Saving Energy in Wi-Fi Hotspots through 802.11 PSM: an Analytical Model*

G. Anastasi[1], M. Conti[2,3], E. Gregori[2,3] and A. Passarella[1]

[1] University of Pisa, Dept. of Information Engineering
Via Diotisalvi, 2 - 56122 Pisa, Italy
g.anastasi, a.passarella@iet.unipi.it
[2] CNR, IIT Institute
Via G. Moruzzi, 1 - 56124 Pisa, Italy
marco.conti, enrico.gregori@iit.cnr.it
[3] CREATE-NET
http://www.create-net.it

Abstract. Wi-Fi hotspots are becoming very popular. Due to limited costs, they are an effective solution to provide wireless Internet access to nomadic users. A key point to exploit the potentialities of Wi-Fi hotspots is managing the scarce energetic resources of mobile devices. To this end, the IEEE 802.11 standard defines a Power-Saving Mode aimed at reducing the energy consumption due to networking activities. In this paper we provide an analytical model of this algorithm. We derive closed formulas for mobile device energy consumption, and highlight its dependence on key parameters, such as the number of users inside the hotspot, the (wired) Internet throughput, and the MAC protocol parameters. The results reported show that, in a standard TCP/IP architecture, the 802.11 PSM scales quite well with respect to the number of users inside the hotspot. However, in some cases, it can even increase the energy consumption obtained without energy management.

1 Introduction and Motivation

Wi-Fi hotspots are limited-sized areas (e.g., a campus, an airport, ...) where wireless coverage is provided by means of 802.11 Access Points. As Access Points are connected to the Internet, users equipped with mobile devices such as notebooks, PDAs, smartphones, can access Internet services wirelessly from the hotspot. The enabling technology for this scenario is IEEE 802.11 [10]. In the last few years the cost of 802.11 Access Points and network cards dropped down significantly, making Wi-Fi hotspots a very appealing business scenario for wireless Internet Service Providers. At the same time, many researchers are working on evaluating and improving the performance of Internet-access through Wi-Fi hotspots (e.g., [5, 6, 11]).

Energy management for the mobile-device networking subsystem deserves special attention ([1, 2, 5, 8, 11, 12]). It has been widely shown that networking activities account for 10% up to 50% of the energy spent by a mobile device ([12]). Since the difference between batteries' capacity and mobile-device energetic requirements is becoming more and more broad ([13]), it is vital to design network architectures that optimize the energy use. To this end, the 802.11 standard includes a Power-Saving Mode (PSM) aimed at exploiting low-power states of the network interface to save energy. Surprisingly, to our knowledge very little effort has been put on evaluating the performance of PSM in a Wi-Fi hotspot scenario. The behavior of PSM with respect to 802.11 without energy management is investigated in [11] by means of simulations, and in [1] by means of experiments. Both [11] and [1] consider a single user inside the hotspot. [11] focuses on Web applications, and shows that PSM energy saving can be significant (about 90%). However, non-negligible delays are introduced in delivering frames to mobile devices. In [1] the hotspot is seen as an extension of the user corporate LAN, and the reference application is the access to files shared via NFS. It is shown that also in this case PSM delays negatively impact the QoS. Furthermore, the energy spent by the device can even increase when PSM is activated. Though [11] and

* This work was carried out under the financial support of the Italian Ministry for Education and Scientific Research (MIUR) in the framework of the Projects: "Internet: Efficiency, Integration and Security", FIRB-VICOM and FIRB-PERF.

[1] highlight some limitations of 802.11 PSM, none of them provides an extensive characterization of this power-saving algorithm. Specifically, these analyses are customized to a specific application. Furthermore, the impact on PSM performance of environment parameters (e.g., the traffic profile, the number of users inside the hotspot, etc.) is not analyzed. Hence, it is hard to clearly understand the PSM strengths and weaknesses.

These observations have motivated us to develop an analytical model to understand the 802.11 PSM performance in depth. In this paper we derive an analytical model and validate it by simulation. With respect to energy saving, our results are aligned with those reported in [11]. Furthermore, our model shows the dependence of the PSM performance on key parameters, such as the Internet throughput, the traffic profile, the MAC-protocol parameters, etc. Specifically, we highlight the impact of other mobile users on the energy spent by a tagged, PSM-enabled, mobile device. Results show a graceful degradation of the PSM performance as the number of users increases. Furthermore, a critical point is found (around 35 users), beyond which PSM is not effective anymore. To summarize, our model contributes to an in-depth characterization of 802.11 PSM behavior and hence provides the basis to improve it beyond its current limitations.

The rest of the paper is organized as follows. Section 2 describes the networking scenario that we consider in our analysis. Section 3 introduces the analytical model. Finally, Section 4 is devoted to the model validation.

2 Reference Scenario

In this work we focus on best-effort Internet applications, i.e., applications without real-time requirements such as Web, e-mail, and file transfer, which are now the most widely used in Wi-Fi hotspots. In detail, our reference scenario is as follows (see Figure 1-left). A tagged PSM-station in a Wi-Fi hotspot runs an application that downloads B bytes from a fixed server connected to the Internet. Data are not downloaded continuously, but as a sequence of *chunks*, spaced by *idle phases* during which no traffic flows between the server and the client. Furthermore, the PSM station uses a standard TCP/IP architecture. The whole download is supported by a unique TCP-connection, i.e., the same connection persists among different chunk downloads. In detail, we assume TCP-Reno, without delayed acks [14]. Finally, other M stations are active in the hotspot. These stations *do not use* the PSM, and generate an asymptotic background traffic, in the sense that they always have a frame ready to be sent. Moreover, no hidden station phenomena occur, i.e., all stations can hear transmissions of each other.

The motivations behind the scenario definition are as follows. First, though very simple, the application-level traffic model captures the typical user behavior. For example, Web users download a page (i.e., a chunk of data) and then read the page contents without generating any traffic on the network. Moreover, assuming persistent connections is aligned with the most recent strategies aimed at boosting Internet servers' performance (see, for example, HTTP/1.1). Finally, by varying the number or active stations (i.e., M) we can analyze the sensitiveness of the PSM to the congestion level in the hotspot, and – therefore – its scalability with respect to the number of users.

As a final remark, it should be noted that the TCP/IP architecture can be highly improved in a Wi-Fi hotspot scenario (see the survey in [4]). However, TCP/IP is currently the only off-the-shelf solution for Wi-Fi hotspots. From this standpoint, our work analyzes the performance and limitations of the currently available technology, and can thus be used as a reference to evaluate novel solutions.

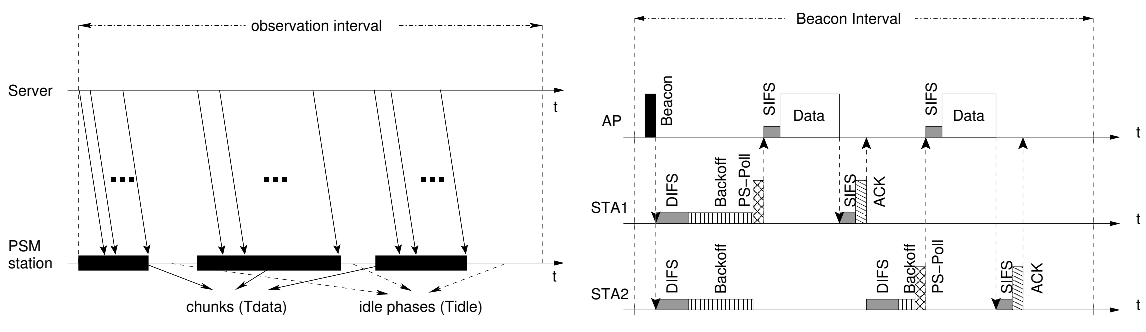

Figure 1. Snapshot of the application-level traffic (left), and example of the 802.11 PSM operations (right).

2.1 Overview of the 802.11 Infrastructure Power-Saving Mode

Before proceeding with the analysis, it is worth recalling the mechanisms introduced by 802.11 PSM. The complete description of 802.11 and of its PSM can be found in [10]. The standard 802.11 defines two possible operating modes for wireless interfaces, i.e. the *active* mode and the *sleep* mode. In active mode, wireless interfaces are able to exchange data, and can be in the *receiving*, in the *transmit*, or in the *idle* state (i.e., they simply overhear the traffic on the channel). The energy consumption of the active mode depends very little on the operating state [9]. Hence, it is typically approximated with a constant value (e.g., $750\,mW$ for Enterasys Networks RoamAbout interfaces [11]). On the other hand, in sleep mode only few components of the wireless interface are supplied by the battery (e.g., the clock that maintains the synchronization with the Access Point). In this mode, wireless interfaces are not able to exchange data, but their energy consumption is at least one order of magnitude lower than in the active mode (e.g., $50\,mW$, [11]). Transition times from the sleep to the active mode are in the order of few $msec$, while opposite transition times are typically assumed to be negligible [11]. During the transition to the active mode, the energy consumption is almost the same than in the active mode itself [11].

The objective of the 802.11 PSM is to let a mobile device in the active mode[4] only for the time necessary to exchange data, and to turn it in the sleep mode as soon as it becomes idle. In a Wi-Fi hotspot (i.e., an infrastructure 802.11 WLAN), this is achieved by exploiting the role of the Access Point. Each station inside the hotspot informs the Access Point whether it utilizes the PSM or not. As the Access Point relays every frame from/to any station, it buffers frames addressed to stations operating in PSM while they are sleeping. Once every Beacon Interval – usually, $100\,msec$ –, the Access Point broadcasts a special frame, named Beacon. This frame contains a Traffic Indication Map (TIM) that indicates PSM stations having frames buffered at the Access Point. PSM stations are synchronized with the Access Point, and wake up to receive Beacons. If they are indicated in the TIM, they download the frames as shown in Figure 1-right. Specifically, a PSM station sends a special frame (PS-Poll) to the Access Point by means of the standard DCF procedure. Upon receiving a PS-Poll, the Access Point sends the first Data frame to the PSM station, and receives the corresponding Ack frame. If appropriate, the Access Point sets the More Data bit in the Data frame to announce other frames addressed to the same PSM station. To download the next frame, the station sends *another* PS-Poll. When, eventually, the station has downloaded all the buffered frames, it switches to the sleep mode. Finally, to send a Data frame, a PSM station wakes-up (if necessary) and performs the standard DCF procedure.

3 Analytical Model of the 802.11 PSM

According to the reference scenario, our aim is to define a model for evaluating the energy spent by the PSM station to download B bytes from the server. Before proceeding, let us introduce some notations used throughout the paper. The energy spent by the PSM station is referred to as E_{PSM}. Moreover, T denotes the time required to download the B bytes. As we will show later, T corresponds also to the time during which we observe the system. Hence, throughout it is also referred to as *observation interval*. Finally, T_{ac} and T_{sleep} denote the total time (in an observation interval) during which the PSM station is in the active[5] and sleep modes, respectively (i.e., $T = T_{ac} + T_{sleep}$); and P_{ac} and P_{sleep} denote the power required by the active and sleep modes, respectively. Based on the results presented in [9], we assume that the power required by the active mode is always the same, whether the PSM station is receiving, transmitting or idle.

A simple expression of E_{PSM} is provided in Proposition 1 (For the sake of space, we omit all proofs. They are available in [3]).

Proposition 1. *The energy spent by the PSM station during the observation interval is*

$$E_{PSM} = T_{ac} \cdot P_{ac} + T_{sleep} \cdot P_{sleep}\ . \tag{1}$$

As T_{sleep} can be obtained as $T - T_{ac}$, the core of this paper consists in characterizing T_{ac} and T. Specifically, Sections 3.2 and 3.3, are devoted to deriving closed-form expressions for the average values of these quantities.

[4] Since we are interested in the wireless-interface energy consumption, when we talk about energy consumption and operating modes of a mobile station we intend energy consumption and operating modes of the *wireless interface* of a mobile station.

[5] T_{ac} also includes transition times from the sleep mode.

3.1 Modeling the Wi-Fi hotspot

As a preliminary step, it is worth describing the WiFi-hotspot model that we assume. In our scenario M background stations are active in the hotspot, and operate in asymptotic conditions. We assume that background stations do not use the standard IEEE 802.11 protocol, but the p-persistent IEEE 802.11 protocol presented in [6]. The p-persistent protocol differs from the standard one in the way backoff intervals are selected. Specifically, after the channel is free for a DIFS interval, each station having frames in the sending queue starts a transmission in the next slot with probability p, and defers to the following slot with probability $1-p$. The value of p depends on the number of active stations and is chosen based on the following line of reasoning. When the standard 802.11 protocol is used, each station experiences several backoff intervals before transmitting a frame successfully. If stations operate in asymptotic conditions, it is possible to derive an average value, say $E\left[Bk\right]$, for the length of these backoff intervals [6]. The value of p is chosen in such a way that the average backoff interval obtained by using the p-persistent protocol is equal to $E\left[Bk\right]$, i.e. $p = 1/(E\left[Bk\right]+1)$ [6]. Assuming the p-persistent protocol yields to closely approximate the channel occupation resulting from the activity of R asymptotic stations, when $R \gg 1$ holds [6]. Hence the p-persistent protocol can be used to characterize the background traffic. However, it does not provide accurate results with respect to the average MAC delay experienced by a tagged station (i.e., the average time required by that station to start a successful transmission). Based on these observations, in our hotspot model the PSM station uses the standard IEEE 802.11 protocol, since its energy consumption significantly depends on the MAC delay it experiences (see below). However, to model the impact on the PSM-station energy consumption of background stations, it is sufficient considering the channel occupation resulting from their activity. Therefore, we assume that background stations use the p-persistent protocol, as it greatly simplifies the analysis. To point out the dependence of p on the number of active stations ($M+1$ in our case), we hereafter refer to p as p_M.

Finally, we neglect frame disruptions due to transient channel fading and interference. Furthermore, we assume that frames sent by the PSM station get never lost, i.e., they are successfully delivered within the maximum number of (re-)transmissions allowed by the MAC protocol. This assumption relies on the the retransmission policy of the 802.11 MAC protocol, that makes the data link service quasi-reliable. To corroborate this hypothesis, let us define p_{loss} as the probability that a PSM-station frame is discarded after being (re-)transmitted for the maximum number of times. Figure 2 plots p_{loss} as a function of the number of active stations in the hotspot, i.e., M. In this figure we show the results obtained from the analytical expression of p_{loss} provided in Section 3.2, and from the simulation model used to validate our analysis (see Section 4 for details).

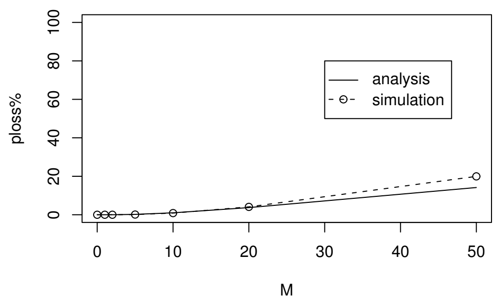

Figure 2. p_{loss} as a function of M.

3.2 Modeling the time spent in the active mode

According to the assumption of using a TCP/IP architecture, the traffic on the WLAN related to the PSM station results from three components: i) the TCP segments[6] coming from the server; ii) the TCP acks sent to the server; and iii) the Beacon frames sent by the Access Point. Thus, T_{ac} is the time spent in the active mode by the PSM station to handle these traffic components. Therefore, in the following we separately focus on each component, highlighting its impact on T_{ac}. Based on this analysis we then derive a closed-form expression for $E\left[T_{ac}\right]$.

[6] For the sake of simplicity, we indicate TCP segments containing application data as TCP segments, while TCP acks denotes TCP segments containing just acknowledgments.

First of all, let us analyze the impact of beaconing. In our model we assume that Beacon frames are always sent successfully at the beginning of the Beacon Interval. In other words, we assume that at the beginning of each Beacon Interval every station freezes any operation, and waits for receiving a Beacon frame. This conforms to the most recent IEEE 802.11 specifications. Based on this assumption, Lemma 1 holds.

Lemma 1. *Let*
- $PIFS$ *be the length in time of a PIFS interval;*
- τ *be the propagation delay between the PSM station and the Access Point;*
- $phyHdrSz$ *be the size in bits of the physical-level header of an 802.11 frame;*
- $phyR$ *and* $baseR$ *be the rate at which the physical and MAC-level headers are transmitted, respectively;*
- $beacSz$ *be the size in bits of a Beacon frame, excluding the physical header.*

Then the time during which the PSM station is active to receive a Beacon frame is

$$T_b = PIFS + \tau + \frac{phyHdrSz}{phyR} + \frac{beacSz}{baseR} \ . \tag{2}$$

It is worth noting that, in principle, $beacSz$ should be variable, as the length of the TIM is variable (see [10]). However, in our model only the tagged station uses the PSM, and hence both the length of the TIM and the size of the Beacon frame are constant. Therefore, T_b is assumed to be a constant term. T_b is not the only component of T_{ac} related to beaconing. Specifically, in our model we assume that, at the end of each Beacon Interval, the PSM station is sleeping. Thus, to resume the wireless interface to receive the next Beacon frame, there is a transition time from the sleep to the active mode (T_{sa}). Hence, $T_{sa} + T_b$ represents the impact on T_{ac} of each Beacon frame.

Let us now focus on the impact of the TCP traffic. To this end, we have to characterize the following quantities:
- N_{seg} (N_{ack}) denotes a random variable measuring the number of TCP segments received (TCP acks sent) by the PSM station during the observation interval;
- $\{T_i^{seg}\}_{i=1,\dots,N_{seg}}$ is a set of random variables where T_i^{seg} measures the time interval spent in the active mode by the PSM station to download the i-th TCP segment from the Access Point;
- $\left\{T_j^{ack}\right\}_{j=1,\dots,N_{ack}}$ is a set of random variables where T_j^{ack} measures the time interval spent in the active mode by the PSM station to upload the i-th TCP ack to the Access Point.

The PSM station downloads each TCP segment from the Access Point inside a distinct Data frame. More precisely, such downloads occur by exchanging (between the Access Point and the PSM station) a sequence of frames composed by a PS-Poll, Data, and Ack frame, as shown in Figure 1-right. Similarly, each TCP ack is uploaded to the Access Point inside a distinct Data frame. In this case i) the PSM station transmits the Data frame by using the standard DCF procedure, and ii) after a SIFS interval the Access Point transmits the corresponding Ack frame. Therefore, T_i^{seg} starts when the PSM station invokes the DCF procedure for the first transmission attempt of the i-th PS-Poll, and finishes when the PSM station has received the corresponding Ack frame. On the other hand, T_j^{ack} starts when the PSM station invokes the DCF procedure for the first transmission attempt of the Data frame containing the j-th TCP ack, and finishes when the PSM station has received the corresponding Ack frame. Based on these observations, and by recalling the way TCP and PSM work, we can prove the following proposition.

Proposition 2. *For each i and for each j, both* T_i^{seg} *and* T_j^{ack} *start at the beginning of the first free slot next to the successful delivery of a frame.*

Proposition 2 allows us to prove a fundamental property of our system, provided by Lemma 2. Lemma 2 can be obtained by recalling that i) T_i^{seg} and T_j^{ack} never overlap, since in both cases the first frame of the sequence is sent by the PSM station; ii) the beginning of each T_i^{seg} and each T_j^{ack} is a regeneration point for the tagged-station (standard 802.11) MAC protocol; and iii) the beginning of each free slot is a regeneration point for the p-persistent 802.11 MAC protocol [6].

Lemma 2. *The sets* $\{T_i^{seg}\}_i$ *and* $\left\{T_j^{ack}\right\}_j$ *are composed by i.i.d. random variables*[7]*. Furthermore, for each couple* (i, j), T_i^{seg} *and* T_j^{ack} *are independent random variables.*

[7] For ease of reading, we use the notation $\left\{T_i^{seg}\right\}_i$ instead of $\left\{T_i^{seg}\right\}_{i=1,\dots,N_{seg}}$ when the range of the index – i in this case – can be trivially derived.

Based on Lemma 2, we can characterize the sets $\{T_i^{seg}\}_i$ and $\{T_j^{ack}\}_j$ by focusing on a single element of each set, throughout referred to as T^{seg} and T^{ack}, respectively. Both T^{seg} and T^{ack} are made up of two components: i) the MAC delay experienced before the successful delivery of the first frame in the respective sequence (i.e., either the PS-Poll or the Data frame containing the TCP ack); and ii) the time required to transmit the frame sequence. Therefore, the following propositions hold.

Proposition 3. *Let*
- T^{MAC} *be a random variable measuring the MAC delay experienced by the PSM station*[8]*;*
- $TcpSegSz$ *be the size in bits of the (tagged) TCP segment;*
- $dataR$ *be the rate used to transmit the DATA field of the Data frame;*
- O^{dl} *be the overhead in time introduced by the MAC protocol to deliver the frames in the PS-Poll/Data/Ack sequence*[9]*;*

then: T^{seg} *is equal to* $T^{MAC} + 3 \cdot \tau + O^{dl} + TcpSegSz/\, dataR$.

Proposition 4. *Let*
- $TcpAckSz$ *be the size in bits of the (tagged) TCP ack;*
- O^{ul} *be the overhead in time introduced by the MAC protocol to deliver the frames in the Data/Ack sequence;*

then: T^{ack} *is equal to* $T^{MAC} + 2 \cdot \tau + O^{ul} + TcpAckSz/\, dataR$.

In our model we assume that both TCP segments and TCP acks have fixed sizes. Hence, all the T^{seg} and T^{ack} components, but T^{MAC}, are constant terms. The last step required to characterize T^{seg} and T^{ack} is thus the analysis of T^{MAC}. For ease of reading, we just provide the closed-form expression of $E\left[T^{MAC}\right]$ in Lemma 3. The complete analysis is postponed to the Appendix. Before deriving $E\left[T^{MAC}\right]$ we need to introduce the following definitions.

- *Equivalent-slot time* (t_{esl}). The time required by the PSM station to decrement the backoff counter by one during the backoff procedure. When no other stations are active t_{esl} is equal to the length of a slot (throughout referred to as t_{sl}). Otherwise it increases, due to transmissions of background stations that freeze the backoff procedure (a closed-form expression for $E\left[t_{sl}^{e}\right]$ is derived in the Appendix).
- *Collision time* (T^{coll}). The time during which the channel is busy when a frame sent by the PSM station undergoes collision.
- *Retransmission limit* (MAX). The maximum number of (re-)transmissions allowed by the standard 802.11 MAC protocol before discarding a frame.
- *Free probability* (p_f). The probability that no other stations transmit in the same time slot used by the PSM station to start a transmission, i.e., $p_f = (1 - p_M)^M$.
- *Loss probability* (p_{loss}). The probability that a frame is discarded after being (re-)transmitted MAX times without success, i.e., $p_{loss} = (1 - p_f)^{MAX}$.

Lemma 3. *Let*
- $DIFS$ *be the length in time of a DIFS interval;*
- $S_N(q)$ *and* $Q_N(q)$ *be the well-known closed-form expressions for the sequences* $\sum_{i=0}^{N} q^i$ *and* $\sum_{i=0}^{N} i \cdot q^i$*, respectively;*
- CW_{min} *be the minimum congestion-window size allowed by the standard IEEE 802.11 MAC protocol, measured in number of slots.*

Then the average value of T^{MAC} *is*

$$
\begin{aligned}
E\left[T^{MAC}\right] = DIFS + \frac{p_f}{1 - p_{loss}} \cdot \Big\{ & E\left[T^{coll}\right] \cdot Q_{MAX-1}(1 - p_f) + \\
& + \frac{E\left[t_{esl}\right] \cdot CW_{min}}{2} \cdot \left(2 \cdot S_{MAX-1}(2 - 2 \cdot p_f) - S_{MAX-1}(1 - p_f)\right) + \\
& + \frac{E\left[t_{esl}\right]}{2} \cdot \left(Q_{MAX-1}(1 - p_f) - S_{MAX-1}(1 - p_f)\right) \Big\} \;.
\end{aligned}
\tag{3}
$$

[8] More precisely, the MAC delays should be indicated by means of two sets of random variables, i.e. $\left\{T_i^{MAC}\right\}_i$ and $\left\{T_j^{MAC}\right\}_j$. However, these random variables are i.i.d., and hence we characterize them by using a single element of whatever set, throughout referred to as T^{MAC}.

[9] O^{dl} includes the (physical, MAC) headers' and trailers' transmission times, and the SIFS intervals.

The results obtained so far are the building blocks that we need to characterize T_{ac}. To complete the analysis, it is worth introducing some modeling assumptions about the reference scenario. As said in Section 2, the mobile device downloads B bytes from the server, interleaving chunks and idle phases. Chunk sizes and idle-phase lengths are assumed to be i.i.d. random variables[10]. Furthermore, the download is supported by a persistent TCP connection. Hereafter, we assume that the connection is in steady state, and we neglect the slow-start phase. Therefore, we can avoid modeling the TCP mechanisms in detail (e.g., slow start, fast retransmit, etc.), and this greatly simplifies the analysis[11]. Finally, we assume that the network status is stationary during the whole data download, i.e., the Internet throughput statistics do not change over time.

To evaluate the average value of T_{ac}, we can use the following line of reasoning. We replicate K times the download of B bytes. During the l-th replication, the PSM station downloads $B^{(l)}$ bytes, and remains in the active mode for $T_{ac}^{(l)}$ seconds. Based on Lemma 2, and on the above assumptions, we can prove that the sets $\{B^{(l)}\}_{l=1,\ldots,K}$ and $\left\{T_{ac}^{(l)}\right\}_{l=1,\ldots,K}$ are composed by i.i.d. random variables. Thus, to characterize T_{ac}, we focus on a single element of the set $\left\{T_{ac}^{(l)}\right\}_l$, throughout referred to as $\hat{T}_{ac}$. The average value of $\hat{T}_{ac}$ provides a closed-form expression for $E\left[T_{ac}\right]$. By recalling that i) for each Beacon frame the PSM station remains active for $T_{sa}+T_b$ seconds; and that ii) T_b, T_i^{seg} and T_j^{ack} never overlap, we can express $\hat{T}_{ac}$ as follows:

$$\hat{T}_{ac}=\sum_{i=1}^{\hat{N}_{seg}} T_i^{seg}+\sum_{j=1}^{\hat{N}_{ack}} T_j^{ack}+\hat{N}_b\cdot\left(T_{sa}+T_b\right) \ , \quad (4)$$

where $\hat{N}_b$, $\hat{N}_{seg}$ and $\hat{N}_{ack}$ are the number of Beacon Intervals, TCP segments and TCP acks during the tagged replication, respectively. It can be proved that also the sets $\left\{N_{seg}^{(l)}\right\}_l$, $\left\{N_{ack}^{(l)}\right\}_l$ and $\left\{N_b^{(l)}\right\}_l$ are composed by i.i.d. random variables. Furthermore, based on Lemma 2, and by assuming that, $\hat{N}_{seg}$ ($\hat{N}_{ack}$) and the elements of $\{T_i^{seg}\}_{i=1,\ldots,\hat{N}_{seg}}$ ($\{T_j^{ack}\}_{j=1,\ldots,\hat{N}_{ack}}$) are independent random variables, we can derive $E\left[T_{ac}\right]$ as shown in Theorem 1.

Theorem 1. *The average value of* T_{ac} *is*

$$E\left[T_{ac}\right]=E\left[N_{seg}\right]\cdot E\left[T^{seg}\right]+E\left[N_{ack}\right]\cdot E\left[T^{ack}\right]+E\left[N_b\right]\cdot\left(T_{sa}+T_b\right) \ . \quad (5)$$

To complete the analysis of T_{ac} we have to derive the unknowns in Equation 5, i.e., $E\left[N_{seg}\right]$, $E\left[N_{ack}\right]$ and $E\left[N_b\right]$. To this end, we assume that the TCP-sender retransmits only lost segments (i.e., the PSM station does not receive duplicated TCP segments). Furthermore, as discussed in Section 3.1, the PSM station transmits its frames successfully within MAX attempts. Therefore, if $\hat{B}$ denotes the number of bytes downloaded during the tagged replication, $\hat{N}_{seg}$ is the minimum number of segments required to transfer $\hat{B}$ bytes, i.e., $\hat{N}_{seg}=\left\lceil \hat{B}\middle/MSS\right\rceil$, where MSS is the Maximum Segment Size of the TCP connection [14]. Furthermore, since we have assumed TCP-Reno without delayed acks, $\hat{N}_{ack}$ is equal to $\hat{N}_{seg}$, and thus $E\left[N_{seg}\right]=E\left[N_{ack}\right]=\left\lceil E\left[B\right]/MSS\right\rceil$ holds. Finally, $\hat{N}_b$ can be evaluated as the number of Beacon Intervals occurring during the tagged replication, i.e. $\hat{N}_b=\left\lceil \hat{T}\middle/BI\right\rceil$, where BI is the length in time of a Beacon Interval, and $\hat{T}$ is the observation interval related to the tagged replication. It can be proved that, if $T^{(l)}$ measures the observation interval related to the l-th replication, the set $\{T^{(l)}\}_l$ is composed by i.i.d. random variables. Therefore, $E\left[N_b\right]=\left\lceil E\left[T\right]/BI\right\rceil$ holds.

3.3 Modeling the observation interval

To evaluate $E\left[T\right]$ we exploit the same line of reasoning used to derive $E\left[T_{ac}\right]$. Specifically, we have shown that, if we replicate the download of B bytes for K times, the elements of the set $\{T^{(l)}\}_l$ are i.i.d. random variables. Thus, we can focus on a single element in the set $\{T^{(l)}\}_l$, throughout referred to as $\hat{T}$. The average value of $\hat{T}$ provides a closed-form expression for $E\left[T\right]$. Hereafter, $\hat{B}$ denotes how

[10] For example, in the Web case this means assuming that Web-page sizes and time intervals during which the user reads the page contents are i.i.d. random variables.

[11] However, as shown in Section 4, the model accuracy is not significantly affected by this approximation.

many bytes are downloaded during that replication. As shown in Figure 1-left, $\hat{T}$ is composed by two factors: (i) the total time during which chunks are downloaded (T_{data}), and (ii) the total time spent in the idle phases (T_{idle}). As discussed in Section 3.2, in our model chunks are downloaded over a single steady-state TCP-connection, and the Internet throughput statistics do not change over time. Hence, the average time required to transfer the $\hat{B}$ bytes over the TCP connection (i.e., $E[T_{data}]$) can be reasonably approximated by $E[B]/\gamma_{TCP}$, where γ_{TCP} is the throughput achieved by the server-client connection. Furthermore, if $N_I^{(l)}$ is a random variable measuring the number of idle phases occurring during the l-th replication, we can prove that the set $\left\{N_I^{(l)}\right\}_l$ is composed by i.i.d. random variables. Moreover, let i) $\hat{N}_I$ be the number of idle-phases occurring during the tagged replication; and ii) $\{idle_i\}_{i=1,\dots,\hat{N}_I}$ be a set of random variables, where $idle_i$ measures the length of the i-th idle phase within the tagged replication. If we assume that, for each i, $\hat{N}_I$ and $idle_i$ are independent random variables, Theorem 2 holds.

Theorem 2. *The average value of T is:*

$$E[T] = E\left[\hat{T}\right] = \frac{E[B]}{\gamma_{TCP}} + E[N_I] \cdot E[idle] \ . \tag{6}$$

4 Model Validation and Results

The validation of our analysis is carried out by comparing the results obtained from the formulas derived so far with the output of a simulation model. Our simulator is an extension of the simulator used in [5] and implements the networking environment described in Section 2. Specifically, stations in the hotspot adopt the standard IEEE 802.11 protocol. Furthermore, the TCP modules at the PSM station and at the fixed server are compliant with the TCP-Reno specifications. The (wired) Internet is modeled by introducing (between the Access Point and the server) exponentially-distributed delays. TCP segments are lost with a probability p_l, representing packets' drops at intermediate routers. Finally, as far as the application traffic, chunk sizes and idle-phase lengths are drawn from the distributions used in [7] to model the Web traffic. Hence, we test our model in the case of Web access from a mobile device. We performed different simulation runs by varying M between 0 and 50. For each M value, we replicated the simulation run for 10 times, and evaluated the confidence interval (with 95% confidence level).

First of all, in the following we validate $E\left[T^{MAC}\right]$. The effects of congestions in the WLAN manifest themselves as an increased T^{MAC}. Hence, this performance figure represents a key point to evaluate the scalability properties of 802.11 PSM. Both the analytical and the simulation results are plotted in Figure 3-left. As expected, $E\left[T^{MAC}\right]$ increases as the load on the WLAN increases. Specifically, $E\left[T^{MAC}\right]$ ranges from about 350 μsec to about 170 $msec$, when M increases from 0 to 50. A very interesting point is the intersection between $E\left[T^{MAC}\right]$ and the Beacon Interval length (the dash-dotted line at 100 $msec$). This point can be seen as a limit for using the 802.11 PSM mechanism. Beyond this point the average time required to access the channel is greater than the Beacon Interval length (BI). Therefore, when data are buffered at the Access Point the PSM station can never switch to the sleep mode, since $E\left[T^{MAC}\right]$ is greater than BI. The PSM is useless, and thus it wastes energy with respect to the standard 802.11 MAC protocol. In this case, PSM saves energy only during idle phases. However, as idle phases are related to the user behavior, power-saving policies exploiting some knowledge about the application behavior are likely to be more effective than PSM, as shown in [2, 1].

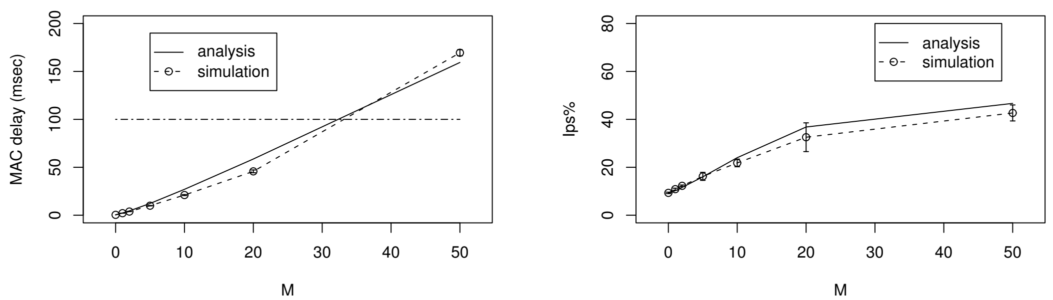

Figure 3. $E\left[T^{MAC}\right]$ (left) and I_{ps} (right) as functions of M.

Figure 3-right allows us to quantify the PSM performance in terms of energy savings. To this end, we define the I_{ps} index as the ratio between the average value of E_{PSM} (Equation 1), and the energy spent to download the same data without the PSM. By definition, I_{ps} represents the percentage of energy spent by using the 802.11 PSM, with respect to the case without energy management. The latter quantity can be approximated[12] by $E[T] \cdot P_{ac}$. Figure 3-right confirms the results reported in [11] (I_{ps} around 10% when $M = 0$), and shows a graceful degradation in the observed region (I_{ps} around 45% when $M = 50$). The asymptotic behavior of I_{ps} when M increases is somewhat counter-intuitive, as one could expect that performance decreases when the number of stations in the Wi-Fi hotspot increases. However, when the congestion level increases, T^{MAC} increases, either the PSM is used or not. Therefore, the more M increases, the more T^{MAC} impacts on the energy spent in both cases, and this explains the I_{ps} profile.

From the above analysis it results that, though the 802.11 PSM shows good performances in terms of energy savings, it presents several weaknesses. Currently, we are investigating the limitations of this power-saving algorithm, and we are studying energy management policies aimed at improving its performance.

References

1. M. Anand, E. Nightingale, and J. Flinn, "Self-Tuning Wireless Network Power Management", in Proc. of the Ninth ACM Annual International Conference on Mobile Computing and Networking (MobiCom 2003), San Diego (CA), Sept. 14-19, 2003.
2. G. Anastasi, M. Conti, E. Gregori and A. Passarella, "Performance Comparison of Power Saving Strategies for Mobile Web Access", Performance Evaluation, Vol. 53, N. 3-4, pp. 273-294, August 2003.
3. G. Anastasi, M. Conti, E. Gregori and A. Passarella, "Saving Energy in Wi-Fi Hotspots through 802.11 PSM: an Analytical Model (extended version)", Tech. Rep. DII03-11, Dept. of Information Engineering, University of Pisa, Nov. 2003 (available at http://www.ing.unipi.it/~o783499/research/docs/WiOpt04_ext.pdf).
4. H. Balakrishnan, V.N. Padmanabhan, S. Seshan and R.H. Katz, "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links", IEEE Transactions on Networking, Vol. 5, N. 6, pp. 756-769, Dec. 1997.
5. R. Bruno, M. Conti and E. Gregori, "Optimization of efficiency and energy consumption in p-persistent CSMA-based wireless LANs", IEEE Transactions on Mobile Computing, Vol. 1, No. 1, pp. 10-31, Jan.-Mar. 2002.
6. F. Calì, M. Conti and E. Gregori, "IEEE 802.11 wireless LAN: capacity analysis and protocol enhancement", IEEE Transactions on Networking, Vol. 8, No. 6, pp. 785-799, Dec. 2000.
7. M.Crovella e A.Bestavros, "Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes", IEEE Transaction on Networking, Vol. 5, No. 6, pp. 835-846, Dec. 1997.
8. J.P. Ebert, B. Stremmel, E. Wiederhold and A. Wolisz, "An energy-efficient power control approach for WLANs", Journal of Communications and Networks (JCN), Vol. 2, No. 3, pp. 197-206, pp. 197-206, 2000.
9. L.M. Feeney and M. Nilsson, "Investigating the energy consumption of a wireless network interface in an ad hoc networking environment", Proc. of the 20^{th} Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2001), Anchorage (Alaska) Apr. 22-26, 2001.
10. IEEE standard for Wireless LAN- Medium Access Control and Physical Layer Specification, 802.11, November 1997.
11. R. Krashinsky and H. Balakrishnan, "Minimizing Energy for Wireless Web Access with Bounded Slowdown", Proceedings of the Eighth Annual ACME/IEEE International Conference on Mobile Computing and Networking (MobiCom 2002), pp.119-130, Atlanta (GA), Sept. 2002.
12. R. Kravets e P.Krishnan, "Power Management Techniques for Mobile Communication", Proceedings of the Fourth Annual ACME/IEEE International Conference on Mobile Computing and Networking (MobiCom 1998), pp. 157-168, Dallas (TX), 1998.
13. T. Starnar, "Powerful Change Part I: Batteries and Possible Alternatives for the Mobile Market", IEEE Pervasive Computing, Vol. 2, No. 4, pp. 86-88, Oct.-Dec. 2003.
14. W.R. Stevens, "TCP/IP Illustrated, Volume 1: The Protocols" , Addison-Wesley, 1994.

Appendix: characterization of T^{MAC}.

The distribution law of T^{MAC} is provided by Lemma 4, based on this line of reasoning. As T^{MAC} starts after the successful delivery of a frame sequence (see Proposition 2), the initial part of T^{MAC} is a DIFS interval during which the channel is free. Then, the backoff procedure starts with the congestion-window size set at the minimum value. Furthermore, collisions of PSM-station frames, and backoff-procedure freezing due to transmissions of background stations, are managed as defined in [10]. Finally, in our model the PSM station sends its frames successfully within MAX attempts.

Lemma 4. *Let*

[12] More precisely, due to PSM delays, T overestimates the time spent in the active mode when PSM is not used. This approximation leads to overestimating the energy spent without PSM.

– $T^{bo}(CW)$ *be the time elapsed to complete the backoff procedure when the congestion-window size is* CW*;*

Then T^{MAC} *is distributed according to the following law:*

$$T^{MAC} = DIFS + \sum_{j=1}^{i+1} T^{bo}(CW_j) + \sum_{j=1}^{i} T_j^{coll} \qquad \frac{(1-p_f)^i \cdot p_f}{1-p_{loss}}, \quad i = 0, \ldots, MAX-1 \; . \tag{7}$$

It should be noted that $T^{bo}(CW)$ also includes the transmission times of background stations, either in the case of successful transmissions, or in the case of collisions. The closed-form expression of Equation 3 is obtained from Equation 7 by characterizing T^{coll} and $T^{bo}(CW)$. These characterizations are provided by Lemmas 6 and 7. As Lemma 6 requires the closed-form expression of $E[t_{esl}]$, we firstly analyze this quantity. The equivalent-slot time (t_{esl}) is the time required by the PSM station to decrement the backoff counter by one. Before decrementing the backoff counter, several transmissions of background stations may occur, either successful or not. If N_t is a random variable measuring the number of these transmissions, then t_{esl} is equal to $\sum_{s=1}^{N_t} T_w^{(s)} + t_{sl}$, where $T_w^{(s)}$ is a random variable measuring the time interval during which the PSM-station backoff procedure remains frozen due to the s-th transmission[13]. Furthermore, it is possible to show that N_t and the elements of $\left\{T_w^{(s)}\right\}_s$ are independent random variables. Therefore, we can derive $E[t_{esl}]$ as shown in the following lemma.

Lemma 5. *The average value of the equivalent-slot time is*

$$E[t_{esl}] = E[N_t] \cdot E[T_w] + t_{sl} = \frac{1-p_f}{p_f} \cdot E[T_w] + t_{sl} \; .$$

Furthermore, if we assume that background stations transmit fixed-size Data frames, and FS *denotes the length in bits of these frames,* T_w *is distributed according to the following law:*

$$T_w = \begin{cases} 2 \cdot \tau + 2 \cdot \frac{phyHdrSz}{phyR} + \frac{macHdrSz + ackSz}{baseR} + \frac{FS}{dataR} + SIFS + DIFS & (1-p_M)^{M-1} \\ \tau + \frac{phyHdrSz}{phyR} + \frac{macHdrSz}{baseR} + \frac{FS}{dataR} + DIFS & 1-(1-p_M)^{M-1} \end{cases} ,$$

where $macHdrSz$ *and* $ackSz$ *are the sizes in bits of a MAC-level header and an Ack frame, respectively, and* $SIFS$ *is the length in time of a SIFS interval.*

Based on Lemma 5 we are now in the position of deriving a closed-form expression for $E\left[T^{bo}(CW)\right]$.

Lemma 6. *Let*

– $X(CW)$ *be the value chosen by the PSM station as the backoff interval when the congestion-window size is* CW*;*

Then $T^{bo}(CW)$ *can be expressed as* $T^{bo}(CW) = \sum_{i=1}^{X(CW)} t_{esl}^{(i)}$*. Furthermore, since* $X(CW)$ *and* t_{esl} *are independent random variables, the average value of* $T^{bo}(CW)$ *is*

$$E\left[T^{bo}(CW)\right] = E[X(CW)] \cdot E[t_{esl}] = \frac{CW-1}{2} \cdot E[t_{esl}] \; .$$

Finally, Lemma 7 provides a closed-form expression for $E\left[T^{coll}\right]$.

Lemma 7. $E\left[T^{coll}\right]$ *can be expressed as*

$$E\left[T^{coll}\right] = \tau + \frac{phyHdrSz}{phyR} + \frac{macHdrSz}{baseR} + \frac{FS}{dataR} + DIFS \; .$$

[13] $T_w^{(s)}$ also includes the DIFS interval during which the channel remains free after the transmission has finished.