Power-Saving in Wi-Fi Hotspots: an Analytical Study*

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 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 "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

*This is an extended version of a paper present in the Proceedings of the Eighth International Conference on Personal Wireless Communications (PWC 2003), published in the LNCS Series.

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.

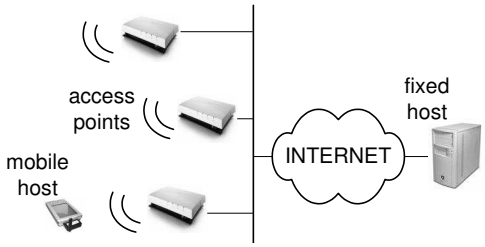

Figure 1: Mobile Internet access in a Wi-Fi hotspot.

Integrating mobile devices in such environment is still an open research problem. Mobile devices typically have limited (computational, storage, energetic) resources, with respect to desktop computers. Moreover, wireless links provide lower bandwidth and higher bit error rates with respect to wired links. Therefore, the use of standard Internet solutions may result in a bad utilization of the 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 [37], the operating system [20, 30, 35], 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 mobile networking is a very pressing requirement. Specifically, the wireless interface drains up to 50% of the total energy spent by a mobile device [27]. 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 [17], the best way to save energy is to supply the interface for the minimum time required by the networking activities. Therefore, 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. Many research works are based on this approach to reduce the energy related to networking [2, 3, 4, 23, 27, 28, 36]. 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 [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 dynamically intercepts the behavior of the network application(s) running on the mobile host. Basing

on this information, it predicts time intervals during which the wireless interface will be idle, and switches it off accordingly.

In [4] we tested our system by utilizing 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. The Web choice is well justified, since it is a highly flexible technology, that is likely to be widely adopted also in mobile environments. However, both the power-saving system of [4], and the models that we here provide, can be used with any kind of best-effort applications.

The first step of our study is 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 that 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. By comparing the performance predictions provided by the model with the experimental results presented in [4], we conclude that our model is highly accurate.

Therefore, we exploit this model to deeply analyze the power-saving system. The results show that, in the Internet conditions experienced by the prototype, 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, we used our model to perform a sensitiveness analysis of the system. This analysis allows us to understand the dependence of the power-saving system on two Internet key parameters, i.e., the throughput and the Round Trip Time (throughout referred to as RTT) between the Web client and server. The results show that power-saving is mainly affected by throughput variations. Specifically, energy savings varies from 48% to 83% when the throughput increases from 0 to ∞, 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 power-saving system never affects the QoS perceived by Web users, since the average additional transfer-time is always less than $0.5sec$.

The paper is organized as follows. Section 2 presents our power-saving system, together with the test environment that we use. In Section 3 we derive the analytical model of the power-saving system. Section 4 is devoted to the model validation. Section 5 presents the sensitiveness analysis and, finally,

Section 6 concludes the paper.

2 Reference environment

2.1 Power-saving management of best-effort traffic

As highlighted in the previous section, our power-saving architecture supports any kind of best-effort applications. Such applications do not continuously exchange data, but data-transfer phases are characterized by *bursts* interleaved by *idle phases* during which data are locally processed. Figure 2 shows a snapshot of a typical data exchange.

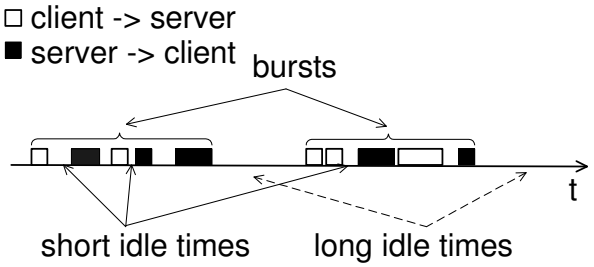

Figure 2: Snapshot of a typical best-effort data exchange.

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

The approach of our power-saving system is based on a dynamic estimate of the duration of idle times. Specifically, we measure at run time the length of idle times. From these measures, we predict the future traffic behavior, and hence we decide whether the network interface should be switched off (or not), and when to resume it. 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.

As it is clear, the core of our approach is a set of smart algorithms for estimating the traffic characteristics. Specifically, to estimate short idle times, we exploit the Variable-Share Update algorithm presented in [21]. The main feature of this algorithm is that it dynamically follows the pattern of the estimated variable, basing on past values assumed by the variable itself. On the other hand, to

estimate long idle times, we use a binary exponential backoff, starting from $1sec$.

In detail, the algorithm for estimating idle times 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 $t_i^{'}$ is less than t_i, the estimate is updated with the 90^{th} percentile of the short idle times, throughout referred to as k. Finally, if t_i is also greater than k, the algorithm supposes that t_i is a *long* idle time. Therefore, it provides the following updates by using a binary exponential backoff starting from $1sec$.

A complete description of the algorithm is provided in [4], and is here omitted due to space reasons.

2.2 Network architecture

To integrate the estimator algorithm in the Wi-Fi hotspot scenario, we define the network architecture shown in Figure 3.

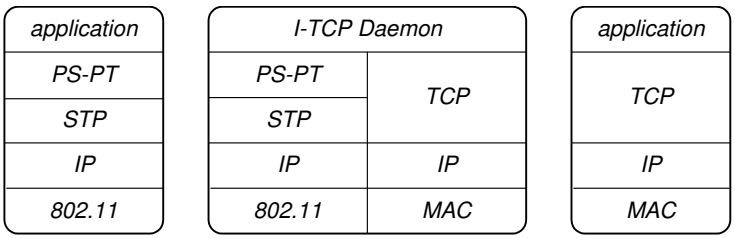

Figure 3: Power-saving network architecture.

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. According to this model, the *Indirect-TCP (I-TCP) Daemon* at the Access Point relays data between the two parts of the splitted connection. Moreover, the transport protocol between the mobile and the fixed host is a *Simplified Transport Protocol* (STP in the figure) which is tuned to the wireless link characteristics. As is shown in [9], this solution outperforms the standard Indirect-TCP architecture [8].

Furthermore, we implement the idle-time estimator in the *Power-Saving Packet Transfer* (PS-PT) protocol. The PS-PT is a simple master-slave protocol. 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 the mobile host to switch the network interface off. 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 for 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 or new data of an update of the idle-time estimate. Updates are derived according to the algorithm presented in the previous section. Upon every update, the Access Point decides if 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.

It is worth noting that the algorithm for estimating idle times is completely executed at the Access Point. As a consequence, the impact of the power-saving system on the mobile-host resources is negligible. Moreover, by using the power-saving network architecture depicted in Figure 3, we approximate the ideal power-saving management of the wireless interface, as defined in Section 1. Specifically, i) data transfers on the wireless link occur at the maximum available throughput, irrespective to the throughput on the (wired) Internet, and ii) the wireless interface remains switched off during idle periods.

Further details on the network architecture shown in Figure 3 are provided in [4].

2.3 Evaluating the Web case

As discussed in Section 1, we test our power-saving architecture when used to support Web access. To this extent, we use SURGE [10] to simulate a mobile user that accesses a Web server from a mobile host (see Figure 1). SURGE is a Web-user simulator developed by Barford and Crovella. Specifically, it is based on the Web traffic statistical model presented in [10, 11], and guarantees that simulated users generate a traffic that meets these statistics. Hence, it can be used to simulate a "typical" Web user.

Furthermore, we compare our solution with a pure Indirect-TCP approach. It must 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 [9].

To measure the performance of our system, we define two indexes. We evaluate the energy saving achieved by power-saving architecture by means of the I_{ps} index, which is defined as follows:

$$I_{ps} = \frac{C_{ps}}{C_{I-TCP}} \ . \tag{1}$$

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. 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 [17]. 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 vital to show that its value is low. To this extent, we use the I_{pd} index, that is defined as follows:

$$I_{pd} = URT_{ps} - URT_{I-TCP} \ . \tag{2}$$

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. Therefore, I_{pd} measures the additional URT that is introduced by the power-saving system to the download of a Web page.

3 System model

The description of the power-saving system provided in Section 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 by the SURGE-user in the average case.

3.1 SURGE-user model

To characterize a client access to a Web server, it is sufficient to focus on a single Web-page download. Figure 4 provides a graphical representation of such download.

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 the ON Time interval while in the Inactive OFF Time (or User Think Time) the user reads the content of the downloaded Web page.

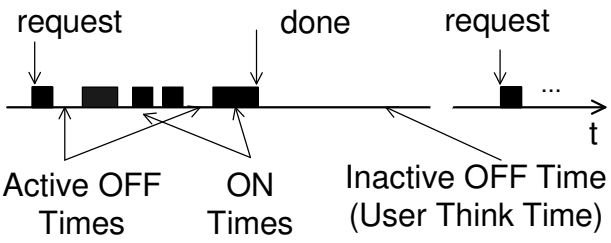

Figure 4: Web-page download.

Finally, during Active OFF Times the browser parses a piece of the main file and sends the request for the next embedded file. Active OFF Times are typically less than User Think Times, since they are due to automatic interactions between computers.

The statistical properties of the Web traffic can be derived by characterizing i) the ON and Active OFF Time lengths; ii) the main file and embedded file sizes; iii) the number of embedded files in a Web page; and iv) the User Think Time length. Many papers in the literature provide such characterizations ([7, 10, 11, 14, 15]).

The SURGE simulator is based on the statistical model presented in [10, 11, 14]. Therefore, we exploit this model to derive the parameters that define the *average* traffic profile generated by the SURGE-user. These parameters are reported in Table 1.

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

Table 1: Parameters that define the SURGE-user average traffic profile.

Specifically, the Web access of a SURGE-user can be thought of as the continuous download of the same *basic block*. The basic block consists of a set of Web pages. It is defined in such a way that the traffic generated to download it meets the statistics of Table 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 $\overline{D}_{mf} \cdot \left(l - 1/p_{emb}\right)$ (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 SURGE-user waits $\overline{UTT}$ seconds before downloading the next page. Figure 5 shows a scheme of the basic block.

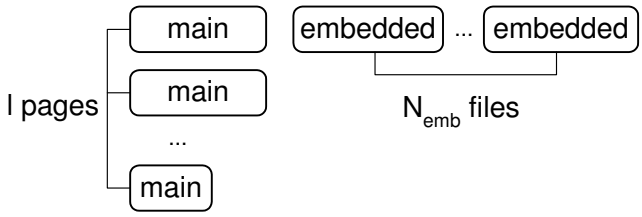

Figure 5: Scheme of the basic block.

3.1.1 Idle times characterization

With respect to the reference network architecture shown in Figure 3, we assume that the application level generates the average traffic profile shown in Figure 5. 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. Moreover, HTTP/1.1 uses persistent connections, i.e., the same TCP connection can be used to sequentially download several files [22]. Typically, Web servers close TCP connections that remain idle for 15 seconds, and do not utilize the same TCP connection to serve more than 150 HTTP requests [6]. Finally, User Think Times generated by SURGE are less than 15 seconds with probability 0.98 [10].

The above remarks show that the SURGE-user opens a new TCP connection with very low probability (about 0.02). Therefore we can reasonably assume that the download of the basic block occurs over a unique TCP connection. We also assume that this connection is already opened when the basic-block download starts, and, in the average case, it is in its steady state (i.e., after the slow-start phase). Moreover, in average, data transfers on a steady-state TCP connection can be seen as transfers of fixed-size groups of TCP segments, interleaved by $RTTs$ [32].

Basing on these remarks, we conclude that *short* idle times in the SURGE-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.

In this section we have defined the average Web-access behavior of a SURGE-user as the continuous download of the same basic block. 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 both the energy consumption and the additional URT introduced.

For easy of reading, we summarize in Table 2 the parameters 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	$\overline{\gamma}$
Average available wireless-link throughput	$\overline{\gamma}_{wl}$
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 UTT	S_2
Average number of switching-on events to send the embedded-file request	S_3
Short idle-time sample	t_i
Short idle-time estimate	$t_i^{'}$
Upper bound of the idle-time distribution	M
90^{th} percentile of short idle-times	k
Error of the idle-time estimator	e

Table 2: Symbols used throughout the paper.

3.2 Energy consumption model

The discussion of the previous section allows us 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 latter case, the energy consumed (referred to as C_{I-TCP}) corresponds to the basic-block transfer-time, since no power-saving strategy is used. Specifically, it is:

$$C_{I-TCP} = \frac{B}{\overline{\gamma}} + l \cdot \overline{UTT} \ , \tag{3}$$

where $\overline{\gamma}$ 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. More precisely, $B = \overline{D}_{mf} \cdot 1 \,/\, p_{emb} + \overline{N}_{emb} \cdot \overline{D}_{emb}$.

On the other hand, the energy consumed by 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. In addition, the mobile host wireless interface consumes t_{so} seconds every time it is switched on. The energy consumption in the power-saving system (throughout referred to as C_{ps}) is thus:

$$C_{ps} = \frac{B}{\overline{\gamma}_{wl}} + t_{so} \cdot S \ , \tag{4}$$

where $\overline{\gamma}_{wl}$ is the average available throughput allowed by the wireless link, and S is the average number of switching-on events during a basic block download. It is worth noting that, thanks to the network architecture design, $t_{so} \cdot S$ is the only contribution to the energy spent related to the power-saving system.

Since the wireless interface is switched off according to the idle-time estimator, its accuracy defines the S term in equation (4). This term is affected by several factors. One of these factors is the number of switching-on events within to a *short* idle time, throughout referred to as S_1.

When a short idle time begins, the Variable-Share Update algorithm provides an estimate, $t_i^{'}$, of the actual idle time, t_i. The wireless interface is switched off if $t_i^{'}$ is greater than t_{so}. If $t_i^{'}$ is less than t_i, the estimate is updated with the 90^{th} percentile of the short idle times, throughout referred to as k. The wireless interface of the mobile host is switched off again only if $k - t_i$ is greater than t_{so}. Finally, if t_i 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}.

Since t_i is less than $1sec$ by definition, there can't be more than 3 switching-on events within an idle-time. Moreover, a specific number of switching-on events is achieved in different cases, according to the relative values of t_i, $t_i^{'}$, 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_i 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 equations (6) to include this case. Specifically, $\chi(k, t_{so})$ is an indicator function, which is defined as follows:

$$\chi(k, t_{so}) = \begin{cases} 1 & \text{if } 1sec - k > t_{so} \\ 0 & \text{otherwise} \end{cases} . \tag{5}$$

Based on the above remarks, the number of switching-on events within a short idle time follows

this distribution:

$$
\begin{aligned}
p\,(1\,\text{switching-on event}) = \;& p\left(t_i^{'} > t_i, t_i^{'} > t_{so}\right) + \\
+ \;& p\left(t_i^{'} < t_i, t_i^{'} \leq t_{so}, k > t_i, k - t_i^{'} > t_{so}\right) + \\
+ \;& p\left(t_i^{'} < t_i, t_i^{'} \leq t_{so}, \right. \\
& \left. k < t_i, k - t_i^{'} \leq t_{so}\right) \chi\,(k, t_{so}) \\
p\,(2\,\text{switching-on events}) = \;& p\left(t_i^{'} < t_i, t_i^{'} \leq t_{so}, \right. \\
& \left. k < t_i, k - t_i^{'} > t_{so}\right) \chi\,(k, t_{so}) + \\
+ \;& p\left(t_i^{'} < t_i, t_i^{'} > t_{so}, k > t_i, k - t_i^{'} > t_{so}\right) + \\
+ \;& p\left(t_i^{'} < t_i, t_i^{'} > t_{so}, \right. \\
& \left. k < t_i, k - t_i^{'} \leq t_{so}\right) \chi\,(k, t_{so}) \\
p\,(3\,\text{switching-on events}) = \;& p\left(t_i^{'} < t_i, t_i^{'} > t_{so}, \right. \\
& \left. k < t_i, k - t_i^{'} > t_{so}\right) \chi\,(k, t_{so})
\end{aligned}
\tag{6}
$$

We are now in the position to evaluate S_1, i.e., the average number of the above distribution:

$$
\begin{aligned}
S_1 = \;& 1 \cdot p\,(1\,\text{switching-on event}) + 2 \cdot p\,(2\,\text{switching-on events}) + \\
+ \;& 3 \cdot p\,(3\,\text{switching-on events}) \;\; .
\end{aligned}
$$

So far, we have proved that each short idle time contributes with S_1 switching-on events to the S term in equation (4). Therefore, we need to evaluate the average number of short idle times that occur during a basic block download. This average number is throughout referred to as r. In Section 3.1.1 we have shown that the basic-block download occurs as transfers of fixed-size TCP-windows of data interleaved by $RTTs$. Therefore, the average number of bytes transferred during each RTT is $\beta \triangleq \overline{\gamma} \cdot \overline{RTT}$. Moreover, since r can be seen as the number of $RTTs$ within a basic block, r is equal to $B \,/ \beta$. Finally, $r \cdot S_1$ is the contribution of short idle times to the S term of equation (4).

Another component of S is the average number of switching-on events within User Think Time (S_2). By definition, User Think Times are longer than $1sec$. Hence, the estimator needs 2 updates to start the exponential backoff procedure. The backoff procedure begins after up to 3 switching-on events. Specifically, following the same line of reasoning used to derive equation (6), it is possible to derive the average number of switching-on events that occur before the backoff procedure starts. This number is throughout referred to as H. Moreover, the average number of additional switching-on

events during the backoff procedure can be evaluated as $i = \lceil \log_2 \overline{UTT} \rceil$. Therefore, S_2 is given by:

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

Finally, a third component (S_3) of S must be considered. Specifically, an additional switching-on events can occur when the SURGE-user downloads a Web page with embedded files. As shown in the next section, the PS-PT module at the Access Point (see Figure 3) detects a short idle-time after the last packet of the main file. Therefore, it bounds a short-idle-time estimate to this packet. The mobile host switches the wireless interface off if the estimate is greater than t_{so}, and switches it on again when the browser sends the embedded-files HTTP request. We include this behavior in our model by defining S_3 as follows:

$$S_3 = p\left(t_i' > t_{so}\right) \ . \tag{7}$$

In conclusion, the S term of equation (4) can be evaluated as:

$$S = r \cdot S_1 + l \cdot S_2 + S_3 = \frac{B}{\overline{\gamma} \cdot \overline{RTT}} \cdot S_1 + l \cdot \left(H + \lceil \log_2 \overline{UTT} \rceil\right) + p\left(t_i' > t_{so}\right) \ , \tag{8}$$

and, hence, C_{ps} becomes as follows:

$$C_{ps} = \frac{B}{\overline{\gamma}_{wl}} + t_{so} \cdot \left\{ \frac{B}{\overline{\gamma} \cdot \overline{RTT}} \cdot S_1 + l \cdot \left(H + \lceil \log_2 \overline{UTT} \rceil\right) + p\left(t_i' > t_{so}\right) \right\} \ . \tag{9}$$

Finally, from equations (1), (3) and (9), a closed formula for I_{ps} can be easily derived.

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 all embedded files are downloaded in the second transaction. Due to the TCP behavior, each transaction consists of several groups of TCP segments interleaved by idle times, as discussed in Section 3.1.1. When an idle-time estimate occurs to be too long, the Access Point stores TCP segments from the Web server in a local buffer, waiting for the mobile host to reconnect (see Section 2.2). In this case, the power-saving system introduces an additional delay to that group of TCP segments. It is worth noting that the additional

delay introduced to a whole transaction is the delay related to the *last* group of that transaction, as shown in Figure 6.

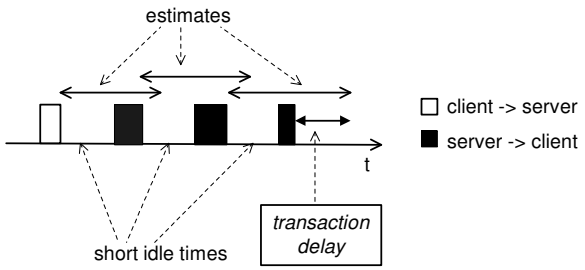

Figure 6: Transaction within a Web-page download.

It must be pointed out that the Access Point is usually connected to the Internet through a 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 to a group is well approximated by the additional delay introduced to 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, a further delay is introduced. The first packet of the transaction experiences a delay of t_{so}, due to the switching-on transient of the wireless interface. Hence, the whole transaction is "shifted" of this time-interval.

In conclusion, the average additional delay introduced to a Web-page download (throughout referred to as $\overline{I_{pd}}$) can be evaluated as follows:

$$\overline{I_{pd}} = t_{so} + \overline{e} + \left\{ t_{so} \cdot p\left(t_i' > t_{so}\right) + \overline{e} \right\} \cdot p_{emb} \ , \tag{10}$$

where $\overline{e}$ is the average value of e and p_{emb} is the probability that a Web-page contains embedded files. In detail, $t_{so} + \overline{e}$ is the additional delay introduced to the main-file transfer-time. If the Web-page contains embedded files, a further additional delay is introduced. This delay is again $t_{so} + \overline{e}$ if the transaction starts when the wireless interface is off, while it is $\overline{e}$ otherwise. Finally, the probability that the embedded-files transaction starts when the wireless interface is off can be computed as $p\left(t_i' > t_{so}\right)$ (see equation (7)).

The evaluation of $\overline{I_{pd}}$ requires the characterization of the idle-time estimator error, e. In the following section we provide a model for the average value of this quantity.

3.3.1 Analytical model of the idle-time-estimator error

To build the estimator of short idle-times (i.e., idle-times less than 1 sec), we utilize the Variable-Share Update algorithm (see Section 2.2). The main feature of this algorithm is that it dynamically adapts to the pattern of the estimated variable (see [20, 21]). Based on this property, hereafter we assume that idle-times and their estimates are distributed according to the same law. Furthermore, we assume a uniform distribution between 0 and a maximum value, M, i.e., $t_i \sim t_i^{'} \in \mathcal{U}[0,M]$. Finally, we assume that t_i and $t_i^{'}$ are independent. It is worth noting that the M value depends on the specific idle-time distribution. Based on the characterization given in Section 3.1.1, we instantiate M to two times the average RTT between the mobile host and the Web server, i.e., $M = 2 \cdot \overline{RTT}$. However, hereafter we provide the estimator-error model as a function of M, to assess the flexibility of our analytical approach.

The average value of the idle-time-estimator error (i.e., $\overline{e}$), can be evaluated as the contribution of two components. Specifically, it is the sum of i) the average estimator error when the estimate is too large (i.e., $t_i^{'} > t_i$); and ii) the average estimator error when it is too short (i.e., $t_i^{'} < t_i$). Since $t_i^{'}$ and t_i are distributed according to the same law, both $p\left(t_i^{'} > t_i\right)$ and $p\left(t_i^{'} < t_i\right)$ are equal to $1/2$. Therefore, $\overline{e}$ can be computed as follows:

$$\begin{aligned} \overline{e} = E[e] = & \quad E\left[e \left| t_i^{'} > t_i \right.\right] \cdot p\left(t_i^{'} > t_i\right) + E\left[e \left| t_i^{'} < t_i \right.\right] \cdot p\left(t_i^{'} < t_i\right) = \\ = & \quad \tfrac{1}{2}\left(E\left[e \left| t_i^{'} > t_i \right.\right] + E\left[e \left| t_i^{'} < t_i \right.\right]\right) . \end{aligned} \tag{11}$$

If $t_i^{'}$ is greater than t_i, and if the mobile host switches off the wireless interface after receiving $t_i^{'}$, the estimator error value is $t_i^{'} - t_i$. However, if $t_i^{'}$ is less than t_{so}, the wireless interface remains on, and hence no additional delay is introduced. Therefore, equation (12) holds:

$$E\left[e \left| t_i^{'} > t_i \right.\right] = E\left[t_i^{'} - t_i \left| t_i^{'} > t_i, t_i^{'} > t_{so} \right.\right] \cdot p\left(t_i^{'} > t_{so}\right) . \tag{12}$$

The second term of equation (12) (i.e., $p\left(t_i^{'} > t_{so}\right)$) can be easily computed from the $t_i^{'}$ distribution law:

$$p\left(t_i^{'} > t_{so}\right) = \frac{M - t_{so}}{M} . \tag{13}$$

Furthermore, the first term of equation (12) can be evaluated as follows:

$$\begin{aligned} E\left[t_i^{'}-t_i\left|t_i^{'}>t_i,t_i^{'}>t_{so}\right.\right] &= \int_{t_{so}}^{M} E\left[t_i^{'}-t_i\left|t_i^{'}>t_i,t_i^{'}>t_{so}\right.,t_i^{'}\right]\cdot p\left(t_i^{'}\right)dt_i^{'} \\ &= \frac{M+t_{so}}{4} \end{aligned} \quad , \tag{14}$$

where the closed formula for the integral is obtained by some algebraic manipulations. Finally, by using equations (13) and (14), equation (12) becomes:

$$E\left[e\left|t_i^{'}>t_i\right.\right]=\frac{M^2-t_{so}^2}{4M}\ . \tag{15}$$

Equation (15) allows us to compute the first component of $\overline{e}$ (see equation (11)). Below we provide a closed formula also for the second component, i.e., $E\left[e\left|t_i^{'}<t_i\right.\right]$.

Specifically, if $t_i^{'}$ is less than t_i, $t_i^{'}$ is updated by using the 90^{th} percentile of the short idle-times, i.e., k. The computation of $\overline{e}$ can be performed by considering the two possible cases in isolation, i.e., i) $t_i \leq k$, and ii) $t_i > k$. If t_i is less than k, the power-saving system introduces $k-t_i$ seconds of additional delay. More precisely, this delay is introduced only if the wireless interface is switched off after updating $t_i^{'}$, i.e., only if $k-t_i^{'}$ is greater than t_{so}. Therefore, the estimator-error value is $k-t_i$ with the joint probability of the events $t_i \leq k$ and $k-t_i^{'}>t_{so}$. Since t_i and $t_i^{'}$ are independent, the joint probability $p\left(t_i \leq k, k-t_i^{'}>t_{so}\right)$ can be computed as the product of the marginal probabilities of the two events.

The same line of reasoning can also be followed when t_i is greater than k. Specifically, the estimator-error value is $1sec-t_i$ if $1sec-k>t_{so}$, while it is 0 otherwise. To include this condition into our model, we use $\chi(k,t_{so})$, as defined in equation (5).

Based on these remarks $E\left[e\left|t_i^{'}<t_i\right.\right]$ becomes:

$$\begin{aligned} E\left[e\left|t_i^{'}<t_i\right.\right] = \ & E\left[k-t_i\left|t_i^{'}<t_i,t_i\leq k,k-t_i^{'}>t_{so}\right.\right]\cdot p\left(t_i\leq k\right)\cdot p\left(k-t_i^{'}>t_{so}\right)+ \\ & +E\left[1sec-t_i\left|t_i^{'}<t_i,t_i>k\right.\right]\cdot p\left(t_i>k\right)\cdot\chi\left(k,t_{so}\right)\ . \end{aligned} \tag{16}$$

We are now in the position to derive all the components of equation (16). Firstly, the terms that

involve the distribution law of t_i and $t_i^{'}$ can be easily computed:

$$\begin{aligned} & p\left(t_i \leq k\right) = 0.9 \\ & p\left(t_i > k\right) = 0.1 \\ & p\left(k - t_i^{'} > t_{so}\right) = p\left(t_i^{'} < k - t_{so}\right) = \frac{k - t_{so}}{M} \end{aligned} \tag{17}$$

Furthermore, by following the same line of reasoning used to derive (14), we can compute the average value of $k - t_i$ of equation (16)[1]:

$$\begin{aligned} E\left[k - t_i\right] &= \int_0^{k-t_{so}} p\left(t_i^{'}\right) \cdot \left(k - E\left[t_i\right]\right) dt_i^{'} = \\ &= \int_0^{k-t_{so}} p\left(t_i^{'}\right) \cdot \left(k - \int_{t_i^{'}}^{k} t_i \cdot p\left(t_i\right) dt_i\right) dt_i^{'} = \\ &= \frac{k + t_{so}}{4} \ . \end{aligned} \tag{18}$$

The steps highlighted in equation 18 are derived as follows: i) the formula on the first line is obtained by fixing $t_i^{'}$ and integrating on its possible values; ii) the formula on the second line is obtained by using the average value definition to expand $E\left[t_i\right]$; and iii) the closed formula on the third line is obtained after simple computations.

The last step to evaluate equation (17) is the computation of $E\left[1sec - t_i \middle| t_i^{'} < t_i, t_i > k\right]$. By applying the same technique used to derive equations (14) and (18), we derive[2]:

$$\begin{aligned} E\left[1sec - t_i\right] &= \int_k^M p\left(t_i\right) \cdot \left(1sec - t_i\right) dt_i \\ &= 1sec - \frac{M + k}{2} \ . \end{aligned} \tag{19}$$

Finally, by introducing equations (17), (18) and (19) into equation (16), we obtain a closed formula for $E\left[e \middle| t_i^{'} < t_i\right]$:

$$E\left[e \middle| t_i^{'} < t_i\right] = 0.9 \cdot \frac{k^2 - t_{so}^2}{4M} + 0.1 \cdot \frac{2sec - M - k}{2} \cdot \chi\left(k, t_{so}\right) \ . \tag{20}$$

Equations (15) and (20) allow us to complete our analysis by deriving a closed formula for the

[1] For easy of reading, we omit of explicitly indicating the conditions who this average value obeys. However, they are explicitly shown in equation 16.

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

average value of e. Equation (11) becomes as follows:

$$\overline{e} = \frac{1}{2}\left(\frac{M^2 - t_{so}^2}{4M} + 0.9 \cdot \frac{k^2 - t_{so}^2}{4M} + 0.1 \cdot \frac{2sec - M - k}{2} \cdot \chi(k, t_{so})\right) . \quad (21)$$

Finally, by recalling the definition of $\chi(k, t_{so})$ (equation (5)), equation (21) can be expressed as:

$$\overline{e} \equiv \overline{e}(M, t_{so}) = \begin{cases} 0.169 \cdot M - 0.238 \cdot \frac{t_{so}^2}{M} + 50msec & \text{if } 1sec - 0.9M > t_{so} \\ 0.216 \cdot M - 0.238 \cdot \frac{t_{so}^2}{M} & \text{otherwise} \end{cases} . \quad (22)$$

As a final remark, it must be noted that all the above equations rely on the assumption that $M > k \geq t_{so}$. More generally, it is easy to show that $\overline{e}$ is as follows:

$$\begin{array}{rcl} \overline{e} & = & \frac{1}{2}\left(\frac{M^2 - t_{so}^2}{4M} \cdot u(M, t_{so}) + 0.9 \cdot \frac{k^2 - t_{so}^2}{4M} \cdot u(k, t_{so}) + \right. \\ & + & \left. 0.1 \cdot \frac{2sec - M - k}{2} \cdot \chi(k, t_{so})\right) \end{array} , \quad (23)$$

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

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

4 Model validation

4.1 The prototype

To validate the analytical model derived in Section 3 we compare its predictions with results from a real Internet prototype. To this purpose, we implemented a Wi-Fi hotspot scenario similar to the one described in Section 1. Specifically, we modified the protocol stacks of both the mobile host and the Access Point according to the network architecture shown in Figure 3.

In our test-bed, we use a real Web server located at the University of Arlington (TX), while the Access Point and the mobile host are located at the University of Pisa (Italy). Moreover, the SURGE simulator (running on the mobile host) acts as the typical Web user.

As a final remark, it must be pointed out that our prototype simulates the wireless LAN between the mobile host and the Access Point. Specifically, it assumes an 11-Mbps throughput on the wireless link, that represents the upper bound of the throughput achievable with the current Wi-Fi technology.

However, we replicated experiments by decreasing the wireless link throughput down to 2Mbps. The obtained results show that energy-saving variations are very small, since the bottleneck between the client and the server is always the wired part of the path.

To obtain accurate measures from the prototype, we ran an extensive set of experiments. Each experiment consists of two (simulated) Web users that download, in parallel, the same set of Web pages. One of the users utilizes the power-saving network architecture depicted in Figure 3, while the other one uses the Indirect-TCP architecture. Moreover, the experiment stops when both users have downloaded 150 files[3]. Finally, from each experiment we compute i) a sample of the I_{ps} index, and ii) the average of the I_{pd} values experienced throughout the entire experiment (i.e., a sample of $\overline{I_{pd}}$). It must be pointed out that this methodology allows us to highlight the behavior of the power-saving system, since both the users download the same data, and experience the same Internet and Web server conditions.

We sequentially replicated the experiments throughout an entire working day, and we replicated a day of experiments for 10 working days. As is deeply described in [5], we guaranteed that experiments are independent. Finally, we group together I_{ps} and $\overline{I_{pd}}$ samples from experiments performed during the same hour (also in different day). As is shown in [5], these samples are i.i.d., and hence we compute the hourly average values of I_{ps} and $\overline{I_{pd}}$, according to the standard statistical method [29]. More details on the experiment test-bed and methodology are provided in [5], and are here omitted due to space reasons.

4.2 Comparing the prototype and the analytical model

The model validation is carried out by comparing the I_{ps} and $\overline{I_{pd}}$ model predictions with the hourly results obtained from the prototype. Specifically, for each hour, i) we instantiate the model parameters with the specific values experienced by the prototype; ii) we derive the model predictions for I_{ps} and $\overline{I_{pd}}$; and finally iii) we compare the predictions with the prototype results.

It is worth noting that, due to the experiment setup, in each experiment the SURGE-users generate a traffic that meets the statistics shown in Table 1. Furthermore, as noted in Section 3.3.1, we use $2\overline{RTT}$ as the maximum value for both t_i and $t_i^{'}$. The $\overline{RTT}$ value experienced by the experiments is almost independent on the specific hour. Hence, the M parameter of equation (22) does not change

[3]This constraint ensures that results are not biased by a particular choice in the file sizes , as explained in [5].

among different hours. Finally, also the values of S_1, S_2 and S_3 are almost identical over the whole day, and $\overline{\gamma}_{wl}$ is assumed equal to $11Mbps$. Therefore, in our test-bed, the only parameter that changes among different hours is the throughput experienced on the Internet, i.e., $\overline{\gamma}$. Table 3 summarizes the values of the other parameters that are used to validate our analytical model.

Definition	**Symbol**	**Value**	**Units**
Total dimension of the basic block	B	49264	bytes
Number of Web pages in the basic block	l	3	-
Average throughput over the wireless link	$\overline{\gamma}_{wl}$	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 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
Upper bound of t_i and $t_i^{'}$ distributions	M	0.6	sec
Switching-on transient interval of the wireless interface	t_{so}	0.1	sec
90^{th} percentile of t_i and $t_i^{'}$ distributions	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 7 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.

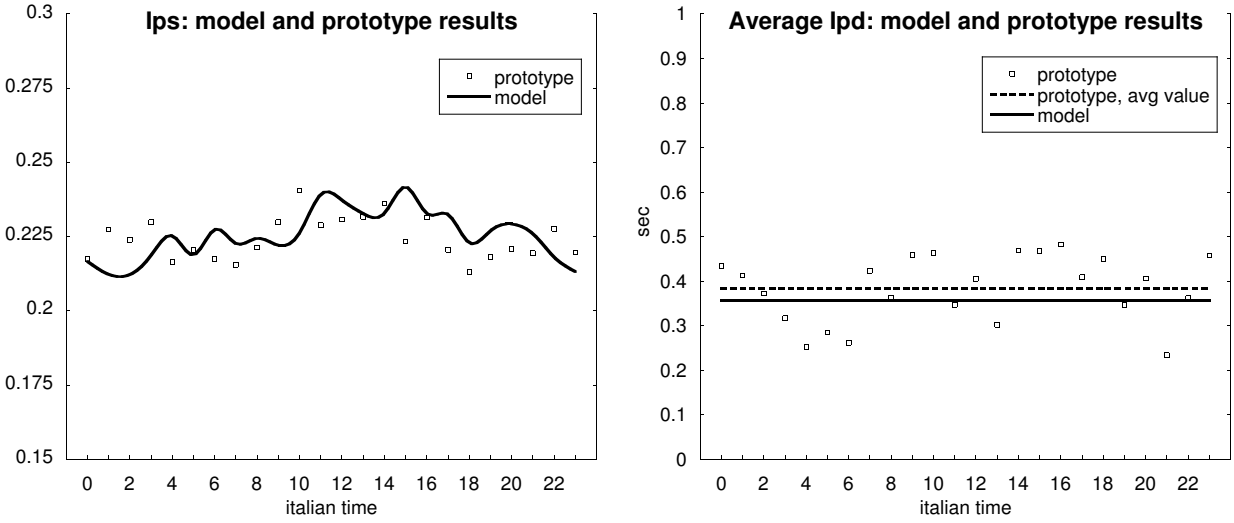

Figure 7: Hourly average I_{ps} and $\overline{I_{pd}}$ obtained from the model and by the prototype, respectively.

As far as the I_{ps} index (left 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 prediction of the model. Specifically, we have set $\overline{\gamma}$ in equation (9) to the average daily Internet throughput experienced by the prototype. 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 plot), the results show that prototype values vary during the day, i.e., the prototype is sensitive to variations of $\overline{\gamma}$. This behavior can be explained by recalling the definition of throughput. Specifically, $\overline{\gamma}$ is sensitive to two parameters, i.e., i) congestions in the Internet, that reduce the TCP window size; and ii) the variations of the RTT between client and server. As discussed in Section 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 result (the average RTT over the whole day is $0.3sec$).

The above results show the accuracy of our analytical model. Hereafter 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., $\overline{\gamma}$) depends on both the network RTT and the TCP window size. Both these parameters affect the I_{ps} index, since it depends on $\overline{\gamma}$ (see equations (3) and (9)). On the other hand, the $\overline{I_{pd}}$ index is only affected by variations of the RTT, as shown by equations (10) and (22). Therefore, below we analyze I_{ps} as a function of $\overline{\gamma}$, and $\overline{I_{pd}}$ as a function of RTT.

Based on equations (3) and (9) we derive the I_{ps} index as a function of $\overline{\gamma}$. Specifically, after simple manipulations, $I_{ps}(\overline{\gamma})$ becomes:

$$I_{ps}(\overline{\gamma}) = \frac{a\overline{\gamma}+b}{c\overline{\gamma}+d} \ , \tag{24}$$

where a, b, c and d group constant terms:

$$\begin{cases} a \triangleq \frac{B}{\overline{\gamma}_{wl}} + t_{so} \cdot (l \cdot S_2 + S_3) \\ b \triangleq \frac{B}{\overline{RTT}} \cdot S_1 \cdot t_{so} \\ c \triangleq l \cdot \overline{UTT} \\ d \triangleq B \end{cases} .$$

Figure 8 shows I_{ps} as a function of $\overline{\gamma}$[4].

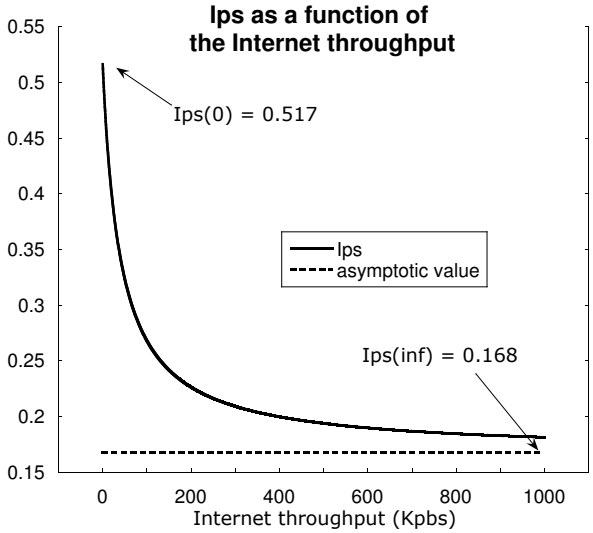

Figure 8: I_{ps} as a function of the Internet throughput $\overline{\gamma}$.

It clearly appears that when $\overline{\gamma}$ increases, I_{ps} decreases, and hence the power-saving system saves more energy. This result is somehow counter-intuitive, since one expects that the best power-saving is achieved when $\overline{\gamma}$ is at its lowest value. 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 (3) and (9), variations of $\overline{\gamma}$ affect both C_{I-TCP} and C_{ps}.

In the Indirect-TCP architecture, when $\overline{\gamma}$ increases, the time needed to fetch the basic block from the Web server (i.e., $B/\overline{\gamma}$) decreases, and C_{I-TCP} decreases accordingly.

On the other hand, the dependence of C_{ps} on $\overline{\gamma}$ is as follows. As highlighted in Section 4.2, $\overline{\gamma}$ is strictly related to the the TCP window size. Since RTT is almost stable, large TCP windows mean high $\overline{\gamma}$ values, while narrow TCP windows correspond to low $\overline{\gamma}$ values. Furthermore, if the TCP window size increases, the number of $RTTs$ needed to fetch the basic block (i.e., r in equation (8)) drops, since more bytes are downloaded in a single RTT. Equations (8) and (9) show that if

[4] As it is clear, $\overline{\gamma} > 0$ is the only $\overline{\gamma}$ range that makes sense in equation (24).

r decreases, the number of switching-on events (i.e., S) decreases. Therefore, we conclude that the more $\overline{\gamma}$ increases, the more C_{ps} decreases.

Since both C_{I-TCP} and C_{ps} benefit from increases of $\overline{\gamma}$, the I_{ps} pattern is defined by the parameters a, b, c and d. Specifically, in the Internet configuration that we experienced, when $\overline{\gamma}$ 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 8 highlights the theoretical lower and upper bounds for I_{ps}. Specifically, in the Internet configuration that we experienced, I_{ps} ranges between 0.517 (when $\overline{\gamma} = 0$) and 0.168 (when $\overline{\gamma}$ approaches ∞). 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%.

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 $\overline{RTT}$.

It must be also 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, it is easy to extend the analysis also to general t_{so} values.

From equations (10) and (23) we derive the plot shown in Figure 9.

In Figure 9 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, ii) the part where $M = 2 \cdot \overline{RTT}$ is greater than $t_{so}/0.9$. It must be pointed out the second region is very small, and can hardly be

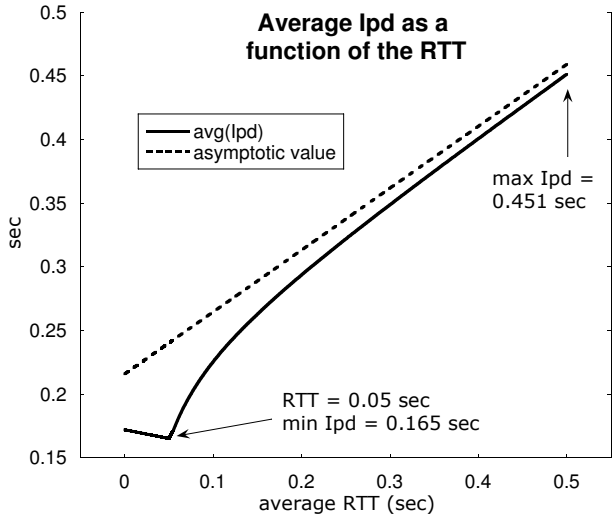

Figure 9: $\overline{I_{pd}}$ as a function of the $\overline{RTT}$ value.

distinguished in Figure 9.

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 ($\min \overline{I_{pd}} = 0.165 sec$) when $\overline{RTT} = 0.05 sec$. This behavior can be explained by recalling the idle-time estimator algorithm. Specifically, since M is less than t_{so}, $p\left(t_i^{'} > t_{so}\right)$ is equal to 0, and hence $\overline{I_{pd}}$ becomes equal to $\overline{e}(1 + p_{emb}) + t_{so}$. Moreover, $\overline{e}$ is basically defined by equation (19), which is a decreasing function of $\overline{RTT}$. In other words, the idle-time estimator experiences an error only when it starts the exponential backoff procedure, since both $t_i^{'}$ and k are always less than t_{so}. Therefore, the additional delay is defined by $1sec - (M + k)/2$, and it is hence at its lowest value when $\overline{RTT}$ 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. There are two differences with respect to the previous region. Since M is greater than t_{so}, i) $p\left(t_i^{'} > t_{so}\right)$ is equal to $(M - t_{so})/M$, and ii) $\overline{e}$ becomes an increasing function of $\overline{RTT}$. Therefore, equation (25) holds:

$$\overline{I_{pd}} = t_{so} + \overline{e} + \left\{ t_{so} \cdot \frac{M - t_{so}}{M} + \overline{e} \right\} \cdot p_{emb} \ , \tag{25}$$

and $\overline{I_{pd}}$ is an increasing function of $\overline{RTT}$.

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 be again evaluated by means of equation (25). However, when $\overline{RTT}$ increases, $\overline{e}$ increases more quickly than in the second region, since $u(k, t_{so})$ is equal to 1 (see equation (23)). Therefore, also $\overline{I_{pd}}$ increases more quickly than in the second region. Moreover, in this region it reaches its maximum value ($\max \overline{I_{pd}} = 0.451 sec$, achieved when $\overline{RTT} = 0.5 sec$).

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 $0.487 \cdot \overline{RTT}$. 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. Our solution operates at the transport and middleware layers. It is based on the Indirect-TCP approach, and is application independent. As such, it can be used with any kind of non real-time network applications. Energy saving is achieved by performing data transfers over the wireless link at the maximum available throughput, and by switching the wireless interface off whenever it is idle. Therefore, the core of our solution is a set of smart algorithms that i) predict the length of idle phases in the application traffic pattern, and ii) manage the wireless interface accordingly.

The analytical model has been used to analyze the performance of our system when used to support mobile Web access. Specifically, we have compared our solution with a pure Indirect-TCP approach by means of 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. Moreover, we have validated the model by comparing its predictions with the results obtained from a real-Internet prototype described in [4]. The results have shown that, in 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, our model has allowed us to highlight the dependence of the performance indexes on key parameters, such as the application-level traffic profile, the throughput on the wireless and wired networks, the RTT between the client and the server. 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 energy saving varies from 48% up to 83%, when the Internet throughput increases from 0 to ∞. However, when focusing on more realistic 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 $0.5sec$.

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: "Web Systems with QoS Guarantees", "Internet: Efficiency, Integration and Security" and FIRB-PERF.

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.Passerella, "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 International Conference on System Sciences (HICSS-36), Big Island, Hawaii, January 03.

[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), 2003.

[6] Apache Web Server on-line documentation, available at http://www.apache.org.

[7] M.Arlitt, C.Williamson, "Internet Web Servers: Workload Characterization and Performance Implication", IEEE/ACM Transactions on Networking, Vol.5, No.5, pp.631-645, October 1997.

[8] A.Bakre, B.R.Badrinath, "Implementation and Performance Evaluation of Indirect TCP", IEEE Transactions on Computers, Vol.46, No.3, March 1997.

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

[10] P.Barford e M.Crovella, "Generating Representative Web Workloads for Network and Server Performance Evaluation", Proceedings of ACM SIGMETRICS ´98, Madison, WI, pp. 151-160, June 1998.

[11] 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.

[12] 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.

[13] 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.

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

[15] C.Cunha, A.Bestavros e M.Crovella, ”Characteristics of WWW Client-Based Traces”, Technical Report TR-95-010, Boston University Department of Computer Science, April 1995.

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

[17] L.M. Feeney and M. Nilsson, ”Investigating the energy consumption of a wireless network interface in an ad hoc networking environment”, Proceedings of the Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2001), 2001.

[18] J.Flinn, S.Park and M.Satyanarayanan, ”Balancing performance, energy, and quality in pervasive computing”, Proc. 22nd IEEE International Conference on Distributed Computing Systems (ICDCS), pp. 217-226, July 2002.

[19] G. H. Forman, J. Zahorjan, ”The Challenges of Mobile Computing”, IEEE Computer, 27(6), April 1994.

[20] D.P. Helmbold, D.E. Long, B. Sherrod ”A Dynamic Disk Spin-down Technique for Mobile Computing”, Proceedings of the Second Annual ACM International Conference on Mobile Computing and Networking, NY, pp. 130 - 142, November 1996.

[21] M. Herbster and M.K. Warmuth, “Tracking the best expert”, Proceedings of the Twelfth International Conference on Machine Learning, Tahoe City, CA, 1995, pp. 286-294.

[22] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, 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, November 1997.

[24] T. Imielinscki B.R. Badrinath ”Wireless Computing”, Communication of the ACM, Vol. 37, No. 10, October 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, pp. 343-358, 2001.

[26] A. Joshi, ”On proxy agents, mobility, and web access”, ACM/Baltzer Mobile Networks and Applications, Vol. 5 (2000), pp. 233-241.

[27] R.Kravets e P.Krishnan, ”Power Management Techniques for Mobile Communication”, Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom’98).

[28] R.Krashinsky and H.Balakrishnan, ”Minimizing Energy for Wireless Web Access with Bounded Slowdown”, Proc. 8th Annaul International Conference on Mobile Computing and Networking (Mobicom 2002), September 23-28, 2002, Atlanta, GA.

[29] A.M. Law and D. Kelton, “Simulation Modeling & Analysis” (Second Edition), McGraw-Hill, 1991.

[30] J.R. Lorch, A.J. Smith, ”Scheduling Techniques for Reducing Processor Energy Use in MacOS”, ACM/Baltzer Wireless Networks, 1997, pp.311-324.

[31] J.R.Lorch e A.J.Smith, ”Software Strategies for Portable Computer Energy Management”, IEEE Personal Communication - June 1998, pp.60-73.

[32] 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.

[33] M. Othman, S, Hailes, ”Power Conservation Strategy for Mobile Computers Using Load Balancing”, ACM Mobile Computing and Communication Review, Vol. 2, N. 1, January 1998, pp. 44-50.

[34] S. H. Phatak, V. Esakki, B. R. Badrinath and L. Iftode, ”Web&: An Architecture for Non-Interactive Web”, Proceedings of the Second IEEE Workshop on Internet Applications, WIAPP’01, July 2001.

[35] C.Poellabauer and K.Schwan, ”Power-Aware Video Decoding using Real-Time Event Handlers”, Proc. 5th ACM International Workshop on Wireless Mobile Multimedia (WoWMoM2002), September 28, 2002, Atlanta,GA.

[36] M.Stemm, and R.H.Katz, ”Measuring and reducing energy consumption of network interfaces in handheld devices”, IEICE Trans. Fund. Electron, Commun. Comp. Sci., Vol. 80, pp. 1125-1131 (Special Issue on Mobile Computing), 1997.

[37] M.Zorzi e R.R.Rao, ”Energy Constrained Error Control for Wireless Channels”, Proceeding of IEEE GLOBECOM ’96, pp.1411-1416, 1996.