A Power Saving Network Architecture for Accessing the Internet from Mobile Computers: Design, Implementation and Measurements

Giuseppe Anastasi, Marco Conti, Willy Lapenna

Abstract **In this paper we propose a power-saving architecture for accessing the Internet from a mobile computer. Firstly, we measure the power consumption of a mobile computer that uses a TCP connection to send/receive data from the Internet. These measurements indicate that, by adopting TCP, the power consumption is negatively affected by the congestion into the fixed-network. To solve this problem we extend the *Indirect TCP model* to achieve both the TCP reliability and an optimal power consumption level. To test our claim we design and implement a *Power Saving Network Architecture* based on an enhanced Indirect TCP model. Experimental results show that our approach is a promising direction, with respect to the classical TCP approach, to reduce power consumption. Specifically, in our experiments the power consumption, by exploiting the enhanced indirect model, is significantly lower than the power consumption measured when using the legacy TCP approach.**

I. INTRODUCTION

A mobile computer operates on a finite battery power that represents one of the greatest limitations to the utility of portable computers [12, 16, 23]. Projections on progress in battery technology show that only small improvements in the battery capacity are expected in next future [30]. If the battery capacity cannot be improved, it is vital that power utilization is managed efficiently by identifying ways to use less power preferably with no impact on the applications. Many researchers have focused on this problem. Strategies for power saving have been investigated at several layers including the physical-layer transmissions, the operating system, and the application levels [17]. Specifically, by focusing on power-saving at the transmission level, some authors have proposed and analyzed policies, based on the monitoring of the transmission error rates, which avoid useless transmissions when the channel noise makes low the probability of a successful transmission [29, 39, 40]. Power-saving policies at the operating system level include strategies for the CPU scheduling [22, 37] and for the hard-disk management [13]. At the application-level, policies that exploit the

This work was partially supported by the Italian Ministry for Education and Scientific Research (MIUR) in the framework of the Project "Internet: Efficiency, Integration and Security", and by the NATO Collaborative Linkage Grant PST.CLG.977405 "Wireless Access to Internet exploiting the IEEE 802.11 technology".

Correspondence Author: Marco Conti, CNR – IIT, Via G. Moruzzi 1, 56124 Pisa, Italy.
Marco Conti and Willy Lapenna are with the Italian National Research Council (CNR), Via G. Moruzzi 1, 56124 Pisa, Italy. e-mail: Marco.Conti@iit.cnr.it, Willy.Lapenna@cnuce.cnr.it.

application semantic[1] [11, 18, 25], or profit of tasks remote execution[2] [15, 16, 24, 28], have been proposed.

In this paper we investigate the relationship between power consumption and data transfer from/to mobile hosts in a mobile Internet environment. In this framework, recently, a lot of research activities concentrated on power saving in ad hoc networks [7, 8, 26]. Another emerging research area is related to the analysis of the energy consumption characteristics of secure protocols [19]. However, ad hoc networking and security are beyond the scope of this paper.

Data transfer from/to mobile hosts contributes to the power management problem as it uses significant power when data are sent and received. Experimental results have shown that power consumption related to networking activities is approximately 10% of the overall power consumption in a laptop computer, but it raises over 50% in current handheld devices [20].

Communication hardware and software, involved in data transfer, can be partitioned in two classes: sub-network and inter-network protocols. Sub-network protocols include network technologies such as LAN and WAN protocols, and are only involved in the information transfer between hosts connected to the same physical network. Inter-network protocols exploit the sub-network services to provide a data transfer between all the computers that have a network access. Currently, the TCP/IP protocol stack is the de-facto standard architecture for inter-network level.

The impact of network technologies on power consumption has been investigated in depth in [31, 32]. The power saving features of the IEEE 802.11 wireless LANs [14] have been analyzed in [27, 36, 38, 10]. The optimal tuning of the IEEE 802.11 MAC protocol for achieving the minimal energy consumption is investigated in [6]. A simple and effective mechanism for minimizing the power consumption in CSMA-based MAC protocols, such as the IEEE 802.11, has been proposed in [5]. Power-saving strategies at the sub-network level are strongly dependent on the network technology and no general solution can be developed. On the other hand, at the inter-network level the TCP/IP protocol stack is a de-facto standard,

---

Giuseppe Anastasi is with the University of Pisa, Department of Information Engineering, Via Diotisalvi 2, 56122 Pisa, Italy, e-mail: g.anastasi@iet.unipi.it.

[1] For example, for the applications involving data access, the base station can periodically broadcast the "hot spot" data, i.e. the information more frequently accessed by mobile users.

[2] For example, user's jobs are transferred from a mobile host to a fixed host to reduce power consumption by the mobile-host CPU.

and hence power-saving strategies at this level apply to almost any information transfer through a computer network.

In this work we investigate power-saving strategies at the Internet-level protocols, i.e., TCP/IP. To evaluate the impact of mobile-host data transfers on the consumption of its battery power we define an index for measuring the battery power consumption. Then, we apply this metric to estimate the power consumption of the TCP/IP protocol stack. Our study focuses on a mobile computer accessing an Internet remote host via an IEEE 802.11 wireless LAN. From this analysis we identify that the congestion inside the Internet, by interacting with the TCP protocol mechanisms [35], forces the IEEE 802.11 network interface in the idle state for most of the time. This is one of the main elements in causing an inefficient utilization of the battery power. For this reason, we consider the Indirect TCP model [3] as an interesting direction for power saving. Specifically, by extending the Indirect TCP model, we define a novel network architecture, named *Power-Saving Network Architecture* (PSNA), for accessing the Internet from a mobile host. In our architecture the transport connection between the mobile host and the fixed host is subdivided in two transport connections: one between the mobile host and an Access Point located at the border between the wireless and the wired networks, and one from the Access Point to the fixed host. The former connection is based on a new transport protocol, named Power-Saving Transport Protocol (PS-TP) that we have designed and implemented. The latter connection is a classical TCP connection. The interaction between the two TCP connections is asynchronous. A daemon on the Access Point acts as a proxy by transferring data between the two connections. In addition, the daemon provides buffers to store data when either a connection is not available or to smooth speed-difference between the connections. Therefore, we extend the Indirect TCP model to a proxy-based architecture.

Experimental results show that, by adopting our PSNA architecture the battery lifetime of a mobile host processing capacity may significantly increase. The actual increase depends on the network conditions. This is achieved by eliminating the consumption of the battery power caused by the network-interface idle state. For example, in the reference scenario adopted in this paper, we show that, with the PSNA architecture, we are able to save up to 98% of the battery power drained by the network interface when the legacy TCP approach is used.

The idea to use the Indirect TCP model for power saving, and to shut down the network interface when the wireless link is idle, is not new [9, 20]. In [9] a scenario similar to the one analyzed in this paper was considered. However, in that work, a serial (voice grade) line is used in the experimental measurements to simulate a wireless link. More precisely, a 28-Kbps modem was adopted thus obtaining an access line that represents a slow access line to the Internet such as a low capacity wireless link. However, it does not provide an adequate representation of the transmission error rates that typically affect a wireless link. Furthermore, the emphasis of that work was mainly on measurements aimed at analyzing the causes of power consumption in the legacy TCP/IP architecture. The possible advantages of an indirect approach for power saving are only sketched. No design and implementation of an architecture based on the indirect approach is provided.

In the present paper we significantly extend the work in [9] as: (i) a real 802.11 WLAN is used for interconnecting the portable computer to the fixed network; (ii) an architecture based on the indirect approach is designed and implemented; (iii) measurements on a real test-bed are used to evaluate the power-saving advantages of the proposed architecture.

In [20] a transport level scheme for power management in mobile communications is proposed. The basic idea is shutting down the communication device after an inactivity timer has expired and resuming it after a sleeping period. Inactivity timeouts and sleeping times are application dependent but fixed. On the other hand, our solution dynamically tunes the power-saving parameters to the network conditions. Specifically, inactivity timeouts and sleeping times are dynamically adjusted to the traffic conditions.

To summarize, this paper provides the following original contributions: *i*) it extends previous measurement results; *ii*) it defines the architecture and protocols of a Power-Saving Network Architecture; *iii*) it provides an implementation of this architecture to evaluate the performance of the indirect approach in a real environment; and *iv*) it presents a solution for dynamically tuning the power-saving parameters to the network conditions.

II. POWER CONSUMPTION MEASUREMENTS

The aim of any power saving strategy is to maximize the battery lifetime, or, in other words, to maximize the amount of work performed by a mobile host with its finite-capacity batteries. In this work we evaluate the amount of battery power drained by the network interface. For this reason we introduce the index, named I_{Power_saving}, to estimate the performance of a power saving strategy aimed at reducing the power consumption of the network interface. I_{Power_saving} is defined as the number of bytes transmitted or received by the mobile host per unit of energy consumed. In the following, we will compute this index as the ratio between the number of bytes transmitted or received by the network interface ($number\_of\_bytes$) and the amount of the battery power drained by the network interface to manage this amount of data ($battery\_consumption$):

$$I_{Power\_saving} = \frac{number\_of\_bytes}{battery\_consumption} \qquad (1)$$

The target of a power saving strategy is to maximize I_{Power_saving}.

To analyze the possible directions to maximize I_{Power_saving}, it is useful to re-elaborate Equation (1). Specifically, by dividing the numerator and the denominator in (1) by the observation time interval, we obtain that I_{Power_saving} is a joint function of the network-interface throughput and of the battery-power consumption per time unit due to the network-interface itself. Therefore, I_{Power_saving} can be improved by increasing the throughput and/or by decreasing the power consumption per time unit. In the following we first investigate separately the two quantities.

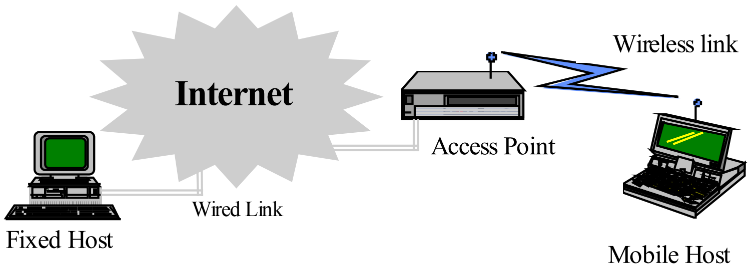

Figure 1a. Network Scenario

Figure 1 shows the network scenario we considered in our experiments. The mobile host was located in Pisa (Italy) while the fixed host was located at the Curtin University of Technology (Perth, Western Australia) in the School of Computing. Figure 1b highlights the route followed by data packets. We utilized a portable computer (with the Linux operating system) connected via a 2Mbps IEEE 802.11 WLAN to an Access Point attached to the Internet. The cards (Wavelan Turbo Card) and the Access Point (WavePoint II) for the WLAN were produced by Lucent Technologies. The portable computer was equipped a group of Li-Io batteries (BTP-31 Sony) providing a 10.8 Volt power, and with a capacity of 4050 mAh. As far as the TCP/IP protocol stack parameters, we used the default values provided by the protocol stack implemented in the adopted Linux release (Debian 2.0.36). The tuning of the TCP parameters to reduce power consumption was beyond the scope of the present paper. A study of these issues can be found in [1].

Figure 1b. Data path for the reference network scenario

II.1 THROUGHPUT ANALYSIS

In this section we investigated the throughput achieved by the mobile host when a TCP connection is used to transfer data to/from the fixed host. We ran an extensive set of experiments to determine the impact of the Internet TCP protocol on the throughput. To this end we implemented a simple application program that generates data with a speed that always guarantees a non-empty buffer at the sender host[3]. In such conditions the throughput of the destination host is only determined by the network traffic conditions.

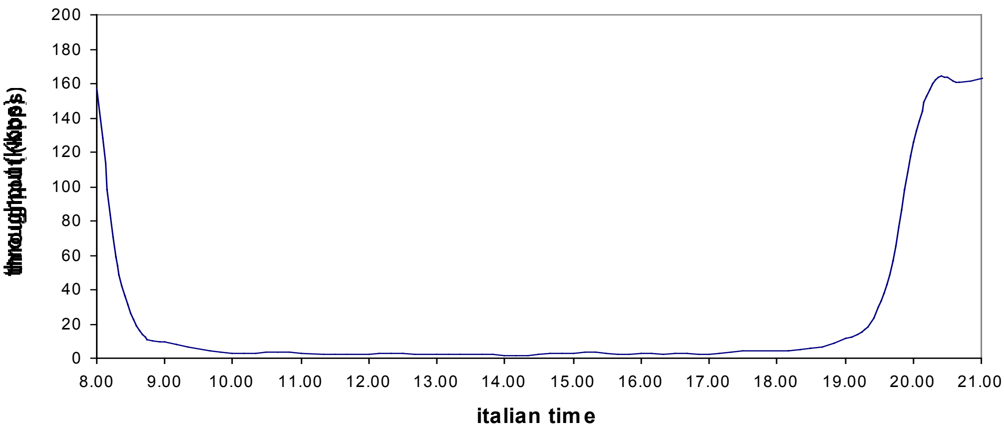

Figure 2. Data transfer from the fixed host to the mobile host

To assess the protocol behavior under various network conditions we performed our experiments for a wide time interval (about two months). Each type of experiment was performed several times (about twenty replications for each type of experiments) during each selected time interval and the statistics we present were averaged on all the performed

[3] This approximates an ftp session with a very large file to be transferred.

experiments. Figure 2 presents the estimated throughput[4] when data are transferred from the fixed host to the mobile host. To better highlight the throughput during the Italian business hours Figure 3 must be analyzed. We can observe that the throughput is always very far from the wireless link capacity (2 Mbps) which means that during business hours the wireless link (and, hence, the network interface) is usually in the idle state.

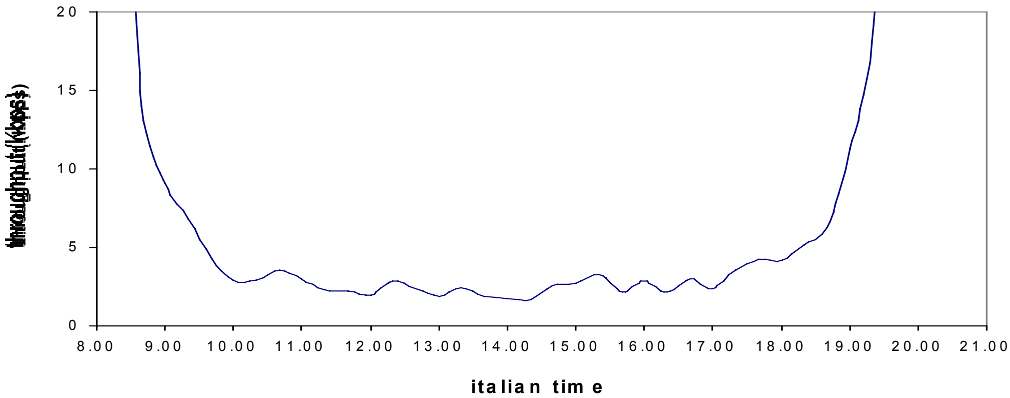

Figure 3. Zooming of Figure 3:TCP Throughput during peak-working hours

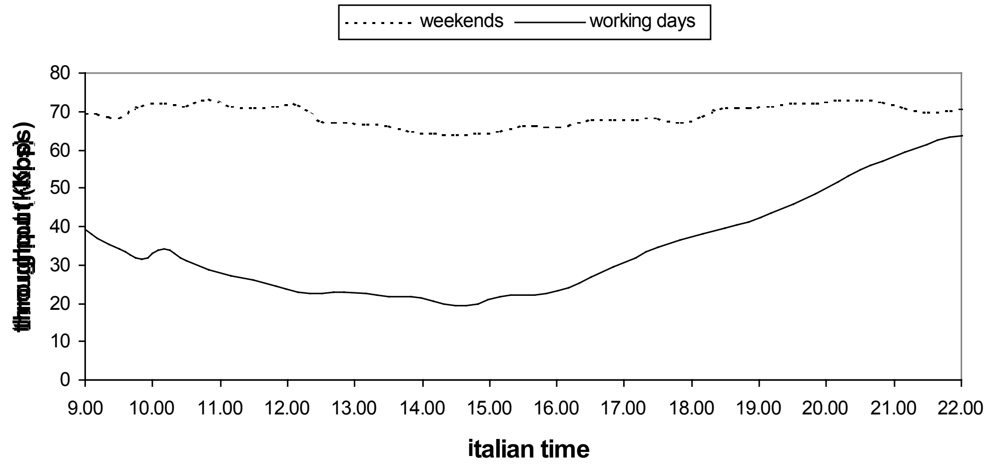

Figure 4. Data transfer from the mobile host to the fixed host

The second set of experiments was aimed at analyzing the throughput in the reverse conditions, i.e. when the data transfer is from the mobile host to the fixed host. The results obtained are shown in Figure 4. We can observe that, as in the previous experiments, the throughput is very low with respect to the wireless link capacity and, hence, the network interface is idle most of the time. In addition, during working days the throughput significantly reduces during peak business hours (i.e. maximum network congestion), while

[4] We sampled the throughput a ten-minute intervals. The throughput estimate at time n is obtained by adopting a moving average estimator with a size-5 window, i.e., the average of the samples: n-5, n-4, …., n-1, n, n+1, …, n+5.

during weekends the throughput is approximately constant and, generally, higher than during working days. Also it must be pointed out that the throughput of a data transfer from the mobile host to the fixed host is much higher than that experienced by the mobile host to download the data from the fixed host. To understand this difference we derived the list of links along the path between Pisa and Perth (and vice-versa) by using the `traceroute` program [33] and, then, we analyzed the traffic statistics collected by the Italian research network manager for the above links. We found that the direct and reverse paths between Pisa and Perth are not symmetric and, therefore, are subject to different traffic conditions.

II.2 NETWORK INTERFACE POWER CONSUMPTION

The results presented in the previous section show that (due to the congestion in the fixed network) the throughput is generally much lower than the wireless network speed. Therefore, the wireless network interface remains in the idle state for a large percentage of time. This means that reducing the battery consumption during idle periods can produce significant power savings, hence contributing to maximize the I_{Power_saving} index. In this section we analyze the power consumption when the network interface is in the idle state. Specifically, we maintain the portable computer in the *stand_by* state (i.e., it does not send or receive packets, but simply schedules the operating system processes) and we analyze the battery lifetime as a function of the state, ON or OFF, of the Wavelan card. In the ON state the network interface is continuously in the idle state, while when it is in the OFF state no power is provided to the interface.

As shown in Figure 5, when the network interface is ON the battery lifetime reduces of about 22 minutes corresponding to about 11.5% of the total battery lifetime. These results are aligned with similar results available in the literature [20].

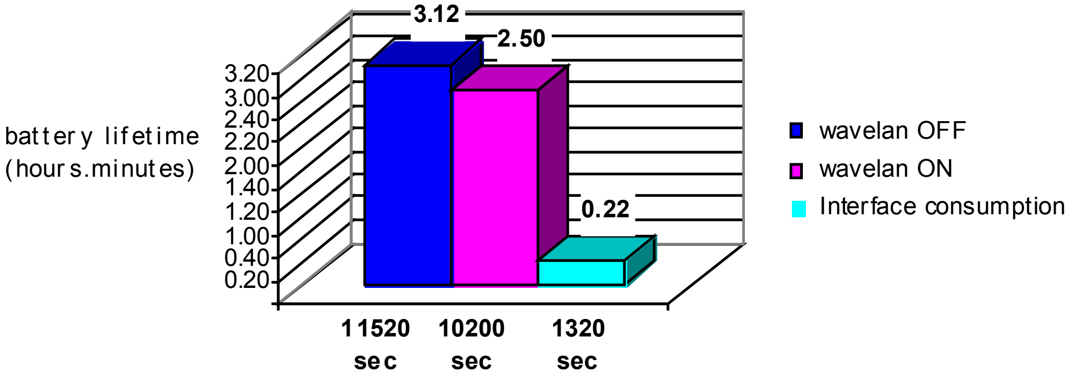

Figure 5. Battery power consumption when the portable computer is in the *stand_by* state

As our target is to define policies that maximize I_{Power_saving}, our aim will be to reduce the amount of the time the network interface remains in the idle state. The percentage of battery lifetime consumed by the network interface when it is idle might seem too low to justify the implementation of specific power saving strategies. However, this percentage (that is in the order of 10% for laptop computers) can represent over 50% of the total system power consumption for current hand-held devices [20].

II.3 EVALUATION OF I_{Power_saving}

We ran an extensive set of experiments to determine the amount of data that a portable computer can transfer to/from the network servers by exploiting its finite battery capacity. As before, we used a simple application program that generates data with a speed that always guarantees a non-empty buffer at the sender host.

The most accurate way to measure power consumption would be to insert appropriate electronic instrumentation between the battery and the computer. Due to practical problems, we utilized a less direct approach. Specifically, we completely charged a battery, and we ran an experiment until the battery dies. To reduce the inaccuracies induced by our measurement approach we ran each experiment several times. The results presented in Table 1 are averaged on all performed experiments. It is worth noting that due to: *i*) the long transfer phase, and *ii*) the throughput stability during the transfer, we observed a low results' variability, less then 5%.

Since the runs we performed caused the battery to discharge and recharge several times we could expect that the battery's characteristics (power storage and consumption) might change over the time. However, our experimental results do not show meaningful differences (from the battery characteristics' standpoint) between the first and last experiment we performed.

Experiments can be classified into two types. In the type A experiments the mobile host is the source of the data while, in the type B experiments, the mobile host receives data. Both types of experiments were performed during the peak working hours in Italy. Specifically, the experiments started around 11 a.m. Italian time.

Type of experiment	Battery lifetime (sec)	Battery Consumption[5] (sec)	Delivered data (KBytes)	Throughput (Kbps)
A	10005	1515	25446	19.87

[5] These values are obtained as a difference of the battery lifetime value in the experiment of II.2 subsection (with the portable computer in the *stand_by* state and wavelan OFF), and that obtained in the A or B experiments.

B	10153	1367	2519	1.82

Table 1: Amount of delivered data by exploiting all the battery capacity

The results presented in Table 1 point out that, from the power saving standpoint, there are no marked differences between maintaining the network interface of the mobile host in the transmitting or in the receiving state. When the network interface is in the transmitting state for most of the time (see type A experiments), the battery lifetime reduces, with respect to type B experiments, of about 150 seconds. However, it is worth noting that, due to the asymmetric characteristics of the two links (see Section II.1), during type A experiments a larger quantity of data is managed on the mobile host. This implies an increase of the power consumption in all the components of the mobile host (disk, CPU, network interface, etc.).

Some interesting observations on the relationship between the battery lifetime, and the state of the network interface, can be derived by comparing the results in Table 1 with those presented in Section II.2. From this comparison, we note that there is not a marked difference, from the power saving standpoint, between maintaining the network interface in the idle state (i.e. the network interface is active but no data are transferred or received) or in the transmitting/receiving state. When the network interface is in the ON state, but no data are transferred, the total battery lifetime is about 10200 seconds. This corresponds to a reduction of the battery lifetime of about 1320 seconds with respect to the case in which the network interface is in the OFF state. On the other hand, we have that the *battery_consumption* (measured in seconds of battery lifetime) in type A and type B is 1515 and 1367 seconds, respectively. This means that the network-interface consumption for transmitting/receiving is only 195 and 47 seconds in type A and type B experiments, respectively. Hence the costs for transmitting/receiving packets are in the two cases about 14% and 3.5% of the power consumption due to the network-interface in the idle state.

Finally, from the results in Table 1, we can also derive the power saving index for type A and type B experiments. Specifically, by applying Equation (1) we have:

$$I_{Power\_saving}(\text{type A}) = \frac{25446}{1515} = 16.8 \text{ Kbytes/seconds\_of\_battery\_lifetime}$$

$$I_{Power\_saving}(\text{type B}) = \frac{2519}{1367} = 1.84 \text{ Kbytes/seconds\_of\_battery\_lifetime} \quad (2)$$

The power saving indices show that type-B experiments have worse performance than those of type A. This is due to the TCP protocol behavior in the presence of network congestion. This protocol, which is the most commonly used by Internet applications, attributes the loss of packets that occurs in the network to congestion. The mechanisms for controlling the congestion of the TCP react to such losses by drastically reducing the speed of data transmission, with a consequent degradation in the performance of the active connections [35]. Thus, when congestion occurs in one of the intermediate routers crossed by a TCP connection, this affects the throughput of the connection on each link it crosses. This means that, in our experiments, the congestion inside the Internet produces an under utilization of the wireless link, and hence an increase of the time the network interface of the mobile host remains in the idle state. Since the power consumption of the network interface depends almost linearly on the time the interface is ON, it follows that the congestion increases the power consumption. In our experiments we observed an asymmetric level of congestion inside the network between Pisa and Perth. Specifically, the highest level of congestion was measured in the path from the fixed host (in Perth) to the mobile host (in Pisa). This causes, in type B experiments, a lower value for the I_{Power_saving} index with respect to type A experiments.

The major outcome deriving from the above analysis on the power saving index is the observation that a power saving policy that wishes to optimize the battery consumption, from the network interface standpoint, should avoid that the network interface remains in the idle state.

III. Power Management through the Indirect TCP Model

Previous results indicate that TCP congestion control mechanisms reduce the utilization of the access link and this negatively affects the battery power consumption.

On the other hand, a reliable communication service, such as that provided by the TCP protocol, is desirable by most Internet applications. Thus, from a power saving standpoint, it would be desirable to have an end-to-end TCP-like service without the interference of the Internet congestion control mechanisms on the access link utilization. Similar requirements occur by looking (from a different standpoint) at a TCP connection between a mobile host (connected to Internet via a wireless link) and an Internet fixed host.

Numerous authors have studied how to separate the effects on TCP of the wireless link and of the fixed-network characteristics. Among the proposed solutions, we identified the Indirect TCP approach as the most suitable for the power saving problem as well [3].

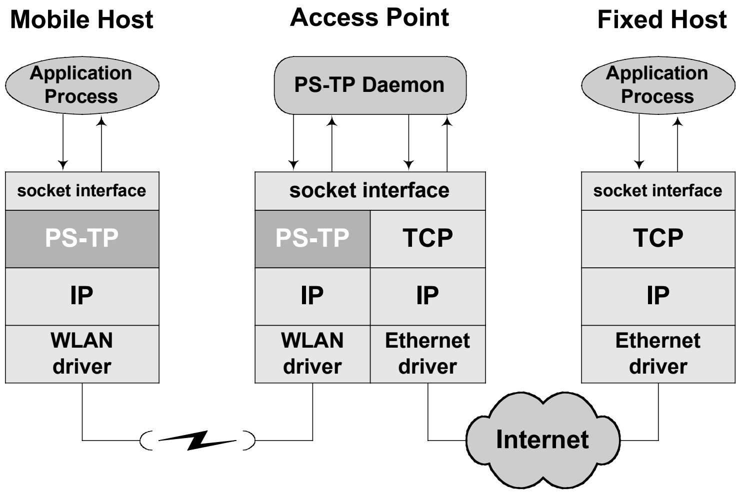

Figure 6. Power-Saving Network Architecture

By exploiting the basic idea of the Indirect TCP approach, we have designed a new network architecture for integrating a mobile host in the Internet by maintaining (from the application standpoint) the TCP type of service but avoiding the TCP negative interference on power saving. Our architecture is shown in Figure 6. According to our architecture, a TCP connection between a mobile host and a fixed host is divided in two connections: a TCP connection from the mobile computer to the Access Point (i.e., a router on the fixed network that provides the access to Internet), and a legacy TCP connection between the Access Point and the fixed host. A daemon (PS-TP Daemon in Figure 6) operating on the Access Point is in charge to transfer the traffic between the two connections. A buffer is used to absorb temporary differences in the speed of the two connections (e.g. periods during which the TCP connection between the access point and the fixed host slows down due to the congestion recovery procedures). As far as the former connection is concerned (i.e. the connection between the mobile host and the Access Point) it must be pointed out that it operates between two directly connected computers (i.e., via a point-to-point link or a wireless LAN, *WLAN*). Therefore, not all the functionalities implemented by a legacy TCP protocol [33] are necessary. For example, congestion control mechanisms are not necessary while the efficiency of the error detection and recovery mechanisms can be enhanced taking into consideration the characteristics of the wireless environment. This means that a simplified transport protocol can be adopted for providing a reliable transport connection between the mobile host and the Access Point. This would reduce the weight of the transport protocol on the CPU and thus would further reduce the power consumption of the mobile host. In our architecture the simplified transport protocol operating between the mobile host and the Access Point is named *Power Saving Transport Protocol* (*PS-TP*).

Specifically, the PS-TP protocol is a reliable connectionless transport protocol. We decided to use the connectionless approach to reduce the amount of work performed by the mobile host whenever it needs to start a data transfer/receive phase with the Access Point. As it will be clarified below, during the data transfer phase the status of the network interface may alternate many times between the ON and OFF states. By adopting a connection-oriented approach we would have to pay several times the costs associated with the set-up and tear down of the transport connection.

Taking into account that the round trip time between the mobile host and the Access Point is almost negligible, to guarantee a reliable data transfer we have included in the PS-TP a simple ARQ mechanism based on the Stop and Wait paradigm [4].

It is worth pointing out the major differences between the PSNA approach and the Indirect-TCP model. Like the legacy TCP, the Indirect TCP maintains synchronous communication between the TCP sender and receiver. On the other hand, the interaction between the two TCP connections with the PSNA approach is asynchronous. The daemon on the Access Point acts as a proxy by transferring data between the two connections. The daemon provides buffers to store data when either a connection is not available or to smooth speed-difference between the connections. Finally, we have designed a simple transport protocol optimized to operate on the wireless-link connection.

To better clarify the interaction between a mobile host and a fixed host by adopting our Power-Saving Network Architecture in the following we discuss the two types of operations that a mobile host can perform: Read access and Send access. We have selected these two operations because they are commonly used by all distributed applications. In the Read case the mobile host downloads some information from the fixed host, e.g. a web server, while in the Send case it sends data to a fixed host. In both cases, we assume to know the amount of data to be delivered/retrieved.

Read Access. The operations performed by the mobile host are executed in (at least) two steps. In the first step the mobile host delivers the read (i.e., download) request to the end system by exploiting the services of the daemon operating on the Access Point. In the second step the mobile host retrieves the requested data from the Access Point. It is worth noting that in this second step the PS-TP protocol is adopted and hence the data are delivered to the mobile host in a continuous way, without any interference from the congestion inside the Internet. To have an optimal behavior from the power saving standpoint, in the time interval between the two steps mentioned above, the network interface should be in the OFF state.

However, the length of this time interval depends on the *Transfer_delay*, i.e. the time it takes the fixed host to deliver its data (through the Internet) to the Access Point on a TCP connection. *Transfer_delay* is obviously a random variable, and its statistics are continuously computed by the Access Point. More precisely, the Access Point estimates the throughput of the TCP connection between the Access Point and the fixed host and, based on these estimates, computes the residual time to complete the data transfer. Specifically, to perform the throughput estimate, the data-transfer phase is subdivided in intervals of size Δ_i, and a throughput sample is obtained from each sub-interval. The throughput sample in the *i*-th time interval, say λ_i, is:

$$\lambda_i = \frac{\text{Data\_Received\_in\_}\Delta_i}{\Delta_i} = \frac{DR_i}{\Delta_i}$$

where DR_i is the amount of data received in the *i*-th time interval and Δ_i is the size of the *i*-th time interval. The size Δ_i is a random quantity that depends on the rate data are received. To have meaningful samples, at least ten data packets must be received in Δ_i.

Finally, from the throughput samples, λ_i, we dynamically estimate the throughput at the end of the *n*-th subinterval, as follows:

$$\overline{\lambda}_n = \alpha \cdot \overline{\lambda}_{n-1} + (1-\alpha) \cdot \lambda_n \qquad (3)$$

where α is a smoothing factor ($0<\alpha<1$). Hereafter, it is assumed $\alpha = 0.9$ that is the value commonly used in Internet estimates [34]. With $\alpha=0.9$, the *n*-th throughput estimate mainly depends on the previous samples, in average 10. At the beginning of the data transfer phase, when we do not have already collected at least 10 samples, the throughput is estimated by the sample mean.

Figure 7 provides a general overview of the interactions between the mobile host, the Access Point and fixed host during a read operation. Furthermore, it highlights the time intervals when the WLAN interface on the mobile host is ON and OFF, respectively. Figure 8 shows the actions performed by the mobile host. For sake of simplicity acknowledgements have been omitted. Furthermore, the pseudo-code is optimized for clarity rather than for efficiency.

Upon receiving the read request from the application the mobile host resumes the WLAN card (if necessary) and sends the request to the Access Point (AP) to open a connection with the remote fixed host FH (see Figure 8, lines 1-4). Then, the mobile host does not turn its wireless network interface in the OFF state but it remains in the receive state waiting for some information on the estimated residual time to complete the data transfer (line 6). Specifically, as soon as the Access Point has computed a meaningful estimate of the residual time returns this information to the mobile host. In Figure 7 this is represented by the message wait(x1), where x1 is the estimated value of the residual time. By exploiting this information the mobile host decides the amount of time its network interface should remain OFF (lines 7-12).

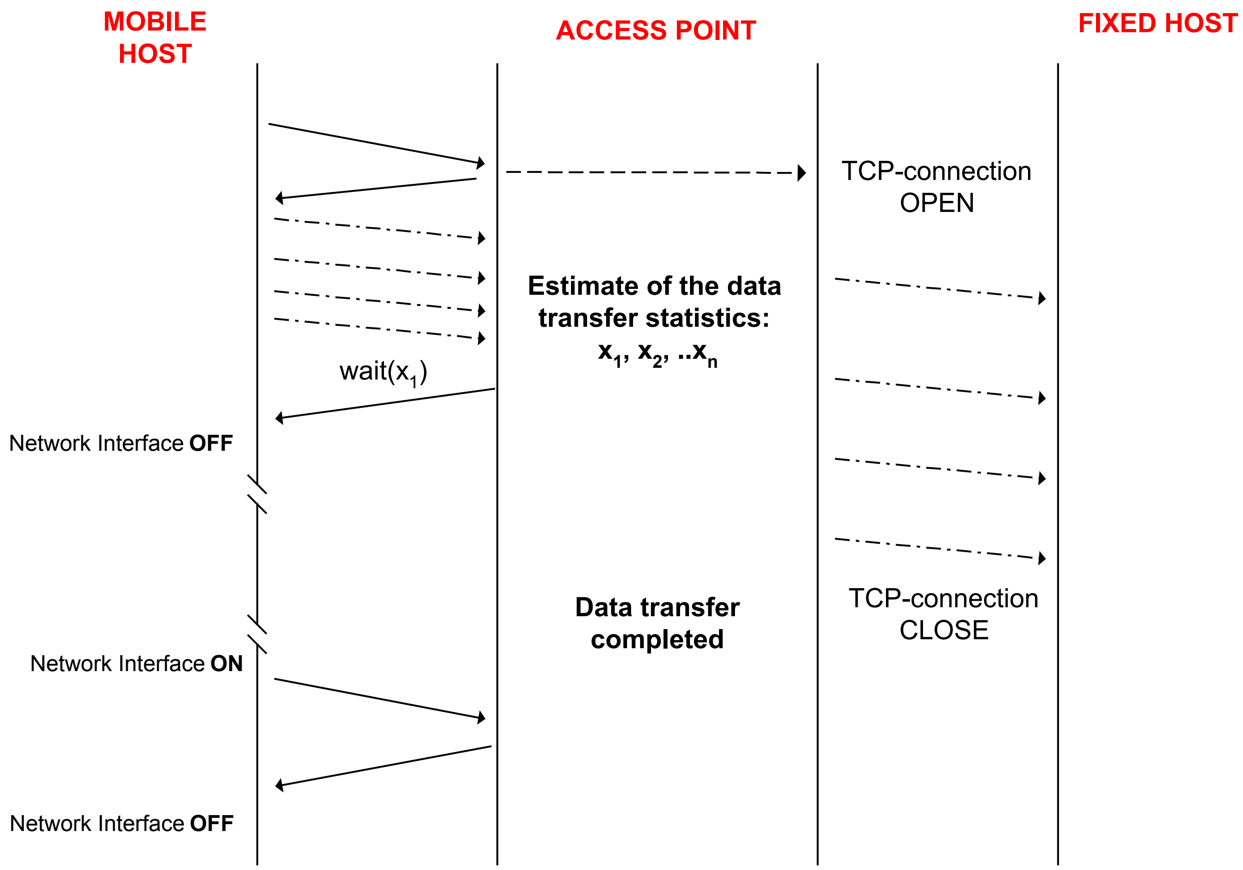

Figure 7. Read-access operations

As the information related to the residual time are generally not precise the mobile host may turn too early its network interface ON. In this case after a poll to the Access Point, it receives a more accurate estimate of the residual time (see wait(x2) in Figure 7, and lines 5-14 in Figure 8). Finally, after some polls, the mobile host receives the indication that all data are ready in the buffer of the Access Point and can be downloaded in a continuous way over the wireless link (line 15). At the end of the download process the WLAN card is turned off again (line 16).

```
Upon reception of a ReadRequest
  do
    if (CardMode=OFF) then ResumeCard();
    send(AP, <Read from FH>);
    repeat
      receive(AP, <Residual_Time>)
      if (Residual_Time >0) then
        begin
          SuspendCard();
          wait(Residual_Time);
          ResumeCard();
          send(AP, <Poll>);
        end
    until (Residual_Time=0);
    DownloadFrom(AP, <data>);
    SuspendCard();
  od
```

Figure 8. Actions performed by the PS-TP at the mobile host for a Read Operation from the remote fixed host.

The above presentation provides a summary of the architecture's features. For example, we have assumed that the data transfer from the Access Point to the mobile host occurs only when *all* data are available in the buffer of the PS-TP daemon. This choice maximizes the utilization of the wireless link during the data transfer phase but introduces a delay that may be excessive for some applications. For delay sensitive applications we need to identify policies that balance between the maximum link utilization and the extra-delay introduced in the data transfer. The analysis of this aspect is beyond the scope of this paper.

```
Upon reception of a SendRequest
  do
    if (CardMode=OFF) then ResumeCard();
    send(AP, <Send to FH>);
    receive(AP, <OK to send>)
    DownloadTo(AP, <data>);
    repeat
      receive(AP, <Residual_Time>)
      if (Residual_Time >0) then
        begin
          SuspendCard();
          wait(Residual_Time);
          ResumeCard();
          send(AP, <Poll>);
        end
    until (Residual_Time=0);
    SuspendCard();
od
```

Figure 9. Actions performed by the PS-TP at the mobile host for a Send Operation to the remote fixed host.

Send Access. The operations performed by the Power-Saving Network Architecture when the mobile host sends data to a fixed host are summarized in Figure 10 while Figure 9 reports the actions performed at the mobile host.

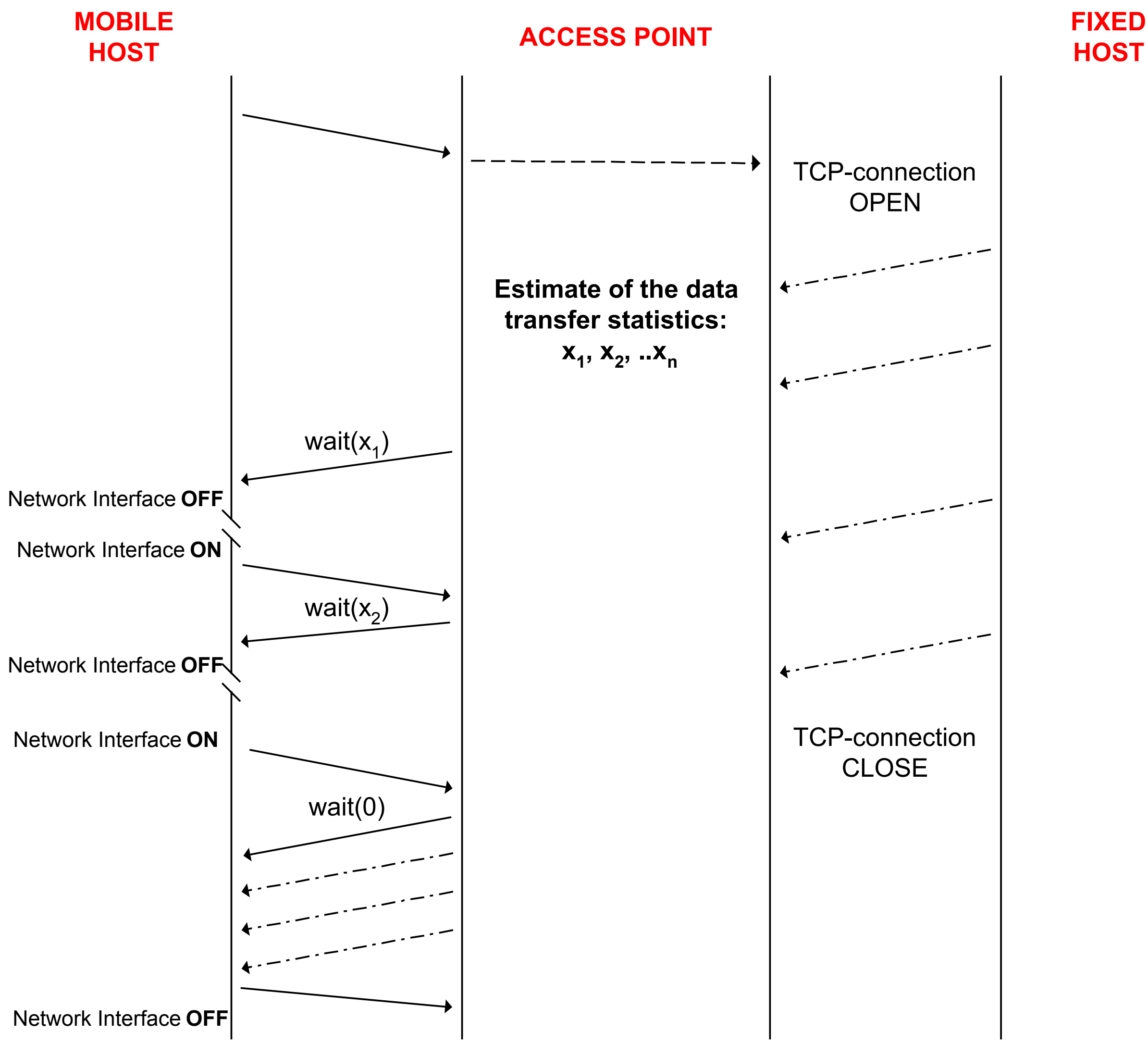

Figure 10. Send-access operations

Upon receiving, from the application, a send request for the remote host FH, the mobile host resumes the WLAN card (if necessary) and sends a signaling packet to the Access Point requiring that a TCP connection is opened between the Access Point and the remote fixed host FH (lines 1-4 in Figure 9). As soon as the mobile host receives the reply from the Access Point (line 5), it starts sending (on the wireless link) all its data packets to the Access Point (line 6). Meanwhile the Access Point transmits the mobile-host data to the fixed host along the TCP connection and, simultaneously, estimates the residual time to complete the data transfer.

In Figure 10 we have assumed that the network interface of the mobile host remains OFF for a time interval that it is enough to complete the remote data transfer. Hence, when the mobile host re-activates its network interface and polls the Access Point, it immediately receives a message from the Access Point indicating that all data have been correctly sent to the fixed host. At this point the data transfer is complete and the mobile host puts its network interface in the OFF state again.

However, it may happen that the mobile hosts polls the Access Point too early, when the remote data transfer (i.e. from Access Point to the fixed host) is still ongoing. In this case the Access Point returns to the mobile host a new estimation of the residual time and the mobile host suspends and resumes the wireless card accordingly. The mobile host will definitely suspend the network interface as soon as it receives the indication that the data transfer to the fixed host has been completed (lines 7-17).

IV. EVALUATION OF THE PSNA ARCHITECTURE

To evaluate the effectiveness of the Power-Saving Network Architecture we developed a prototype implementation of its components. Specifically, we implemented: *i*) the PS-TP protocol on the mobile computer used for the experiments presented in Section II.1-II.3, and *ii*) the PS-TP protocol and the PS-TP daemon on the Access Point. The details of this implementation are presented in [21].

Type of experiment	Architecture	Battery lifetime (sec)	Battery Consumption (sec)	Delivered data (KBytes)	Throughput (Kbps)	Data transfer length (sec)
A	legacy TCP	10005	1515	25446	19.87	10005
A	PSNA	11501	19	25446	1525.48	130
B	legacy TCP	10153	1367	2519	1.82	10153
B	PSNA	11512	8	2519	1525.48	13

Table 2: Comparison of the legacy and Indirect TCP approaches.

By exploiting the PSNA implementation we evaluated the I_{Power_saving} index to estimate the performance from the power-saving standpoint of the PSNA architecture. Specifically, we replicated for the PSNA architecture the two types of experiments (type A and B) performed in Section II.3 to evaluate the performance (from the power-saving standpoint) of the legacy TCP/IP architecture.

More precisely, by using the PSNA architecture, we measured the battery lifetime when the mobile host sends (type A experiments) or receives (type B experiments) the same amount of data that the mobile host itself can send or receive with the legacy TCP approach by exploiting all the battery capacity (see Section II.3).[6] Then, by remembering that the battery lifetime when the network interface is OFF corresponds to 11520 seconds (see Section II.2), we derived the *battery_consumption* in the PSNA architecture for type A and type B experiments, respectively. The results of the experiments are presented in Table 2. In this

[6] In the experiments, at the end of the data transfer, the network interface is turned OFF.

table we report results averaged on all the performed experiments. As we observed in Section II.3, the results variability is very low, less than 5%.

As shown in Table 2, the *battery_consumption* is 19 and 8 seconds for type A and type B experiments, respectively. Hence, by applying Equation (1) we can estimate the power saving index for the PSNA architecture.

$$I_{Power\_saving}(\text{type A, PSNA}) = \frac{25446}{19} = 1339 \text{ Kbytes/seconds\_of\_battery\_lifetime} \quad (3)$$

$$I_{Power\_saving}(\text{type B, PSNA}) = \frac{2519}{8} = 314.8 \text{ Kbytes/seconds\_of\_battery\_lifetime} \quad (4)$$

In both cases we have a significant improvement with respect to the corresponding experiments performed with the legacy TCP approach. Specifically, by using the PSNA architecture we increase the power saving index, and, hence, the amount of delivered/received data per unit of battery power, of about 80 and 171 times in the type A and B experiments, respectively. In addition, we have that the battery consumption due to the network interface reduces more than 98%.

V. DISCUSSION

In this paper we have shown the power-saving effectiveness of the PSNA architecture. The results presented above have been obtained by considering a high-speed wireless link that guarantees a throughput on the access link much higher than that available inside the Internet. It could be questioned to what extent the effectiveness of the proposed architecture depends on the characteristics of wireless access link. Our claim is that the advantage of the indirect model still exists even with low-speed access links (or congested WLANs), i.e., when the throughput discrepancy between the access and backbone links is small. The results presented in [9] support our claim. In that paper it is shown that a significant power saving is obtained by using an access link with a maximum speed of 28 Kbps, i.e., a speed comparable with the values of the average throughput estimated in the backbone Internet (see, for example, Section II.1). More precisely, the indirect approach always provides an advantage over the legacy TCP approach. The reason is the TCP congestion control mechanism that prevents to fully utilize the capacity of the access link, even for low-speed links. In fact, when congestion occurs inside the Internet, packets are discarded, and this often generates a time-out at the TCP sender. The time-out interval is long when accessing a remote server, and for all this period no useful traffic is transmitted on the access link. Furthermore, after the time-out the sender cannot utilize the link at its maximum speed because the transfer rate increases according to the slow-start algorithm. To summarize, each time-out experienced by the TCP sender forces an underutilization of the access link. During this period no (or a small amount of) data is transmitted on the link while the network interface is ON and this wastes battery power.

The PSNA architecture has several potentialities that have not been investigated in this paper. One interesting extension of this architecture is presented in [2] where it is shown that the PSNA approach is able to provide power saving for mobile Web access still guaranteeing an acceptable QoS.[7]

In this work we have not investigated the impact of user mobility on our architecture. We think that our architecture can be extended to efficiently manage user mobility. This can be done in several ways. If the PS-TP Daemon is implemented at the AP, we need to add a cooperation protocol among daemons. Every time the user moves from one AP to another, packets must be re-routed to the AP (PS-TP daemon) currently in charge of the user. This approach however may generate a lot of traffic among APs that may be critical if a user moves very frequently during a session. This problem can be alleviated by implementing the PS-TP daemon at a proxy server located in some place inside the access network[8]. In this case, as long as the user remains inside the same access network, no traffic redirection is required.

VI. CONCLUSIONS

In this work we have shown that the transport protocol policy has a significant impact on the length of the data-transfer phase observed by a mobile host, and that by reducing the time the network interface is ON we can extend significantly the battery lifetime. To this end we have defined and implemented a Power Saving Network Architecture (PSNA) that extends the Indirect TCP approach. Specifically, we have added to the Access Point a proxy-like functionality that enables both synchronous and asynchronous transport-level communications. The asynchronous communications paradigm enables communication from/to a mobile host also when it is disconnected. Another novel aspect of PSNA is the use of an ad-hoc transport protocol for transferring data to/from the mobile host and the Access Point. This transport protocol is tuned to provide power saving on wireless links.

[7] The additional delay introduced in the transfer of Web pages is in the order of 1 sec or less.
[8] In this case the PS-TP daemons operates similarly to a Web proxy cache.

We have performed a set of experimental measures to estimate the power consumed to transfer the same amount of data (on a large distance Internet connection) by using the legacy TCP approach and the PSNA architecture, respectively. The results show that, by using the PSNA architecture, it is possible to increase the amount of data transferred per unit of battery power of approximately 100 times with respect to the legacy TCP approach. The advantage of PSNA can also be expressed, by looking at the battery consumption. Specifically, with the indirect approach to transmit a given amount of data we save up to 98% of the network-interface battery consumption. One may argue that, even though our policy is really effective for saving the network-interface battery consumption, the network interface only contributes to a small percentage (11.5% in our experiments) of the overall (mobile-computer) power consumption. However, it is worth noting that the network interface can represent over 50% of the total system power consumption for current hand-held devices [20]. For this reason it is an interesting research issue to test the PSNA architecture when a PDA is used as the mobile device. The work still to be performed consists in the implementation of the client-side portion of the PSNA architecture, and it is the objective of further studies.

In this paper we have shown the PSNA effectiveness, from the power saving standpoint, by analyzing an ftp-like application. The ftp-like client, running on a mobile host, performs either a Read (i.e., a `get` command) or a Send (i.e., a `put` command) operation from/to a remote ftp-like server. An example of concurrent use of Read and Send operations is presented in [2] where the PSNA approach has been recently extended also for Web browsing that uses both Read and Send operations inside the same session.

It is also worth noting that using the PSNA approach for integrating a mobile host in the Internet, beyond power saving, presents additional advantages such as:

i) maximization of the throughput on the wireless link thus minimizing the time the mobile host needs to use the wireless link itself;

ii) separation between the mobile and the fixed network. These environments are not homogeneous and the separation avoids that the problems on the mobile network (e.g., packets lost due to errors on the wireless link) negatively affect the TCP connection in the fixed network.

Acknowledgements

The authors wish to express their gratitude to Mohan Kumar for giving the opportunity to use the computing facilities of the Curtin University of Technology, Perth (Western Australia).

References

[1] Agrawal, S., Singh, S. (2001) An Experimental Study of TCP's Energy Consumption over the Wireless Link. Proceedings of the 4th European Personal Mobile Communications Conference, Vienna, Austria, February 20-22.

[2] Anastasi, G., Conti, M., Gregori, E., Passarella, A. (2002) A Power Saving Architecture for Web Access from Mobile Computers. Proceedings of the 2nd IFIP TC-6 Networking Conference (Networking 2002), Pisa, Italy, May 19-24, pp. 240-251, Springer-Verlag, Berlin.

[3] Bakre, A., Badrinath, B. R. (1997) Implementation and Performance Evaluation of Indirect TCP. *IEEE Transactions on Computers*, Vol. 46, No. 3, 260-278.

[4] Bertsekas, D., Gallager, R. (1992) Data Networks. Prentice-Hall, Inc. Upper Saddle River, NJ, USA.

[5] Bononi, L., Conti, M., Donatiello, L. (2001) A Distributed Mechanism for Power Saving in IEEE 802.11 Wireless LANs. *ACM/Kluwer Mobile Networks and Applications (MONET)*, Vol. 6. N.3, 211-222.

[6] Bruno, R., Conti, M., Gregori, E. (2002) Optimization of Efficiency and Energy Consumption in p-Persistent CSMA-Based Wireless LANs. IEEE Transactions on Mobile Computing, Vol.1, N.1, 10-31.

[7] Campbell, A., Conti, M., Giordano, S. (2003) Special Issue on "Mobile Ad Hoc Network". *ACM/Kluwer MONET,* Vol. 8, No. 5.

[8] Conti, M., Giordano, S. (2002) Special Issue on "Mobile Ad Hoc Networking". Cluster Computing Vol. 5, No. 2, 117-131.

[9] Conti, M., Donatiello, L. , Mazzoni, F. and Zobbi, A. (1998) Exploiting an Indirect TCP Model for Power Saving: Remarks and Experimental Results. Proceedings of *MoMuC'98* Fifth International Workshop on Mobile Multimedia Communication, Berlin, Germany, October 12-15, pp. 213-222.

[10] Ebert, J.P., Stremmel, B., Wiederhold, E. and Wolisz, A. (2000) An Energy-efficient Power Control Approach for WLANs. *Journal of Communications and Networks (JCN)*, vol. 2, no. 3, 197-206.

[11] Flinn, J., Park, S., and Satyanarayanan, M. (2002) Balancing Performance, Energy, and Quality in Pervasive Computing. Proceedings of the *22nd IEEE International Conference on Distributed Computing Systems (ICDCS)*, Vienna, Austria, July 2-5,217-226.

[12] Forman, G. H., Zahorjan, J. (1994) The Challenges of Mobile Computing. IEEE Computer, Vol. 27, No. 4., pp. 38-47.

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

[14] IEEE Std 802.11-1997 (1997) IEEE Standard For Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.

[15] Imielinski, T., Vishwanathan, S., Badrinath, B.R. (1994) Power Efficient Filtering of Data on air. *Proceedings of the EDBT,* Cambridge, United Kingdom, March 28-31, pp. 245-258, Springer.

[16] Imielinski, T., Badrinath, B.R. (1994) Wireless Computing. *Communication of the ACM*, Vol. 37, No. 10, 18-28.

[17] Jones, C., Sivalingam, K., Agarwal, P. and Chen, J. C. (2001) A Survey of Energy Efficient Network Protocols for Wireless and Mobile Networks. *In ACM/Kluwer Wireless Networks (WINET)*, vol. 7, No. 4, 343-358.

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

[19] Karri, R., Mishra (2002), P. Optimizing the energy consumed by secure wireless session – Wireless Transport Layer Security case study. *ACM/Kluwer MONET*, to appear.

[20] Kravets, R., Krishnan, P. (1998) Power Management Techniques for Mobile Communication. Proceedings of The Fourth Annual *ACM/IEEE International Conference on Mobile Computing and Networking* (MOBICOM'98), Dallas, Texas, United States, pp. 157-168, ACM Press New York, NY, USA.

[21] Lapenna, W. (2000) Politiche di Power Saving per l'accesso a server Web da computer mobili. *Laurea Thesis*, Pisa , Italy.

[22] Lorch, J.R. and Smith, A.J. (1997) Scheduling Techniques for Reducing Processor Energy Use in MacOS. *ACM/Baltzer Wireless Networks*, Vol. 3, No. 5, 311-324.

[23] Lorch, J. R. and Smith, A. J. (1998) Software Strategies for Portable Computer Energy Management. *IEEE Personal Comunications*, Vol. 5, N. 3, 60-73.

[24] Othman, M. and Hailes, S. (1998) Power Conservation Strategy for Mobile Computers Using Load Balancing. *ACM Mobile Computing and Communication Review*, Vol. 2, N. 1, 44-50.

[25] Phatak, S. H., Esakki, V., Badrinath, B. R. and Iftode, L. (2001) Web&: An Architecture for Non-Interactive Web", Proceedings of the Second IEEE Workshop on Internet Applications, WIAPP'01, San Jose, CA, 23-24 July.

[26] Petrioli, C., Rao, R. and Redi, J. (2001) Special Issue on "Energy Conserving Protocols. *ACM/Kluwer MONET,* Volume 6 , Issue 3.

[27] Rohl, C., Woesner, H. and Wolisz, A. (1997) A Short Look on Power Saving Mechanisms in the wireless LAN Standard Draft IEEE 802.11. The 6th WINLAB Workshop on third generation Wireless Systems, New Brunswick, NJ, USA, 21-23 March.

[28] Rudenko, A., Reiher, P., Popek, G.J., and Kuenning, G.H. (1998) Saving Portable Computer Battery Power through Remote Process Execution. *ACM Mobile Computing and Communication Review*, Vol. 2, N. 1, 19-26.

[29] Rulnick, M. and Bambos, N. (1996) Mobile Power Management for Wireless Communication Networks. *ACM/Baltzer Wireless Networks*, Vol. 3, No. 1.

[30] Sheng, S., Chandrakasan, A. and Brodersen, R.W. (1992) A Portable Multimedia Terminal", *IEEE Communications Magazine*, Vol. 30, No. 12.

[31] Stemm, M. and Katz, R.H. (1996) Measuring and Reducing Energy Consumption of Network Interfaces in Hand-Held Devices. Proc. 3rd International Workshop on Mobile Multimedia Communications (*MoMuC-3*), Princeton, NJ, 25-27 September.

[32] Stemm, M. and Katz, R. H. (1997) Measuring and reducing energy consumption of network interfaces in handheld devices. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Science*, Special Issue on Mobile Computing, Vol. 80, No. 8, 1125-1131.

[33] Stevens, W.R. (1994) *TCP/IP Illustrated*, Vol. 1, *The Protocols* Addison-Wesley, Reading, Massachusetts.

[34] Stevens, W.R. (1996), *TCP/IP Illustrated*, Vol 3, *TCP for Transactions, HTTP, NNTP, and the UNIX Domain Protocols*, Addison-Wesley, Reading, Massachusetts.

[35] Request for Comments 2001, Stevens, W.R. (1997) TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms.

[36] Weinmiller, J., Schlager, M., Festag, A. and Wolisz, A. (1997) Performance Study of Access Control in Wireless LANs-IEEE 802.11 DFWMAC and ETSI RES 10 HIPERLAN. *ACM/Baltzer Mobile Networks and Applications*, Vol 2, 55-67.

[37] Weiser, M., Welch, B., Demers, A. and Shenker, S. (1994) Scheduling for Reducing CPU Energy. USENIX Association, *First Symposium on Operating System Design and Implementation*, Monterey, CA, pp 13-23.

[38] Woesner, H., Ebert, J. P., Schlaeger, M. and Wolisz, A. (1998) Power-Saving Mechanisms in Emerging Standards for Wireless LAN's: The MAC-Level Perspective. *IEEE Personal Communications* (Special Edition on Power Saving), vol. 5, no. 3, pp. 40-48.

[39] Zorzi, M. and Rao, R. R. (1996) Energy Constrained Error Control for Wireless Channels. *Proceeding of IEEE GLOBECOM '96*, London, UK, November, pp.1411-1416.

[40] Zorzi, M., Rao, R. R. and Milstein, L. B. (1997) ARQ Error Control on Fading Mobile Radio Channels. *IEEE Transactions. Vehicular. Technol*., 46, 445-455.