the delay queue are processed after all the

Algorithm 1 AP identifies the poor PSM clients

```
Inputs: PSM clients $(C_1, T_{C_1}), (C_2, T_{C_2}), \ldots, (C_m, T_{C_m})$
    CAM clients $(C_{m+1}, T_{C_{m+1}}), (C_{m+2}, T_{C_{m+2}}), \ldots, (C_{m+n}, T_{C_{m+n}})$

#At the beginning of beacon period
sort(PSM clients);
for each $i$ in {1 to m} do
    $T_p += T_{C_i}$;
    $T_w = t_{c_{i+1}} + t_{c_{i+2}} + \cdots + t_{c_m} + t_{c_{m+1}} + t_{c_{m+2}} + \cdots + t_{c_{m+n}}$;
    if ($T_p + T_w \le T_{beacon} - \Theta$) then
        putInPoorGroup(client $i$);
    else
        break;
    end if
end for

#AP serves a poor client
getPoorClient();
if PS_Poll is from the poor client then
    transmitPacket();
else
    enqueueJobInDelayQueue();
end if

#AP completes to serve a poor client
$T_r$=getRemainderTime();
$T_p = T_{C_i}$;
update($T_w$);
if ($T_p + T_w \le T_r - \Theta$) then
    serve($C_i$);
else
    emptyPoorGroup();
end if

#AP completes to serve all poor clients
processDelayJobs();
```

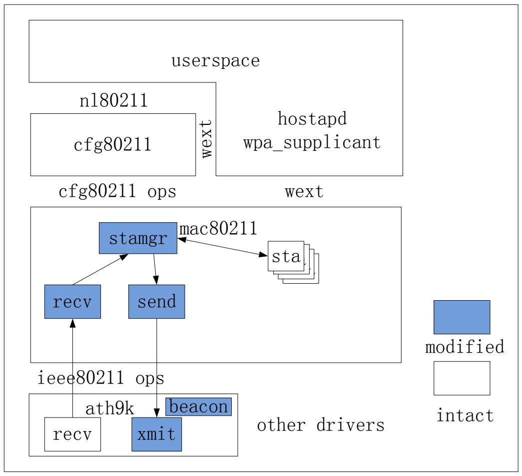


Fig. 6: Implementation in mac802.11.

poor clients are served.

B. Implementation

We implement the prototypes of HPSM based on Mad-wifi [1] in a real environment and NS-2 with power saving extension [2] in a simulation environment, respectively. Mad-wifi provides stable performance and good hardware compatibility. The system parameters are described in Section IV, and newly introduced parameters are described in Section VI. We install the latest driver ath9k on openSUSE. Since it has multiple sending queues with different priorities, we can easily isolate PSM traffic from CAM traffic. Figure 6 shows the main architecture of Mad-wifi and our modifications to the components. The 802.11 in NS-2 has the similar architecture. We implement the prototype on NS-2 mainly for validating the efficacy of HPSM with more PSM clients. We made changes in the drivers' (mac80211 and ath9k) source code. As shown in Figure 6, we created a scheduler inside mac802.11. At the beginning of a beacon period, we classify the clients based on their buffered packets. When AP gets a request from a "sta", we will process the request if it is from a poor client under service. Otherwise, the request will be put in a pending queue. After the poor client is served, AP checks whether the next client is still waiting for the uninterrupted service. When all poor clients are served, AP then processes the pending tasks. The functions are called when the corresponding events are triggered in drivers. The major functions are implemented in "sta" (station management), "rx" (receiving), and "tx" (sending) in mac802.11, as well as "xmit" (transmit) and "beacon" in ath9k.

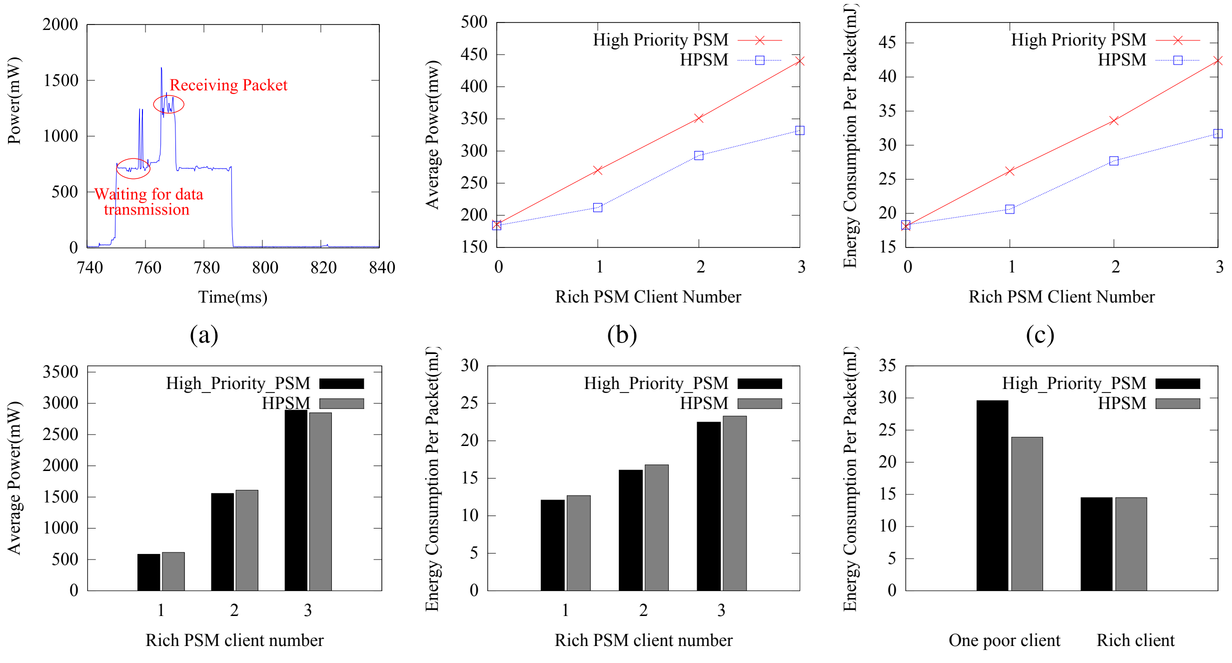
VI. EVALUATION

In this section, we first evaluate how HPSM impacts the power consumption and network performance in a real testbed environment. In particular, we compare HPSM to the existing standards: normal PSM and high priority PSM. The configuration of our testbed is already described in Section IV-A, and we have four HTC smartphones running as PSM clients. Since HPSM aims to reduce the contention problem among PSM clients, we mainly evaluate the performance of PSM clients and no CAM traffic is introduced. Then, we conduct simulation experiments based on NS-2 with more clients involved.

A. Controlled Experiments in Real Testbed

In the controlled experiments, we set one client as the poor and three other clients as the rich. We send one packet every beacon period to the poor under bit rate of 44 Kbps, and we send five packets to each rich client every beacon period under bit rate of 220 Kbps. The number of the rich PSM clients varies from 0 to 3, representing the different PSM traffic contention levels.

1) Evaluation of Power Consumption: In Section IV-B, we can see that the poor has to wait for a long period of time (33.76ms in a beacon) to get AP's response under high priority PSM. However, HPSM can hold the rich's requests and reduce the contention. Thus, the poor client is able to send out PS_Poll and receive AP's response as early as possible. After the data transmission completes, the poor client switches to sleep status. Figure 7 (a) shows the poor client's power consumption measurement in a beacon period under HPSM. Even when there are contentions from other three rich PSM clients, the poor client just waits about 15ms for receiving packet in the beacon period. Clearly HPSM



(a) (b) (c)

(d) (e) (f)

Fig. 7: Power consumption in a beacon period (a), power consumption of the poor client (b), energy consumption per packet of the poor client (c), power consumption of the rich clients (d), energy consumption per packet of the rich clients (e), and energy consumption in a 2v1 system (f).

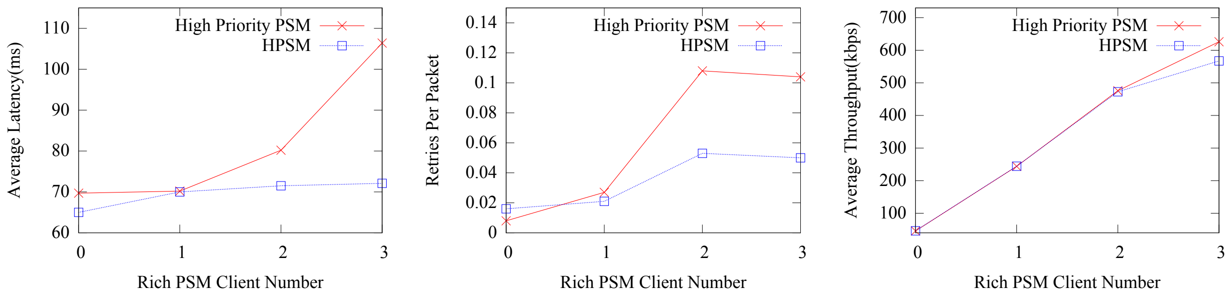
brings two benefits. The first is that the poor can save more power. The second is that traffic contention is reduced in the following beacon periods, since each rich client just needs to face the contention from the other rich clients.

With the number of rich PSM clients increases from 0 to 3, the poor client will encounter heavier and heavier traffic contention. Figure 7 (b) shows that the power consumption of the poor client under HPSM is obviously smaller than that under high priority PSM, and the difference between HPSM and high priority PSM is enlarged with the intensifying contention. The experimental result shows that HPSM assists the poor client to save up to 25% power when it faces the contention from three rich clients. In Figure 4 (a), the virtual AP technology brings higher energy conserving in case “Virtual AP 1:1:1:1" when the number of clients is less than the number of virtual APs. We want to stress that virtual AP cannot handle traffic contention when multiple clients associate to the same virtual AP. However, HPSM is capable of helping the poor client conserve energy in such a case.

We further investigate the power consumption from a more detailed perspective: the energy consumption for receiving a packet. Because there are packet losses during data transmission, energy consumption per packet is an accurate indicator of power efficiency of data transmission. For example, a client receives x packets during a period of t ms. The average power consumption during this period is p mW. Later on, it receives $y(x > y)$ packets during the same period of t ms. The average power consumption is still p mW. However, the client's energy consumption per packet of the former period $(\frac{p\cdot t}{1000\cdot x})$ is smaller than that $(\frac{p\cdot t}{1000\cdot y})$ of latter period, which means that the client's power efficiency in the former period is higher. Figure 7 (c) displays the poor client's energy consumption of receiving a packet. Similarly, compared with high priority PSM, HPSM is able to help the poor client improve its energy consumption per packet by 25%.

Besides the power consumption of a poor client, we also study the energy consumption of rich clients. Since there is only one power monitor, we do separate experiments to measure the power consumption for each rich client. In Figure 7 (d), the x-axis represents the number of rich clients in one set experiment; and y-axis shows the sum of rich clients' power consumption. The rich clients' power consumption under HPSM is quite similar to that under high priority PSM in the three cases. The reason is that all rich clients stay at active status in the whole beacon period. Although their tasks are put into the pending queue by HPSM, it brings negligible impacts on the power consumption of the rich clients. Figure 7 (e) shows that the energy consumption per packet of rich clients under HPSM increases up to 3% in comparison to that under high priority PSM. The reason is that HPSM reserves the network resource for the poor client, which decreases the number of packets received by the rich clients. We regard the increase of the energy consumption per packet as the cost that the rich clients pay for the more bandwidth they consumed.

In the following, we further investigate the scenario in which two poor clients and one rich client associate to a single AP. Figure 7 (f) shows the energy consumption per packet of one poor PSM, which is 23.95mJ under HPSM. It is larger than the energy consumption of the poor client (20.58mJ) when it faces the contention from only one rich



(a) (b) (c)

Fig. 8: Average latency of the poor client (a), retries per packet of the poor client (b), and throughput of the system (c).

client under HPSM. In the former case, the two poor clients need to share the benefits provided by the rich client; however in the latter case, the poor client takes all the benefits from the rich client.

Given a certain number of rich clients, when the number of poor clients increases, the gain of a poor client decreases because it has to share benefits with the other poor clients. On the other hand, although the increase of rich client population will intensify the contention problem, HPSM can effectively protect a poor client's performance from serious degradation by favoring the poor with higher scheduling priority over the rich, under this intensified contention scenario.

2) Evaluation of Network Performances: The PSM traffic contention causes the poor client to spend extra time to win the chance of receiving a packet under high priority PSM. Thus, the poor client's average latency in packet delivery will increase with the contention getting more serious. However, HPSM allows a poor client's packet to be sent ahead of the rich clients' packets, and hence the average latency of poor client under HPSM can sustain at a low level. Figure 8 (a) shows the average latency of the poor client: under high priority PSM, it significantly increases when the traffic contention is intensified; but it does not change much under HPSM. Compared with high priority PSM, HPSM shortens the latency by about 40% when there are three rich clients.

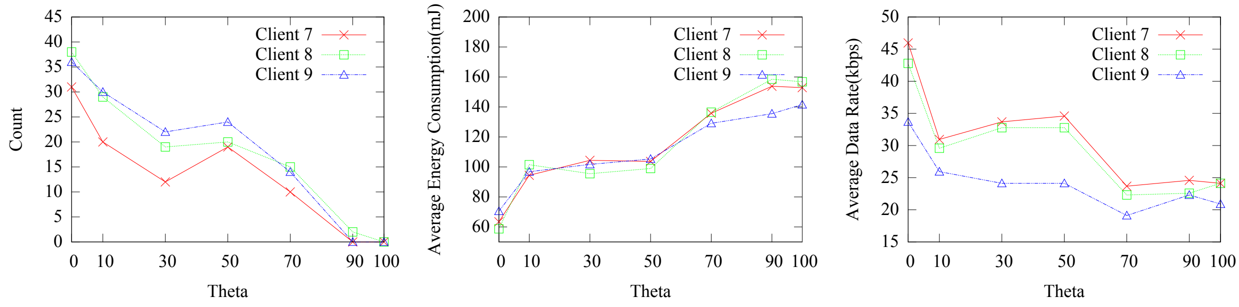
With more and more rich clients involved in the data transmission, the contention problem becomes intensified. In such a circumstance, the link will get to be saturated, which increases the probability of packet loss. Both the clients and AP will suffer packet loss under the contention. Consequently, they need to retransmit the dropped packets. The unwanted retransmission will impose extra workload on the system and incur more packet losses later on. Figure 8 (b) shows the poor client's retries per packet increases with the traffic contention getting heavier under high priority PSM. HPSM reserves the channel to serve the poor client, hence reduces the contention problem during its data transmission. In comparison with the high priority PSM, HPSM can help the poor client reduce retries per packet by up to 50% when there are three rich clients.

Although HPSM can assist the poor client to save more power and gain much higher performance, it inevitably affects the performance of rich PSM clients. Besides the impact from HPSM, the contention also impacts the average latency and retransmission of the rich clients. For a specific rich client, its network performance cannot be repeatedly determined given the same experimental setup, because it will contend with the other rich clients. Therefore, we regard all rich clients as a group and observe the HPSM's impacts on their network performance. As HPSM reserves network resource for the poor client, the consequence of reservation is that the rich's packets cannot be sent out during the period of reservation. So, the whole system's throughput degrades while the poor's performance is improved. In Figure 8 (c), the experimental results show that the system's throughput decreases up to 9.4% comparing to high priority PSM when 4 clients associate to the AP. However, in a long run, the reservation won't happen in every beacon period. The throughput degradation is much less than the worst case. When AP sends one packet to the poor client every two beacon periods, HPSM helps the poor client save up to 20% energy while the throughput only decreases by 3.6%.

B. Experiments in Simulation

We further evaluate HPSM with more clients by conducting two sets of simulation experiments. In the first set, we mainly measure the Θ's impact on the performance of clients under HPSM. In the second set, we evaluate HPSM by involving the data flows of different packet sizes, real traffics, and the scenario in which only poor clients exist.

1) Impacts of Θ: The experiments in our real testbed show that HPSM brings significant benefits to the poor clients. We observe the similar results in our simulation experiments. In the following, we mainly evaluate how the factor Θ impacts a poor client's performance. Θ reflects the client's requirement for timely data transmission in a beacon period. The maximal constraint can fulfill all clients' requirement for receiving packets in a timely manner. However, in some cases users may prefer power saving to network performance, while network performance could be more important in some other cases. We vary Θ to observe its impacts upon the system's performance. When $\Theta = 0$, the system has no constraint on scheduling packets. In such a case, a client consuming few bandwidth has the highest probability to be classified as poor. On the other hand, $\Theta = 100$ means that any packet switch will violate the



(a) (b) (c)

Fig. 9: The counts identified as the poor (a), energy consumption per packet (b), and the counts identified as the poor (c).

constraint and all packets are transmitted following the normal standards.

We set up the simulation environment with 10 nodes, one of which is the AP. Two CAM clients (C_1, C_2) and seven PSM clients (C_3, C_4, . . ., C_9) associate to the AP. We establish CBR UDP flows from C_1 to the other eight clients. The link rate is set to 2Mbps. The default packet size is 512 Bytes and the traffic information of the nine clients is listed in Table II. According to our measurement in Section IV, we set the power of idle status and the power of receiving packet to 653.5 mW and 1039.5 mW, respectively. We run the experiments for about 90 beacon periods.

TABLE II: System parameter setup

Client	C_1	C_2	C_3, C_4 C_5, C_6	C_7, C_8 C_9
Flow rate (Kbps)	1344	128	256	64
Flow direction	outgoing	incoming	incoming	incoming

In this experiment, C_3, C_4, C_5, and C_6 could be classified as poor, which happens at the beginning of data transmission. This is because, during that period, the link is not saturated and there are just few packets of these clients buffered at AP. However, C_7, C_8, and C_9 consume the fewest bandwidth among the eight clients that have downlink traffic. They are the potential candidates that HPSM should help. Thus, we measure how Θ impacts the performance of these three clients.

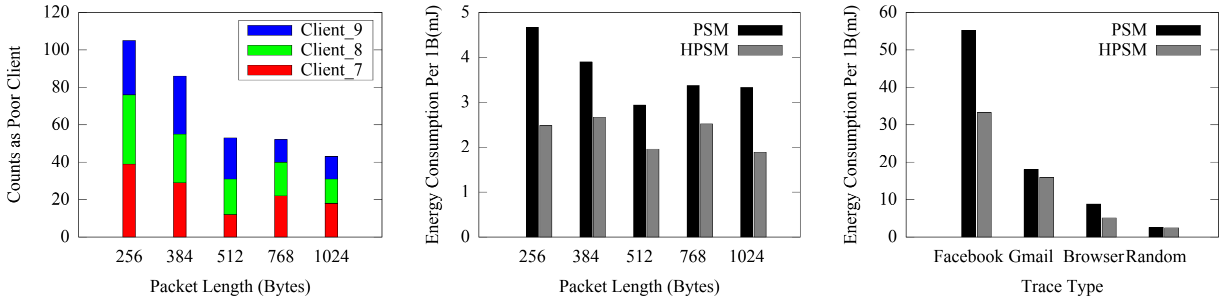
The y-axis in Figure 9 (a) shows how many times a client is classified as poor. For all the three clients, the number of times being classified as poor decreases while Θ increases from 0 to 100. The high constraint causes the clients to be classified as poor with low chance. When some clients cannot wait for a long time to receive packets, they will request high constraint. Consequently, AP has little time to serve the poor clients, and thus for a client, the chance of being classified as poor is low.

When a client is classified as poor, it can finish packet receiving in a beacon period with high priority. In other words, a poor client can save power in the beacon period. Figure 9 (b) shows the average energy consumption per packet of the three clients. When the constraint is low, the clients have high chance to be classified as poor. Thus, they can save more power and the energy consumption per packet is low. The energy consumption increases with the increase of the constraint.

When Θ decreases from 100 to 0, for clients C_7, C_8 and C_9, the counts of being identified as poor increase from 0 to 105 (270 is the maximal number), which means that the three clients are classified as poor for 105 times in total. Their energy consumption per packet decreases from 1.51 to 0.64. For clients C_3, C_4, C_5, and C_6, the counts of being identified as poor increase from 0 to 19, but their energy consumption per packet increases from 0.53 to 0.57. Although C_3, C_4, C_5, and C_6 are classified as poor for a few times, they are identified as rich in most times, during which AP serves C_7, C_8 and C_9 with high priority. Therefore, their energy consumption per packet increases slightly with that of the poor decreases significantly. Overall, the clients can save power in the beacon periods if they are classified as the poor. In a long term, those clients that are identified as poor for more times than the others will save power. Meanwhile, the delay of the other clients is bounded by Θ. We regard the degradation of the rich clients' performance as the tax paid for using more link resources.

The schedule strategy not only helps the poor clients save power but also improves their data rates of receiving packets. According to Table II, the overall data rate (incoming + outgoing) excesses the link's data rate. In such a case, AP cannot transmit all the packets sent from C_1 in a beacon period. The packets will be buffered in the queue before they are scheduled for transmission. When AP applies the standard schedule strategy, the poor clients have to spend a long time to receive all packets. Differently, HPSM allows the poor clients to receive packets with high priority. They can finish receiving packets in a shorter period. Thus, HPSM improves the downlink data rate for the poor clients. The more times a client is classified as poor, the higher data rate it can achieve. In Figure 9 (c), the downlink data rates of the three clients are high when they can be classified as poor for many times. Their data rates decrease while Θ increases.

When Θ increases, the clients have little chance of being identified as the poor. The direct consequence is that the clients' energy efficiency will decrease. Their data rates will also be affected, because they will face the intensified contention from other clients. However, the count of being identified as the poor is not the only factor that can impact the data rate. The other factors like channel condition can



(a) (b) (c)

Fig. 10: The counts identified as the poor (a), energy consumption per Byte of the clients (b), and energy consumption in trace driven experiment.

also impact the data rate. So, the data rate in Figure 9 (c) may not strictly follow the curve in Figure 9 (a). Besides, in Figure 9 (a), the individual performance varies because each of them encounters contention to different extents. Overall, we can observe that the total counts and data rate decrease as Θ increases in Figures 9 (a) and 9 (c).

We also observe that all the three clients are classified as poor in some beacon periods, but as rich in some other beacon periods. At the beginning of the experiments, the link is not saturated, and AP can transmit packets before the buffer queue is fully filled. The clients C_3, C_4, C_5, and C_6 could be classified as poor at this period. As the experiment goes on, the link gets to be saturated, which impacts both the uplink and downlink traffics. The interference and contention make AP spend longer time to serve a poor client. Then, the rest of poor clients could be re-classified as rich if the time is not long enough for serving them uninterruptedly.

In an ideal case, Θ should well balance the power saving of the poor and the network performance of the rich when it equals the maximal constraint required by all the clients. However, Θ can be set to a smaller value to satisfy the poor clients, or set to a larger value to guarantee the rich's performance. Note that in any case, it should not be too large, say 50 (half of the beacon period based on our experiments); otherwise, HPSM would bring negligible benefits to the poor.

2) Extended Experiments: With the fixed packet size, it is easy to demonstrate how much benefit or negative impact brought by HPSM in the experiments above. But scheduling packets with different sizes could result in a different impact to some extent. For example, when the other parameters remain unchanged, transmitting a large packet will incur more fairness loss comparing to that of transmitting a small packet. In either case, scheduling a client's packets with higher priority will benefit its performance and negatively impact the performance of the other clients. Our algorithm classifies a client by the data transmission time, which depends on both the number and size of its packets. Thus, it can work for packets with different lengths. In the following, we extend our evaluation for more application scenarios, which include the data flows of different packet sizes, real traffics, and the case where only poor clients exist.

First, we use the same experimental setup of Section VI-B1 except varying the packet size from 256 Bytes to 1024 Bytes for all clients. Based on our discussion in Section VI-B1, HPSM is able to bring considerable benefits to the poor by setting $\Theta \leq 50$. We measure how HPSM impacts the performance of clients C_7, C_8 and C_9 by setting $\Theta = 30$. Figure 10 (a) shows how many times the three clients are classified as the poor. The results show that HPSM can identify the poor clients when they receive packets in different sizes. When packet size increases, the number of packets will decrease because the bitrate remains the same. In the CBR flow, fewer large packets will be sent under light traffic at the beginning period of the experiment. Therefore, for the packet sizes of 512, 768 and 1024, the counts of being identified as poor are smaller than the counts of the other two packet sizes. We also measure the energy efficiency of the three clients under PSM and HPSM. Figure 10 (b) shows how much energy is consumed for receiving one byte data. We can see that HPSM improves the energy efficiency of the poor clients comparing to PSM, no matter what size a packet has.

Second, to evaluate the HPSM's performance based on the data flows in which packets have different sizes, we generate traffic traces and conduct the trace driven experiments. We introduce one CAM client, four PSM clients, and the target client that receives the trace data, and set $\Theta = 50$. The CAM client, the first three PSM clients, and the fourth PSM client receive CBR flows at the bitrates of 128 Kbps, 256 Kbps, and 64 Kbps, respectively. We generate four types of traces: the first date trace "Random" contains the data flows under the bitrate of 64 Kbps in which the packets sizes are set randomly. Then, in a WiFi system under 2 Mbps bandwidth, we use tcpdump [3] to capture the other three traces generated by browsing web pages, using Facebook and Gmail apps on a smartphone. The details of the four traces are listed in Table III. The items in the row of "Packet size (Bytes)" represent the minimum and maximum packet sizes in the data flows. For real traffics, packets are not generated constantly because of the user interaction with the apps. Consequently, the average flow rate of the real traces is not as high as that of the CBR data flow. Figure 10 (c) shows the energy consumption under PSM and HPSM when we replay these traces. The target client consumes the lowest bandwidth comparing to the others, therefore, HPSM will serve it as the poor and

TABLE III: Trace details

Trace Name	Random PktLen	Browse	Facebook	Gmail
Flow rate (Kbps)	64	17.73	7.24	3.15
Packets size (Bytes)	[1:999]	[54:1514]	[66:1464]	[66:1484]
Trace length (beacons)	90	600	800	500

improve its energy efficiency. For the trace "Random", the energy consumption per byte of the client is 2.45 mJ under HPSM, which is slightly smaller than that (2.53 mJ) under PSM. We note that the energy consumptions per byte of the other three traces are higher than that of "Random". This is because the lengths of the three traces are much longer than that of "Random", and a longer trace has more idle periods in which energy is wasted. Overall, we see that HPSM improves the energy efficiency of a client in comparison to PSM.

At last, we consider the scenario in which only poor clients exist. When the link is not saturated, it is possible that all clients are classified as the poor. A client can be classified as the poor if it satisfies the two requirements mentioned before. In this case, the clients are classified as the poor once they satisfy the requirement of Θ. As we have discussed in Section V, when all clients are classified as the poor, AP will serve them in sequence and the overall energy consumption should be smaller than that under the original schedule. We introduce three poor clients, which have the downlink CBR under the bitrate of 64 Kbps, in the experiments. As expected, the overall energy consumption under PSM is 63.66 J and that under HPSM is 60.01 J. Since the link is not saturated, the contention is not as serious as that under a saturated link. Therefore, the benefit brought by HPSM cannot be as high as that under the contention from the rich.

VII. DISCUSSION

To provide a comprehensive view of this work, we discuss HPSM's limitations and potential improvements in this section.

When AP is serving a client, there could be unpredictable uplink and downlink traffics. We dynamically check whether the remaining time is long enough for serving the next poor client. If the downlink traffic goes to the client that AP is serving, AP will serve the client till all packets are transmitted uninterruptedly. Then, other clients have to wait for a longer period. There is a potential solution which could mitigate the problem caused by the unpredictable traffic. We limit the time period, in which AP serves a poor client, to be the sum of service time of the current poor client and the next one in the queue. For example, assume AP is going to serve the poor client C_i, and AP should spend $T_{C_i} + T_{C_{i+1}}$ to serve C_i and C_{i+1}; but when there is burst traffic going to C_i, AP can spend at most $T_{C_i} + T_{C_{i+1}}$ to serve C_i. Once the client runs out of the service time, AP stops to serve the client and switches to serve the next one.

HPSM handles the general unfair energy consumption under the contention. In HPSM, the clients consuming a small proportion of link resources have higher priority for receiving packets than those clients that consume a large proportion of link resources. In real environments, there could be different definitions and requirements on the priority. In fact, HPSM can work with other solutions in wireless networks for mixed applications. For example, if a session of a rich client has higher priority for receiving packets, AP can deliver packets to it first, and then continue to serve the poor clients.

Although we set Θ manually to observe its impacts upon the performance of the system in this work, it is possible to adjust the value of Θ automatically. One approach is to check the data flows' protocols, since different protocols have different requirements on the delay constraint. For example, we can identify a flow by observing the packet's port, and then calculate the constraint (the longest time it can wait). After all flows are handled, we will choose the biggest one as Θ.

In a system containing few poor clients but many rich clients, HPSM can help the poor gain significant benefits. With the number of the poor clients increases, the benefits decreases unavoidably because the poor clients have to share the benefits. In our solution, we sort the poor PSM clients based on the service time they need. Therefore, the poor clients share the benefits by following the rule that a client pays more if it consumes more public resource. In addition, a client could be classified as the rich in some beacon periods, but as the poor in some other beacon periods. In a long run, any PSM client could benefit from HPSM.

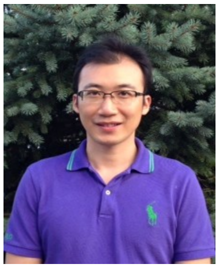
VIII. CONCLUSION

The PSM traffic contention has negative impact on PSM's performance. It causes a PSM client consuming little link resource suffer from lower energy efficiency than a PSM client consuming much link resource. In this paper, we introduce an energy conserving model for tackling PSM traffic contention problem. We show that optimizing energy conserving for multiple PSM clients is an NP-Complete problem. Then, we propose a new mechanism called HPSM to appropriately handle PSM traffic contention and address the unfairness problem among multiple PSM clients. We define the PSM client's battery as private resource and the link resource as public resource. HPSM classifies a client as poor or rich based on the link resource it consumes, and then endows the poor clients with higher priority over the rich clients on receiving packets. To validate the efficacy of HPSM, we develop a prototype of HPSM and conduct a series of experiments. Our experimental results show that HPSM helps the poor client save about 25% power when it faces the contention from the rich clients. In addition, its network performances like latency and packet re-transmission rate are also significantly improved.

REFERENCES

[1] http://madwifi-project.org/.

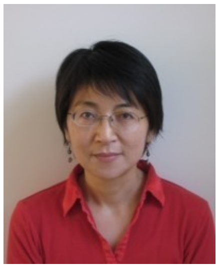
[2] http://sourceforge.net/projects/nspme/files/.
[3] http://www.androidtcpdump.com/.
[4] http://www.msoon.com/labequipment/powermonitor/.
[5] M. Anand, E. Nightingale, and J. Flinn. Self-tuning wireless network power management. ACM MobiCom 2003.
[6] G. Ananthanarayanan and I. Stoica. Blue-fi: enhancing wi-fi performance using bluetooth signals. ACM MobiSys 2009.
[7] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani. Energy consumption in mobile phones: a measurement study and implications for network applications. ACM IMC 2009.
[8] Xavier Pérez Costa and Daniel Camps-Mur. Au-apsd: Adaptive ieee 802.11e unscheduled automatic power save delivery. IEEE ICC 2006.
[9] L.Benini D.Bertozzi and B.Ricco. Power aware network interface management for streaming multimedia. IEEE WCNC 2002.
[10] F. Dogar, P. Steenkiste, and K. Papagiannaki. Catnap: exploiting high bandwidth wireless interfaces to save energy for mobile devices. ACM MobiSys 2010.
[11] G. Fisher. Remembering mollie orshansky-the developer of the poverty thresholds. *Social Security Bulletin*, 2008.
[12] H. Han, Y. Liu, G. Shen, Y. Zhang, Q. Li, and C. Tan. Design, realization, and evaluation of dozyap for power-efficient wi-fi tethering. *IEEE/ACM Trans. Netw.*, 22(5):1672–1685, October 2014.
[13] Y. He and R. Yuan. A novel scheduled power saving mechanism for 802.11 wireless lans. *IEEE Transactions on Mobile Computing*, pages 1368–1383, 2009.
[14] 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. *IEEE Communications Letters*, 13(3):164–166, 2009.
[15] Y. He, R. Yuan, X. Ma, J. Li, and C. Wang. Scheduled psm for minimizing energy in wireless lans. IEEE ICNP 2007.
[16] M. Khan, V. Dave, Y. Chen, O. Jensen, L. Qiu, A. Bhartia, and S. Rallapalli. Model-driven energy-aware rate adaptation. ACM MobiHoc 2013.
[17] R. Krashinsky and H. Balakrishnan. Minimizing energy for wireless web access with bounded slowdown. ACM MobiCom 2002.
[18] J. Lee, C. Rosenberg, and E. Chong. Energy efficient schedulers in wireless networks: design and optimization. *Mob. Netw. Appl.*, 11(3):377–389, June 2006.
[19] X. Lei and S. Rhee. Enhancement of the ieee 802.11 power saving mode by prioritized reservations. *Int. J. Distrib. Sen. Netw.*, pages 9:9–9:9, January 2015.
[20] H. Lin, S. Huang, and R. Jan. A power-saving scheduling for infrastructure-mode 802.11 wireless lans. *Comput. Commun.*, 29(17):3483–3492, November, 2006.
[21] J. Liu and L. Zhong. Micro power management of active 802.11 interfaces. ACM MobiSys 2008.
[22] F. Lu, G. Voelker, and A. Snoeren. Slomo: Downclocking wifi communication. USENIX NSDI, 2013.
[23] J. Manweiler and R. Roy Choudhury. Avoiding the rush hours: Wifi energy management via traffic isolation. ACM MobiSys 2011.
[24] S. Nedevschi, L. Popa, G. Iannaccone, S. Ratnasamy, and D. Wetherall. Reducing network energy consumption via sleeping and rate-adaptation. USENIX NSDI 2008.
[25] A. Pyles, X. Qi, G. Zhou, M. Keally, and X. Liu. Sapsm: Smart adaptive 802.11 psm for smartphones. ACM UbiComp 2012.
[26] D. Qiao and K. Shin. Smart power-saving mode for ieee 802.11 wireless lans. In *IEEE INFOCOM 2005*.
[27] E. Rozner, V. Navda, R. Ramjee, and S. Rayanchu. Napman: network-assisted power management for wifi devices. ACM MobiSys 2010.
[28] E. Tan, L. Guo, S. Chen, and X. Zhang. Psm-throttling: Minimizing energy consumption for bulk data communications in wlans. IEEE ICNP 2007.
[29] E.J. Vergara and S. Nadjm-Tehrani. Watts2share: Energy-aware traffic consolidation. IEEE GreenCom 2013.
[30] Y. Xie, X. Luo, and R. Chang. Centralized psm: an ap-centric power saving mode for 802.11 infrastructure networks. IEEE SARNOFF 2009.
[31] Z. Zeng, Y. Gao, and P.R. Kumar. Sofa: A sleep-optimal fair-attention scheduler for the power-saving mode of wlans. IEEE ICDCS 2011.
[32] C. Zhang, X. Zhang, and R. Chandra. Energy efficient wifi display. ACM MobiSys 2015.
[33] X. Zhang and K. Shin. E-mili: energy-minimizing idle listening in wireless networks. ACM MobiCom 2011.



Dachuan Liu received his Bachelor and Master in Computer Science from Sichuan University at Chengdu, Sichuan, China. He is a Ph.D. candidate of Computer Science at the College of William and Mary, Williamsburg, VA, USA. His research interests include wireless communication, user authentication and privacy protection on smartphone.

Haining Wang received his Ph.D. in Computer Science and Engineering from the University of Michigan at Ann Arbor, MI, USA, in 2003. He is a Professor of Electrical and Computer Engineering at the University of Delaware, Newark, DE, USA. His research interests lie in the areas of security, networking system, and cloud computing. He is a senior member of IEEE.

Gang Zhou received his Ph.D. degree from the University of Virginia at Charlottesville, VA, USA, in 2007. He is an Associate Professor, and also Graduate Director, in the Computer Science Department at the College of William and Mary, Williamsburg, VA, USA. He has published over 80 academic papers in the areas of sensors & ubiquitous computing, mobile computing, body sensor networks, internet of things, and wireless networks. The total citations of his papers are more than 5000 according to Google Scholar, among which five of them have been transferred into patents and the MobiSys'04 paper has been cited more than 800 times. He is currently serving in the Journal Editorial Board of IEEE Internet of Things as well as Elsevier Computer Networks. He received an award for his outstanding service to the IEEE Instrumentation and Measurement Society in 2008. He also won the Best Paper Award of IEEE ICNP 2010. He received NSF CAREER Award in 2013. He received a 2015 Plumeri Award for Faculty Excellence. He is a Senior Member of ACM and also a Senior Member of IEEE.



Weizhen Mao is a Professor of Computer Science at the College of William and Mary, Williamsburg, VA, USA. Her research interest is in algorithms and complexity with application in wireless and sensor networks, sequential and parallel job scheduling, and any problems that involve resource allocation and optimization.

Boyang Li received his Bachelor from Shandong Normal University at Jinan, Shandong, China and Master from Miami University at Oxford, OH, USA. He is a Ph.D. candidate of Computer Science at the College of William and Mary, Williamsburg, VA, USA. His research interests are in software engineering, natural language processing, and source code analysis. He is a student member of IEEE.