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

G. Anastasi*, M. Conti•, E. Gregori•, A. Passarella*

*University of Pisa, Dept. of Information Engineering
Via Diotisalvi, 2 - 56122 Pisa, Italy
{g.anastasi, a.passarella}@iet.unipi.it}

•CNR - IIT Institute
Via G. Moruzzi, 1 - 56124 Pisa, Italy
{marco.conti, enrico.gregori}@iit.cnr.it}

Abstract. Wi-Fi hotspots are becoming very popular. Due to limited costs, they are an optimal 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 resoureces 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 the main factors that impact energy consumption, and investigate their 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. Thus, our analysis can be used as a bottom-line to enhance energy management beyond the current limitations of the 802.11 PSM.

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 the IEEE 802.11 standard [8]. In the last few years the cost of 802.11 Access Points and network cards has 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 ([4, 5, 9, 11]).

Energy management for the mobile-device networking subsystem deserves special attention ([2, 4, 7, 9, 10, 14]). It has been widely shown that networking activities account for 10% up to 50% of the energy spent by a mobile device ([10]). Since the difference between batteries' capacity and mobile-device energetic requirements is becoming more and more broad ([12]), it is vital to design network architectures that optimize 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 (see [8] for a detailed description of the PSM mechanisms). Surprisingly, to our knowledge very little effort has been put on evaluating the performance of PSM in a Wi-Fi hotspot scenario [9, 1]. Both papers consider a single user inside the hotspot. The behavior of PSM with respect to 802.11 without energy management is investigated by means of simulation [9], and experiments [1]. [9] focuses on Web applications, and shows that PSM energy saving can be as big as 90%. However, significant delays are introduced to frames addressed to the mobile device. In [1] the hotspot is seen as a mere extension of the user corporate LAN, and the reference application is the access to files shared via NFS. It is shown that in this case too significant delays are introduced. Furthermore, the energy spent by the whole device can even increase when PSM is activated. Though [9] and [1] highlight some limitations of 802.11 PSM, none of them provides an extensive characterization of this power-saving algorithm. Specifically, their analysis is 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 remarks have motivated us to achieve a more comprehensive understanding of 802.11 PSM performance. In this paper we derive an analytical model of the PSM behavior and validate it by simulation. With respect to energy saving, our results are aligned with those reported in [9]. Furthermore, our model highlights the dependence of the PSM performance on key parameters, such as the throughput, the traffic profile, the MAC-protocol parameters, etc. For example, in this paper we focus on scalability issues. 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 ceases to be effective. In conclusion, our model contributes to an extensive characterization of 802.11 PSM. This should be the first step 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. This choice relies on the observation that these applications (e.g., Web, e-mail, file transfer, ...) are now the most widely used in Wi-Fi hotspots. In detail, our reference scenario is as follows. A tagged PSM-station in a Wi-Fi hotspot 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 idle phases no traffic flows between the server and the client. Furthermore, the PSM station uses a standard TCP/IP architecture, and the whole download is supported by a unique TCP-connection[1]. We assume that the TCP connection is in its steady state. 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 have always a frame ready to be sent.

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 a steady-state TCP-connection avoids modeling the TCP mechanisms in detail, thus simplifying the analysis. However, the model accuracy is not affected by this approximation (see Section 4). 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 [3]). However, TCP/IP is currently the only off-the-shelf solution that can be used to deploy a Wi-Fi hotspot. From this standpoint, our work analyzes the performance and limitations of the currently available technology, and can thus be used as a bottom-line to evaluate novel solutions.

3 Analytical Model of the 802.11 PSM

Mettere una frase per dire che si assume la conoscenza dell'algoritmo PSM riportato in appendice. Le proof le lasciamo in un tech.rep. Dopo introdurre solo la notazione per il teorema 1.

Before proceeding, it is worth recalling the strategy implemented by the 802.11 PSM policy. The 802.11 defines two operating modes for wireless interfaces, i.e., the active mode and the sleep mode. In the active mode, interfaces are able to send and receive data, while in sleep mode they are not. Moreover, due to the CSMA/CA MAC-protocol, the power required in the active mode is almost the same, whether the interface is transmitting, receiving, or idle. On the other hand, the power required in the sleep mode is at least one order of magnitude below [9]. Therefore, the aim of the PSM strategy is letting the PSM station in the sleep mode[2] whenever it has no data to exchange on the WLAN.

[1] We assumed TCP-Reno, without delayed ACKs.

[2] Since we are interested in the energy consumption related to the wireless interface, 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.

According to the reference scenario, we have to evaluate the energy spent by a PSM station that downloads B bytes from a fixed server. To this end, let us introduce some notation used throughout the paper. Hereafter, this quantity is referred to as E_{PSM}. Moreover, i) T denotes the time required to download the B bytes, ii) T_{ac} and T_{sleep} denote the total times during which the PSM station is in the active

An initial formula for E_{PSM} is provided by Theorem

Theorem 1: The energy spent by the PSM station to download the B byte from the server is

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

where: i) T_{ac} and T_{sleep} are the time intervals during

As a first step, we need to evaluate the total time (T) needed to complete the download. only for a portion of T. For the rest of time the station is put in the sleep mode. Thus, the energy spent when using the PSM (E_{PSM}) can be evaluated by deriving the above two time intervals, hereafter referred to as T_{ac} and T_{sleep}, respectively. Specifically, it is

As T_{sleep} can be obtained as $T - T_{ac}$, the core of this paper consists in deriving closed formulas for T and T_{ac}.

3.1 Total length of the data download

From Section 2 it emerges that T can be seen as made up of two components: (i) the overall time during which data flows on the TCP connection (T_{data}); and (ii) the overall time spent in the idle phases (T_{idle}). Specifically, T_{data} includes the time intervals during which chunks are downloaded. Secondly, the time spent in the idle phases, named T_{idle}. As an idle phase starts when a chunk is completely downloaded, T is equal to $T_{data} + T_{idle}$. A visual scheme of these components is shown in Figure 1. Based on the above remark, the following theorem holds.

Theorem 2: Let

- γ_{TCP} be the average throughput experienced by data flowing on the TCP connection;
- N_I be the number of chunks used to download the B bytes;
- $\{idle_i\}_{i=1,\dots,N_I}$ a set of random variables, $idle_i$ measuring the length of the i-th idle phase.

Then T is equal to $B/\gamma_{TCP} + \sum_{i=1}^{N_I} idle_i$, and its average value is as follows:

$$E\left[T\right] = \frac{E\left[B\right]}{\gamma_{TCP}} + E\left[N_I\right] \cdot E\left[idle\right] \ . \tag{2}$$

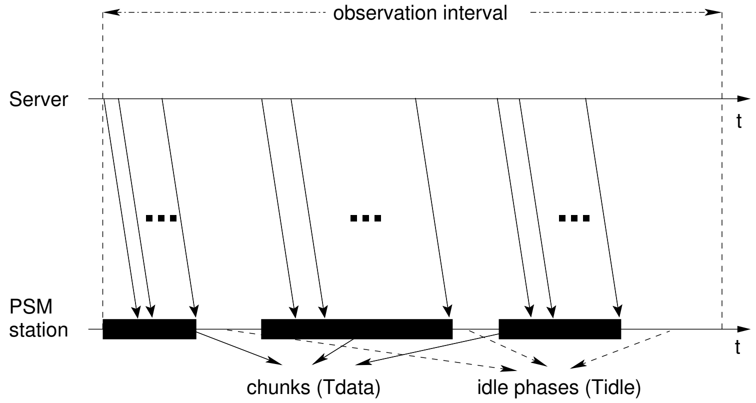

Figure 1: T as composed by T_{data} and T_{idle}.

Proof: By definition, T_{data} is equal to B/γ_{TCP}, and T_{idle} is equal to $\sum_{i=1}^{N_I} idle_i$. The expression of T immediately follows. Finally, Equation 2 can be derived by assuming that $idle_i$ are i.i.d. random variables, and are independent of N_I. ■

3.2 Time spent in the active mode

As a preliminary step, we make two assumptions that will be used during the following analysis. We assume that (i) the TCP-sender retransmits only segments that have been actually lost, and (ii) DATA frames exchanged between the Access Point and the PSM-station get never lost. The latter assumption relies on the the retransmission policy of the 802.11 MAC, that makes the data link service quasi-reliable. Figure 2-right plots p_{loss} as a function of the number of active stations in the hotspot, i.e., M. In this figure we present the results achieved from both the analytical and the simulation models, as explained in the rest of the paper. As it is clear, p_{loss} is negligible when the congestion in the WLAN is not very high. Therefore, based on the plot in Figure 2-right, in this paper we show the results achieved when M is below 50.

The first step of the analysis consists in evaluating how many DATA frames the PSM station exchanges with the Access Point during the total time, T. Specifically, the following lemma holds.

Lemma 1: The number of TCP segments downloaded by the PSM station from the Access Point is $N_{seg} = \lceil B/MSS \rceil$, where MSS is the Maximum Segment Size used over the TCP connection. Moreover, each TCP segment is downloaded inside a corresponding DATA frame. Finally, the PSM station sends N_{seg} TCP ACKs, each one inside a corresponding DATA frame.

Proof: As TCP tries to avoid IP fragmentation, each segment generated at the TCP-sender potentially results in a DATA frame at the Access Point. Therefore each TCP segment is downloaded in a distinct DATA frame. Moreover, based on point (i) above, the PSM never receives a duplicated TCP segment. Hence, the PSM station receives the *minimum* number of segments that are required

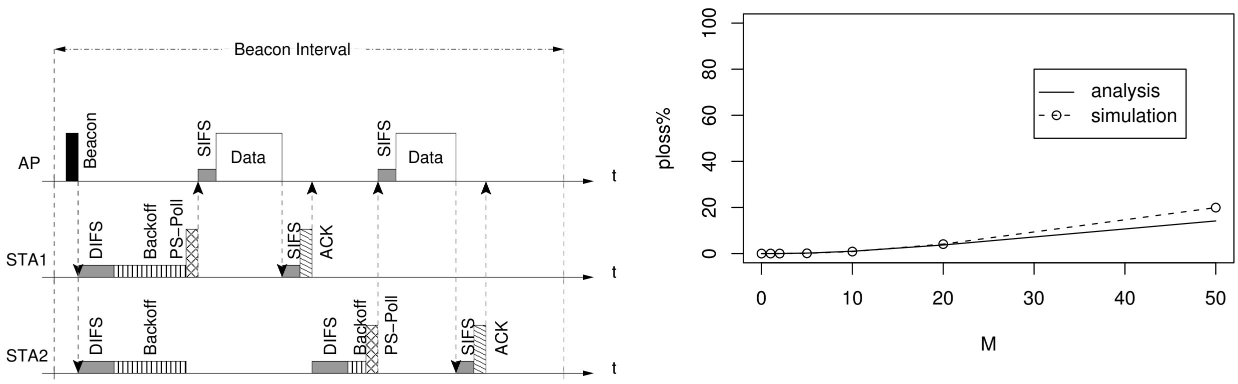

Figure 2: Example of the 802.11 PSM operations(left), and p_{loss} as a function of M (right).

to transfer B bytes over the TCP connection, i.e., $\lceil B/MSS \rceil$. Based on point (ii), $\lceil B/MSS \rceil$ is also the number of DATA frames downloaded from the Access Point, i.e., N_{seg}. Furthermore, each segment generates a TCP ACK at the PSM station, that is immediately sent towards the TCP-sender. Hence, N_{seg} is also the number of DATA frames that are uploaded by the PSM station. ■

From the 802.11 specifications [8] and Lemma 1, it emerges that each TCP segment is downloaded by means of a *distinct* PS-Poll/DATA/ACK frame sequence, as shown in Figure 2-left. Furthermore, each TCP ACK is uploaded by means of a DATA/ACK frame sequence[3]. These remarks allows to evaluate the component of T_{ac} related to the download of the TCP segments and to the upload of the TCP ACKs, hereafter referred to as T_{ac}^{TCP}. The following lemma holds.

Lemma 2: Let

- $\{T_i^{seg}\}_{i=1,\ldots,N_{seg}}$ be a set of random variables, where T_i^{seg} measures the time interval required to download the i-th TCP segment, from the instant when the corresponding PS-Poll is ready to be sent for the first time;
- $\{T_i^{ack}\}_{i=1,\ldots,N_{seg}}$ be a set of random variables, where T_i^{ack} measures the time interval required to upload the i-th TCP ACK, from the instant when the corresponding DATA frame is ready to be sent for the first time.

Then, the sets $\{T_i^{seg}\}$ and $\{T_i^{ack}\}$ are composed by i.i.d. random variables. Furthermore, T_{ac}^{TCP} is equal to $\sum_{i=1}^{N_{seg}} T_i^{seg} + \sum_{i=1}^{N_{seg}} T_i^{ack}$.

Proof: As highlighted in Section 2, the background stations generate an asymptotic traffic. Moreover, due to the 802.11 specifications [8], every time a PS-Poll is ready to be sent for the first time,

[3]The PS-Poll is sent by using the standard DCF and backoff procedures. Moreover, the DATA and the ACK frames are free from collisions, as they are sent a SIFS interval after the previous frame. Finally, the DATA frame containing a TCP ACK is sent by using the standard DCF and backoff procedures.

the PSM station behaves as follows. It sets the congestion window at its minimum value and executes the backoff procedure. Then, it executes the standard DCF access procedure. Furthermore, from [8] it emerges that the same steps are followed when a DATA frame containing a TCP ACK becomes ready to be sent for the first time. Hence, we can reasonably assume that the system is regenerative, with respect to the point in time when either a PS-Poll or a DATA frame containing a TCP ACK is ready to be sent for the first time. Therefore, we can assume that elements of both$\{T_i^{seg}\}$ and $\{T_i^{ack}\}$ are i.i.d. The expression of T_{ac}^{TCP} derives immediately from these characterizations. ■

Lemma 2 highlights the time spent in the active mode related to data generated by the application. In addition, a PSM station must wake up periodically to receive Beacons. For the sake of simplicity, in our model we assume that the PSM station wakes up once *every* Beacon Interval. It is easy to show that this assumption may result in a maximum overestimation of T_{ac} of about 1%. Finally, the contribution of "Beaconing" to T_{ac} is provided by the following lemma.

Lemma 3: Let

- N_b be the number of Beacon Intervals within the total time T;
- $\{T_i^b\}_{i=1,\dots,N_b}$ be a set of random variables, where T_i^b measures the time spent to receive the i-th Beacon, starting from the instant when the Beacon is ready to be sent;
- T_{so} be the time required by the circuitry of the wireless interface to switch from the sleep to the active mode[4].

Then N_b is equal to $\lceil T/BI \rceil$, where BI is the length in time of the Beacon Interval. The set $\{T_i^b\}$ is composed by i.i.d. random variables. Finally, the time spent in the active mode to receive Beacons is $T_{ac}^{beac} = \sum_{i=1}^{N_b} T_i^b + N_b \cdot T_{so}$.

Proof: The value of N_b can be derived immediately by definition. [8] states that, in order to broadcast a Beacon frame, the Access Point must sense the medium as free for a PIFS interval. Moreover, in our model the PSM station must switch to the active mode to receive Beacons, meaning that it is neither sending nor receiving other frames at the beginning of a Beacon Interval. Hence, only the asymptotic traffic generated by background stations is present on the WLAN at this point in time. Therefore, we can reasonably assume that the system is regenerative with respect to beginning of a Beacon Interval, and, ultimately, that the elements of $\{T_i^b\}$ are i.i.d. Finally, the form of T_{ac}^{beac} immediately derives from this characterization. ■

Based on Lemma 2 and 3 we are now in the position of deriving T_{ac}. Specifically, the following theorem holds.

[4]We assume a negligible time to perform the opposite transition.

Theorem 3: The time spent by the PSM station in the active mode is $T_{ac} = \sum_{i=1}^{N_{seg}} \left(T_i^{seg} + T_i^{ack}\right) + \sum_{i=1}^{N_b} T_i^b + N_b \cdot T_{so}$. Moreover, the average value of T_{ac} is

$$E\left[T_{ac}\right] = E\left[N_{seg}\right] \cdot \left(E\left[T^{seg}\right] + E\left[T^{ack}\right]\right) + E\left[N_b\right] \cdot \left(E\left[T^b\right] + T_{so}\right) \ , \tag{3}$$

where T^{seg}, T^{ack} and T^b belong to sets $\{T_i^{seg}\}$, $\{T_i^{ack}\}$ and $\{T_i^b\}$, respectively.

Proof: the expression of T_{ac} derives immediately from the composition of Lemma 2 and Lemma 3. Furthermore, its average value is obtained as follows.

- $\{T_i^{seg}\}$, $\{T_i^{ack}\}$ and $\{T_i^b\}$ are sets of i.i.d. random variables; hence all their elements have the same statistical properties.
- We assume that the elements of both $\{T_i^{seg}\}$ and $\{T_i^{ack}\}$ are independent of N_{seg}; this assumption derives from the regenerative properties of the system, discussed in Lemma 2.
- In the same way, based on Lemma 3, we assume that the elements of $\{T_i^b\}$ are independent of N_b.

■

The most challenging problem to derive a closed form of Equation 3 is evaluating the average values of T^{seg}, T^{ack} and T^b, since it requires an accurate modeling of the contention at the 802.11 MAC layer. With respect to T^{seg} and T^{ack}, it is worth noting that the PSM station follows almost identical steps to send either a PS-Poll and a DATA frame containing a TCP ACK. More precisely, if the MAC delay experienced by a frame is defined as the time interval elapsed from the instant when the frame is ready to be sent for the first time, up to the instant when its successful transmission starts, the following lemma holds.

Lemma 4: The MAC delays experienced by both PS-Poll frames and DATA frames are distributed according to the same law.

Proof: Whenever a PS-Poll or a DATA frame containing a TCP ACK becomes ready to be sent for the first time, the PSM station behaves as follows [8]. It (i) resets the congestion window to its minimum value, (ii) executes a backoff procedure, and (iii) follows the standard DCF method to access the shared medium. Moreover, in our model each PS-Poll and each DATA frame containing a TCP ACK are successfully transmitted within the maximum number of attempts allowed by the 802.11 MAC protocol. Therefore both those frames experience MAC delays that are distributed according to the same law. ■

Based on this result, in the following section, we derive a closed form for $E\left[T^{seg}\right]$ and $E\left[T^{ack}\right]$, by emphasizing the contribution of the MAC delay, hereafter referred to as T^{MAC}. Finally, we derive a closed form also for $E\left[T^b\right]$. By substituting these formulas in Equation 3, a closed form of $E\left[T_{ac}\right]$ can be derived . This formula and Equation 2 allow to derive a closed form of $E\left[E_{PSM}\right]$ (Equation 1).

3.2.1 Evaluation of the T^{seg} and T^{ack}

From an analytical standpoint, the most challenging component of T^{seg} and T^{ack} is T^{MAC}. Furthermore, T^{seg} and T^{ack} are composed by additional factors, as shown in the following propositions.

Proposition 1: T^{seg} is equal to $T^{MAC} + 3 \cdot \tau + O^{dl} + frSz/\, dataR$, where τ is the propagation delay between the PSM station and the Access Point, $frSz$ is the size of the DATA field of the DATA frame, $dataR$ is the rate used to transmit this field, and O^{dl} is the overhead in time related to the MAC protocol. Specifically, $O^{dl}$is a follows:

$$O^{dl} = 3 \cdot \frac{phyHdrSz}{phyR} + \frac{pollSz + macHdrSz + ackSz}{baseR} + \frac{fcsSz}{dataR} + 2 \cdot SIFS\ .$$

Proof: T^{seg} is defined as the time required to exchange a PS-Poll/DATA/ACK frame sequence, as shown in Figure 2-left. After the PSM station has gained access to the medium, those frames are exchanged based on the 802.11 MAC and PHY layer protocols [8]. O^{dl} groups all constant terms involved, and its form can be derived as follows. The sequence consists in three frames, resulting in three PHY headers ($phyHdrSz$) being sent at the PHY rate ($phyR$). The PS-Poll and the ACK frames do not contain DATA fields, and they consist only in the MAC header and the FCS field. As such, they are sent at the rate used to send the headers of MAC frames ($baseR$). Obviously, the same rate is also used for the MAC header of the DATA frame. Furthermore, the FCS field of the DATA frame is sent at the same rate used for the DATA field ($dataR$). Finally, as shown in Figure 2-left, the DATA and ACK frames are spaced by SIFS intervals from the previous frames. In conclusion, the expression of T^{seg} immediately follows from the definition of T^{MAC}, τ, O^{dl}, $frSz$ and $dataR$. ■

Proposition 2: T^{ack} is equal to $T^{MAC} + 2 \cdot \tau + O^{ul} + tcpAckSz/\, dataR$, where τ is the latency on the WLAN, $tcpAckSz$ is the size of the DATA field of the DATA frame, $dataR$ is the rate used to transmit this field, and O^{ul} is the overhead in time related to the MAC protocol. Specifically, $O^{ul}$is a follows:

$$O^{ul} = 2 \cdot \frac{phyHdrSz}{phyR} + \frac{macHdrSz + ackSz}{baseR} + \frac{fcsSz}{dataR} + SIFS\ .$$

Proof: We omit this proof, as it is almost identical to the proof of Proposition 1. ■

Based on the above propositions, we can now concentrate on evaluating T^{MAC}. As T^{MAC} is a shared component of both T^{seg} and T^{ack}, in the following either a PS-Poll or a DATA frame containing a TCP ACK are referred to as *tagged frame*.

Evaluation of T^{MAC}

To evaluate T^{MAC} we need to model the contention between the PSM station and the M background stations. To this end, we substitute the 802.11 DCF access procedure with an equivalent model, which fits better our analytical approach. Specifically, the standard 802.11 DCF access procedure is equivalent to a p-persistent access protocol ([5]). After the channel is free for a $DIFS$ interval, each station having frames in the sending queue starts a transmission on the next slot with probability p, while defers to the following one with probability $1-p$. The value of p is chosen based on the number of active stations, i.e., $M+1$. To make this dependence explicit, hereafter we refer to p as p_M. Finally, inter-frame spaces are as in the standard DCF.

We are now in the position of deriving the distribution of T^{MAC}. As shown in the proof of Lemma 4, a PSM station willing to send a tagged frame starts a backoff procedure with the congestion window at its minimum value. Then, the backoff and the DCF procedures are executed in the standard way. Since in our model it is assumed that the tagged frame is successfully sent within MAX attempts, the following theorem holds.

Theorem 4: Let

- $T^{bo}(CW)$ be the time to complete the backoff procedure when the congestion window value is CW;
- T^{coll} be the time during which the channel is busy when the tagged frame collides;
- p_f be the probability that no other stations transmit in the same time slot used by the PSM station to start the transmission of the tagged frame.

Then p_f is equal to $(1-p_M)^M$, p_{loss} is equal to $(1-p_f)^{MAX}$, and T^{MAC} is distributed according to the following law:

$$T^{MAC} = DIFS + \sum_{j=1}^{i+1} T^{bo}(CW_j) + \sum_{j=1}^{i} T_j^{coll} \qquad \frac{(1-p_f)^i \cdot p_f}{1-p_{loss}}, \quad i = 0, \ldots, MAX-1 \,. \tag{4}$$

Proof: First of all, it is worth noting that a PS-Poll is scheduled immediately after receiving a Beacon, or immediately after exchanging a previous PS-Poll/DATA/ACK frame sequence [8]. In both cases, all stations in the Wi-Fi hotspot wait a DIFS interval before doing anything. The same remark also applies when a DATA frame containing a TCP ACK becomes ready to be sent for the first time. This results in an (initial) DIFS interval in T^{MAC}. The rationale behind the further components of Equation 4 is as follows. i denotes the number of collisions occurring when the tagged frame is sent, and j represents a specific retransmission attempt within the overall number, $i+1$. Provided that the PSM station experiences i collisions, it performs $i+1$ backoff procedures, each one with a corresponding value of the congestion window. In the first i attempts, after the backoff is expired, a collision occurs, and then the next backoff procedure starts. In the last attempt, the PSM station performs the backoff procedure and then (successfully) accesses the channel. Moreover, it is easy to show that the probability of delivering the tagged frame after i collisions a geometric law with parameter p_f. The correcting factor $1 - p_{loss}$ derives from the assumption that no losses occur due to an excessive number of retransmissions. Finally, the expressions of p_f and p_{loss} derives immediately by their definitions. ■

Equation 4 requires the models of $T^{bo}(CW)$ and T^{coll}, that are provided in the following of this section. By using these models, we have derived a closed form for $E\left[T^{MAC}\right]$ (and both $E\left[T^{seg}\right]$ and $E\left[T^{ack}\right]$, by exploiting Propositions 1 and 2). It is assumed that both $T^{bo}(CW)$ and T^{coll} are independent of the number of retransmission attempts performed by the PSM station (i.e., i in Equation 4). Thus, $E\left[T^{MAC}\right]$ is:

$$\begin{aligned} E\left[T^{MAC}\right] &= DIFS + \frac{p_f}{1-p_{loss}} \cdot \left\{ E\left[T^{coll}\right] \cdot Q_{MAX-1}(1-p_f) + \right. \\ &+ \frac{E\left[t_{sl}^e\right] \cdot CW_{min}}{2} \cdot \left(2 \cdot S_{MAX-1}(2 - 2 \cdot p_f) - S_{MAX-1}(1-p_f)\right) + \\ &+ \left. \frac{E\left[t_{sl}^e\right]}{2} \cdot \left(Q_{MAX-1}(1-p_f) - S_{MAX-1}(1-p_f)\right) \right\} . \end{aligned} \tag{5}$$

Equation 5 requires the definition of the following components.

- t_{sl}^e is the the length of the "equivalent-slot". Specifically, it can be seen as the average time required by a station to decrement the backoff counter during the backoff procedure. When no other stations are active, t_{sl}^e is equal to the length of a slot. Otherwise, it increases, due to the transmissions of the other stations freezing the backoff procedure.
- $S_N(q)$ and $Q_N(q)$ are the well-known closed forms of the successions $\sum_{i=0}^{N} q^i$ and $\sum_{i=0}^{N} i \cdot q^i$, respectively.

Evaluation of T^{coll}

By definition, T^{coll} is the interval elapsed from the instant when the PSM station ends a backoff procedure, up to the instant when it starts the next backoff procedure, when a collision occurs. Let assume that C background stations collide with the PSM station, and let $frSz_i$, $i = 1, \ldots C$ be the size of the frame sent by the i-th station during the collision. Then, the following lemma holds.

Lemma 5: T^{coll} can be expressed as:

$$T^{coll} = \tau + \frac{phyHdrSz}{phyR} + \frac{macHdrSz}{baseR} + \max_i \left\{ \frac{frSz_i}{dataR} \right\} + DIFS . \tag{6}$$

Proof: In our model it is assumed that the payloads of DATA frames sent by background stations are shorter than TCP ACKs. Furthermore, the PS-Poll is composed by the MAC header and the FCS field. Thus, when a collision occurs, the PSM station transmits a tagged frame and immediately senses the medium as busy, since the colliding stations are still transmitting. The overall time during which the medium remains busy is equal to the time interval required to transmit the longest DATA frame. When the medium becomes free it remains idle for a DIFS interval, due to the MAC protocol definition. Finally, the PSM station starts the next backoff procedure. From this line of reasoning Equation 6 immediately follows. ■

For the sake of simplicity, in our model it is assumed that DATA frames sent by background stations are all of the same (fixed) size (FS). Hence, $E\left[T^{coll}\right]$ is as follows

$$E\left[T^{coll}\right] = \tau + \frac{phyHdrSz}{phyR} + \frac{macHdrSz}{baseR} + \frac{FS}{dataR} + DIFS \ . \tag{7}$$

Evaluation of $T^{bo}(CW)$ By definition, $T^{bo}(CW)$ is the time required by the PSM station to execute the backoff procedure, when the congestion window is CW. The first step of the backoff procedure is choosing a value between 0 and $CW-1$, according to a uniform distribution. This value (hereafter referred to as $X(CW)$) represents the number of *free* slots the PSM station must wait before starting the next transmission attempt. Furthermore, during the backoff procedure, background stations may transmit frames, as usual. This causes the PSM station to freeze the procedure until the shared medium returns free. From the above remarks, Lemma 6 follows.

Lemma 6: Let

- t_{sl} be the length in time of a slot;
- T^w be the time interval during which the PSM station freezes the backoff procedure due to a transmission on the shared medium;

Then T^{bo} is

$$T^{bo}(CW) = X(CW)\cdot t_{sl} + X(CW)\cdot \frac{1-p_f}{p_f}\cdot T^w \triangleq X(CW)\cdot t^e_{sl} \ , \tag{8}$$

Furthermore, by assuming that $X(CW)$ and T^w are i.i.d. random variables, the average value of T^{bo} is

$$E\left[T^{bo}(CW)\right] = E\left[X(CW)\right]\cdot E\left[t^e_{sl}\right] = \frac{CW-1}{2}\cdot E\left[t^e_{sl}\right] \ . \tag{9}$$

Proof: Equation 8 is obtained by recalling that: (i) the backoff procedure ends after $X(CW)$ free slots; and (ii) for each slot that remains free, $\frac{1-p_f}{p_f}$ slots are used by other stations to start a transmission. Finally, Equation 9 holds as $X(CW)$ is a random variable following a uniform law between 0 and $CW-1$. ■

It is worth noting that Equation 8 gives the expression of the equivalent slot, t_{sl}^{e}. Furthermore, its average value is

$$E\left[t_{sl}^{e}\right] = t_{sl} + \frac{1-p_f}{p_f} \cdot E\left[T^{w}\right] \ . \tag{10}$$

Based on the above results, the last step to derive $T^{bo}\left(CW\right)$ is characterizing T^{w}. This is achieved by the following lemma.

Lemma 7: The distribution of T^{w} is

$$\begin{cases} 2\cdot\tau + 2\cdot\frac{phyHdr}{phyR} + \frac{macHdr+ackSz}{baseR} + \frac{FS}{dataR} + SIFS + DIFS & \left(1-p_M\right)^{M-1} \\ \tau + \frac{phyHdr}{phyR} + \frac{macHdr}{baseR} + \frac{FS}{dataR} + DIFS & 1-\left(1-p_M\right)^{M-1} \end{cases} .$$

Proof: When the medium becomes busy, either a successful transmission occurs, or frames from different stations collide. As M background stations are active in the Wi-Fi hotspot, the probability of a successful transmission is the probability that just one station accesses the medium during the (initial) slot, i.e., $\left(1-p_M\right)^{M-1}$. In this case, a standard DATA/ACK frame sequence occurs, and then the PSM station continues the backoff procedure after and additional DIFS interval [8]. Otherwise, the transmitting stations send only the DATA frames without receiving any ACK. Thus, when those transmissions end, the PSM station continues the backoff procedure after an additional DIFS interval.

■

In conclusion, the above equations allow to derive the closed form of $E\left[T^{MAC}\right]$ shown in Equation 5.

3.2.2 Evaluation of $E\left[T^{b}\right]$

To transmit a Beacon the Access Point must sense the channel free for a PIFS interval. Then, the Beacon is sent as a broadcast frame, without using neither the standard DCF nor the backoff procedures [8]. At the beginning of Beacon Intervals background stations are not obliged to let the channel free for the transmission of the Beacon. Thus, the Access Point may need to wait the end of a transmission that either is ongoing at the beginning of the Beacon Interval, or starts during a PIFS interval from the beginning of the Beacon Interval. For the same reason, the transmission of a Beacon may collide with other frames sent by background stations. For the sake of simplicity, in our model it is assumed that none of the above events occur. Said in other words, it is assumed that the Beacon is successfully transmitted after a PIFS interval from the beginning of the Beacon Interval. Hence T^{b} is as follows:

$$T^{b} = PIFS + \tau + \frac{beacSz}{baseR} \ ,$$

where $beacSz$ is the size of the Beacon. It is worth noting that, in principle, $beacSz$ could be variable, as the length of the TIM is variable (see the Appendix, or [8]). 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 are constant.

4 Model Validation

The validation of our analysis is carried out by comparing the results obtained with the output of a simulation model. Our simulator implements the networking environments described in Section 2. The model of the Wi-Fi hotspot is totally compliant to the IEEE specifications [8], and is an extension of the simulator used in [4]. Furthermore, the TCP modules at the PSM station and at the fixed server are compliant with the TCP-Reno specifications [13]. The (wired) Internet is modeled as exponentially-distributed delays between the Access Point and the server, and DATA-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 [6] to model the Web traffic. Hence, we test our models in the case of the Web-access from a mobile computer.

First of all, in the following we validate $E\left[T^{MAC}\right]$. As shown in the previous section, 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 offered to the WLAN increases. Specifically, $E\left[T^{MAC}\right]$ ranges from about $350\mu sec$ to about $170msec$, when M increases from 0 to 50. A very interesting point in Figure 3-left is the intersection between $E\left[T^{MAC}\right]$ and the Beacon Interval length (the dash-dotted line at $100msec$). 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 wireless medium is greater than the Beacon Interval length. Therefore, when data are buffered at the Access Point, the PSM station can never switch to the sleep mode, and the PSM itself becomes an overhead with respect to the standard 802.11 MAC protocol.

Figure 3-right allows us to understand the performance of the 802.11 PSM in terms of energy saving. To this end, we define the I_{ps} index as the ratio between the average value of E_{PSM} (Equation 1), and of the energy spent to perform the same data download without the PSM. The latter quantity is well approximated as $E\left[T\right]\cdot P_{ac}$. In fact, as the 802.11 PSM introduces some delay to frames addressed to the PSM station, the total time T overestimates the time spent in the active model when the PSM is not used. By definition, I_{ps} represents the percentage of energy spent by using the 802.11 PSM, with respect to the case without energy management.

Figure 3-right confirms the results reported in [9] (I_{ps} around 10% when $M = 0$), and shows

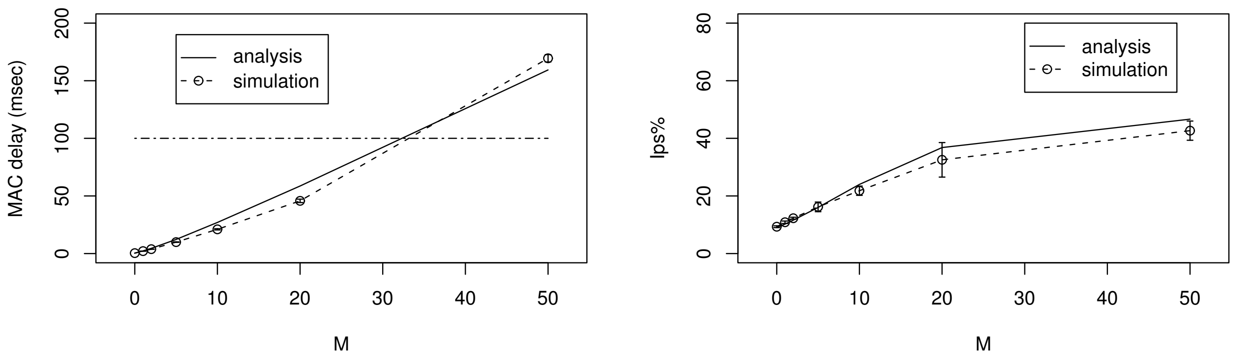

Figure 3: $E\left[T^{MAC}\right]$ (left) and I_{ps} (right) as functions of M.

a pretty good scalability 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 the larger the number of stations are in the Wi-Fi hotspot, the worse performance in terms of power-saving. However, when the congestion in the WLAN increases, both T_{ac} and T increase. T increases because the throughput on the TCP connection decreases, resulting in an increase of T_{data} (see Theorem 2). In conclusion, when M increases, the energy consumption achieved by the PSM increases. This behavior clearly appears by observing $E\left[T^{MAC}\right]$ (Figure 3-left). However, due to its definition, I_{ps} tends to saturate, achieving an asymptotic value.

From the above analysis it results that, though the 802.11 PSM shows good performances in terms of energy savings, it is not likely the best solution for energy saving. Currently, we are deeply investigating the limitations of the 802.11 PSM, 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

[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, August 2003, pp. 273-294. 1

[3] H. Balakrishnan, V.N. Padmanabhan, S. Seshan and R.H. Katz, "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links", IEEE/ACM Transactions on Networking, Vol. 5, N. 6, December 1997. 3

[4] 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, 2002, Vol. 1, pp. 10-31. 1, 14

[5] F. Calì, M. Conti and E. Gregori, “IEEE 802.11 wireless LAN: capacity analysis and protocol enhancement”, ACM Transactions on Networking, Dec. 2000. 1, 10

[6] M.Crovella e A.Bestavros, "Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes", IEEE/ACM Transaction on Networking, Vol.5, No.6, pp.835-846, December 1997. 14

[7] J.P. Ebert, B. Stremmel, E. Wiederhold and A. Wolisz, “An energy-efficient power control approach for WLANs”, J. Comm. Networks, 2000, Vol. 2, pp. 197-206. 1

[8] IEEE standard for Wireless LAN- Medium Access Control and Physical Layer Specification, 802.11, November 1997. 1, 2, 6, 7, 8, 9, 10, 13, 14

[9] R. Krashinsky and H. Balakrishnan, “Minimizing Energy for Wireless Web Access with Bounded Slowdown”, Proceedings of Proceedings of the Eighth Annual ACME/IEEE International Conference on Mobile Computing and Networking (MOBICOM’02), 2002. 1, 2, 3, 14, 16, 17

[10] 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’98), 1998. 1

[11] S. Pilosof, R. Ramjee, D. Raz, Y. Shavitt and P. Sinha, “Understanding TCP fairness over Wireless LAN”, Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM’03), 2003. 1

[12] 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. 1

[13] W.R. Stevens, “TCP/IP Illustrated, Volume 1: The Protocols” , Addison-Wesley, 1994. 14

[14] H. Woesner, J.P. Ebert, M. Schlaeger and A. Wolisz, “Power-saving mechanisms in emerging standards for wireless LAN s: the MAC level perspective”, IEEE Personal Communications, Vol. 5, pp. 40-48 (Special Edition on Power Saving), 1998. 1

Appendix: Overview of the 802.11 Infrastructured Power-Saving Mode

The standard 802.11 wireless interfaces can work in two different modes, i.e. the *active* mode and the *sleep* mode. While in active mode, the wireless interface is able to exchange data, and can be in the *receiving*, in the *transmit*, or in the *idle* state (i.e., it simply overhears the traffic on the channel). Due to the 802.11 MAC protocol, the energy consumption of the active mode depends very little on the operating state. Hence, it can be typically approximated with a constant value (e.g., $750mW$ for Enterasys Networks RoamAbout interfaces [9]). On the other hand, while in sleep mode, only few components of the wireless interface are supplied by the battery (e.g., the clock that maintains synchronization with the Access Point). Therefore, the wireless interface is not able to exchange data, but its energy consumption is at least one order of magnitude lower than in the active mode (e.g., $50mW$, [9]).

The objective of the 802.11 PSM is to let a mobile in the active mode only for the time necessary to exchange data, and to turn it in sleep mode whenever it becomes idle. In a Wi-Fi hotspot (i.e., an infrastructured 802.11 WLAN), this is achieved by exploiting the central role of the Access Point. Each station within 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 the frames destined to stations operating in the PSM. Then, once every Beacon Interval – usually, $100msec$ –, the Access Point broadcasts a special frame, named Beacon. This frame contains a Traffic Indication Map (TIM) that indicates PSM stations having some frame 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 is shown in Figure 2-left. 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 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 switch to the sleep mode.

To send a DATA frame, a PSM station (if the case) wakes-up and performs the standard DCF procedure.