

# Power-Saving in Wi-Fi Hotspots: an Analytical Study<sup>\*</sup>

G.Anastasi<sup>1</sup>, M.Conti<sup>2</sup>, E.Gregori<sup>2</sup>, and A.Passarella<sup>1</sup>

<sup>1</sup> University of Pisa, Dept. of Information Engineering  
Via Diotisalvi, 2 - 56122 Pisa, Italy

{g.anastasi, a.passarella}@iet.unipi.it

<sup>2</sup> CNR - IIT Institute

Via G. Moruzzi, 1 - 56124 Pisa, Italy

{marco.conti, enrico.gregori}@iit.cnr.it

**Abstract.** Wi-Fi hotspots are one of the most promising scenarios for mobile computing. In this scenario, a very limiting factor is the scarcity of mobile-device energetic resources. Both hardware and software architectures of current devices are very inefficient from this standpoint, mainly the networking subsystem. This work analyzes a power-saving network architecture for the mobile-Internet access through Wi-Fi hotspots. Specifically, this solution supports any kind of best-effort network applications, since it is application-independent. In this paper we derive a complete analytical model of the power-saving system when applied to mobile Web access. Furthermore, we use this model to compare our solution with a well-known approach, i.e., the Indirect-TCP. The comparison is performed by considering two performance figures: the energy saved in downloading a Web page and the related transfer-time. The results show that, in the average, our solution saves up to 78% of the energy. Furthermore, the power-saving system introduces an additional average transfer-time of 0.4 sec, and hence it does not significantly affect the QoS perceived by the users. Finally, we assess the sensitiveness of the power-saving system with respect to Internet key parameters, such as the available throughput and the RTT.

**Keywords.** Wi-Fi, Power Saving, Web, Mobile Internet, Analytical Models.

## 1 Introduction

In this work we analyze a power-saving network architecture for 802.11 “Wi-Fi” hotspot environments. This is today one of the most promising scenarios for mobile computing, and is rapidly becoming a key business area. In the typical deployment of such scenario, Internet Service Providers guarantee wireless Internet access in a limited-size environment, such as a campus or a mall (i.e., a

---

<sup>\*</sup> 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: FIRB-PERF and FIRB-VICOM.

“hotspot”). Wireless coverage is achieved by means of Access Points which build a 802.11 WLAN. Moreover, Access Points are connected to the Internet through a standard high-speed LAN. Mobile users subscribe a contract with an ISP, and are allowed to access the Internet on-the-move inside the hotspot.

Integrating mobile devices in such an environment is still an open research issue. Since mobile devices are battery-fed, their energetic resources are a very critical factor. Furthermore, the development of energy-aware solutions for mobile networking is a very pressing requirement [2–4, 11, 12, 17, 18]. Specifically, the wireless interface may drain up to 50% of the total energy spent by a mobile device [17]. Moreover, legacy Internet protocols (such as TCP/IP) are very inefficient from this standpoint [1, 2]. Due to the specific consumption pattern of 802.11 wireless interfaces [14], the optimal power-saving strategy consists in transferring data at the maximum throughput allowed on the wireless link, and switching the wireless interface off (or to put it in a “doze” mode) whenever it is idle [2–4, 16–18]. As it is clear, the key point of such an approach is the identification of idle periods within the network traffic pattern.

In this paper we provide an analytical study of the power-saving networking solution for Wi-Fi hotspots that we developed in [4]. This solution follows an Indirect-TCP approach [7, 8], and operates at the transport and middleware layers. Furthermore, it is application independent in the sense that it does not rely on any a-priori knowledge of the application behavior. Specifically, our system tracks dynamically the behavior of the network application(s) running on the mobile host. Based on the application behavior, it predicts time intervals during which the wireless interface will be idle, and switches it off accordingly.

In [4] we tested our system by considering the Web application. Therefore, in this paper we provide an analytical model of the system behavior when it is used to support mobile Web access. However, both the power-saving system in [4], and the models provided here are quite general and can be used with any kind of best-effort applications.

The first step of our study consists in building a model of the traffic generated by a typical Web user. Then, we provide closed formulas that describe the average behavior of the power-saving system when used to support such a traffic. Moreover, we compare our solution with a pure Indirect-TCP approach. The comparison is carried out by focusing on two main performance figures, i.e., i) the energy saved in downloading a Web page, and ii) the related additional transfer-time.

After assessing the model accuracy against results presented in [4], we exploit the model to deeply analyze the power-saving system. The results show that, under the Internet conditions experienced in [4], our solution saves up to 78% of the energy spent when using the pure Indirect-TCP approach. Moreover, the average additional transfer-time is less than 0.4sec, and hence we can conclude that the power-saving system does not significantly affect the QoS perceived by the users.

Finally, by using our model, we perform a sensitiveness analysis to investigate the dependence of the power-saving system on two Internet key parameters, i.e.,

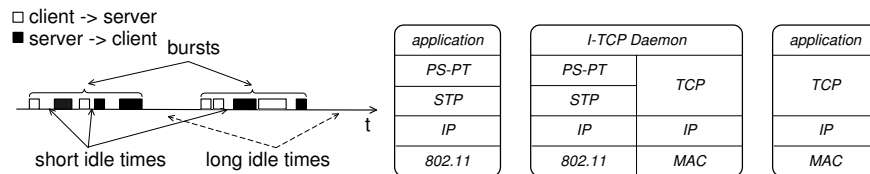
the throughput and the Round Trip Time ( $RTT$ ) between the Web client and server. The results show that power-saving is mainly affected by throughput variations. Specifically, the energy saving varies from 48% to 83% when the throughput increases from 0 to  $\infty$ , i.e., from the lower to the upper theoretical bound. On the other hand, the additional transfer-time is a slightly increasing function of  $RTT$ . However, the average additional transfer-time is always less than  $0.5sec$ .

The paper is organized as follows. Sections 2 and 3 present the power-saving system and its analytical model. Section 4 is devoted to the validation of this model. Finally, Sect. 5 presents the sensitiveness analysis.

## 2 Reference Environment

### 2.1 Power-Saving Network Architecture for Best-Effort Traffic

As highlighted in the previous section, our power-saving architecture supports any kind of best-effort applications. In such applications data-transfer phases are characterized by *bursts* separated by *idle phases* during which data are locally processed. Figure 1-left shows a snapshot of a typical data exchange.



**Fig. 1.** Snapshot of a typical best-effort data exchange (left) and the power-saving network architecture (right).

Specifically, bursts can be seen as made up of data packets. Packets are separated by *short idle times*, while bursts are separated by *long idle times*. Short idle times are due to automatic interactions between computers, while long idle times are related to human reaction times. Therefore, short idle times are typically shorter than long idle times. A typical cut-off value used in the literature is  $1sec$ .

Our power-saving system relies upon the dynamic estimation of (short and long) idle times based on the traffic behavior observed in the past. Specifically, short idle times are estimated by exploiting the Variable-Share Update algorithm [15], while a binary exponential backoff, starting from  $1sec$ , is used for long idle times.

In detail, the estimation algorithm works as follows. When a short idle time begins, the Variable-Share Update algorithm provides an estimate,  $t'_i$ , of the actual idle time,  $t_i$ . If  $t'_i$  occurs to be less than  $t_i$ , the estimate is updated with

the 90<sup>th</sup> percentile of the short idle times, throughout referred to as  $k$ . Finally, if  $t_i$  too occurs to be greater than  $k$ , the algorithm assumes that  $t_i$  is a *long* idle time. Therefore, it provides the next updates by using a binary exponential backoff starting from  $1sec$ .

The network interface of the mobile host is managed according to these estimates. It is worth noting that the network interface has a transient in getting on (throughout referred to as  $t_{so}$ ) during which it drains power from the battery but is not available for exchanging data. Therefore, for idle times less than  $t_{so}$  it is energetically convenient to leave the network interface on. A complete description of the algorithm is provided in [4], and is here omitted for the sake of space.

To integrate the estimation algorithm described above in a real Wi-Fi system, we defined the network architecture shown in Fig. 1-right. This power-saving architecture exploits the Indirect-TCP model, since it splits the transport connection between the (mobile) client and server at the Access Point. In addition, the transport protocol between the mobile and the Access Point is a *Simplified Transport Protocol* (STP in the figure) that is tailored to the characteristics of the wireless link. As shown in [8], this solution outperforms the standard Indirect-TCP architecture [7].

In our architecture the idle-time estimation algorithm is implemented in the *Power-Saving Packet Transfer* (PS-PT) protocol. The PS-PT protocol follows a simple master-slave scheme. When there are no more data to be exchanged (i.e., when an idle time occurs), the PS-PT module at the Access Point generates a short idle time estimate and decide whether it is convenient to switch off the network interface at the mobile host. If so, it sends a “shutdown” command to the mobile host including an indication of the time interval during which it should remain disconnected. Possible data destined to the mobile host that become available while it is disconnected are buffered at the Access Point. When the mobile host reconnects, it polls the Access Point to obtain either new data or an update of the idle-time estimate. Updates are derived according to the algorithm described above. Upon deriving the update, the Access Point decides whether it is convenient to the mobile host to switch the wireless interface off. Finally, if the mobile host generates new data while it is disconnected, the wireless interface is immediately switched on and data are sent. Details about the network architecture shown in Fig. 1-right can be found in [4].

## 2.2 Performance Figures

To evaluate the performance of our system, we define two indexes. We measure the energy saving achieved by the power-saving architecture – with respect to a pure I-TCP architecture – by means of the  $I_{ps}$  index, defined as

$$I_{ps} = \frac{C_{ps}}{C_{I-TCP}} . \quad (1)$$

Specifically,  $C_{ps}$  is the energy spent to download a set of Web pages when using the power-saving system.  $C_{I-TCP}$  is the energy spent to download the

same set of Web pages when using a pure Indirect-TCP approach<sup>3</sup>. Due to the 802.11 wireless interface consumption patterns, the energy spent is proportional to the time interval during which the wireless interface remains switched on [14]. Therefore, we express both  $C_{ps}$  and  $C_{I-TCP}$  in seconds.

Finally, it must be noted that the power-saving system can introduce an additional delay to the Web-page transfer-time. Specifically, additional delays may be introduced when idle-time estimates are greater than the actual idle times. Web users might perceive the additional transfer-time as a degradation of the QoS, and hence it is important to quantify it. To this end, we use the  $I_{pd}$  index, defined as

$$I_{pd} = URT_{ps} - URT_{I-TCP} , \quad (2)$$

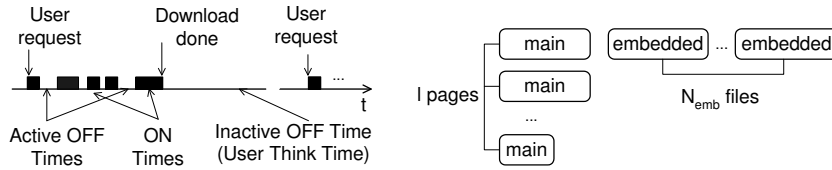
where  $URT$  is the User Response Time, i.e., the time elapsed to download a Web-page, between the user request and the page rendering at the mobile host.

### 3 System Model

The description of the power-saving system (Sect. 2) shows that several parameters affect the system behavior. In particular, the throughputs on the wireless and wired networks, the accuracy of the idle-time estimates, the application-level traffic profile play an important role. To clearly understand their influence on the system performance, we describe the average behavior of the power-saving system by means of an analytical model. Since the system performance is related to the application-level traffic, it is necessary a preliminary characterization of the traffic generated, in the average case, by a generic Web-user.

#### 3.1 Web-User Model

To characterize a client access to a Web server, it is sufficient to focus on a single Web-page download (see Fig. 2-left).



**Fig. 2.** Web-page download (left) and scheme of the basic block (right).

As is well known a Web page consists of a main file and zero or more embedded files (e.g., figures). All files composing a Web page are transferred during

<sup>3</sup> It should be noted that, throughout our work, we use an Indirect-TCP architecture with a simplified transport protocol (i.e., the STP) between the mobile host and the Access Point, as discussed in [8].

the ON Time interval while in the Inactive OFF Time (or User Think Time) the user reads the content of the downloaded Web page. Finally, during Active OFF Times the browser parses a piece of the main file and sends the request for the next embedded file(s)<sup>4</sup>. Active OFF Times are typically less than User Think Times, since they are due to automatic interactions between computers.

In [4] we used the SURGE simulator to simulate a typical Web user. The SURGE is based on the statistical model of the Web traffic presented in [9, 10, 13]. Therefore, we exploit this model to derive the parameters that define the *average* traffic profile generated by the generic Web-user (see Tab. 1).

**Table 1.** Parameters that define the Web-user average traffic profile.

Definition	Symbol	Value
Probability that a Web page contains embedded files	$p_{emb}$	0.44
Average number of embedded files in a Web page	$N_{emb}$	1.50
Average size of the embedded files (bytes)	$\overline{D}_{emb}$	6348
Average size of the main files (bytes)	$\overline{D}_{mf}$	17496
Average User Think Time (seconds)	$\overline{UTT}$	3.25

Specifically, the Web session of the simulated user can be thought of as the continuous download of the same *basic block*. The basic block consists of a set of Web pages defined in such a way that the traffic generated for downloading them meets the statistics of Tab. 1. In detail, the basic block is made up of  $l \triangleq \lceil 1/p_{emb} \rceil$  Web pages. The first page contains the main file and  $N_{emb}$  embedded files, while the others pages are composed by the main file only. The dimension of each main file is  $\overline{D}_{mf}$ , except for the main file of the last page, whose dimension is  $\overline{D}_{mf} \cdot (l - 1/p_{emb})$  (for instance, if  $1/p_{emb}$  is 3.4, the basic block is made up of 4 pages, the dimension of the last main file being  $0.4 \cdot \overline{D}_{mf}$ ). The dimension of each embedded file is  $\overline{D}_{emb}$ . Finally, the user waits  $\overline{UTT}$  seconds before downloading the next page. Figure 2-right shows a scheme of the basic block.

**Idle Times Characterization** With respect to the reference network architecture shown in Fig. 1-right, we assume that the application layer generates the average traffic profile shown in Fig. 2-right. The power-saving system exploits idle times in this traffic to save energy. The characterization of these idle times is thus a necessary step to model the energetic behavior of the power-saving system.

It is worth recalling that Web utilizes TCP as the transport protocol. In particular, HTTP/1.1 uses persistent connections. As discussed in [6], we can reasonably assume that the download of the basic block occurs over a unique, steady-state, TCP connection. Moreover, in average, data transfer on a steady-state TCP connection can be seen as the transfer of fixed-size groups of TCP

<sup>4</sup> Several files can be requested at the same time, by exploiting the pipelining mechanism of the HTTP.

segments, separated by *RTT*s [19]. Based on these remarks, we can conclude that *short* idle times in the user traffic profile are produced by the TCP behavior, and can thus be considered as samples of the *RTT* between the client and the Web server. On the other hand, *long* idle times (i.e., idle times between bursts) can be considered as samples of the User Think Time.

Hereafter we focus on a single basic-block download to derive the analytical model of the power-saving system. Specifically, we provide a model for evaluating both the energy consumption and the additional *URT* introduced. For the reader's convenience, we summarize in Tab. 2 the parameters used throughout the paper.

**Table 2.** Symbols used throughout the paper.

Definition	Symbol
Total dimension of the basic block	$B$
Number of pages in the basic block	$l$
Average available Internet throughput	$\bar{\gamma}$
Average available wireless-link throughput	$\bar{\gamma}_{wl}$
Switching-on transient interval of the wireless interface	$t_{so}$
Average number of switching-on events during the basic block download	$S$
Average number of switching-on events during a short idle time	$S_1$
Average number of short idle times during the basic block download	$r$
Average number of switching-on events during a <i>UTT</i>	$S_2$
Average number of switching-on events to send the embedded-file(s) request	$S_3$
Short idle-time sample	$t_i$
Short idle-time estimate	$t'_i$
Upper bound of the idle-time distribution	$M$
90 <sup>th</sup> percentile of short idle-times	$k$
Error of the idle-time estimator	$e$

### 3.2 Energy Consumption Model

We are now in the position to evaluate the energy spent to download a single basic block, by using either the power-saving system or the pure Indirect-TCP approach. In the following we claim a set propositions that allow us to provide closed formulas for  $I_{ps}$  and  $I_{pd}$ , respectively. Proofs of these propositions are omitted for the sake of space. They can be found in [6].

The energy spent when using an Indirect-TCP approach (i.e.,  $C_{I-TCP}$ ) is the total time required to download the basic block, since no power-saving strategy is used. Therefore, the following theorem holds.

**Theorem 1.** *The energy spent to download a single basic block by using a pure Indirect-TCP approach is*

$$C_{I-TCP} = \frac{B}{\bar{\gamma}} + l \cdot \overline{UTT} = \frac{\overline{D}_{mf} \cdot 1/p_{emb} + \overline{N}_{emb} \cdot \overline{D}_{emb}}{\bar{\gamma}} + l \cdot \overline{UTT} , \quad (3)$$

where  $\bar{\gamma}$  is the average available Internet throughput between the client and the Web server and  $B$  is the total size (in bytes) of the basic block.

On the other hand, the energy consumed when using the power-saving system is made-up of two components. The first one corresponds to the time required to transfer the basic block over the wireless link. The second one is due to the wireless network interface at the mobile host that consumes  $t_{so}$  seconds every time it is switched on. Therefore, the following lemma hold.

**Lemma 1.** *The energy spent by using the power-saving system is*

$$C_{ps} = \frac{B}{\bar{\gamma}_{wl}} + t_{so} \cdot S \triangleq \frac{B}{\bar{\gamma}_{wl}} + r \cdot S_1 + l \cdot S_2 + S_3, \quad (4)$$

where  $\bar{\gamma}_{wl}$  is the average available throughput allowed by the wireless link, and  $S$  is the average number of switching-on events during the download of a basic block.

It is worth noting that, thanks to the network architecture design,  $t_{so} \cdot S$  is the only energy contribution that is introduced by the power-saving mechanisms. The Lemmas presented belows allow us to derive a closed formula for this contribution.

Firstly,  $S$  includes three components, i.e., i) the number of swithing-on events within *short* idle times (i.e.,  $r \cdot S_1$  in (4)); ii) the number of switching-on events within *long* idle times (i.e.,  $l \cdot S_2$  in (4)); and iii) the number of switching-on events within the download of the first page of the basic block, between the end of the main-file download and the beginning of the embedded-files download (i.e.,  $S_3$  in (4)). Hence,  $S$  can be evaluated as shown in (4).

According to the characterization of short idle-times (see Sect. 3.1), the  $r$  term in (4) can be seen as the number of  $RTT$ s within the download of a single basic block, i.e.  $r$  is equal  $B / (\bar{\gamma} \cdot \overline{RTT})$ .  $S_1$  can be derived by considering the algorithm for estimating short idle-times, and the distributions of short idle-times and their estimates (i.e.,  $t_i$  and  $t'_i$ , respectively). Details are reported in [6].

Furthermore, the  $S_2$  term in (4), can be derived as follows. In the initial part of a User Think Time (1 sec), the network interface is managed according to the estimator of *short* idle times (see Sect. 2.1). Then, it is managed according to a binary exponential backoff. Hence, the following lemma holds.

**Lemma 2.** *The average number of switching-on events within a User Think Time is*

$$S_2 = H + \lceil \log_2 \overline{UTT} \rceil ,$$

where  $H$  is the average number of switching-on events within the first second of the User Think Time.



A formula for  $H$  can be obtained by following the same approach used to derive  $S_1$ , and is thus omitted (see [6] for details). Finally, since the PS-PT protocol detects a short idle-time after the last packet of the main file, the  $S_3$  term of (4) can be evaluated as follows.

**Lemma 3.** *The average number of switching-on events between the end of the main-file download and the beginning of the embedded-files download is*

$$S_3 = p(t'_i > t_{so}) .$$

The above lemmas allow us to prove the following theorem.

**Theorem 2.** *The energy spent to download a single basic block by using the power-saving system is*

$$C_{ps} = \frac{B}{\gamma_{wl}} + t_{so} \cdot \left\{ \frac{B}{\gamma \cdot RTT} \cdot S_1 + l \cdot (H + \lceil \log_2 \overline{URT} \rceil) + p(t'_i > t_{so}) \right\} . \quad (5)$$

### 3.3 Additional URT Model

In the HTTP/1.1 version, the download of a Web page can be seen as the sequence of two non-overlapping transactions between the client and the server, each of which starts with a client request. During the first transaction the client downloads the main file, while in the second transaction it fetches all embedded files, if any. Due to the TCP behavior, each transaction consists of several groups of back-to-back TCP segments separated by idle times, as discussed in Sect. 3.1. When an idle-time estimate occurs to be too long, the Access Point stores TCP segments received from the Web server in a local buffer, waiting for the mobile host to reconnect (see Sect. 2.1). In this way, the power-saving system introduces an additional time to the delay experienced by that group of TCP segments. It is worth noting that the additional delay introduced in a whole transaction is only the delay related to the *last* group of TCP segment in that transaction.

It must be pointed out that the Access Point is usually connected to the Internet through an high-speed LAN (e.g., a 100-Mbps fast-Ethernet). Hence, the time interval required by the Access Point to receive a group of TCP segments is typically negligible. As a consequence, the additional delay introduced for a group is well approximated by the additional delay introduced for the first TCP segment within the group, i.e., the error of the idle-time estimator (throughout referred to as  $e$ ). Finally, if the transaction starts when the wireless interface of the mobile host is off, it is delayed of further  $t_{so}$  seconds.

Based on these remarks, we can evaluate the average additional delay introduced in a Web-page download ( $\overline{I_{pd}}$ ) as follows:

**Theorem 3.** *The average additional delay introduced by the power-saving system when downloading of a Web page is*

$$\overline{I_{pd}} = t_{so} + \bar{e} + \left\{ t_{so} \cdot p(t'_i > t_{so}) + \bar{e} \right\} \cdot p_{emb} , \quad (6)$$

where  $\bar{e}$  is the average value of  $e$  and  $p_{emb}$  is the probability that a Web-page contains embedded files.

In detail,  $t_{so} + \bar{e}$  is the additional delay introduced in the transfer time of the main-file. If the Web-page contains embedded files, a further delay is introduced. This delay is  $\bar{e}$  if the wireless interface is initially switched on,  $t_{so} + \bar{e}$  otherwise. Thus, to evaluate  $\overline{I_{pd}}$  we need to derive the idle-time estimator error,  $e$  whose formula is provided by the following theorem. As above, the related proof can be found in [6], and is here omitted.

**Theorem 4.** *The average error of the short idle-time estimator is*

$$\bar{e} = \frac{1}{2} \left( \frac{M^2 - t_{so}^2}{4M} \cdot u(M, t_{so}) + 0.9 \cdot \frac{k^2 - t_{so}^2}{4M} \cdot u(k, t_{so}) + 0.1 \cdot \frac{2sec - M - k}{2} \cdot \chi(k, t_{so}) \right), \quad (7)$$

where  $u(x, y) = \begin{cases} 1 & \text{if } x \geq y \\ 0 & \text{otherwise} \end{cases}$ , and  $\chi(k, t_{so}) = \begin{cases} 1 & \text{if } 1sec - k > t_{so} \\ 0 & \text{otherwise} \end{cases}$ .

## 4 Model Validation

In [4] we implemented a prototype of the power-saving network architecture. We used a real-Internet path to connect the Access Point and the Web server on the fixed host and simulated an 11-Mbps wireless link between the Access Point and the mobile host. We performed an extensive set of experiments, and each experiment provided us with a sample of  $I_{ps}$  and  $\overline{I_{pd}}$ . We replicated the experiments by considering different wireless-link throughputs, starting from 11 Mbps down to 2 Mbps. We observed very small variations in energy saving since the bottleneck between the client and the server is always the wired part of the path (see [5] for details on the methodology).

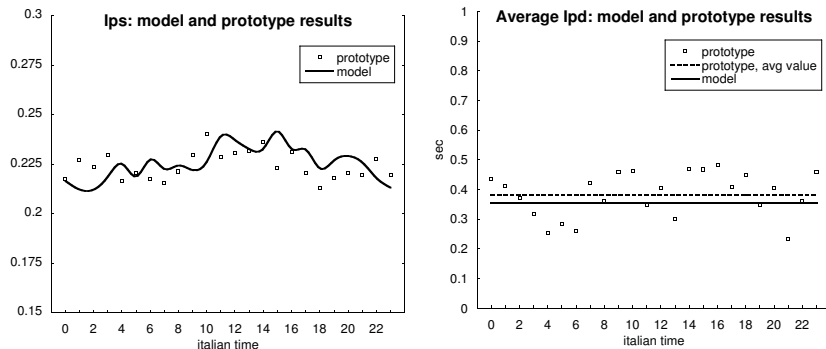
The model validation is carried out by comparing the analytical results for  $I_{ps}$  and  $\overline{I_{pd}}$  provided by the model with the average hourly results obtained from the prototype<sup>5</sup>. Specifically, for each hour of the day, i) we instantiate the model parameters with the average values experienced by the prototype; ii) we derive the model predictions for  $I_{ps}$  and  $\overline{I_{pd}}$ ; and finally iii) we compare these predictions with the prototype results. It is worth noting that in each experiment the Web user generates a traffic that meets the statistics shown in Tab. 1. Furthermore, according to the remarks discussed in Sect. 3.1, we use  $2 \cdot \overline{RTT}$  as the maximum value for both  $t_i$  and  $t'_i$ . The  $\overline{RTT}$  experienced in the experiments is almost independent on the specific hour. Hence, the  $M$  parameter in (7) does not vary over different hours. Finally,  $S_1$ ,  $S_2$  and  $S_3$  values are almost identical over the whole day, and  $\overline{\gamma}_{wl}$  is assumed equal to 11Mbps. Therefore, in our testbed, the only parameter varying over different hours is the throughput experienced in the (wired) Internet, i.e.,  $\overline{\gamma}$ . Table 3 reports the values of the other parameters used to validate our analytical model.

<sup>5</sup> As discussed in [5],  $I_{ps}$  and  $\overline{I_{pd}}$  samples collected within the same hour (also in different days) are i.i.d.

**Table 3.** Parameters used to validate the analytical model.

Definition	Symbol	Value
Average number of switching-on events in a short idle time	$S_1$	1.55
Average number of switching-on events in a $UTT$	$S_2$	5
Number of switching-on events to send the embedded files request	$S_3$	1
Network RTT between the client and the Web server	$\overline{RTT}$	0.3 sec
Switching-on transient interval of the wireless interface	$t_{so}$	0.1 sec
Indicator function of $k$ and $t_{so}$ relative values	$\chi(k, t_{so})$	1

Figure 3 shows the hourly average values for  $I_{ps}$  and  $\overline{I_{pd}}$  measured by using the prototype. We also plot the  $I_{ps}$  and  $\overline{I_{pd}}$  figures derived from the analytical model.



**Fig. 3.** Hourly average  $I_{ps}$  and  $\overline{I_{pd}}$  obtained from the model and by the prototype, respectively.

As far as the  $I_{ps}$  index (left-side plot), the model and the prototype provide very close results: the difference is always less than 9% of the prototype results. Furthermore, we have compared the daily average of  $I_{ps}$  values with the corresponding analytical predictions provided by the model. The results obtained (not reported here) show that the difference between the model and the prototype is less than 1% of the prototype average daily value.

As far as the  $\overline{I_{pd}}$  index (right-side plot), the results show that experimental values vary during the day, i.e., the prototype is sensitive to variations of  $\overline{\gamma}$ . Specifically,  $\overline{\gamma}$  depends on two factors, i.e., i) congestion in the Internet, that reduce the TCP window size; and ii) variations of the  $RTT$  between the client and the server. As discussed in Sect. 3.3, the additional URT is affected by  $RTT$  variations. However, we have no sufficient information to include the precise  $RTT$  pattern in the analytical model. As a consequence, the  $\overline{I_{pd}}$  model allows us to measure the additional URT related to the average  $RTT$ . Specifically, if we compare the model prediction with the daily average value of  $\overline{I_{pd}}$ , the difference

is about 7% of the prototype results (the average  $RTT$  over the whole day is 0.3sec).

The above results show the accuracy of our analytical model. In the next session we use this model to investigate the sensitiveness of the power-saving system to two Internet key parameters, i.e. the available throughput and the  $RTT$ .

## 5 System Sensitiveness

### 5.1 Throughput Analysis

As noted in the previous section, the Internet throughput (i.e.,  $\bar{\gamma}$ ) depends on both the network  $RTT$  and the TCP window size. Since the  $I_{ps}$  index depends on  $\bar{\gamma}$  (see (3) and (5)), it follows that it is influenced by both the  $RTT$  and the TCP window size. On the other hand, the  $\overline{I_{pd}}$  index is only affected by variations in the  $RTT$ , as shown by (6) and (7). Therefore, below we analyze  $I_{ps}$  as a function of  $\bar{\gamma}$ , and  $\overline{I_{pd}}$  as a function of  $RTT$ .

Firstly, we derive the  $I_{ps}$  index as a function of  $\bar{\gamma}$ . Starting from (3) and (5), after some algebraic manipulations,  $I_{ps}(\bar{\gamma})$  becomes:

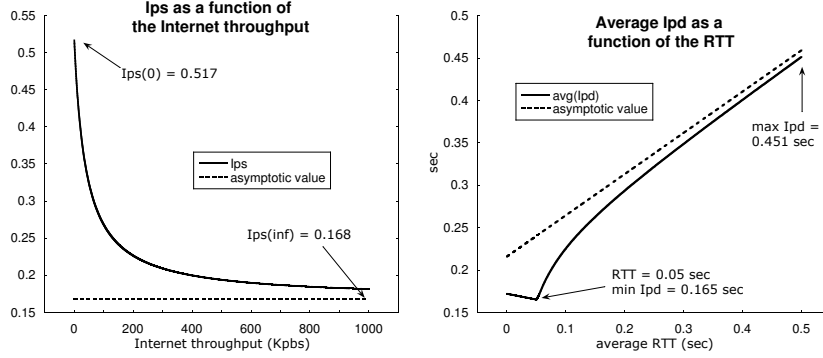
$$I_{ps}(\bar{\gamma}) = \frac{a\bar{\gamma} + b}{c\bar{\gamma} + d}, \quad (8)$$

where  $a$ ,  $b$ ,  $c$  and  $d$  are constant terms. Figure 4-left shows  $I_{ps}$  as a function of  $\bar{\gamma}$ <sup>6</sup>.

It clearly appears that when  $\bar{\gamma}$  increases  $I_{ps}$  decreases and, hence, the power-saving system saves more energy. This result is somehow counter-intuitive, since one would expect that the best power saving is achieved when  $\bar{\gamma}$  is at its lowest value. In this case the overall idle time during the download of the basic block is at its maximum value. However, the  $I_{ps}$  behavior in the above plot can be explained as follows. As shown in (3) and (5), variations of  $\bar{\gamma}$  affect both  $C_{I-TCP}$  and  $C_{ps}$ . In the Indirect-TCP architecture, when  $\bar{\gamma}$  increases, the time needed to fetch the basic block from the Web server (i.e.,  $B/\bar{\gamma}$ ) decreases, and  $C_{I-TCP}$  decreases accordingly. On the other hand, the dependence of  $C_{ps}$  on  $\bar{\gamma}$  is as follows. As highlighted in Sect. 4,  $\bar{\gamma}$  is strictly related to the TCP window size. Since  $RTT$  is almost stable, large TCP windows mean high  $\bar{\gamma}$  values, while narrow TCP windows correspond to low  $\bar{\gamma}$  values. Furthermore, if the TCP window size increases, the number of  $RTT$ s needed to fetch the basic block (i.e.,  $r$  in (4)) drops, since more bytes are downloaded in a single  $RTT$ . Equations (4) and (5) show that if  $r$  decreases, the number of switching-on events (i.e.,  $S$ ) decreases. Therefore, we can conclude that the more  $\bar{\gamma}$  increases, the more  $C_{ps}$  decreases. Since both  $C_{I-TCP}$  and  $C_{ps}$  benefit from increases of  $\bar{\gamma}$ , the  $I_{ps}$  pattern is defined by the parameters  $a$ ,  $b$ ,  $c$  and  $d$ . Specifically, in the Internet configuration we experienced, when  $\bar{\gamma}$  increases,  $C_{ps}$  decreases more than  $C_{I-TCP}$  does, and hence  $I_{ps}$  drops.

<sup>6</sup> Clearly  $\bar{\gamma} > 0$  is the only  $\bar{\gamma}$  range that makes sense in (8).

As a final remark, it is worth noting that Fig. 4-left highlights the theoretical lower and upper bounds of  $I_{ps}$ . Specifically, in the Internet configuration we experienced,  $I_{ps}$  ranges between 0.517 (when  $\bar{\gamma} = 0$ ) and 0.168 (when  $\bar{\gamma}$  approaches  $\infty$ ). Therefore, with respect to the I-TCP architecture, our power-saving system guarantees energy savings that are always above 48%, and raise up to 83%. However, if we focus on realistic throughput values (i.e., between  $50Kbps$  and  $1Mbps$ ), energy savings are almost stable, since they vary between 68% and 82%.



**Fig. 4.**  $I_{ps}$  as a function of the Internet throughput  $\bar{\gamma}$  (left-side plot) and  $\overline{I_{pd}}$  as a function of the  $\overline{RTT}$  value (right-side plot).

## 5.2 $RTT$ Analysis

In this section we analyze the dependence of  $\overline{I_{pd}}$  on the average Round Trip Time, i.e.,  $\overline{RTT}$ . As a preliminary step, it is necessary to define the range of valid  $\overline{RTT}$  values. Specifically, our power-saving system defines  $1sec$  as the upper bound of short idle-times. Therefore,  $1sec$  is also the upper bound of both the short idle-time and estimate distributions (i.e.,  $M \leq 1sec$ ). Since in our model  $M$  is equal to  $2 \cdot \overline{RTT}$ ,  $\overline{RTT}$  must be less than  $0.5sec$ . On the other hand, we can use 0 as the lower bound – or, more precisely, as the theoretical lower limit – of  $\overline{RTT}$ .

Also, it must be pointed out that  $t_{so}$  is the lower bound of  $\overline{I_{pd}}$ . Specifically, even if the estimator error is always equal to 0, the mobile host switches the wireless interface on at least once every Web-page download (i.e., when the user sends a new Web page request). Therefore  $t_{so}$  represents an additional URT that can never be eliminated when using our power-saving system.

Finally, for the sake of simplicity, hereafter we assume that  $t_{so} = 0.1sec$  holds. Therefore, since  $t_{so}$  is equal to  $0.1sec$  and  $\overline{RTT}$  is less than  $0.5sec$ ,  $\chi(k, t_{so})$  is always equal to 1. However, the analysis can be easily extended to any  $t_{so}$  value.

From (6) and (7) we derive the plot shown in Fig. 4-right. In this plot we can observe three regions corresponding to i)  $M$  less than  $t_{so}$ ; ii)  $M$  between  $t_{so}$  and  $t_{so}/0.9$ ; and, iii)  $M$  greater than  $t_{so}/0.9$ . The second region is very small and, actually, can be hardly distinguished in Fig. 4-right.

In the first region, the  $\overline{RTT}$  is so short that  $t'_i$  is always less than  $t_{so}$ . Hence, the wireless interface remains idle for  $k$  seconds and then (if necessary) it is switched off for the next  $1 - k$  seconds. This latter value also defines the estimation error. Therefore, in this region, both  $\bar{e}$  and  $\overline{I_{pd}}$  are decreasing functions of  $\overline{RTT}$ , and  $\overline{I_{pd}}$  reaches its minimum value ( $\min \overline{I_{pd}} = 0.165sec$ , when  $\overline{RTT} = 0.05sec$ ). On the other hand, in the second and third regions  $t'_i$  may be greater than  $t_{so}$ . Therefore,  $\bar{e}$  and  $\overline{I_{pd}}$  become increasing functions of  $\overline{RTT}$ , and hence  $\overline{I_{pd}}$  reaches its maximum value ( $\max \overline{I_{pd}} = 0.451sec$ , achieved when  $\overline{RTT} = 0.5sec$ ). The complete analysis of Fig. 4-right is provided in [6], and is here omitted for the sake of space.

As a final remark, it is worth noting that the  $\overline{I_{pd}}$  figure in the third region can be well approximated by a linear increasing function, that grows as  $0.487 \cdot \overline{RTT}$ . Therefore, we can conclude that increases of  $\overline{RTT}$  have a moderate impact on the additional URT.

## 6 Conclusions

In this work we have derived an analytical model of the power-saving architecture developed in [4]. This solution is tailored to Wi-Fi hotspot scenarios, and is aimed at reducing the energy consumed by a mobile host running non-real-time network applications. The analytical model has been used to analyze the performance of our system when used to support mobile Web access. The results have shown that, under the Internet conditions experienced by the prototype, the power-saving system saves up to 78% of the energy consumed by using a pure Indirect-TCP approach. Furthermore, the additional transfer-time of a Web page is about 0.4sec, and, hence, Web users do not perceive the presence of the power-saving system as a significant degradation of the QoS.

Furthermore, we have performed a sensitiveness analysis with respect to two Internet key parameters, i.e., the throughput on the wired network and the  $RTT$ . This analysis has shown that, when focusing on typical throughput ranges (i.e., between 50Kbps and 1Mbps), the energy saved is always greater than 68%. Finally, we have found that the average additional transfer-time is a slightly increasing function of the average  $RTT$ . However, we can conclude that the power-saving never affects the QoS perceived by Web users, since the average additional transfer-time is always less than 0.5sec.

## References

1. S.Agrawal, S.Singh, "An Experimental Study of TCP's Energy Consumption over a Wireless Link", 4th European Personal Mobile Communications Conference, February 20-22, 2001, Vienna, Austria.
2. G. Anastasi, M. Conti, W. Lapenna, "A Power Saving Network Architecture for Accessing the Internet from Mobile Computers: Design, Implementation and Measurements", The Computer Journal, Vol. 46, No.1, 2003, pp. 3-15.

3. G.Anastasi, M.Conti, E.Gregori and A.Passarella, "A power saving architecture for web access from mobile computers", Proc. 2nd IFIP TC-6 Networking Conf. (Networking 2002), 2002, Pisa, Italy, LNCS# 2345, pp. 240-251.
4. G. Anastasi, M. Conti, E. Gregori and A. Passarella, "Balancing Energy Saving and QoS in the Mobile Internet: An Application-Independent Approach", Proceedings of the 36th Hawaii Int. Conf. on System Sciences (HICSS-36), January 03.
5. G.Anastasi, M.Conti, E.Gregori and A.Passarella, "Performance Comparison of Power Saving Strategies for Mobile Web Access", Performance Evaluation, to appear.
6. G.Anastasi, M.Conti, E.Gregori and A.Passarella, "Power-Saving in Wi-Fi Hotspots: an Analytical Study (Extended Version)", Tech. Rep. DII03-5, Dept. of Information Engineering, University of Pisa, May 2003 (available at [http://www.ing.unipi.it/~o783499/research/docs/PWC-WiFi\\_ext.pdf](http://www.ing.unipi.it/~o783499/research/docs/PWC-WiFi_ext.pdf)).
7. A.Bakre, B.R.Badrinath, "Implementation and Performance Evaluation of Indirect TCP", IEEE Trans. on Computers, Vol.46, No.3, March 1997.
8. H. Balakrishnan, V. N. Padmanabhan, S. Seshan, R. H. Katz, "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links", IEEE/ACM Trans. on Networking, Vol. 5, N. 6, December 1997.
9. P.Barford e M.Crovella, "Generating Representative Web Workloads for Network and Server Performance Evaluation", Proc. of ACM SIGMETRICS '98, pp. 151-160, June 1998.
10. P.Barford, A.Bestavros, A.Bradley e M.Crovella, "Changes in Web Client Access Patterns", World Wide Web (Special Issue on Characterization and Performance Evaluation), 1999.
11. L.Bononi, M.Conti and M.Donatiello, "A distributed mechanism for power saving in IEEE 802.11 wireless LANs", Mobile Networks Applic. (MONET), 2001, Vol. 6, pp. 211-222.
12. R.Bruno, M.Conti and E.Gregori, "Optimization of efficiency and energy consumption in p-persistent CSMA-based wireless LANs", IEEE Trans. Mobile Comput., 2002, Vol. 1, 10-31.
13. M.Crovella e A.Bestavros, "Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes", IEEE/ACM Trans. on Networking, Vol.5, No.6, pp.835-846, Dec. 1997.
14. L.M. Feeney and M. Nilsson, "Investigating the energy consumption of a wireless network interface in an ad hoc networking environment", Proceedings of the 20th IEEE Computer and Communications Societies Conf. (INFOCOM 2001), 2001.
15. M. Herbster and M.K. Warmuth, "Tracking the best expert", Proceedings of the 12th Int. Conf. on Machine Learning, 1995, pp. 286-294.
16. IEEE standard for Wireless LAN- Medium Access Control and Physical Layer Specification, P802.11, November 1997.
17. R.Kravets e P.Krishnan, "Power Management Techniques for Mobile Communication", Proc. of the 4th ACM/IEEE Int. Conf. on Mobile Computing and Networking (Mobicom'98).
18. R.Krashinsky and H.Balakrishnan, "Minimizing Energy for Wireless Web Access with Bounded Slowdown", Proc. 8th Int. Conf. on Mobile Computing and Networking (Mobicom 2002), 2002.
19. M. Mathis, J. Semke, J. Mahdavi and T.Ott, "The macroscopic behavior of the TCP Congestion Avoidance Algorithm", Computer Communication Review, Vol. 27, N. 3, July 1997.