

A Performance Study of Power-Saving policies for Wi-Fi Hotspots

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 one of the most promising scenarios for mobile computing. In this scenario, a very limiting factor is the shortage of energetic resources in mobile devices. Legacy networking protocols are very inefficient in terms of energy management. This work focuses on a network architecture for energy-efficient mobile-Internet access through Wi-Fi hotspots (PS-WiFi). The proposed architecture is able to support any kind of (best effort) applications. In this paper we derive an analytical model of PS-WiFi to analyze the performance of the architecture and to tune its parameters. The model is validated exploiting measurements obtained by using an Internet prototype implementation. The validation shows that the model is able to predict the energy saved by the PS-WiFi architecture with a very good accuracy. Therefore, we used this model to better understand the behavior of PS-WiFi, and to assess its sensitiveness to the main Internet parameters, i.e., the 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 “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 “hotspot”). Wireless coverage is achieved by means of 802.11 Access Points. 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. Figure 1 shows a scheme of this scenario.

An efficient integration of mobile devices in this environment presents many research problems. Mobile devices typically have limited (computational, storage, energetic) resources, with respect to

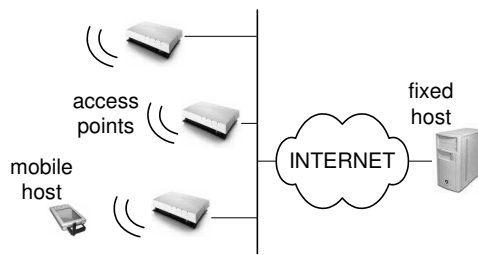


Figure 1: Mobile Internet access in a Wi-Fi hotspot.

desktop computers. In addition, wireless links provide lower bandwidth and higher bit error rates with respect to wired links. Therefore, the use of legacy Internet solutions may result in a low utilization of the scarce system resources. Among the others, the energetic resources are a very critical factor, since mobile devices are battery-fed ([19, 24, 31]). Power-saving policies for mobile devices have been proposed at different levels [25], including the physical transmissions [38], the operating system [20, 30, 36], the network protocols [2, 3, 4, 12, 13, 16, 27, 28] and the applications [18, 26, 33, 34].

The development of energy-aware solutions for the networking subsystem is a key requirement. The wireless interface drains up to 50% of the total energy spent by a mobile device [27]. In addition, 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 [17], the best way to save energy is concentrating the networking activities. Therefore, the ideal power-saving strategy consists in transferring data at the maximum throughput allowed on the wireless link, and switching the wireless interface off (or putting it in a “doze” mode) as soon as it becomes idle. Many papers propose this approach to reduce the energy consumption of the wireless interface [2, 3, 4, 23, 27, 28, 37].

In this paper we investigate the performance of the power-saving system for Wi-Fi hotspots (throughout referred to as PS-WiFi) that we designed in [4]. PS-WiFi follows an Indirect-TCP approach [8, 9], and operates at the transport and middleware layers. Moreover, it is application independent since it does not rely on any a-priori knowledge about the application behavior. Specifically, our system continuously monitors the behavior of the network application(s) running on the mobile device. Based on it, PS-WiFi predicts when and how long the wireless interface will be idle, and switches it off accordingly.

In [4] we tested the efficiency of our system by developing a PS-WiFi prototype and performed some measurements. The experimental results shown that PS-WiFi is very effective. It saves about 80% of the energy spent when using an Indirect-TCP architecture without energy management [8]. However, as the measurement study was carried out using the real Internet, the environment pa-

rameters were not completely under our control. Hence, the measurements reported in [4] do not allow analyzing the performance of PS-WiFi extensively. To better understand the potentialities of the PS-WiFi approach in this paper we develop an analytical model of PS-WiFi. We considered the Web browsing as the reference application. However, the PS-WiFi model can be used with any non real-time application.

The first step of our study is the definition of a traffic model for a typical Web user. This model is then exploited to provide closed formulas that describe the performance indexes of PS-WiFi under this load conditions. Specifically, we compare the performance of our system with those of an Indirect-TCP architecture without any power management. 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 transfer-time.

Our model is very accurate. The difference between the analytical results and the measurements on the real testbed is, on average, about 1% for the energy-saving performance index, and about 7% for the additional transfer-time. Therefore, the analytical model is a flexible tool to analyze the behavior of PS-WiFi. We used this model to better understand: (i) which parameters determine the performance of PS-WiFi; and (ii) how parameter values affect the PS-WiFi performance. Specifically, in this paper we analyze the sensitiveness of the system with respect to the main Internet parameters, i.e., the throughput and the Round Trip Time (throughout referred to as *RTT*) between the Web client and the Web server. The results show that power saving is mainly affected by throughput variations. Specifically, power saving varies from 48% to 83% when the throughput increases from 0 to ∞ . However, for typical throughput values (i.e., between 50Kbps and 1Mbps) variations of power saving are limited (between 68% and 82%). On the other hand, the additional transfer-time is a slightly increasing function of *RTT* and never affects significantly the QoS perceived by Web users, since the average additional transfer-time is always less than 0.5 sec.

The paper is organized as follows. Section 2 presents our power-saving system along with the test environment we used. In Section 3 we derive the analytical model of PS-WiFi. Section 4 is devoted to the model validation. Section 5 presents the sensitiveness analysis, and Section 6 concludes the paper.

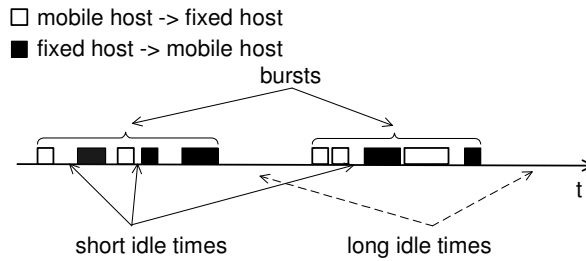


Figure 2: Snapshot of a typical best-effort data exchange.

2 The PS-WiFi system

2.1 Power-saving management of best-effort traffic

Our power-saving architecture was designed to support any network applications with the following traffic pattern: data-transfer phases are characterized by *traffic bursts* interleaved by *idle phases* (during which data are processed locally). During each burst several packets are exchanged. Packets inside a burst are separated by *short idle times*, while consecutive bursts are separated by *long idle times*. Short idle times and long idle times are generated by different phenomena. Short idle times are driven by network protocols and data processing (e.g., a TCP-sender waiting for receiving a acks after sending segments), while long idle times are related to human times (e.g., a new Web page is requested after the user has read the previous one). Due to their different nature, short idle times are much smaller than long idle times, and 1 *sec* is typically assumed as the cut-off value between the two classes. Figure 2 shows a snapshot of a typical data exchange. It should be noted that idle times are measured assuming the mobile device perspective: an idle time starts whenever the mobile device has no more packets to exchange, and finishes when the mobile device sends or receives a new packet.

The ideal power management would switch off the wireless interface during idle times and resuming it whenever a new packet is ready to be exchanged. Two factors make this ideal policy unfeasible in practice. First, the duration of an idle time – i.e., the instant in time when the wireless interface must be resumed – is unknown a priori. Second, the wireless interface has a switching-on transient time (throughout referred to as t_{so}) during which it consumes energy but can not communicate. Hence, for idle times less than t_{so} , it is energetically convenient to leave the network interface on. To overcome these challenges, the PS-WiFi approach is based on a dynamic estimate of the duration of idle times. Specifically, i) we measure at run time the duration of idle times; ii) we use these measurements to predict the length of the next idle time; and iii) we use this next idle-time prediction to decide

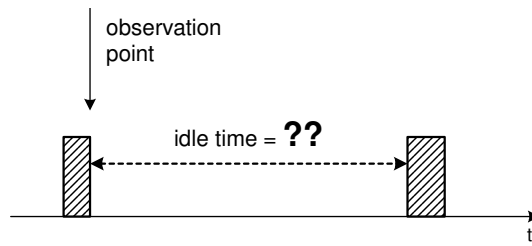


Figure 3: PSA scenario

whether the wireless interface should be switched off or not. If the wireless interface is switched off, the next idle-time prediction is also used to decide when to resume it. Therefore, the core component of PS-WiFi is a set of algorithms for predicting the duration of the next idle time.

As short and long idle times are generated by different phenomena, PS-WiFi includes two distinct algorithms for estimating them. Short idle times are estimated by means of the Variable-share Update Algorithm (VUA) [21]. On the other hand, long idle times are estimated by means of a binary exponential backoff policy (see below). It must be noted that the estimator accuracy is very important for short idle times, while it is less crucial for long idle times. Long idle times typically finish when the user of the mobile device sends a message to the fixed host. Hence, the first packet after a long idle time is sent by the mobile device to the fixed host (see Figure 2). Therefore, overestimates of the long idle times have no impact on performance since the mobile device resumes the wireless interface as soon as a new packet is generated. On the other hand, packets inside a burst also travel on the opposite direction (i.e, from the fixed host to the mobile device). In this case, if the mobile device switches the wireless interface off it is not aware when packets coming from the Web server arrive and becomes aware only when it resumes the wireless interface, after the *estimated* idle time has elapsed (see the next section for details). Therefore, an overestimate may increase the transfer time experienced by that packet. For these reasons, PS-WiFi includes a very accurate estimator for short idle times (VUA), and a simpler estimator for long idle times.

Based on these two estimators, we have designed a Power Saving Algorithm (PSA). Let us assume that idle times are detected by PS-WiFi as soon as they start. Then, the question to answer is: “How can we estimate the length of that idle time?” (Figure 3 gives a graphical representation of this scenario). When an idle time occurs, we have just received a packet, and hence we are likely to be inside a burst. Thus, a *short* idle time is assumed, and VUA provides an estimate, say t' , of the (supposed) short idle time. If this estimate occurs to be too short, an update must be provided¹. To this

¹The way PS-WiFi detects that an estimate is too short depends on the implementation in the network architecture,

end, by exploiting the distribution of short idle times estimated from the history, the 90th percentile (throughout referred to as k) is used as the updated estimate. As t' seconds have already elapsed, the new packet is expected within the next $k - t'$ seconds. If a new packet is still not available after this time interval, the idle time is greater than k seconds. This means that the probability of the idle time being a short idle time is below 10%, and hence a *long* idle time is assumed. The estimate is set to the the minimum possible value of long idle times², i.e., 1 *sec*. As k seconds have already elapsed, the new packet is expected within the next $1 - k$ seconds. If needed, further updates are generated using a binary exponential backoff procedure³, until the long idle time finishes.

When the idle time ends, if it was a short idle time, then its value is provided to VUA to update its parameters.

To summarize, PSA exploits both estimators of short and long idle times. If $u^{(i)}$ denotes the sequence of estimates provided by PSA when an idle time occurs, and $z^{(i)}$ denotes the corresponding sequences of intervals within which new packets are expected (see Figure 4), the following equations hold:

$$\left\{ \begin{array}{l} u^{(0)} = t' \\ u^{(1)} = k \\ u^{(2)} = 1 \\ u^{(3)} = 2 \\ u^{(4)} = 4 \\ \dots \\ u^{(n)} = 2^{n-2} \end{array} \right. , \left\{ \begin{array}{l} z^{(0)} = t' \\ z^{(1)} = k - t' \\ z^{(2)} = 1 - k \\ z^{(3)} = 1 \\ z^{(4)} = 2 \\ \dots \\ z^{(n)} = u^{(n)} - u^{(n-1)} \end{array} \right. . \quad (1)$$

Specifically, $u^{(0)}$ and $u^{(1)}$ are provided by VUA, while $u^{(2)}, \dots, u^{(n)}$ are provided by the long idle-time estimator. It is worthwhile noting that PSA is memory-less, in the sense that both $u^{(i)}$ and $z^{(i)}$ related to a particular idle time are independent of sequences related to previous idle times.

Finally, it should be noted that the assumption that idle times are detected by PS-WiFi as soon as they starts may not hold. For example, due to an overestimate of the previous idle times, the PSA may be executed when the idle time has already started. We will explain how PSA operates in this case.

and hence it is discussed in the next section.

²1 *sec* corresponds also to the upper bound of the short idle times.

³Updates are equal to 2^k , with $k = 1, 2, 3, \dots$

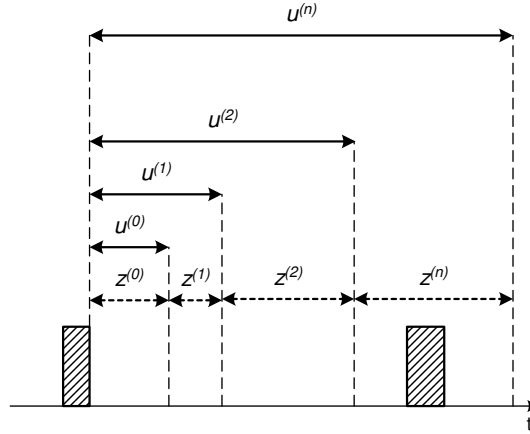


Figure 4: Sequence of updates provided by PSA to estimate an idle times

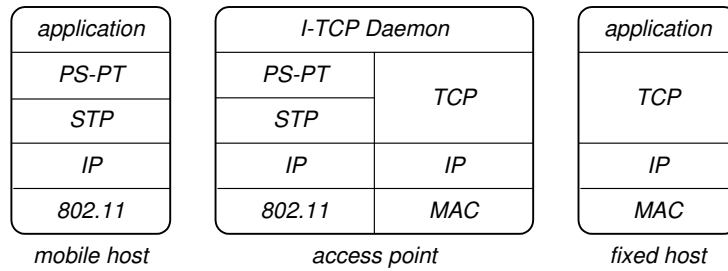


Figure 5: PS-WiFi network architecture.

2.2 Network architecture

To integrate PSA in the Wi-Fi hotspot scenario, we define the network architecture shown in Figure 5. This power-saving architecture exploits the Indirect-TCP model: it splits the transport connection between the (mobile) client and server at the Access Point. The *Indirect-TCP (I-TCP) Daemon* at the Access Point relays data between the two parts of the splitted connection. In addition, the transport protocol between the mobile and the fixed host is a *Simplified Transport Protocol (STP)* tailored to the wireless link characteristics. As shown in [9], using an “ad hoc” transport protocol improves the performance of the Indirect-TCP architecture [8].

We implement the PSA in the *Power-Saving Packet Transfer (PS-PT)* protocol. As a design choice, the PSA is completely executed at the Access Point. Hence, the impact of the power-saving system on the mobile-device computing resources is negligible. In detail, PS-WiFi works as follows. When there are no more data to be exchanged on the WLAN (i.e., when an idle time occurs), the PS-PT module at the Access Point generates a short idle-time estimate (i.e., $u^{(0)}$) and decides whether it is convenient to the mobile device to switch the network interface off (i.e., if $u^{(0)}$ is greater than t_{so}). If so, it sends a “shutdown” command to the mobile device including an indication of the time

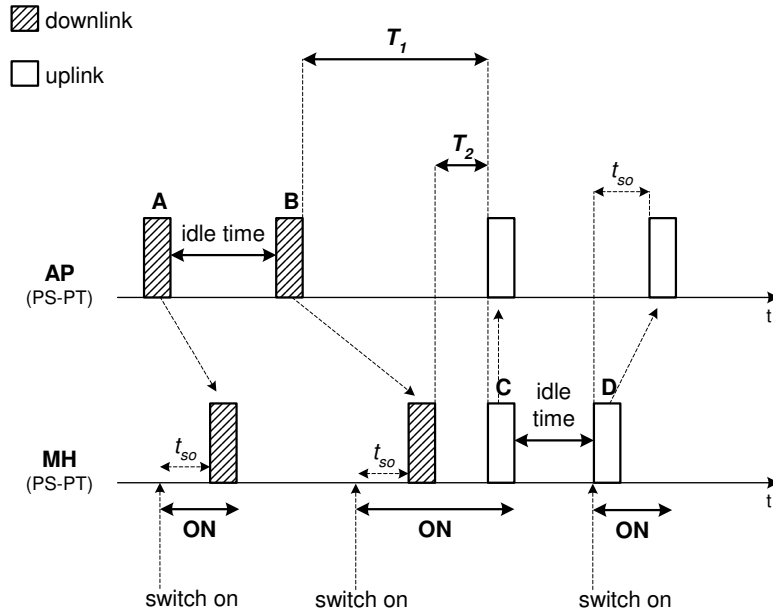


Figure 6: Example of packets exchange in a PS-WiFi system (focus on idle-time measurements).

interval during which it should remain disconnected. The data addressed to the mobile device that may become available while it is disconnected are buffered at the Access Point. When the mobile device reconnects, it polls the Access Point to obtain queued data. If no data are available, the Access Point updates the idle time estimate (i.e., $u^{(1)}$, $u^{(2)}$, and so on), and decides if it is convenient to the mobile device to switch the wireless interface off (i.e., $z^{(i)}$ is greater than t_{so}). Finally, if the mobile device generates new data while it is disconnected, the wireless interface is immediately switched on and data are sent.

To complete the description of PS-WiFi we discuss how idle times are measured. For ease of understanding we assumed so far (see Section 2.1) that idle times are measured at the mobile device. However, measuring idle times at the mobile device is not always the right choice. Let us focus on the example presented in Figure 6 (for simplicity in the figure we neglect transfer times on the WLAN; in addition, time intervals during which the wireless interface is on are defined by PSA). Due to possible overestimation, packets A and B may be delayed at the Access Point and, hence, the idle time between A and B measured by the mobile device is affected by the estimate error. In the ideal case, PS-WiFi should be transparent to the traffic generated by the applications, i.e., it should not modify idle times with respect to the case when no power management is used. Hence, idle-time measures should not include additional components introduced by PS-WiFi. Thus, idle times between consecutive packets in the downlink direction (e.g., A and B in Figure 6) are measured at the Access Point. On the other

hand, consecutive packets in the uplink direction (e.g., C and D in Figure 6) are measured at the mobile device. Indeed, t_{so} seconds are added by PS-WiFi to that idle times, if the second packet (i.e., D) is generated when the wireless interface is off. Measuring idle times between packets flowing in opposite directions (e.g., B and C in Figure 6) requires a mixed approach. To clarify this concept, let us focus on packets B and C in Figure 6. If we assume that C is generated independently by B (e.g., B and C are related to different applications running at the mobile device concurrently), then the idle time to be measured is T_1 . However, the Access Point might measure $T_1 + t_{so}$ if the wireless interface was off when C was generated, while the mobile device would measure T_2 . The right idle-time value could be measured by i) synchronizing the mobile device with the Access Point⁴; ii) recording the time instant when B arrived at the Access Point; and iii) recording the time instant when C arrived at the mobile device. It must be noted that this strategy is no longer correct if C is generated by the mobile device in response to B, since in this case the right idle time would be T_2 . However, typical applications used in WiFi hotspot (e.g., Web browsing, e-mail, file transfer) generate a traffic that is *almost mono-directional*, and, hence, consecutive packets related to the same application flowing in opposite directions can be considered as an exception. To summarize, idle times are measured by PS-WiFi as follows: i) the mobile device and the Access Point are assumed to be synchronized, ii) the instant in time when each packet arrives at the PS-PT layer is recorded, and iii) idle times are measured as the difference between those time instants related to consecutive packets.

The strategy used to measure idle times also impacts on the way idle-time estimates are used. Specifically, since idle times are seen as inter-arrival times, idle-time estimates must be considered accordingly. Let us focus on Figure 7, where u_j denotes the final update provided by PS-WiFi for the j -th idle time (u_j is an item of the sequence $u^{(i)}$ shown in Equation 1). Packet A is sent to the mobile device when the estimate u_0 has elapsed. After A has been sent (point K in the figure) no more data is available and, hence, PSA is invoked. As idle times are inter-arrival times measured at the PS-PT layer, estimated idle times must start at the point in time when the previous packet (A in our example) has arrived at the PS-PT layer. Therefore, in our example, the estimated idle time (i.e., u_1) starts at time H , though the estimate is generated at time K . It should be noted that PSA may update the first estimate related to an idle time before sending it to the mobile device. The Access Point exploits the knowledge that the idle time is at least $K - H$. Hence, when packet A is sent, PSA

⁴Please note that in a 802.11 WLAN the MAC layer requires synchronization and, hence, synchronizing the PS-PT layers has no cost.

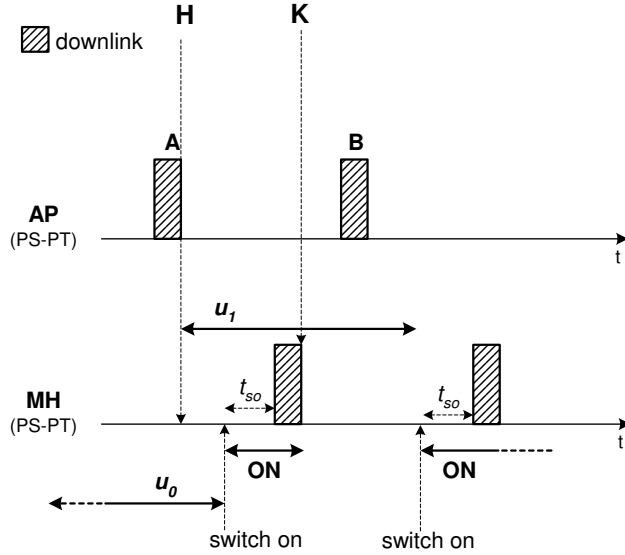


Figure 7: Use case of idle-time estimates.

considers the first item in the sequence $u^{(i)}$ that is greater than $K - H$, and sends this estimate to the mobile device along with packet A (see Figure 7)⁵.

As a final remark, it is worth noting that PS-WiFi approximates the ideal power-saving strategy for the wireless interface: i) data transfers on the wireless link occur at the maximum available throughput on it and are not affected by the throughput on the (wired) Internet; and ii) the wireless interface remains switched off during idle periods.

3 Modeling PS-WiFi behavior

To analyze the performance of PS-WiFi, we compare our system to an Indirect-TCP system without power management. In the latter case, we used the same architecture shown in Figure 5, without the PS-PT layer. The performance in terms of power-saving are evaluated by means of the I_{ps} index:

$$I_{ps} = \frac{C_{ps}}{C_{I-TCP}}, \quad (2)$$

where C_{ps} is the energy spent to transfer a given set of data when using the power-saving system, and C_{I-TCP} is the energy spent to transfer the same data when using the Indirect-TCP without power management.

Due to the 802.11 wireless interface consumption patterns, the energy spent by the wireless in-

⁵Clearly, the estimate is sent only if the residual (estimated) idle time is greater than t_{so} .

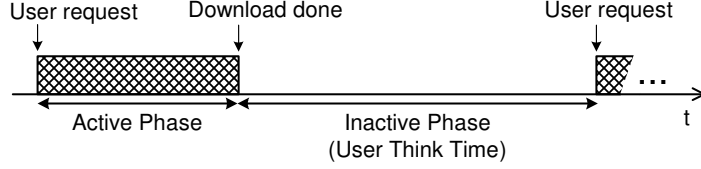


Figure 8: Scheme of the Web traffic as composed by Active and Inactive phases.

terface of the mobile device is almost proportional to the time interval during which the wireless interface remains on [17]. Therefore, we express both C_{PS} and C_{I-TCP} in seconds.

As shown above, PS-WiFi may introduce an additional transfer-time (i.e., a delay) to packets exchanged between the mobile device and the fixed host. Users might perceive these delays as a degradation of the QoS and, hence, it is important to minimize them. It must be pointed out that the right index to measure the QoS degradation depends on the particular application, and the single-packet delay is not the best performance delay figure. Let us focus, for example, on Web users, and let us define the User Response Time (URT) as the time elapsed to download a Web page, i.e., the time interval between the user request and the page rendering at the mobile device. For this specific application, the QoS degradation introduced by PS-WiFi is better represented by the additional URT rather than by the additional delay related to a single packet. Since in our studies we used the Web as the reference application, we measured the QoS degradation perceived by the user by means of the I_{pd} index:

$$I_{pd} = URT_{ps} - URT_{I-TCP} , \quad (3)$$

where URT_{ps} is the User Response Time in the PS-WiFi system, and URT_{I-TCP} is the User Response Time in the Indirect-TCP architecture.

The description of PS-WiFi (see Section 2) shows that several parameters affect the system behavior: the throughputs on the wireless and wired networks, the accuracy of the idle-time estimates, the application traffic profile, etc. To better understand their influence on the system performance, in the next section we present an analytical model of PS-WiFi. By solving this model we derive closed formulas for I_{ps} and I_{pd} .

3.1 Web-traffic model

Since the system performance is related to the application-level traffic, it is necessary a preliminary characterization of Web traffic. In this paper we exploit the model used in the SURGE simulator

Definition	Symbol	Value	Unit
Probability that a Web page contains embedded files	p_{emb}	0.44	-
Average number of embedded files in a Web page	\bar{N}_{emb}	1.50	-
Average size of an embedded file	\bar{D}_{emb}	6348	bytes
Average size of a main file	\bar{D}_{mf}	17496	bytes
Average User-Think-Time length	\overline{UTT}	3.25	sec

Table 1: Parameters defining the Web-traffic profile.

[10, 11]. SURGE creates (off-line) a set of Web pages to be stored in a real Web-server. Furthermore, it simulates a typical Web-user that downloads these Web-pages from real (mobile) hosts, and defines the sequence of Web-page downloads in such a way that the traffic generated by the simulated user meets the statistical model of Web traffic presented in [10, 11]. Hereafter, we exploit this model to characterize the traffic generated by a Web user. Figure 8 and Table 1 describe the traffic model and related parameters derived from SURGE.

Let us focus on a Web-page download. A Web page consists of a main file and zero or more embedded files (e.g., figures). The user requests the Web page to the browser, which downloads all the files from the Web server. When the download is complete, the user reads the contents of the Web page, and then issues another request to the browser. Hence, the Web traffic can be modeled as the sequence of consecutive Web-page downloads. The traffic related to each Web page presents an Active Phase during which the browser downloads the files from the Web server, and an Inactive Phase (or User Think Time, UTT) during which the user reads the contents of the Web page. While the Inactive Phase is completely characterized by the UTT parameter, the Active Phase depends on parameters related to the Web pages. Specifically, the probability that a Web page contains embedded files is hereafter referred to as p_{emb} . Furthermore, the average number of embedded files contained in a Web page⁶ is referred to as \bar{N}_{emb} . Finally, the average size of a main file is \bar{D}_{mf} , while the average size of an embedded file is \bar{D}_{emb} .

In this work we exploit these parameters (summarized in Table 1) to define the reference application-level traffic (a pictorial representation is provided in Figure 9). In our model the Web user downloads continuously a set of Web pages, referred to as *basic block*. Between two consecutive downloads, the user waits \overline{UTT} seconds. The pages that compose the basic block are defined to meet the parameters p_{emb} , \bar{N}_{emb} , \bar{D}_{mf} and \bar{D}_{emb} . Specifically, in our model the first page contains embedded files, while the other pages do not. The average total size of the embedded files must be $\bar{N}_{emb} \cdot \bar{D}_{emb}$, and the number

⁶ \bar{N}_{emb} is evaluated by considering only Web pages that contain embedded files.

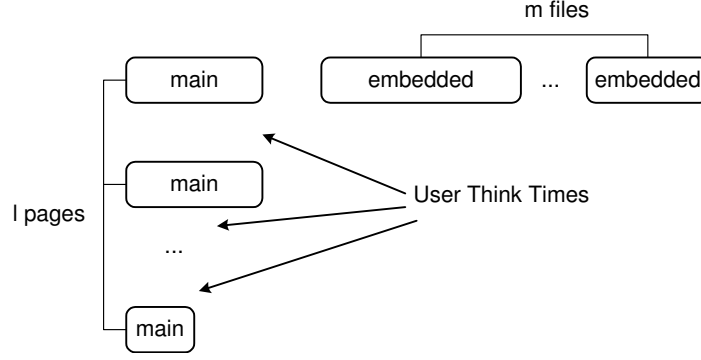


Figure 9: Scheme of the basic block.

of embedded files must be an integer number. To this end, in our model the first page contains m embedded files, where m is equal to $\lceil \bar{N}_{emb} \rceil$. The size of the first $m - 1$ files is \bar{D}_{emb} , while the size of the last file is $\bar{D}_{emb} \cdot (m - \bar{N}_{emb})$. Furthermore, as just the first page contains embedded files, the number of Web pages composing the basic block should be $1/p_{emb}$. Moreover, the total size of the basic-block pages must be $\bar{D}_{mf} \cdot (1/p_{emb}) + \bar{D}_{emb} \cdot \bar{N}_{emb}$, i.e., the total size of the main files must be $\bar{D}_{mf} \cdot (1/p_{emb})$. However, since the number of Web pages must be an integer number, the basic block contains l Web pages, where l is equal to $\lceil 1/p_{emb} \rceil$. The main files of the first $l - 1$ pages are long \bar{D}_{mf} bytes, while the main file of the last page is long $\bar{D}_{mf} \cdot (l - 1/p_{emb})$ bytes.

In conclusion, in our model a Web user downloads continuously the Web pages contained in the basic block, interleaving each download with a User Think Time. The User Think Time and the basic block definitions guarantee that the application-level traffic has the same average statistics of traffic generated by the SURGE simulator [10, 11]. Therefore, hereafter we characterize the PS-WiFi behavior by focusing on the download of a single basic block.

3.1.1 Idle times characterization

As discussed in Section 2.1, PS-WiFi predicts idle-time lengths occurring in the application-level traffic, and manages the wireless interface of the mobile device accordingly. Therefore, at this point, we need to characterize the idle times of the Web traffic we have modeled.

Let us focus on Figure 8. Clearly, a first class of idle times is represented by User Think Times. By recalling the definition of long idle time (Section 2.1), it is easy to show that User Think Times fall in this category. Furthermore, we need to identify idle times that may occur during the Active Phases. First of all, it should be noted that during Active Phases the browser and the Web server exchange data without user interventions, and hence idle times must be seen as *short* idle times. By

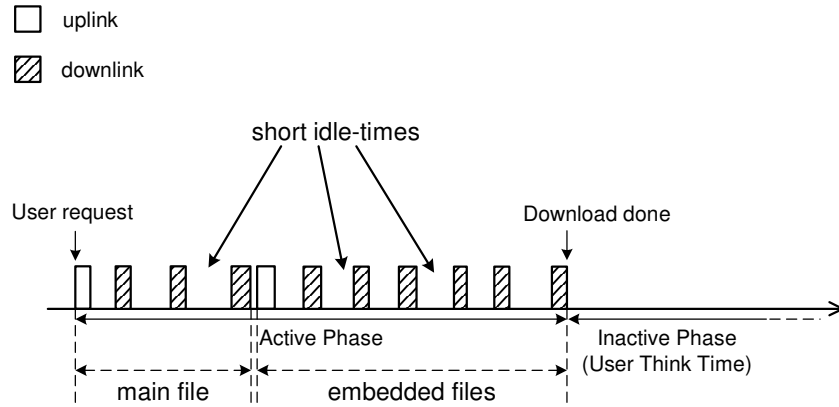


Figure 10: Application-level traffic during the Active Phase

definition, short idle times are related to the behavior of the network protocols, therefore we have to recall which network protocols are used by the Web applications, and how they work in the PS-WiFi architecture (Figure 5). Today, Web uses the HTTP/1.1 as the application-level protocol. In our model, we assume that the browser at the mobile device downloads a Web page in two steps (see Figure 10): i) the main file is downloaded first, and the list of the embedded files is extracted from it; ii) then, all the embedded files are requested (and downloaded) together⁷. In each step, the shape of the application-level traffic is defined by the underlying protocols, and hence, by the joint effect of PS-PT, STP and TCP. However, as appears from Figure 10, the Web traffic is almost mono-directional, in the downlink direction. Therefore, in our model we consider only idle times between packets flowing in the downlink direction, and hence, as discussed in Section 2.2, we have to characterize these idle times as observed at the Access Point. Moreover, thanks to the Indirect-TCP architecture, the behavior of the TCP between the Access Point and the fixed host is completely independent of whatever protocol running between the Access Point and the mobile device (i.e., the STP and the PS-PT). Therefore, we conclude that in our model short-idle times are determined only by the behavior of the TCP between the Access Point and the fixed host. To characterize the TCP behavior, it is worth noting that i) HTTP/1.1 uses persistent connections, i.e., the same transport connection can be used to download several files sequentially [22]; ii) Web servers autonomously close persistent TCP connections that remain idle for 15 seconds, and utilize the same TCP connection to serve up to 150 HTTP Requests [6]; and iii) in the model defined by SURGE, User Think Times are less than 15 seconds with very high probability (98%). Therefore, in our model all Web pages composing the basic

⁷This assumption relies on the fact that HTTP/1.1 allows pipelining requests, i.e., consecutive HTTP Requests can be sent back-to-back.

Definition	Symbol
Total size of the basic block	B
Number of Web pages in the basic block	l
Average throughput on the wired Internet	$\bar{\gamma}$
Average throughput on the WLAN	$\bar{\gamma}_{wl}$
Energy spent during idle times when using PS-WiFi	C_{it}
Time spent in the idle mode when using PS-WiFi	T_{ON}^{idle}
Transient interval of the wireless interface to switch on	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 switching-on events during a long idle time, before the backoff procedure starts	F
Random variable measuring short idle-time lengths	t
Random variable measuring initial estimates of short idle-time lengths	t'
Upper bound of the short idle-time distribution	M
90 th percentile of short idle-time lengths	k
Delay added by PS-WiFi to a packet flowing in the downlink direction	d

Table 2: Symbols used throughout the paper.

block are downloaded by means of a single TCP connection. Moreover, we neglect possible slow-start phases, and hence we assume that the TCP connection is in steady-state. Under this hypothesis, the TCP behavior can be modeled as follows [32, 35]: i) once every RTT the TCP at the Web server sends a fixed number of back-to-back TCP segments⁸; and ii) these back-to-back TCP segments arrive at the Access Point together. Therefore, in our model we assume that short idle times during Active Phases corresponds to Round Trip Times between the Access Point and the Web server.

3.2 Energy Consumption Modeling

We are now in the position to evaluate the energy spent to download a single basic block, by using either PS-WiFi or the Indirect-TCP architecture without any power management. Hereafter, we assume that the network status is stationary during the download of the basic block (i.e., the throughput and the RTT are stationary). Furthermore, we develop a model of the system behavior in the *average* case, and hence the quantities we derive must be intended as average values. For easy of reading, we summarize in Table 2 the parameters used hereafter.

The energy spent when using an Indirect-TCP architecture without power management (i.e., C_{I-TCP}) is (proportional to) the total time required to download the basic block. Therefore, the

⁸The number of TCP segment is defined by the average size of the congestion window.

following theorem holds.

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

$$C_{I-TCP} = \frac{B}{\bar{\gamma}} + l \cdot \overline{UTT} = \frac{\overline{D}_{mf} \cdot 1 / p_{emb} + \overline{N}_{emb} \cdot \overline{D}_{emb}}{\bar{\gamma}} + l \cdot \overline{UTT}, \quad (4)$$

where B is the total size (in bytes) of the basic block, and $\bar{\gamma}$ is the average throughput of the (wired) Internet.

Proof: C_{I-TCP} is the time spent to download the basic block, i.e., the time spent downloading the actual data, and the User Think Times. The first term is the ratio between the basic-block size (i.e., B) and the average throughput experienced by the client. It is worth recalling that in an Indirect-TCP architecture, the throughput achieved at the transport layer is the minimum throughput of the two connections. Thus, if we assume that the fixed Internet is the bottleneck between the server and the client, the average throughput experienced by the client is the throughput of the fixed Internet, i.e., $\bar{\gamma}$. The second term is the sum of the User Think Times occurring within the basic block, i.e., $l \cdot \overline{UTT}$. The final form of C_{I-TCP} follows from the basic-block definition provided in Section 3.1. ■

Following the scheme in Figure 2, the energy spent using PS-WiFi (i.e., C_{ps}) can be seen as made-up of two components. The first one is related to actual transfers of data composing the basic block. The other one is the energy spent during idle times, throughout referred to as C_{it} . Since idle times are managed as shown in Section 2, two factors impact on C_{it} : i) every time the network interface is shut down, t_{so} seconds are paid the next time it is switched on, and ii) when an idle-time estimate (more precisely, an item in the sequence $z^{(i)}$, see Equation 1) occurs to be less than t_{so} , the network interface remains in the idle state⁹. Therefore, the following proposition holds.

Proposition 1: The energy spent by using PS-WiFi is

$$C_{ps} = \frac{B}{\bar{\gamma}_{wl}} + C_{it} = \frac{B}{\bar{\gamma}_{wl}} + t_{so} \cdot S + T_{ON}^{idle}, \quad (5)$$

where: i) $\bar{\gamma}_{wl}$ is the average throughput available on the WLAN; ii) S is the average number of times the mobile-device wireless interface is switched on during the basic-block download; and iii) T_{ON}^{idle} is the average time during which the mobile-device wireless interface remains idle when PS-WiFi is adopted.

⁹Since the RTT between the Access Point and the mobile device is typically negligible, and due to the simplicity of the PS-PT protocol [4], the overhead of the PS-PT protocol is assumed to be negligible.

Proof: Transfers between the Access Point and the mobile device occur at the throughput available on the WLAN. Hence, the energy spent to receive the basic block is $B/\bar{\gamma}_{wl}$. Any other contribution is related to idle times (see Figure 2), and is thus comprised in C_{it} . Whenever an idle time occurs, PS-WiFi behaves as shown in Section 2. Two kinds of events can happen, i.e., i) the wireless interface is shut down (one or more times), and ii) the wireless interface is let idle, because the (residual) idle time is supposed to be too short. Hence, a first component of C_{it} is defined by the average number of switching-on events occurring during the basic-block download (S), and a second component is the total time during which the wireless interface remains idle (T_{ON}^{idle}). ■

Below we derive a closed form of Equation 5, under the assumption of T_{ON}^{idle} being negligible. This hypothesis is true in the experimental testbed used for the validation (Section 4). In general, it can be shown that this assumption holds if the average value of the short idle-time estimates is greater than t_{so} . Otherwise, our analysis provides optimistic results, i.e., the upper bound of the energy saved with PS-WiFi. The goal of the following analysis is deriving a closed form for S . To this end, we have now to make some assumptions explicit. Hereafter, we assume that short (long) idle-times are i.i.d. random variables. Moreover, we assume that a short (long) idle-time is independent of any previous long (short) idle-time. Furthermore, we assume that VUA is precise, and hence short idle-time estimates follow the same distribution of real short idle times. Finally, we assume that the sequence of estimates (i.e., $u^{(i)}$, Equation 1) related to an idle time is independent of the sequences related to previous idle times. To clarify this assumption, let us focus on Figure 7. In that example, the estimate of the idle time between packets A and B is evaluated at time K , and hence PSA evaluates the first item in the sequence $u^{(i)}$ being greater than $K - H$. In other words, that estimate depends on the previous estimate error. In our model we neglect this dependence, and assume that estimates are generated at the point in time when the idle time starts (i.e., H in the example). Since this assumption leads to considering a greater number of switching-on events with respect to the real case, the closed form of S that we derive is an overestimate of the real value. According to these assumptions, we conclude that in our model PS-WiFi regenerates with respect to all points in time when an idle time starts. Hence, to characterize its behavior, we focus on the beginning of an idle time, and analyze the two possible events that may occur: a short idle time or a long idle time. The main result of this analysis is the closed form of C_{ps} provided by Theorem 2. For ease of reading, proofs are postponed to Appendix A.

Theorem 2: The energy spent to download a single basic block by using PS-WiFi is

$$C_{ps} = \frac{B}{\bar{\gamma}_{wl}} + t_{so} \cdot \left\{ \frac{B}{\bar{\gamma} \cdot RTT} \cdot S_1 + l \cdot (F + \lceil \log_2 \overline{URT} \rceil) + p(u^{(0)} > t_{so}) \right\}, \quad (6)$$

where i) S_1 is the average number of switching-on events occurring during a short idle-time; ii) F is the number of switching-on events occurring during a long idle-time, before the long idle-time estimator is invoked (i.e., after $u^{(2)}$ of Equation 1 is generated); and iii) $p(u^{(0)} > t_{so})$ is the probability of $u^{(0)}$ being greater than t_{so} .

3.3 Modeling the I_{pd} index

The I_{pd} index measures the additional URT introduced by PS-WiFi to download a Web page. To characterize I_{pd} , it is worth focusing on Figure 10. Let us introduce the definition of a *transaction* between the Web client and the Web server. A transaction denotes a piece of Web traffic starting with one or more HTTP Requests sent by the client back-to-back, and including all packets sent by the Web server in response to that HTTP Request(s). Hence, the download of a Web page can be modeled as the sequence of two transactions, where i) the main file is downloaded during the first transaction, and ii) all the embedded files are downloaded during the second transaction. Let us now analyze the delay introduced by PS-WiFi to a transaction (see Figure 11). When the HTTP Request(s) is generated by the browser, the wireless interface of the mobile device may be shut down. In this case, the HTTP Request(s) is delayed of t_{so} seconds. A further delay can be added at the end of the transaction. As noted in Section 3.1.1, the inter-arrival times between packets sent by the Web server depend only on the TCP protocol between the Access Point and the fixed host. In other words, the time instants when these packets arrive at the Access Point are the same whether PS-WiFi is used or not. Therefore, a further delay is added by PS-WiFi if the *last* packet of the transaction is delayed at the Access Point. Hereafter we neglect the possibility that two consecutive packets (e.g., A and B in Figure 11) are transferred back-to-back on the WLAN, due to overestimation of the previous idle time (e.g., T in Figure 11). That is, *all* short idle times are detected and estimated by PS-WiFi. Therefore, a further delay can be introduced to the transaction only as a side effect of the estimate error related to the last short idle time. This delay is hereafter referred to as d .

The analysis of the delay introduced to a transaction allows us to characterize the I_{pd} index. Specifically, from Figure 3.1.1 it appears that the embedded files are requested only when the main

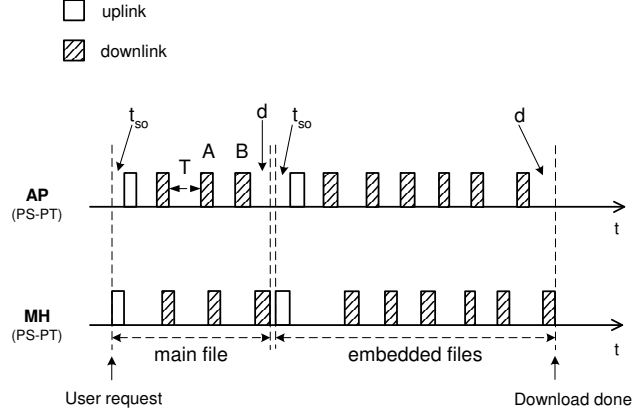


Figure 11: Components of the additional URT.

file is completely downloaded at the mobile device, and hence the second transaction starts only after the last packet of the first transaction arrives at the mobile device. Therefore, I_{pd} can be evaluated as the sum of the delays introduced to each transaction. A closed form for the average value of I_{pd} is provided by the following theorem.

Theorem 3: The average URT introduced by PS-WiFi to download a Web page is

$$\overline{I_{pd}} = (t_{so} + \bar{d}) + \left\{ t_{so} \cdot p \left(u^{(0)} > t_{so} \right) + \bar{d} \right\} \cdot p_{emb} , \quad (7)$$

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

Proof: In our model (since User Think Times are assumed to be larger than 1 second) the transaction related to the main file starts when the wireless interface is switched off. Hence, $t_{so} + \bar{d}$ is the average additional delay introduced to the main-file transaction. If the Web-page contains embedded files (i.e., with a probability equal to p_{emb}), the delay related to the embedded-files transaction must be included. A delay equal (on average) to \bar{d} is introduced at the end of the transaction. A further delay equal to t_{so} may be introduced if the transaction starts when the wireless interface is shut down. The probability that this occurs can be derived as follows. When the last packet of the main file is sent to the mobile device, no more data are available to be exchanged on the WLAN. Hence an idle time is detected, and PS-WiFi generates $u^{(0)}$ as its (initial) estimate. However, the browser generates the HTTP Requests for the embedded files immediately, and the second transaction starts. Therefore, the probability that the wireless interface is shut down at this point in time is the probability of $u^{(0)}$ being greater than t_{so} . ■

The evaluation of $\overline{I_{pd}}$ requires the characterization of d . In the following section, we sketch the line

of reasoning used to model this quantity, and we provide a closed form for it. For ease of reading, the detailed derivation is presented in Appendix C.

3.3.1 Analytical model of d

To evaluate $\overline{I_{pd}}$ we need to characterize d , that is the additional delay PS-WiFi introduces to the last packet of a transaction, due to overestimation of the previous short idle time. As PS-WiFi regenerates with respect to the point in time when an idle time starts, we can analyze d by focusing on the PS-WiFi behavior on a generic short idle time, followed by a packet addressed to the mobile device (i.e., flowing in the downlink direction). To this end, we introduce some assumptions that we exploit hereafter. The estimator of short idle times, is based on the VUA. As noted in Section 3.2, VUA allows us to assume that short idle times and their estimates follow the same distribution. In the following, t denotes a random variable measuring short idle times, and t' denotes a random variable measuring the estimates provided by VUA (i.e., t' is equal to $u^{(0)}$ of Equation 1). We assume that i) t and t' follow a uniform distribution between 0 and a maximum value, M (i.e., $t \sim t' \in \mathcal{U}[0, M]$); and ii) t and t' are independent. Moreover, based on the characterization given in Section 3.1.1, we instantiate M to two times the average RTT between the mobile device and the Web server, i.e., $M = 2 \cdot \overline{RTT}$. However, to make our analytical approach flexible, hereafter we provide the delay model as a function of M .

The average value of d can be evaluated as the contribution of two components. Specifically, it is the sum of i) the average delay when the initial estimate is too large (i.e., when $t' > t$); and ii) the average delay when the initial estimate is too short (i.e., when $t' < t$). Since t' and t are distributed according to the same law, both $p(t' > t)$ and $p(t' < t)$ are equal to $1/2$. These remarks allow to prove the following proposition.

Proposition 2: The average delay added by PS-WiFi to a packet flowing in the downlink direction can be expressed as

$$\begin{aligned} \overline{d} = E[d] &= E[d|t' > t] \cdot p(t' > t) + E[d|t' < t] \cdot p(t' < t) = \\ &= \frac{1}{2} \cdot (E[d|t' > t] + E[d|t' < t]) . \end{aligned}$$

The rest of the analysis is devoted to deriving $E[d|t' > t]$ and $E[d|t' < t]$. This is achieved by analyzing the behavior of PS-WiFi in both cases. As this task just requires simple (but quite long) algebraic manipulations, here we only provide the final results, while proofs are postponed to Appendix

C.

Lemma 1: The average delay when t' is greater than t is

$$E [d | t' > t] = \frac{M^2 - t_{so}^2}{4M} .$$

Lemma 2: The average delay when t' is less than t is

$$E [d | t' < t] = 0.9 \cdot \frac{k^2 - t_{so}^2}{4M} + 0.1 \cdot \frac{2sec - M - k}{2} \cdot \chi(k, t_{so}) ,$$

where $\chi(k, t_{so})$ is an indicator function, defined as

$$\chi(k, t_{so}) = \begin{cases} 1 & \text{if } 1sec - k > t_{so} \\ 0 & \text{otherwise} \end{cases} . \quad (8)$$

Finally, the above lemmas allow us to provide a closed form for \bar{d} , as follows.

Theorem 4: The average delay added by PS-WiFi to a packet flowing in the downlink direction is

$$\begin{aligned} \bar{d} = & \frac{1}{2} \left(\frac{M^2 - t_{so}^2}{4M} \cdot u(M, t_{so}) + 0.9 \cdot \frac{k^2 - t_{so}^2}{4M} \cdot u(k, t_{so}) + \right. \\ & \left. + 0.1 \cdot \frac{2sec - M - k}{2} \cdot \chi(k, t_{so}) \right) , \end{aligned} \quad (9)$$

where $u(x, y)$ is the classical step function:

$$u(x, y) = \begin{cases} 1 & \text{if } x \geq y \\ 0 & \text{otherwise} \end{cases} . \quad (10)$$

4 Model validation

To validate the analytical model derived in Section 3.1 we compare its predictions with measurements taken from a real Internet prototype (details about the prototype can be found in [4]). Specifically, we compare the I_{ps} and $\overline{I_{pd}}$ measurements presented in [4] with the predictions provided by our model. To avoid fluctuations in the experimental results, we averaged measurements on one-hour windows (details about the methodology used to aggregate measurements can be found in [5]). It is worth noting that, due to the experiment setup, the only parameter that changes significantly among different hours is the throughput of the wired Internet, i.e., $\bar{\gamma}$. Indeed, i) in each experiment the application-level

Definition	Symbol	Value	Unit
Total size of the basic block	B	49264	bytes
Number of Web pages in the basic block	l	3	-
Average throughput over the wireless link	$\bar{\gamma}_{wl}$	11	Mbps
Average number of switching-on events in a short idle time	S_1	1.55	-
Average number of switching-on events in a long idle time before the backoff procedure starts	F	3	-
Probability that the initial estimate of a short idle time is greater than t_{so}	$p(u^{(0)} > t_{so})$	1	-
Network RTT between the client and the Web server	RTT	0.3	sec
Upper bound of t and t' distributions	M	0.6	sec
Switching-on transient interval of the wireless interface	t_{so}	0.1	sec
90 th percentile of t and t'	k	0.54	sec
Indicator function of k and t_{so} relative values	$\chi(k, t_{so})$	1	-

Table 3: Parameters used to validate the analytical model.

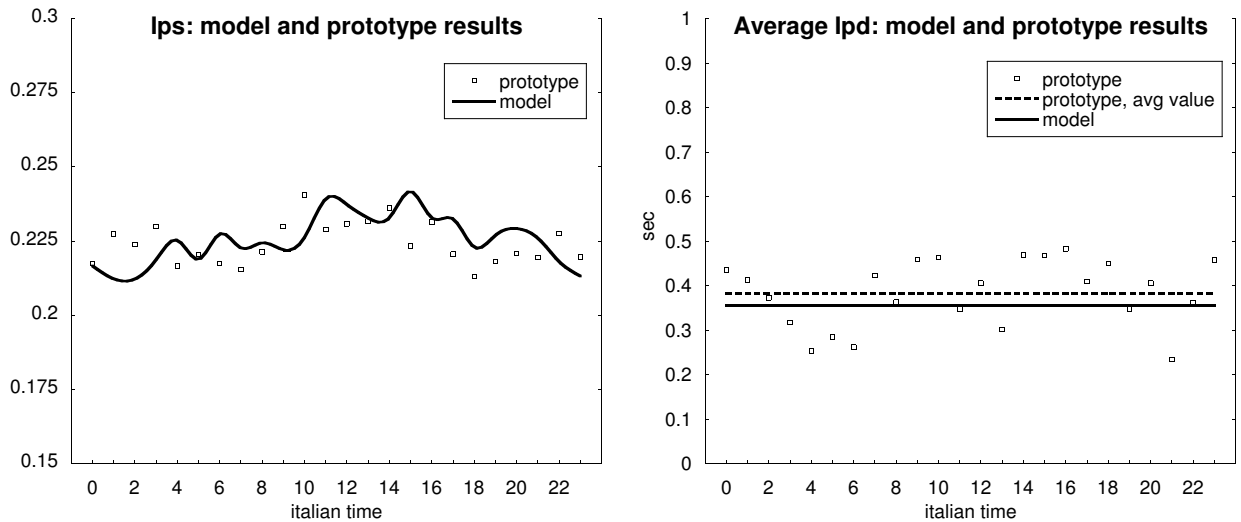


Figure 12: Hourly average I_{ps} and \bar{I}_{pd} obtained from the model and the prototype, respectively.

traffic meets the statistics shown in Table 1; ii) $\bar{\gamma}_{wl}$ and \overline{RTT} show very little fluctuations (and, thus, also M and k can be assumed to be constant); and iii) due to the assumptions about the distribution of t and t' , the parameters S_1 , F and $p(u^{(0)} > t_{so})$ in Equation 6 can be approximated as constant terms. Table 3 summarizes the values of the the parameters that we use to validate our analytical model.

Figure 12 shows the hourly average values of I_{ps} and \bar{I}_{pd} measured by using the prototype. We also plot the I_{ps} and \bar{I}_{pd} figures derived from the analytical model. 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 values of I_{ps} with the predictions of the model. Specifically, we have set $\bar{\gamma}$ in Equation 6 to the average daily throughput experienced by the prototype. The results obtained (not reported here) show that the difference between the model

results and the prototype measurements is less than 1%.

As far as the $\overline{I_{pd}}$ index (right-side plot in Figure 12), the results show that prototype values vary during the day, i.e., the prototype is sensitive to variations of $\bar{\gamma}$. Indeed, $\bar{\gamma}$ is sensitive to two parameters, i.e., i) congestions in the Internet, that reduce the TCP window size; and ii) variations of the RTT between client and server. As discussed in Section 3.3.1, 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, we compare the model prediction with the daily average value of $\overline{I_{pd}}$ (see the dashed line in the plot). The difference is about 7% of the prototype result.

5 Sensitiveness Analysis

The above results show the accuracy of our analytical model. Hereafter, we use this model to investigate the sensitiveness of PS-WiFi to two Internet key parameters, i.e. the (wired) Internet throughput (i.e., $\bar{\gamma}$) and the RTT . As noted in the previous section, the throughput depends on both the network RTT and the TCP window size. Both these parameters affect the I_{ps} index, since it depends on $\bar{\gamma}$ (see Equations 4 and 6). On the other hand, the $\overline{I_{pd}}$ index is only affected by RTT variations, as shown by Equations 7 and 9. Therefore, below we analyze I_{ps} as a function of $\bar{\gamma}$, and $\overline{I_{pd}}$ as a function of RTT .

5.1 I_{ps} sensitiveness

Based on Equations 4 and 6 we derive the I_{ps} index as a function of $\bar{\gamma}$. Specifically, after simple manipulations, $I_{ps}(\bar{\gamma})$ becomes:

$$I_{ps}(\bar{\gamma}) = \frac{a\bar{\gamma} + b}{c\bar{\gamma} + d}, \quad (11)$$

where a , b , c and d group terms that we have assumed to be constant (see Section 4):

$$\begin{cases} a \triangleq \frac{B}{\bar{\gamma}_{wl}} + t_{so} \cdot \left\{ l \cdot (F + \lceil \log_2 \overline{UTT} \rceil) + p \left(u^{(0)} > t_{so} \right) \right\} \\ b \triangleq \frac{B}{\overline{RTT}} \cdot S_1 \cdot t_{so} \\ c \triangleq l \cdot \overline{UTT} \\ d \triangleq B \end{cases} .$$

Figure 13 shows I_{ps} as a function of $\bar{\gamma}$. It clearly appears that when $\bar{\gamma}$ increases, I_{ps} decreases,

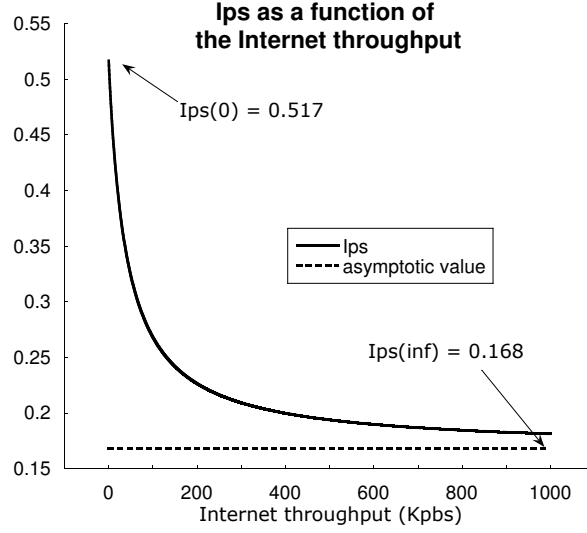


Figure 13: I_{ps} as a function of the Internet throughput $\bar{\gamma}$.

and this means that the PS-WiFi saves more energy. This result is somehow counter-intuitive, since one expects that the best power-saving is achieved when $\bar{\gamma}$ is low. In this case the overall idle time during the basic block download is at its maximum value. However, the I_{ps} behavior in the above plot can be explained as follows. As shown in Equations 4 and 6, 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 Section 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 drops, since more bytes are downloaded in a single RTT . Thus, the number of switching-on events during the basic-block download decreases, and hence, 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 of our experiments, when $\bar{\gamma}$ increases, C_{ps} decreases more than C_{I-TCP} does, and hence I_{ps} drops.

As a final remark, it is worth noting that Figure 13 highlights the theoretical lower and upper bounds of I_{ps} . In the Internet configuration that we experienced, I_{ps} ranges between 0.517 (when $\bar{\gamma} \rightarrow 0$) and 0.168 (when $\bar{\gamma} \rightarrow \infty$). Therefore, with respect to the Indirect-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 50 Kbps and 1 Mbps), energy saving does not

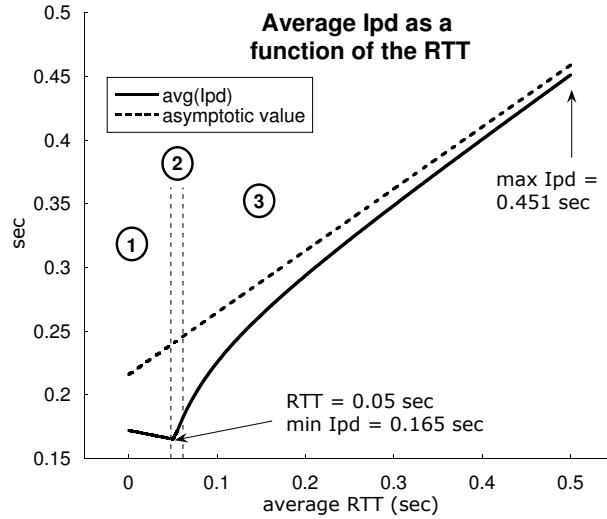


Figure 14: $\overline{I_{pd}}$ as a function of the \overline{RTT} value.

vary sharply, since it ranges between 68% and 82%.

5.2 I_{pd} sensitiveness

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, PS-WiFi 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} .

It must be also pointed out that t_{so} is the lower bound for $\overline{I_{pd}}$. Specifically, even if the estimator error were always equal to 0, the mobile device would switch 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 PS-WiFi. 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, it is easy to extend the analysis also to general t_{so} values.

From Equations 7 and 9 we derive the plot shown in Figure 14. In Figure 14 we can observe three regions, i.e., i) the part where $M = 2 \cdot \overline{RTT}$ is less than t_{so} , ii) the part when $M = 2 \cdot \overline{RTT}$ is between t_{so} and $t_{so}/0.9$, and, finally, iii) the part where $M = 2 \cdot \overline{RTT}$ is greater than $t_{so}/0.9$. Indeed, the second

region is very small, and can hardly be distinguished in Figure 14.

In the first region, both $u(M, t_{so})$ and $u(k, t_{so})$ are equal to 0. Furthermore, $\overline{I_{pd}}$ is a decreasing function of \overline{RTT} , and reaches its minimum value ($\min \overline{I_{pd}} = 0.165 \text{ sec}$ when $\overline{RTT} = 0.05 \text{ sec}$). This behavior can be explained by recalling the idle-time estimator algorithm. In this region, an additional delay is added to a packet flowing in the downlink direction only when the idle time is greater than k , and hence PS-WiFi provides $u^{(2)} = 1 \text{ sec}$ as the updated estimate. Otherwise, since both $z^{(0)} = t'$ and $z^{(1)} = k - t'$ are less than t_{so} , no delay is introduced. Since M is less than t_{so} , $p(t' > t_{so})$ is equal to 0, and hence $\overline{I_{pd}}$ becomes equal to $\overline{d} \cdot (1 + p_{emb}) + t_{so}$. Moreover, \overline{d} is basically defined by Equation 23 (see Appendix C), which is a decreasing function of \overline{RTT} . Specifically, \overline{d} is equal to $1 \text{ sec} - (M + k) / 2$, and it is at its lowest value when \overline{RTT} (and thus M) is at its maximum value.

In the second region, $u(M, t_{so})$ is equal to 1, while $u(k, t_{so})$ is equal to 0. Since M is greater than t_{so} , there are two differences with respect to the previous region, i.e. i) $p(t' > t_{so})$ is equal to $(M - t_{so}) / M$, and ii) \overline{d} becomes an increasing function of \overline{RTT} . Therefore, the following equation holds:

$$\overline{I_{pd}} = t_{so} + \overline{d} + \left\{ t_{so} \cdot \frac{M - t_{so}}{M} + \overline{d} \right\} \cdot p_{emb} . \quad (12)$$

Therefore it is easy to show that $\overline{I_{pd}}$ increases with \overline{RTT} , as \overline{d} is an increasing function of \overline{RTT} (see Appendix C).

Finally, in the third region, both $u(M, t_{so})$ and $u(k, t_{so})$ are equal to 1. In this case, $\overline{I_{pd}}$ can again be evaluated by means of Equation 12. However, when \overline{RTT} increases, \overline{d} increases more quickly than in the second region, since $u(k, t_{so})$ is equal to 1 (see Equation 9). Therefore, also $\overline{I_{pd}}$ increases more quickly than in the second region. Moreover, in this region it reaches its maximum value ($\max \overline{I_{pd}} = 0.451 \text{ sec}$, achieved when $\overline{RTT} = 0.5 \text{ sec}$).

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 $0.487 \cdot \overline{RTT}$. To summarize, we can conclude that increases of \overline{RTT} have a moderate impact on the additional URT.

6 Conclusions

In this paper we have presented an analytical model of the power-saving system (PS-WiFi) developed in [4]. The analytical model is used to analyze the performance of our system assuming mobile

Web access applications. We have compared our solution with an Indirect-TCP architecture without power management by evaluating two performance indexes, i.e., the energy spent in downloading a Web page and the related transfer-time. We have derived closed formulas that allowed us to evaluate both performance indexes. We have validated the model by comparing its predictions with the results obtained from an Internet prototype. Experimental results have shown that our model is highly accurate, since the average difference with respect to experimental measurements is about 1% for the energy-saving index, and about 7% for the transfer-time index. Therefore, our model is a valid tool to characterize the behavior of PS-WiFi with respect to key parameters, such as the application traffic profile, the throughput on the wireless and wired networks, the *RTT* between the client and the server, etc. Specifically, in this work we have presented 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 energy saving varies from 48% up to 83%, when the throughput increases from 0 to ∞ . However, when focusing on more realistic throughput ranges (i.e., between 50Kbps and 1Mbps), the energy saving does not vary sharply, and is always greater than 68%. Finally, the average additional transfer-time is a slightly increasing function of the average *RTT*. However, we can conclude that PS-WiFi never affects the QoS perceived by Web users, since the average additional transfer-time is always less than 0.5sec.

Acknowledgments

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.

References

- [1] S. Agrawal and S. Singh, "An Experimental Study of TCP's Energy Consumption over a Wireless Link", Proc. of the 4th European Personal Mobile Communications Conference, Vienna (Austria), Feb. 20-22, 2001.
- [2] G. Anastasi, M. Conti and 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, pp. 3-15, Jan. 2003.

- [3] G. Anastasi, M. Conti, E. Gregori and A. Passarella, "A Power Saving Architecture for Web Access from Mobile Computers", Proc. of the 2nd IFIP TC-6 Networking Conference (Networking 2002), LNCS No. 2345, pp. 240-251, May 19-24, 2002.
- [4] G. Anastasi, M. Conti, E. Gregori and A. Passarella, "Balancing Energy Saving and QoS in the Mobile Internet: An Application-Independent Approach", Proc. of the 36th Hawaii International Conference on System Sciences (HICSS-36), Big Island (Hawaii), Jan. 6-9, 2003.
- [5] G. Anastasi, M. Conti, E. Gregori and A. Passarella, "Performance Comparison of Power Saving Strategies for Mobile Web Access", Performance Evaluation (Special Issue on Networking 2002), Vol. 53, No. 3-4, pp. 273-294, Aug. 2003.
- [6] Apache Web Server on-line documentation, available at <http://www.apache.org>.
- [7] M. Arlitt and C. Williamson, "Internet Web Servers: Workload Characterization and Performance Implication", IEEE Transactions on Networking, Vol. 5, No. 5, pp. 631-645, Oct. 1997.
- [8] A. Bakre and B.R. Badrinath, "Implementation and Performance Evaluation of Indirect TCP", IEEE Transactions on Computers, Vol. 46, No. 3, pp. 260-278, Mar. 1997.
- [9] 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, No. 6, pp. 756-769, Dec. 1997.
- [10] P.Barford and M.Crovella, "Generating Representative Web Workloads for Network and Server Performance Evaluation", Proc. of the 1998 ACM SIGMETRICS Joint International Conference on Measurement and Modeling of Computer Systems, pp. 151-160, June 1998.
- [11] P. Barford, A. Bestavros, A. Bradley and M. Crovella, "Changes in Web Client Access Patterns: Characteristics and Caching Implications", ACM World Wide Web, Vol 2, No. 1-2, pp.15-28, 1999.
- [12] L. Bononi, M. Conti and M. Donatiello, "A distributed mechanism for power saving in IEEE 802.11 wireless LANs", ACM Mobile Networks Applications, Vol. 6, No. 3, pp. 211-222, 2001.
- [13] 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.
- [14] M. Crovella and A. Bestavros, "Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes", IEEE Transactions on Networking, Vol. 5, No. 6, pp. 835-846, Dec. 1997.
- [15] C. Cunha, A. Bestavros and M. Crovella, "Characteristics of WWW Client-Based Traces", Tech. Rep. TR-95-010, Boston University Department of Computer Science, Apr. 1995.
- [16] 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, 2000.
- [17] L.M. Feeney and M. Nilsson, "Investigating the energy consumption of a wireless network interface in an ad hoc networking environment", Proc. of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2001), Anchorage (Alaska) Apr. 22-26, 2001.
- [18] J. Flinn, S. Park and M. Satyanarayanan, "Balancing performance, energy, and quality in pervasive computing", Proc. of the 22nd IEEE International Conference on Distributed Computing Systems (ICDCS02), pp. 217-226, Vienna (Austria), July 2-5, 2002.
- [19] G.H. Forman and J. Zahorjan, "The Challenges of Mobile Computing", IEEE Computer, Vol. 27, No. 4, pp. 38-47, Apr. 1994.
- [20] D.P. Helmbold, D.E. Long and B. Sherrod, "A Dynamic Disk Spin-down Technique for Mobile Computing", Proc. of the 2nd Annual ACM International Conference on Mobile Computing and Networking (MobiCom '96), pp. 130-142, Nov. 1996.
- [21] M. Herbster and M.K. Warmuth, "Tracking the best expert", Proc. of the 12th International Conference on Machine Learning, pp. 286-294, 1995.
- [22] R. Fielding, J. Gettys, J. Mogul, H. Frystyk and T. Berners-Lee, "Hypertext Transfer Protocol - HTTP/1.1", RFC 2068, 1997.

- [23] IEEE standard for Wireless LAN- Medium Access Control and Physical Layer Specification, P802.11, Nov. 1997.
- [24] T. Imielinski and B.R. Badrinath, "Mobile Wireless Computing", Communications of the ACM, Vol. 37, No. 10, pp. 18-28, Oct. 1994.
- [25] C. Jones, K. Sivalingam, P. Agarwal and J.C. Chen, "A survey of energy efficient network protocols for wireless and mobile networks", Wireless Networks, Vol. 7, No. 4, pp. 343-358, 2001.
- [26] A. Joshi, "On proxy agents, mobility, and web access", ACM/Baltzer Mobile Networks and Applications, Vol. 5, No. 4, pp. 233-241, 2000.
- [27] R. Kravets and P. Krishnan, "Power Management Techniques for Mobile Communication", Proc. of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '98), pp. 157-168, Dallas (TX), 1998.
- [28] R. Krashinsky and H. Balakrishnan, "Minimizing Energy for Wireless Web Access with Bounded Slowdown", Proc. of the 8th Annual International Conference on Mobile Computing and Networking (MobiCom '02), pp. 119-130, Atlanta (GA), Sep. 2002.
- [29] A.M. Law and D. Kelton, "Simulation Modeling & Analysis" (Second Edition), McGraw-Hill, 1991.
- [30] J.R. Lorch and A.J. Smith, "Scheduling Techniques for Reducing Processor Energy Use in MacOS", ACM/Baltzer Wireless Networks, Vol. 3, No. 5, pp.311-324, 1997.
- [31] J.R. Lorch and A.J. Smith, "Software Strategies for Portable Computer Energy Management", IEEE Personal Communication, Vol. 5, No. 3, pp. 60-73, June 1998.
- [32] M. Mathis, J. Semke, J. Mahdavi and T.Ott, "The macroscopic behavior of the TCP Congestion Avoidance Algorithm", Computer Communication Review, Vol. 27, No. 3, July 1997.
- [33] M. Othman and S. Hailes, "Power Conservation Strategy for Mobile Computers Using Load Balancing", ACM Mobile Computing and Communication Review, Vol. 2, No. 1, pp. 44-50, Jan. 1998.
- [34] S.H. Phatak, V. Esakki, B.R. Badrinath and L. Iftode, "Web&: An Architecture for Non-Interactive Web", Proc. of the 2nd IEEE Workshop on Internet Applications (WIAPP01), S. Jose (CA), July 23-24, 2001.
- [35] S. Pilosof, R. Ramjee, D. Raz1, Y. Shavitt and P. Sinha2, "Understanding TCP fairness over Wireless LAN", Proc. of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2003), S. Francisco (CA), Mar. 30 - Apr. 3, 2003.
- [36] C. Poellabauer and K. Schwan, "Power-Aware Video Decoding using Real-Time Event Handlers", Proc. of the 5th ACM International Workshop on Wireless Mobile Multimedia (WoWMoM02), Atlanta (GA), Sep. 28, 2002.
- [37] M. Stemm and R.H. Katz, "Measuring and reducing energy consumption of network interfaces in handheld devices", IEICE Trans. Fund. Electron, Commun. Comp. Sci. (Special Issue on Mobile Computing), Vol. 80, No. 8, pp. 1125-1131, 1997.
- [38] M. Zorzi and R.R. Rao, "Energy Constrained Error Control for Wireless Channels", IEEE Personal Communications, Vol. 4, No. 6, pp. 27-33, Dec. 1997.

Appendix A

This Appendix contains the proof of Theorem 2.

Theorem 2: The energy spent to download a single basic block by using PS-WiFi is

$$C_{ps} = \frac{B}{\gamma_{wl}} + t_{so} \cdot \left\{ \frac{B}{\bar{\gamma} \cdot RTT} \cdot S_1 + l \cdot (F + \lceil \log_2 \overline{UTT} \rceil) + p \left(u^{(0)} > t_{so} \right) \right\} ,$$

where i) S_1 is the average number of switching-on events occurring during a short idle time; ii) F is the number of switching-on events occurring during a long idle time, before the long idle-time estimator is invoked (i.e., after $u^{(2)}$ of Equation 1 is generated); and iii) $p(u^{(0)} > t_{so})$ is the probability of $u^{(0)}$ being greater than t_{so} .

Proof: Based on Equation 5, to characterize C_{ps} we have to derive a closed form for the average number of switching-on events occurring during a basic-block download, i.e., S . S can be expressed as in the following proposition:

Proposition 3: The average number of switching-on events during the download of the basic block is

$$S = r \cdot S_1 + l \cdot S_2 + S_3, \quad (13)$$

where: i) r and l are the number of short and long idle times during the basic-block download, respectively; ii) S_1 and S_2 are the number of switching-on events during a short and a long idle time, respectively; and iii) S_3 is the number of switching-on events occurring during the download of the first Web page, after the main file has arrived at the mobile device, and before the HTTP Request(s) for the embedded files have been sent.

Proof: As discussed in Section 3.2, PS-WiFi regenerates with respect to the point in time when an idle time occurs. Therefore, S can be derived as follows: i) we evaluate the number of switching-on events related to a generic short idle time, and related to a generic long idle time (i.e., S_1 and S_2); ii) we evaluate the number of short idle times and long idle times occurring during the basic-block download (i.e., r and l); and iii) due to the regenerative property, we derive the total number of switching-on events as $r \cdot S_1 + l \cdot S_2$. Finally, it must be noted that a further idle time is detected by PS-WiFi when the main file of the first page has been downloaded. In fact, in this case there are no more data to be exchanged on the WLAN, and hence the estimator is invoked. Therefore, an additional switch-on event may occur, and the term S_3 accounts for this case. ■

To achieve a closed form of S we have to characterize the terms composing Equation 13. The following lemmas are devoted to this task. For easy of reading, the distribution of the number of switching-on events occurring during a short-idle time is postponed to Appendix B. S_1 is the average value of this distribution.

Lemma 3: The average number of short idle times occurring during the download of the basic

block (i.e., r) is

$$r = \frac{B}{\bar{\gamma} \cdot \overline{RTT}} .$$

Proof: As shown in Section 3.1.1, in our model the data transfers over the TCP connection between the I-TCP Daemon and the Web server occur as follows: i) once every RTT the TCP at the Web server sends a fixed number of back-to-back TCP segments; and ii) these back-to-back TCP segments arrive at the Access Point together. Therefore, the average number of bytes transferred during each RTT is $\beta \triangleq \bar{\gamma} \cdot \overline{RTT}$. Moreover, since r can be seen as the number of RTT s within a basic block, r is equal to B/β . ■

Lemma 4: The average number of switching-on events during a long idle time (i.e., S_2) is

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

where F is the average number of switching-on events occurring before the backoff procedure starts.

Proof: By definition, User Think Times are greater than 1sec . From the sequence $u^{(i)}$ shown in Equation 1, it appears that PS-WiFi needs 2 updates to start the exponential backoff procedure. This procedure starts when the idle time results greater than 1sec , i.e., after the PS-WiFi has generated: i) $u^{(0)}$ as the first estimate; ii) $u^{(1)}$ (i.e., the 90th percentile of short idle times) as the first update of the estimate; and iii) $u^{(2)}$ (i.e., 1sec) as the second update. Therefore, at this point in time, up to 3 switching-on events have occurred. The analysis of the number of switching-on events that occur up to this point (i.e., F) is very similar to the analysis of the number of switching-on events that occur during a short idle time (i.e., S_1), derived in Appendix B. For the sake of space, we here omit this analysis. Further estimate updates are derived according the binary exponential backoff procedure, as shown in Equation 1. It is easy to show that, if Q denotes the length of a long idle time, the number of estimate updates generated by the backoff procedure is $\lceil \log_2 Q \rceil$. Moreover, as upon each such update the wireless interface of the mobile device is shut down (i.e., each value of the $z^{(i)}$ sequence is greater than 1sec , and hence it can be reasonably assumed to be greater than t_{so}), $\lceil \log_2 Q \rceil$ is also the number of switching-on events that occur during the backoff procedure. The expression of S_2 can be derived immediately by recalling that, on average, Q is equal to \overline{UTT} . ■

Lemma 5: The average number of switching-on events occurring after the download of the first

main file of the basic block, before the request for the embedded files is:

$$S_3 = p\left(u^{(0)} > t_{so}\right) .$$

Proof: When the Access Point sends the last packet of the main file, no more data are available to be exchanged on the WLAN, and hence an idle time is detected. Therefore, PS-WiFi generates an idle-time estimate (i.e., $u^{(0)}$), and the mobile device switches the wireless interface off if this estimate is greater than t_{so} . Furthermore, as the packet arrives at the application level, the browser sends the HTTP Request(s) for the embedded files. Thus, a single switching-on event may occur in this case, if the estimate provided by the PS-PT at the Access Point is sufficiently long. The average number of switching-on events in this case is hence the probability of $u^{(0)}$ being greater than t_{so} . ■

Finally, the closed form for C_{ps} claimed in Theorem 2 is derived by substituting results from the above lemmas in Equation 5. This concludes the proof. ■

Appendix B

In this Appendix the distribution of the number of switching-on events occurring during a short idle time is derived.

Lemma 6: The distribution of the switching-on events occurring during a short idle time is as follows:

$$\begin{aligned}
p(0 \text{ switching-on event}) &= p(t' \leq t_{so}, t' \geq t) + \\
&\quad + p(t' \leq t_{so}, t' < t \leq k, k - t' \leq t_{so}) + \\
&\quad + p(t' \leq t_{so}, k - t' \leq t_{so}) \cdot (1 - \chi(k, t_{so})) \\
p(1 \text{ switching-on event}) &= p(t' > t, t' > t_{so}) + \\
&\quad + p(t' < t, t' \leq t_{so}, k > t, k - t' > t_{so}) + \\
&\quad + p(t' < t, t' \leq t_{so}, \\
&\quad\quad k < t, k - t' \leq t_{so}) \cdot \chi(k, t_{so}) \\
p(2 \text{ switching-on events}) &= p(t' < t, t' \leq t_{so}, \\
&\quad k < t, k - t' > t_{so}) \cdot \chi(k, t_{so}) + \\
&\quad + p(t' < t, t' > t_{so}, k > t, k - t' > t_{so}) + \\
&\quad + p(t' < t, t' > t_{so}, \\
&\quad\quad k < t, k - t' \leq t_{so}) \cdot \chi(k, t_{so}) \\
p(3 \text{ switching-on events}) &= p(t' < t, t' > t_{so}, \\
&\quad k < t, k - t' > t_{so}) \cdot \chi(k, t_{so})
\end{aligned} \tag{14}$$

where $\chi(k, t_{so})$ is defined by Equation 8.

Proof: When a short idle time begins, the Variable-Share Update algorithm provides an estimate, t' , of the actual idle time, t . The wireless interface is switched off if t' is greater than t_{so} . If t' is less than t , the estimate is updated with the 90th percentile of the short idle times, i.e. k . The wireless interface of the mobile device is switched off again only if $k - t$ is greater than t_{so} . Finally, if t is even greater than k , the algorithm executes a binary exponential backoff procedure starting from 1 second, and hence the wireless interface is switched off if $1sec - k$ is greater than t_{so} . Therefore, since t is less than $1sec$ by definition, there can't be more than 3 switching-on events within a short idle time. Moreover, a specific number of switching-on events is achieved in different cases, according to the relative values of t , t' , k and t_{so} . Therefore, each term of the switching-on-event distribution must be computed as the sum of the marginal probabilities of each case. It is worth noting that, when t is greater than k , the wireless interface is switched off only if $1sec - k$ is greater than t_{so} . Since k and t_{so} are not random variables, we introduce $\chi(k, t_{so})$ in Equation 14 to include this case. The single expressions provided in Equation 14 can be easily derived from these remarks. ■

Appendix C

This Appendix contains the proof of Theorem 4.

Theorem 4: The average delay added by PS-WiFi to a packet flowing in the downlink direction is

$$\begin{aligned} \bar{d} = & \frac{1}{2} \left(\frac{M^2 - t_{so}^2}{4M} \cdot u(M, t_{so}) + 0.9 \cdot \frac{k^2 - t_{so}^2}{4M} \cdot u(k, t_{so}) + \right. \\ & \left. + 0.1 \cdot \frac{2sec - M - k}{2} \cdot \chi(k, t_{so}) \right) \end{aligned}$$

where $u(x, y)$ and $\chi(k, t_{so})$ are defined as follows:

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

Proof: The starting point of the proof is Proposition 2, that has been discussed in Section 3.3.1.

Proposition 2: The average delay added by PS-WiFi to a packet flowing in the downlink direction can be expressed as

$$\begin{aligned} \bar{d} = E[d] = & E[d|t' > t] \cdot p(t' > t) + E[d|t' < t] \cdot p(t' < t) = \\ = & \frac{1}{2} \cdot (E[d|t' > t] + E[d|t' < t]) . \end{aligned} \quad (15)$$

To obtain the closed form of Theorem 4, we have to separately analyze the two components highlighted in Proposition 2, i.e., i) the average delay when the initial short idle-time estimate is too large (i.e., when $t' > t$); and ii) the average delay when the initial short idle-time estimate is too short (i.e., when $t' < t$). The following lemmas provide closed forms for these components.

Lemma 1: The average delay when t' is greater than t is

$$E[d|t' > t] = \frac{M^2 - t_{so}^2}{4M} . \quad (16)$$

Proof: When t' is greater than t , an additional delay is added only if t' is greater than t_{so} . Otherwise, the wireless interface of the mobile device remains on, and the new packet is received without being delayed. If t' is greater than t , the delay is $t' - t$. Therefore, the following equation

holds:

$$E [d | t' > t] = E [t' - t | t' > t, t' > t_{so}] \cdot p(t' > t_{so}) . \quad (17)$$

The second term of equation 17 (i.e., $p(t' > t_{so})$) can be computed from the t' distribution law:

$$p(t' > t_{so}) = \frac{M - t_{so}}{M} . \quad (18)$$

Furthermore, the first term of Equation 17 can be evaluated as follows:

$$\begin{aligned} E [t' - t | t' > t, t' > t_{so}] &= \int_{t_{so}}^M E [t' - t | t' > t, t' > t_{so}, t'] \cdot p(t') dt' , \\ &= \frac{M + t_{so}}{4} \end{aligned} \quad (19)$$

where the closed formula for the integral is obtained by some algebraic manipulations. By substituting Equations 18 and 19 in Equation 17 we obtain the closed form of Lemma 1. ■

Lemma2: The average delay when t' is less than t is

$$E [d | t' < t] = 0.9 \cdot \frac{k^2 - t_{so}^2}{4M} + 0.1 \cdot \frac{2sec - M - k}{2} \cdot \chi(k, t_{so}) .$$

Proof: When t' is less than t , t' is updated by using the 90th percentile of the short idle-time distribution, i.e., k . In this case \bar{d} can be evaluated by considering two possible cases in isolation, i.e., i) $t \leq k$, and ii) $t > k$. If t is less than k , the additional delay is $k - t$ seconds. More precisely, this delay is introduced only if the wireless interface is switched off after updating t' , i.e., only if $k - t'$ is greater than t_{so} (see Equation 1). Therefore, the probability that the delay is $k - t$ can be expressed as the joint probability of the events $t \leq k$ and $k - t' > t_{so}$. Moreover, since t and t' are independent, the joint probability $p(t \leq k, k - t' > t_{so})$ can be computed as the product of the marginal probabilities of the two events. The same line of reasoning can also be followed to evaluate the average delay when t is greater than k . Specifically, the delay is $1sec - t$ if $1sec - k$ is greater than t_{so} , while it is 0 otherwise. To include this condition into our model, we use $\chi(k, t_{so})$, as defined in Equation 8. Therefore, $E [d | t' < t]$ can be expressed as follows:

$$\begin{aligned} E [d | t' < t] &= E [k - t | t' < t, t \leq k, k - t' > t_{so}] \cdot p(t \leq k) \cdot p(k - t' > t_{so}) + \\ &+ E [1sec - t | t' < t, t > k] \cdot p(t > k) \cdot \chi(k, t_{so}) . \end{aligned} \quad (20)$$

We are now in the position to derive all the components of Equation 20. Firstly, the terms related to distribution law of t and t' can be easily computed:

$$\begin{aligned}
p(t \leq k) &= 0.9 \\
p(t > k) &= 0.1 \\
p(k - t' > t_{so}) &= p(t' < k - t_{so}) = \frac{k - t_{so}}{M}
\end{aligned} \tag{21}$$

Furthermore, by following the same line of reasoning used to derive (19), we can compute the component¹⁰ $E[k - t | t' < t, t \leq k, k - t' > t_{so}]$ of Equation 20:

$$\begin{aligned}
E[k - t] &= \int_0^{k - t_{so}} p(t') \cdot (k - E[t]) dt' = \\
&= \int_0^{k - t_{so}} p(t') \cdot \left(k - \int_{t'}^k t \cdot p(t) dt \right) dt' = \\
&= \frac{k + t_{so}}{4}
\end{aligned} \tag{22}$$

The steps highlighted in Equation 22 are derived as follows: i) the formula on the first line is obtained by fixing t' and integrating on its possible values; ii) the formula on the second line is obtained by using the average value definition to expand $E[t]$; and iii) the closed formula on the third line is obtained after simple computations.

The last step to evaluate Equation 21 is the computation of $E[1sec - t | t' < t, t > k]$. By applying the same technique used to derive Equations 19 and 22, we obtain the following form¹¹:

$$\begin{aligned}
E[1sec - t] &= \int_k^M p(t) \cdot (1sec - t) dt \\
&= 1sec - \frac{M + k}{2}
\end{aligned} \tag{23}$$

Finally, by substituting Equations 21, 22 and 23 in Equation 20, we obtain a closed formula for $E[e | t' < t]$:

$$E[e | t' < t] = 0.9 \cdot \frac{k^2 - t_{so}^2}{4M} + 0.1 \cdot \frac{2sec - M - k}{2} \cdot \chi(k, t_{so}) \tag{24}$$

Equations 16 and 24 allow us to complete our analysis by deriving a closed formula for the average

¹⁰For easy of reading, we omit of explicitly indicating the conditions who this average value obeys. However, they are explicitly shown in Equation 20.

¹¹Also in this case, we omit of explicitly indicating the conditions who this average value obeys.

value of d . Specifically, Equation 15 becomes as follows:

$$\bar{d} = \frac{1}{2} \left(\frac{M^2 - t_{so}^2}{4M} + 0.9 \cdot \frac{k^2 - t_{so}^2}{4M} + 0.1 \cdot \frac{2sec - M - k}{2} \cdot \chi(k, t_{so}) \right). \quad (25)$$

Then, by recalling the definition of $\chi(k, t_{so})$, Equation 25 can be expressed as:

$$\bar{d} \equiv \bar{d}(M, t_{so}) = \begin{cases} 0.169 \cdot M - 0.238 \cdot \frac{t_{so}^2}{M} + 50msec & \text{if } 1sec - 0.9 \cdot M > t_{so} \\ 0.216 \cdot M - 0.238 \cdot \frac{t_{so}^2}{M} & \text{otherwise} \end{cases}. \quad (26)$$

Finally, it must be noted that all the above equations rely on the assumption that $M > k \geq t_{so}$ holds. More generally, it is easy to show that \bar{d} is as shown in Theorem 4. This concludes the proof of Theorem 4. ■