Energy Efficient Wifi Tethering on a Smartphone

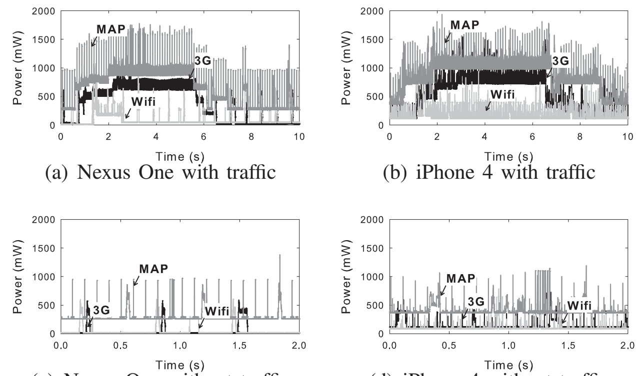
Kyoung-Hak Jung†, Yuepeng Qi†, Chansu Yu†‡, and Young-Joo Suh†
†Department of Computer Science and Engineering & Division of IT Convergence Engineering
Pohang Univ. of Science and Tech., Pohang, 790-784, Republic of Korea
Email: {yeopki81, yuepengqi, yjsuh}@postech.ac.kr
‡Department of Electrical and Computer Engineering
Cleveland State University, Cleveland, Ohio 44115
Email: c.yu91@csuohio.edu

***Abstract*—While numerous efforts have been made to save energy of "client" devices but it has not been addressed for access points (APs) as they are assumed to be supported by AC power. This paper proposes E-MAP, which is an energy saving algorithm for a tethering smartphone that plays a role of mobile AP (MAP) temporarily. It saves MAP's energy by introducing the sleep cycle as in power save mode (PSM) in 802.11 but successfully keeps clients from transmitting while it sleeps. One important design goal of E-MAP is backward compatibility, *i.e.*, it requires no modification on the client side and supports PSM and adaptive PSM (A-PSM) as well as normal constant awake mode (CAM) clients. Experiments show that E-MAP reduces the energy consumption of a Wifi tethering smartphone by up to 54% with a little impact on packet delay under various traffic patterns derived from real-life traces.**

I. Introduction

While the coverage of cellular networks is much larger than that of Wifi networks in the US (99% vs. 49%) [1], [2], 90% of customers prefer Wifi-only tablets over the 3G versions according to a market research [3]. Wifi-only devices including laptops, iPads, iPod touches, and e-books can still access Internet via *tethering* even though there is no Wifi access point (AP) in the neighborhood. They use a smartphone as a mobile AP (MAP) and utilize its 3G/LTE interface to access Internet. This feature is available on most of recent smartphones such as iPhone 4, 4S and 5 (iOS 4.2.5 or later) [4], Windows Phone 7 and 8 [5] and many Android phones [6]. There are also dedicated products such as Verizon's Jetpack mobile hotspot, which can connect up to five (ten) Wifi-enabled devices simultaneously in 3G (LTE) [7].

However, tethering could significantly shorten the battery lifetime of a smartphone. While client's energy consumption can be reduced by using *power save mode* (PSM) [8] of 802.11 and its variants such as *adaptive PSM* (A-PSM) [9], [10], AP's energy has largely been ignored because it is assumed to be supported by AC power. To better understand the power consumption of a MAP, Fig. 1 depicts the measurement results of two smartphones, HTC Nexus One (Android 2.3.7) and iPhone 4 (iOS 6.1.0) when they function as a MAP (both 3G and Wifi enabled), as a 3G client and as a Wifi client. It is observed in all four figures that the MAP consumes 260∼820mW more energy than the combination of the 3G client and the Wifi client. This is mainly because the MAP is not allowed to sleep.



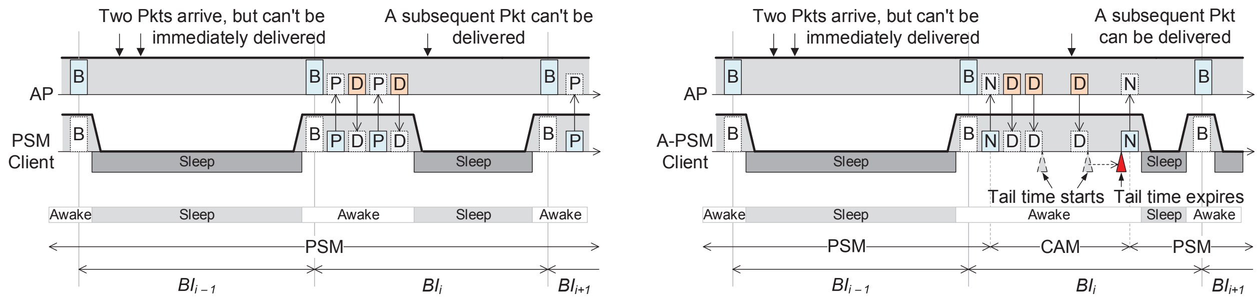
(a) Nexus One with traffic
(b) iPhone 4 with traffic
(c) Nexus One without traffic
(d) iPhone 4 without traffic

Fig. 1: Power consumption measurements using Monsoon Power Monitor [11]. (Each figure compares power consumption of a smartphone when it is used as a 3G client, a Wifi client, and a MAP with both Wifi and 3G enabled. Note that periodic spikes in the figures represent beacon transmission and reception over Wifi.)

This paper proposes an energy-efficient MAP mechanism (E-MAP) that conserves battery power of a MAP by turning off its Wifi interface when there is no traffic. Two important considerations are (i) it should not increase the packet delay substantially and (ii) it should not assume any firmware modifications or protocol addition on client devices, and should be compatible with PSM and A-PSM clients as well as normal constant awake mode (CAM) clients. For the former, packets that arrive while E-MAP sleeps would be delivered later, increasing the packet delay for downlink traffic. To minimize the impact, E-MAP adaptively determines the sleep start time as well as the sleep duration based on network activity and buffered packets. For the latter, the main difficulty comes from the fact that the MAP cannot sleep unless there is a power negotiation that all associated clients could not send uplink traffic. This has been approached in a recent work, called DozyAP [12], but may not be practically feasible because it requires modifications at both AP and clients.

Experiments based on our prototype implementation on HTC Nexus One smartphone running Android 2.3.6 with a variety of client devices (IBM Thinkpad X201 and Samsung Nexus S) show that E-MAP reduces the energy consumption

978-1-4799-3360-0/14/$31.00 ©2014 IEEE
978-14799-3360-0/14/$31.00 ©2014 IEEE



(a) PSM client (b) A-PSM client

Fig. 2: Power saving operations of PSM and A-PSM clients (B: beacon frame, P: PS-Poll frame, D: Data frame, N: Null frame).

of a Wifi tethering smartphone by up to 54% with a little impact on packet delay under various traffic patterns obtained from real-life traces [13]. DozyAP [12] saves more energy (3.6∼17.4%) than E-MAP mainly due to a longer sleep duration but is critically disadvantaged in terms of packet delay (as much as 12 times higher delay increase).

The contributions of this paper are four-fold: First, this paper proposes a mechanism to save AP's power (rather than client's power) without increasing the packet delay substantially. Second, the proposed E-MAP protocol demands no modification at the client side and is compatible with PSM and A-PSM as well as CAM clients. To the best of our knowledge, there is no such work in the literature. Third, this paper analyzes the packet delay in the context of E-MAP scenario. This will help further research in developing more efficient MAP algorithms. Fourth, this paper proposes to employ the AP virtualization technique for the implementation of the proposed E-MAP mechanism. This minimizes the modifications at the AP side.

The rest of this paper is as follows. Section II explains PSM and A-PSM. Section III describes the proposed E-MAP protocol followed by the delay analysis in Section IV. In Section V, we present our experiment environments and evaluation results. Section VI provides an overview of existing power save mechanisms. Finally we conclude this paper in Section VII.

II. Background

A. Energy Saving of Wifi Clients

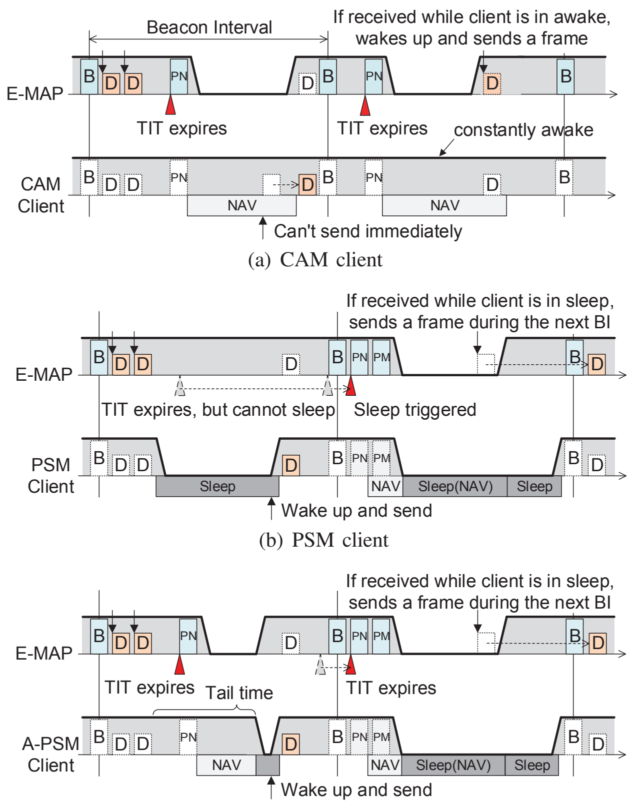
According to the IEEE 802.11 standard [8], a PSM client can conserve its power by switching its Wifi interface between a low power sleep state and an awake state, as shown in Fig. 2(a). In other words, a PSM client staying in the low power state wakes up periodically (*beacon interval* or BI) to receive a *beacon frame* to which all clients must synchronize. This beacon frame contains identifiers of PSM clients for which there are buffered packets in the AP. This is called *traffic indication map* (TIM). For example, two packets that arrived during $i-1^{th}$ beacon interval (BI_{i-1}), shown in Fig. 2(a), will be buffered at the AP and announced in the form of TIM in the beginning of BI_i. If a PSM client is not addressed in the TIM, it goes back to a low power sleep state to conserve power. Otherwise, it remains in an awake state and sends a PS-Poll frame to request the buffered packet(s). After receiving them all, the PSM client immediately goes back to a sleep state as in the figure. However, if a subsequent packet thereafter arrives at the AP, it should be buffered until the next BI.

Therefore, PSM saves energy by reducing the amount of time that the Wifi interface is in the awake state but it increases the packet delay. The maximum delay would be BI (*e.g.*, 100 msec) when the client device wakes up every BI. In the case of multimedia traffic such as VoIP, Skype and FaceTime, this could be intolerably long. For this reason, most recent smartphones and tablets adopt *adaptive PSM* (A-PSM) mechanisms [9], [10]. As shown in Fig. 2(b), a PSM client momentarily switches to CAM in order not to miss any immediately following frames such as delay-sensitive multimedia frames or TCP-ACK [14]. It switches back to PSM only after a timeout has passed without traffic, which is referred to as *tail time*. For example, in Fig. 2(b), the third data packet generated during BI_i will be delivered immediately, which is not the case in Fig. 2(a).

To switch between the two power modes, a client sends an uplink frame (or a Null frame) with the power management bit (PwrMgt) in the frame header set or reset. The AP becomes aware of the switch, and it either starts to buffer incoming packets for that client or sends the received packets immediately. Fig. 2(b) shows that the client sends a PwrMgt-disabled (enabled) Null frame to switch to CAM (PSM). With the help of the tail time, A-PSM reduces the packet delay but consumes more energy than PSM. The amount of the tail time varies widely depending on manufacturers and software versions. For example, HTC Nexus One uses 200 msec and Apple iPhone 4 uses 90 msec.

B. Energy Saving of a Wifi AP

Energy saving of an AP has not been considered important until recently with the popularity of Wifi tethering. Considering the scenarios in Fig. 2, an AP or a MAP constantly remains in a high power awake state regardless of the traffic activity and depletes its battery power rapidly. Nevertheless, very little work has been reported in the literature. One such study is DozyAP [12], in which the power consumption of a MAP is reduced by putting its Wifi interface into a low power state when all its associated clients agree not to send any traffic while the MAP sleeps. This is facilitated in DozyAP by introducing "sleep request" and "sleep response" messages, and a new protocol on both AP and client devices. As discussed in

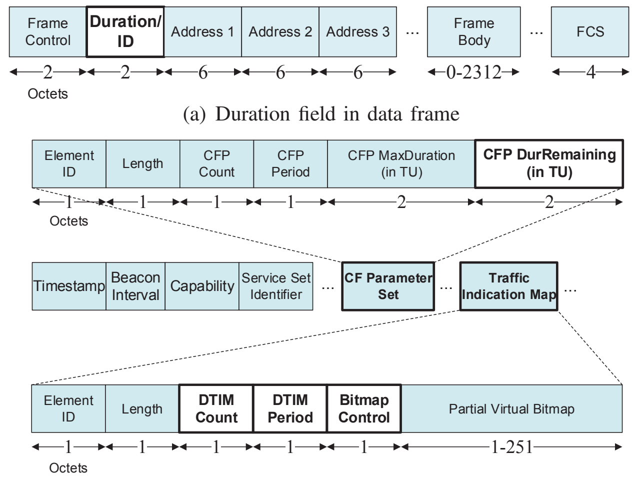


(a) CAM client

(b) PSM client

(c) A-PSM client

Fig. 3: E-MAP sleeps to save energy. (B: beacon, D: data, PN: pseudo null. For simplicity, ACK and other control and management frames are not drawn in the figure.)



(a) Duration field in data frame

(b) CFP duration and DTIM in beacon frame

Fig. 4: Structure of IEEE 802.11 MAC frames.

Introduction, this is not practically feasible because we cannot mandate all existing as well as future clients to implement the new protocol.

III. Energy-Efficient Mobile AP

A. E-MAP

This section proposes an energy-efficient MAP mechanism (E-MAP), which saves AP's energy by introducing the sleep cycle as in PSM. It uses standard features of 802.11 for compatibility as discussed in Introduction and they are integrated into *pseudo null* (PN), *pseudo multicast* (PM) or *pseudo beacon* (PB) frame. Those standard features are Duration field in the MAC header, CFP DurRemaining field of Point coordination function (PCF) and Delivery traffic indication map (DTIM) field in the beacon frame, which will be explained in detail later in this section.

PN frame for a short sleep. E-MAP goes to sleep when it observes an idle medium for more than a certain period, which is similarly approached in A-PSM (for clients) [9], [10] and DozyAP (for MAP) [12]. E-MAP maintains *traffic inactivity timer* (TIT) for this purpose. When it expires, E-MAP sends a *pseudo null* (PN) frame and begins to sleep as shown in Fig. 3(a). Note that the Duration field of the PN frame is set to a desired value of sleep duration which is typically until the next beacon time. Since the Duration field denotes the communication duration (see Fig. 4(a)), client devices set their NAV (*network allocation vector*) accordingly and would not attempt to send uplink traffic until NAV expires. This standard operation makes sure the absence of uplink traffic during the designated duration and thus, E-MAP can sleep safely.

PN-PM frame combination for PSM/A-PSM clients. In the Wifi tethering scenario, clients are typically laptops and tablets that operate in constantly awake mode (CAM). However, it is possible that E-MAP needs to support PSM as well as A-PSM clients. As discussed in Section II, a PSM client goes into a sleep state immediately after receiving a beacon frame when it is not addressed in the TIM. E-MAP does not have a chance to send a PN frame and thus, cannot sleep because the PSM client can wake up any time and send an uplink traffic.

To send a PN frame to all and every associated client, E-MAP uses a *pseudo multicast* (PM) frame. When E-MAP sends a beacon frame along with the indication of the presence of pending multicast frame(s), all the clients remain in the awake state to receive the multicast frame(s). E-MAP utilizes this all-awake duration to send a PN frame as shown in the right side of Fig. 3(b). This is followed by a PM frame to allow PSM clients to complete receiving the anticipated multicast frame and to sleep. Note that this procedure conforms to the delivery mechanism of multicast frames based on *Delivery traffic indication map* (DTIM) in the 802.11 standard. Fig. 4(b) shows the format of a TIM. When E-MAP sends a beacon frame with the DTIM count of 0 and the Bitmap control set, it denotes that the current TIM is DTIM and that a multicast frame is present.

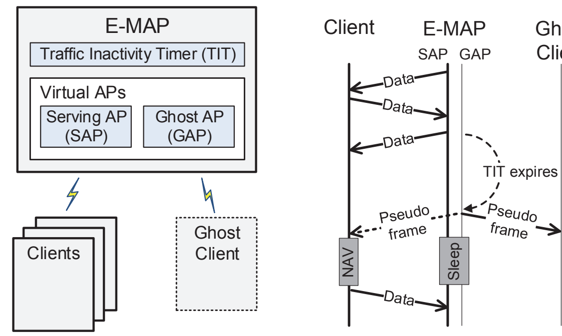
In case of an A-PSM client, it keeps its Wifi interface awake for the predefined tail time as discussed in Section II-A. E-MAP can utilize this time period to immediately send a PN frame if TIT has been expired, as shown in Fig. 3(c). Otherwise, E-MAP resorts to the PN-PM combination.

PB frame for a long sleep. Since the Duration field is 2 octets in size with MSB (most significant bit) reserved, it can represent at most 33 msec. E-MAP may want to sleep more

Algorithm 1 E-MAP Algorithm.

```
▷ Called when E-MAP algorithm starts
Procedure BEGINEMAP()
    Change Wifi into a high power state
    Start TIT                          ▷ execute the traffic inactivity timer

▷ Called whenever TIT expires due to no traffic activity
Procedure TITEXPIRED()
    if (all clients are in awake)
        sleepTime = SENDPSEUDOFRAME()
        Change Wifi into a low power state for sleepTime
    else
        Wait until the next DTIM beacon transmission

▷ Called when E-MAP sends PN/PB/PM frame
Procedure SENDPSEUDOFRAME()
    ▷ calculate remaining time until the next beacon
    remainTime = NextBeaconTime − CurrentTime
    if (PCF is available and remainTime > 33 msec)
        Send a PB frame with remainTime
    else
        remainTime = min(remainTime, 33 msec)
        Send a PN frame with remainTime
    return remainTime

▷ On sending a DTIM beacon frame
Procedure ISDTIMBEACONINTERVAL()
    Send a DTIM beacon frame
    if (TIT is expired)
        sleepTime = SendPseudoFrame()
        Send a PM frame with MoreData disabled
        Change Wifi into a low power state for sleepTime

▷ On receiving data frame from 3G network
Procedure RECEIVEFRAMEFROM3G()                      ▷ downlink
    if (client is in awake)                ▷ if the client is awake
        Change Wifi into a high power state (if necessary)
        Send the received data to the client
        Restart TIT                ▷ reset the traffic inactivity timer
    else                                  ▷ if client is in PS mode
        Buffer the received data

▷ On receiving a frame from a Wifi client
Procedure RECEIVEFRAMEFROMWIFI()                       ▷ uplink
    Restart TIT                    ▷ reset the traffic inactivity timer
    if (frame is data)
        Forward the received frame toward 3G
    else
        if (frame is PS-Poll)
            Send a buffered data to the corresponding client
        if (frame contains the PwrMgt bit enabled)
            Change information for the client to PS mode
        else
            Change information for the client to awake mode
            Send all remaining buffered data to the client
```

than this because BI is usually 100 msec. For this purpose, E-MAP uses a *pseudo beacon* (PB) frame with the indication of *contention free period* (CFP) longer than 33 msec. Client devices are not allowed to send uplink traffic during CFP unless it is polled by the AP and thus, E-MAP can sleep safely. Note that CFP is part of the PCF (*point coordination function*) mechanism of 802.11, in which a *superframe* is divided into CFP and *contention period* (CP). The CFP parameter Set is



(a) E-MAP architecture (b) Operations

Fig. 5: E-MAP architecture and the basic operation.

shown in Fig. 4(b). In particular, a *CFP DurRemaining* field is 2 octets in size and denotes the CFP duration in time units of 1,024 usec. Therefore, it can represent at most 67 sec.

Unfortunately, few commercial APs support PCF as it is designated as an optional service in the standard. However, since E-MAP does not require the full functionality of the PCF, this can be implemented with a minor addition to the existing DCF protocol. Of course, E-MAP can send another PN frame to sleep again in the same BI if PCF is not available.

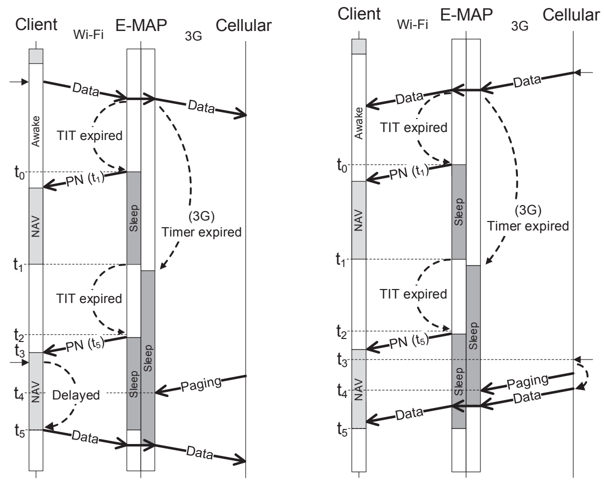
Mixture of CAM and PSM clients. When CAM and PSM clients coexist, E-MAP sends the PN-PM combination because it needs to make sure every client receives a PN frame. CAM clients would not be bothered with a PM frame.

B. *Calculation of TIT Threshold*

Putting a Wifi interface in a low power state after a certain idle time is not common although it is usually for client devices [9], [10]. This idle time threshold in the context of an A-PSM mechanism is 90 msec in iPhone 4 and 200 msec in Nexus One. A low threshold may impact the delay while a high threshold reduces the energy saving.

In E-MAP, the TIT threshold (sleep start time) is initially set to 150 msec as in DozyAP [12]. However, when E-MAP wakes up and finds no buffered traffic during the previous sleep duration, it sets it to the minimum inter-frame gap (*e.g.*, 20.51 msec or 1,023 backoff slots plus a succeeding DIFS time) because the network is known to be lightly loaded. Otherwise (presence of buffered traffic), it is set to 150 msec. The sleep duration is set to the remaining time until the next BI or 33 msec, whichever is smaller, in case of a PN frame. It is set to the remaining time until the next BI in case of a PB frame.

Pseudo-code of the E-MAP is shown in Algorithm 1. It does not show the operations for clients to discover and associate with the E-MAP as they are the same as the standard procedure defined in 802.11. One important consideration in this regard is that E-MAP needs to remain awake for at least one entire BI to make sure all clients in the tethering environment connect to the E-MAP. For those who wish to join later may not be connected because E-MAP frequently sleeps unlike a regular AP. However, it is not unreasonable to assume that all the clients start to operate from the beginning in the tethering scenario.



(a) CAM client with uplink (b) CAM client with downlink

Fig. 6: Interactions between an E-MAP and a CAM client.

C. E-MAP Architecture

One of design goals of E-MAP is to demand no changes on the client side and little changes on the MAP or the tethering device. To achieve the latter goal, E-MAP employs the AP virtualization technique as shown in Fig. 5(a). This allows a simpler design because *serving AP* (SAP) is just a normal AP and *ghost AP* (GAP) executes most of extra operations for energy efficiency. In other words, GAP transmits a PN or PB frame when it is triggered by the TIT as depicted in Fig. 5(b).

IV. Delay Analysis

A. Impact on Packet Delay

E-MAP causes no additional delay for downlink traffic regardless of the client type (CAM, PSM or A-PSM) but it does for uplink traffic. This section analyzes the delay.

CAM clients (Uplink traffic). Fig. 6(a) represents the interactions between an E-MAP and a CAM client with uplink traffic. Here is the scenario.

- At time t_0, TIT expires because there is no traffic for a while. E-MAP sends a PN frame with the sleep duration until time t_1 and then transits its Wifi interface to a low power state. A CAM client overhears the PN frame and sets NAV accordingly.
- At time t_1, E-MAP's Wifi interface wakes up and restarts TIT. TIT restarts to monitor the traffic inactivity.
- At time t_2, TIT expiry reoccurs and E-MAP sends a PN and decides to sleep again until t_5.
- For the packet generated at time t_3, it cannot be transmitted until the NAV expires. The CAM client buffers the packet until t_5 before transmitting to E-MAP as shown in Fig. 6(a). This results in an additional delay and is caused by E-MAP.

CAM clients (Downlink traffic). Let us consider the downlink traffic in a similar scenario mentioned above. The first packet in Fig. 6(b) is delivered immediately as both 3G and Wifi interfaces of E-MAP are awake. For the packet generated at time t_3, E-MAP does not cause any additional delay because it can wake up the Wifi interface and deliver it immediately to the CAM client. On the other hand, when the 3G interface sleeps as in the figure, there will be an additional delay. According to the common 3G network operation, the cellular network periodically sends a paging message indicating that there is a data to be delivered and the client wakes up periodically to listen for an incoming page. In the figure, the 3G interface of E-MAP receives the page at t_4 and receives the following data from the 3G network. Then, it wakes up the Wifi interface to deliver the packet to the CAM client. However, it is important to note that this additional delay is caused by the typical 3G power saving operation and E-MAP does not contribute to this end.

PSM/A-PSM clients. The impact on uplink packet delay in the case of PSM and A-PSM clients is very similar to the CAM clients discussed above. We skip this for brevity in this paper. Please refer to [15] for details.

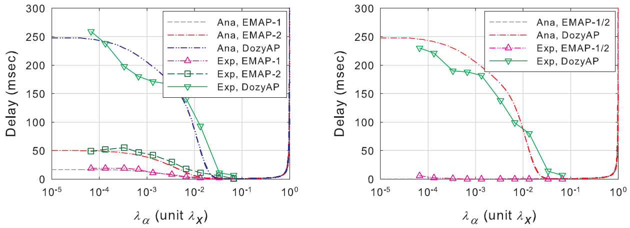
B. Mathematical Analysis of Packet Delay

In a wireless network, packet delay is oftentimes determined by queueing delay not only due to the bursty nature of network traffic but also due to the sleep cycle operation in either the 3G or Wifi network. The E-MAP scenario can be modeled using M/G/1 queue with multiple vacations because the sleep duration can be considered a vacation. The key difference from the conventional M/G/1 vacation model is that the server (tethering device or E-MAP) goes to a vacation (sleep) after some time (TIT expiry) rather than immediately when the queue is empty. In fact, similar scenarios have been studied elsewhere. For exmaple, [16] studied the effect of the sleep periods on the queueing delay for UMTS networks. Thus, we shall utilize the results from [16] to analyze the packet delay of the Wifi connection in an E-MAP network.

Uplink delay of E-MAP. The analysis begins with the derivation of queue length distribution. Let L_n be the queue length after a client sends n^{th} packet to E-MAP. ($L_n = 0$ implies that TIT starts.) And, let Q_n be the number of packets arrived at the client before TIT expires. (E-MAP goes into sleep when $Q_n = 0$.) Now, if we let R_n and A_n be the number of packets arrived during sleep and during the service time of the n^{th} packet, respectively, it is possible to express L_{n+1} as a function of L_n, Q_n, R_n and A_{n+1} and the sequence $\{L_n; n = 1, 2, \cdots\}$ constitutes a Markov chain [16]. Solving this Markov chain to derive the expected packet delay $E[t_d]$ results in:

$$E[t_d] = \frac{\lambda_a e^{-\frac{\lambda_a}{\lambda_I}}(1 + V_D\lambda_D^2)}{2\{\lambda_D^2[1 - e^{-\frac{\lambda_a}{\lambda_I}}][1 - e^{-\frac{\lambda_a}{\lambda_D}}] + \lambda_a\lambda_D e^{-\frac{\lambda_a}{\lambda_I}}\}} + \frac{1}{\lambda_x} + \frac{(1 + V_x\lambda_x^2)\rho}{2\lambda_x(1 - \rho)}. \quad (1)$$

Here, λ_x, λ_a, λ_I, λ_D, V_x, V_D, and ρ represent the service rate ($1/t_x$), packet arrival rate, TIT expiration rate ($1/TIT$), sleep rate ($1/NAV$), variance of service rate ($1/\lambda_x^2$), variance



(a) Uplink (b) Downlink

Fig. 7: Analysis and experimental delay results as a function of the packet arrival rate λ_a (TIT threshold: 150 msec, Sleep duration (EMAP-1): 33 msec, Sleep duration (EMAP-2): 100 msec, Sleep duration (DozyAP): 500 msec).

of sleep rate $(1/\lambda_D^2)$, and λ_a/λ_x, respectively. Please refer to [15] and [16] for a more detailed derivation.

Downlink delay of E-MAP. As discussed before, E-MAP does not have a negative impact on downlink packet delay. Upon receipt of a packet from 3G, E-MAP immediately wakes up its Wifi interface and forwards the packet to the client. That is, only the packets buffered until waking up are not be considered since the sleep of E-MAP does not make any trouble in the downlink traffic. Thus, the expected packet delay $E[t_d]$ can be derived as:

$$E[t_d] = \frac{(1+V_x\lambda_x^2)\rho}{2\lambda_x(1-\rho)[1-(1+\rho)e^{-\frac{\lambda_a}{\lambda_I}}]} + \frac{1-e^{-\frac{\lambda_a}{\lambda_I}}}{\lambda_x[1-(1+\rho)e^{-\frac{\lambda_a}{\lambda_I}}]} - \frac{e^{-\frac{\lambda_a}{\lambda_I}}[2+\rho+3\lambda_x V_x\lambda_a+2V_x\lambda_x^2(1+V_x\lambda_x\lambda_a)]\rho}{2\lambda_x(1-\rho)[1-(1+\rho)e^{-\frac{\lambda_a}{\lambda_I}}]}. \quad (2)$$

Analysis results. Fig. 7 shows the variation of the expected packet delay with respect to the packet arrival rate λ_a. While the packet delay usually increases with increasing λ_a, this is not the case with a low traffic condition in the tethering scenario as shown in Fig. 7(a). This is because a client device most likely observes a sleeping E-MAP (in fact, the client thinks E-MAP is busy with other clients) and thus, waits until it wakes up, resulting in a high packet delay. Observe that the expected packet delay approaches 16.5 and 50 msec for EMAP-1 and EMAP-2, respectively, which is the half of the sleep duration (33 and 100 msec). However, as packet arrival rate increases, E-MAP has a better chance to be in an awake state and to receive the uplink traffic immediately, resulting in less delay. Of course, this trend does not continue when λ_a is sufficiently large. In other words, the delay rapidly increases when λ_a becomes much higher than the service rate λ_x. Fig. 7 includes the results of DozyAP. In DozyAP, uplink or downlink packet(s) have to be buffered until the sleep duration finishes. The delay of both directions can be derived as (1) and (2).

V. Performance Evaluation

A. Evaluation Methodology

To choose an experimental platform for this research, the followings are considered. It should (i) support a tethering function, (ii) allow modification on network kernel driver, and (iii) provide a detachable battery to connect the power monitor to the device. Among the possible candidates, we chose a HTC Nexus One smartphone running Android 2.3.6 and implemented E-MAP and DozyAP on it. IBM Thinkpad X201 and Samsung Nexus S are used as CAM and PSM/A-PSM clients. Note that no modification is needed at the client side for E-MAP as discussed above while some changes are necessary for DozyAP.

TABLE I: Average packet interval and standard deviation (SD) of six different traces. L, M, and H indicate low, moderate, and high traffic intensity, respectively.

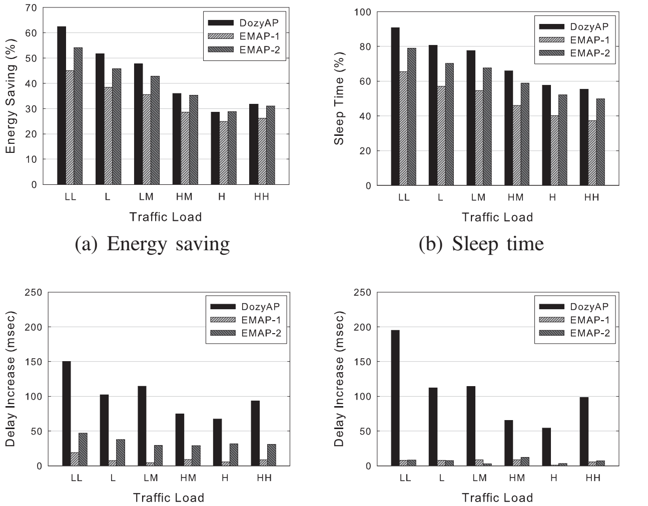
	LL	L	LM	HM	H	HH
Interval(s)	5.26	2.61	1.40	0.93	0.68	0.54
SD	17.90	8.47	5.37	2.96	2.25	1.81

We measured the power consumption of a MAP using Monsoon Power Monitor, an external, real-time power monitoring device [11]. Since it measures the total power consumption, it is necessary to minimize the contributions from other components in a MAP (HTC Nexus One). We turned off the backlight, disabled all other communication interfaces such as Bluetooth, and initialized it in a factory default setting with no application running in the background. To be more accurate, we also measured the sleep time of Wifi interface and found that the total power consumption can reasonably be used as a metric for the energy efficiency of the Wifi interface. (See Fig. 8(a) and 8(b).)

Another important measure is packet delay. A simple java program has been developed to measure the end-to-end delay between a client and a corresponding server, which are connected via a tethering MAP (HTC Nexus One). For every packet, the java program at both ends and the MAP record the timestamp and calculate delays at 3G/buffer/Wifi as well as the end-to-end delay. Before the measurement, we ran the NTP daemon on the server and synchronized the clock of the client and the MAP with the server for more accurate measurements.

Packets are communicated according to the same time and order as in a real traffic trace obtained from [13]. This trace has the history of the Internet usage for 24 iPhone 3GS users, including user's unique ID, the timestamp, a specific URL address that the users visited through a 3G network. Among them, we selected six traces with different traffic patterns to evaluate the impact of traffic on the energy saving. Table I shows the statistics of the six traffic traces.

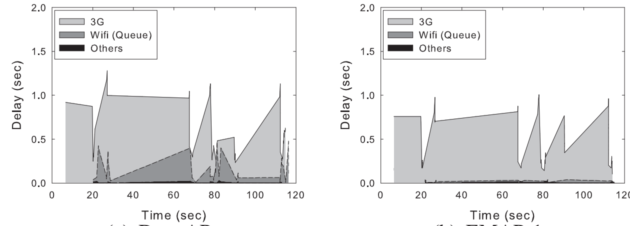
For comparison, we measured the power consumption and the packet delay of a normal MAP, DozyAP, and two variants of E-MAP (EMAP-1 and EMAP-2). EMAP-1 uses a PN frame to realize a short sleep (at most 33 msec per sleep as discussed in Section III-A). EMAP-2 uses a PB frame for a longer sleep (as long as 100 msec per sleep because BI is set to 100 msec during our experiments). EMAP-2 saves more energy but increases the packet delay more. DozyAP constitutes an extreme where the packet delay is the highest. The measurement of a normal MAP provides the baseline



(a) Energy saving (b) Sleep time

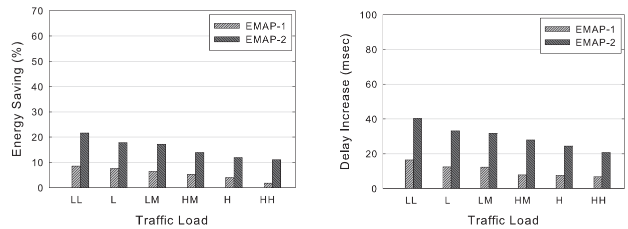
(c) Uplink delay (d) Downlink delay

Fig. 8: Energy and delay performance when a CAM client is connected to a tethering phone. (Energy saving in Fig. (a) shows the difference in comparison to a normal MAP. Similarly, Figs. (c) and (d) show the delay increase in comparison to a normal MAP. Packet delay in Figs. (c) and (d) denotes the end-to-end delay between a Wifi-only client and a corresponding server via a DozyAP or an E-MAP.)



(a) DozyAP (b) EMAP-1

Fig. 9: Breakdown of uplink delay for DozyAP and EMAP-1 with LM traffic. (We omit the result of EMAP-2 since it is almost similar to that of EMAP-1.)



(a) Energy saving (b) Uplink delay

Fig. 10: Energy and delay performance when a PSM client is connected to a tethering phone. (DozyAP is missing because it does not support PSM clients.)

performance for comparison.

B. Performance with a CAM Client

Fig. 8 shows the energy and delay performance of a MAP with a CAM client. It is shown in Fig. 8(a) that DozyAP saves more energy (28.5∼62.4% in comparison to a normal MAP) than EMAP-1 (24.9∼45.0%) and EMAP-2 (28.7∼54.1%) for six traffic patterns but the difference is at most 17.4% and 8.3% for EMAP-1 and EMAP-2, respectively. One important observation is that the energy saving measurement in Fig. 8(a) closely matches the sleep time measurement shown in Fig. 8(b) as mentioned earlier.

On the contrary, DozyAP increases the packet delay significantly as shown in Fig. 8(c) and Fig. 8(d). It is as much as 12 and 5 times higher than EMAP-1 and EMAP-2, respectively, in case of uplink traffic. This is mainly due to the longer sleep duration of DozyAP and is hardly be justified with the relatively smaller benefits in terms of energy savings. This is even more significant in case of downlink traffic shown in Fig. 8(d). Since DozyAP uses explicit sleep request/response messages, it does not forward data packets from cellular network while the MAP sleeps. On the other hand, E-MAP immediately wakes up its Wifi interface upon a downlink traffic and forwards the received packet to the corresponding client. This saves E-MAP's energy while minimizing the delay increase.

To see it more accurately, the breakdown of the end-to-end delay is shown in Fig. 9. From the figure, it is evident that the 3G interface in a tethering network is the dominant factor which determines the overall delay on wireless path to the client. However, DozyAP's queueing delay at the Wifi interface is not negligible and EMAP-1 shows a significantly lower queueing delay than DozyAP.

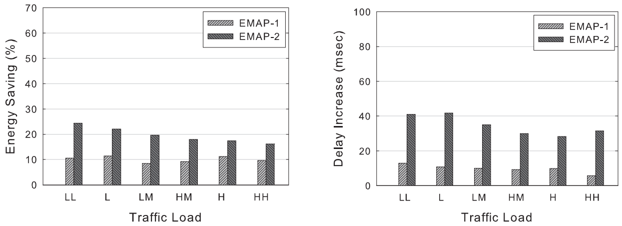
C. Performance with a PSM/A-PSM Client

The energy and delay performance of a tethering phone with a PSM client is shown in Fig. 10. The performance of DozyAP is not included because it does not support PSM clients. Comparing it with Fig. 8, both energy saving and delay increase are reduced. The reason is that E-MAP cannot sleep when a PSM client already entered a sleep mode, reducing the energy saving. Nevertheless, EMAP-1 and EMAP-2 saves 21% and 9% of total energy, respectively, compared to a normal MAP. Fig. 10(b) shows that EMAP-1 and EMAP-2 cause uplink delay increase by 8∼18 msec and 20∼40 msec, respectively, which is slightly smaller than in Fig. 8(c). In both cases, it is less than 50 msec on the average because a MAP is not allowed to sleep more than one BI (100 msec).

Fig. 11 presents the energy and delay performance of EMAP-1 and EMAP-2 with an A-PSM client. As shown in Fig. 11(a) in comparison to Fig. 10(a), it enables a tethering phone to conserve more energy than with a PSM client due to the tail time mentioned in Section III-A. The A-PSM client works like as a CAM client under a frequent traffic activity. One interesting observation is that energy saving is reduced as traffic load increases in both Fig. 10(a) and Fig. 11(a) but this is not as dramatic in the latter. Energy performance of E-MAP is insensitive to traffic load in case of an A-PSM client. The same is true with the packet delay as shown in Fig. 11(b).

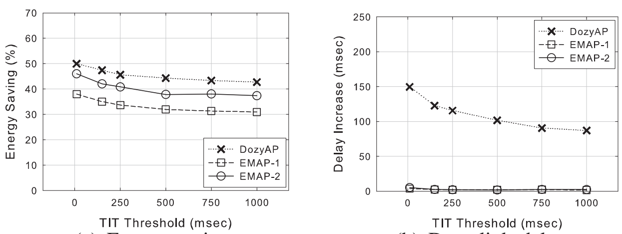
D. Impact of TIT Threshold

TIT threshold determines the sleep start time in both DozyAP and E-MAP and thus can affect the energy and delay



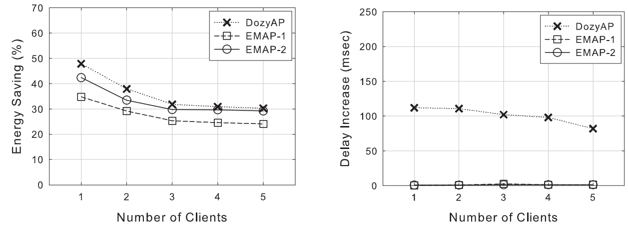
(a) Energy saving (b) Uplink delay

Fig. 11: Energy and delay performance when an A-PSM client is connected to a tethering phone. (DozyAP is missing because it does not support A-PSM clients.)



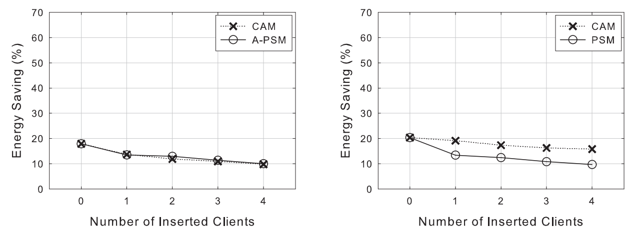
(a) Energy saving (b) Downlink delay

Fig. 12: Energy and delay performance with various TIT threshold values. (LM traffic pattern is used for this experiment.)



(a) Energy saving (b) Downlink delay

Fig. 13: Energy and delay performance with multiple clients. (LM traffic pattern is used for this experiment.)



(a) One PSM client and multiple CAM or A-PSM clients (b) One A-PSM client and multiple CAM or PSM clients

Fig. 14: Energy saving of E-MAP with a mixture of CAM and PSM clients.

performance. Fig. 12(a) shows that energy saving decreases as the TIT threshold increases although the difference is not significant. This is obvious that a higher TIT threshold leads to a more infrequent sleep transition and thus, less energy saving. However, as shown in Fig. 12(b), a higher TIT threshold forces a tethering phone to keep awake for a longer time, reducing the packet delay. This mitigates DozyAP's delay increase caused by the sleep process of Wifi interface. As discussed before, EMAP-1 and EMAP-2 do not have a negative impact on downlink delay and cause a negligible difference with various TIT thresholds. It can be concluded that E-MAP is not critically sensitive to the choice of TIT threshold value.

E. Multiple Clients

The tethering scenario could be complicated with more than one client as they cause a heavier traffic activity, forcing the tethering AP to stay awake longer, and thereby, reducing the chance to sleep and save energy. Fig. 13 shows the experiment results with 1∼5 clients. According to the results, all the three schemes can still reduce a lot of energy consumption up to more than 20%. Specifically, DozyAP can achieve 30% of energy saving even though five clients are connected to the AP while it is 29% and 24% with EMAP-1 and EMAP-2, respectively. However, again, DozyAP results in a much higher delay than E-MAP. Note that the delay decreases as the number of clients increases. With the large number of clients, a MAP is more likely to remain awake due to the increased network activity, and thus the additional packet delay disappears.

F. Mixture of CAM and PSM Clients

Fig. 14 illustrates the performance results of E-MAP under the coexistence environments. To better understand the impact of the mixture of CAM and PSM clients, we tested two scenarios. Fig. 14(a) shows the results with one PSM client and a different number of CAM or A-PSM clients. Fig. 14(b) shows the results with one A-PSM client and a different number of CAM or PSM clients. As in the previous results, the increasing number of clients reduces the energy saving. This is due to the increasing traffic intensity in case of Fig. 14(a), which is similarly the case with multiple CAM clients in Fig. 14(b). However, a significant reduction in energy saving with one or multiple PSM clients in Fig. 14(b) is simply due to the existence of a PSM client. In other words, the coexistence of different PSM types does not affect the performance if a PSM client already exists. But, the addition of a PSM client makes a difference when there was none.

VI. Related Work

Energy efficient Wifi operation has been a very hot topic for more than a decade. This section presents a brief survey on the variants of PSM.

Packet scheduling. There has been a lot of research based on packet scheduling approach to save power of mobile devices. PSM-throttling [17] reduces a significant amount of energy wasted on idle channel listening without degrading the user-perceived performance by reshaping the traffic into periodic bursts. Catnap [18] minimizes power consumption of clients by allowing them to sleep during the small idle intervals between packets by leveraging the different bandwidth between wired and wireless links at the AP. [19] forces a client to sleep while the level of the client's buffer is above a certain threshold, and saves its energy. [14] uses a web proxy at the AP splitting TCP flows coming from a server, and regulates the relaying of the packets back to the client to maximize the energy saving with the minimal extra delay. All these solutions are aiming at extending sleep duration by handling transmission intervals. To conserve more energy, [20] reduces the number of wireless data transfers by using the client's cache and the bursty Web pages automatically downloaded from a proxy.

MAC parameter adaptation. [21] adjusts scanning intervals depending on the amount of time the client spent in a disconnected or idle state, and thereby reduces energy consumption caused by periodic scanning process. C-PSM [22] saves energy by choosing optimal PSM parameters (*e.g.*, beacon and listen intervals) depending on traffic patterns. Further, STPM [23] adaptively determines client's power state by estimating the costs (*e.g.*, time and energy) required to transit between CAM and PSM, and the expected costs after the transition.

Contention avoidance. Contention avoidance scheduling has been extensively studied in [9], [24], [25] in a way to save energy. However, those algorithms usually consider a single network and could result in energy waste in a highly populated area with multiple APs. SleepWell [26], an alternative to this problem, forces PSM clients to wake up at different times by spreading APs' beacon start time, and thereby reduces contention.

Sleep scheduling. A number of studies have been done to avoid power drawn during idle by putting into a sleep state more effectively. BSD [27] reduces unnecessary energy usage due to the periodic wake up while bounding an extra delay based on a user-supplied parameter. SPSM [28] adjusts clients' sleep cycle to provide guarantees on a desired delay for each user. SiFi [10] saves energy during the silence period of a VoIP call by predicting the length of future silence periods based on historical data. Similarly, μPM [29] predicts the arrival time of next incoming frame and outgoing frame, and enables clients to enter the power saving mode during the short idle intervals between MAC frames.

VII. Conclusion

PSM has been an important technique to save energy of power-hungry mobile devices. More sophisticated PSM schemes reduce energy consumption while regulating delay increase but most of them just focused on the client side. With the increasing popularity of Wifi tethering, it is imperative to have an energy efficient MAP algorithm which is not limited by applicability and compatibility. This paper presents the design, implementation, and evaluation of a new energy efficient Wifi tethering for a MAP that increases battery life by putting its Wifi interface into a low power state. Further, it minimizes the delay increase by adaptively choosing the sleep start time and duration. From experimental evaluation, E-MAP results in battery life improvement of up to 54%.

Acknowledgement

This research was supported in part by the National Science Foundation under Grant No. CNS-1338105, the Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (2013R1A1A2065379), and the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIP) (2011-0029034).

References

[1] O. B. Yetim and M. Martonosi, "Adaptive Usage of Cellular and WiFi Bandwidth: An Optimal Scheduling Formulation," CHANTS, 2012.
[2] A. Rahmati and L. Zhong, "Context-for-Wireless: Context-Sensitive Energy-Efficient Wireless Data Transfer," ACM MobiSys, 2007.
[3] Chetan Sharma Consulting, http://www.chetansharma.com/.
[4] Apple iOS, https://developer.apple.com/technologies/ios/.
[5] Windows Phone, www.windowsphone.com/.
[6] Android Phone, http://www.android.com/.
[7] VerizonWireless, http://www.verizonwireless.com/.
[8] IEEE 802.11, *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, Standard, IEEE, Aug. 1999.
[9] E. Rozner, V. Navda, R. Ramjee, and S. Rayanchu, "NAPman: Network-Assisted Power Management for WiFi Devices," ACM MobiSys, 2010.
[10] A. J. Pyles, Z. Ren, G. Zhou, and X. Liu, "SiFi: Exploiting VoIP Silence for WiFi Energy Savings in Smart Phones," UbiComp, 2011.
[11] Monsoon Solutions Inc. http://www.msoon.com/LabEquipment/PowerMonitor/.
[12] H. Han, Y. Liu, G. Shen, Y. Zhang, and Q. Li, "DozyAP: Power-Efficient Wi-Fi Tethering," ACM MobiSys, 2012.
[13] R. LiKamWa, Y. Liu, N. D. Lane, and L. Zhong, "Can Your Smartphone Infer Your Mood?," PhoneSense workshop, 2011.
[14] N. Ding, A. Pathak, D. Koutsonikolas, C. Shepard, Y. C. Hu, and L. Zhong, "Realizing the Full Potential of PSM using Proxying," IEEE Infocom, 2012.
[15] K. H. Jung, Y. Qi, and C. Yu, "Energy Efficient Tethering for a Mobile AP," Technical Report, http://monet.postech.ac.kr/doc/tr-emap.pdf
[16] S.-R. Yang and Y.-B. Lin, "Modeling UMTS Discontinuous Reception Mechanism," IEEE TWC, 2005.
[17] E. Tan, L. Guo, S. Chen, and X. Zhang, "PSM-throttling: Minimizing Energy Consumption for Bulk Data Communications in WLANs," ICNP, 2007.
[18] F. R. Dogar, P. Steenkiste, and K. Papagiannaki, "Catnap: Exploiting High Bandwidth Wireless Interfaces to Save Energy for Mobile Devices," ACM MobiSys, 2010.
[19] D. Bertozzi, L. Benini, and B. Ricco, "Power Aware Network Interface Management for Streaming Multimedia," IEEE WCNC, 2002.
[20] T. Armstrong, O. Trescases, C. Amza, and E. deLara, "Efficient and Transparent Dynamic Content Updates for Mobile Clients," ACM MobiSys, 2006.
[21] A. Gupta and P. Mohapatra, "Energy Consumption and Conservation in WiFi Based Phones: A Measurement-Based Study," IEEE SECON, 2007.
[22] Y. Xie, X. Luo, and R. K. C. Chang, "Centralized PSM: An AP-centric Power Saving Mode for 802.11 Infrastructure Networks," SARNOFF, 2009.
[23] M. Edmund, E. Nightingale, and J. Flinn, "Self-Tuning Wireless Network Power Management," ACM MobiCom, 2003.
[24] Y. Heand R. Yuan, "A Novel Scheduled Power Saving Mechanism for 802.11 Wireless LANs," IEEE TMC, 2009.
[25] H.-P. Lin, S.-C. Huang, and R.-H. Jan, "A Power-Saving Scheduling for Infrastructure-Mode 802.11 Wireless LANs," Computer Communications, 2006.
[26] J. Manweiler and R. R. Choudhury, "Avoiding the Rush Hours: WiFi Energy Management via Traffic Isolation," ACM MobiSys, 2011.
[27] R. Krashinsky and H. Balakrishnan, "Miniminzing Energy for Wireless Web Access Using Bounded Slowdown," ACM MobiCom, 2002.
[28] D. Qiao and K. Shin, "Smart Power-Saving Mode for IEEE 802.11 Wireless LANs," IEEE Infocom, 2005.
[29] J. Liu and L. Zhong, "Micro Power Management of Active 802.11 Interfaces," ACM MobiSys, 2008.