Power Saving Policies for wireless access to TCP/IP Networks

Giuseppe Anastasi, Marco Conti, Willy Lapenna

Abstract **In this paper we examine the relationship between transport-layer protocols and mobile-host processing capacity. Firstly, we measure the power consumption of a mobile host that uses a TCP connection to send/receive data from Internet. These measurements indicate that by adopting the TCP protocol the power consumption is negatively affected by the congestion that occurs into the fixed-network. To solve this problem we propose to exploit the *Indirect TCP model* for achieving both the TCP reliability and an optimal power consumption level. Experimental results indicate that the Indirect TCP model 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 indirect model, is significantly lower than the power consumption measured when we use the legacy TCP approach.**

I. INTRODUCTION

A mobile computer operates on a finite battery power that represents one of the greatest limitation to the utility of portable computers [2,6]. Projections on progress in battery technology show that only small improvements in the battery capacity are expected in next future [16]. If the battery capacity cannot be improved, it is vital that power utilization is managed efficiently by identifying way 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. 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 ([12-15] and [21-25]). Power-saving policies at the operating system level include the study of strategies for the CPU scheduling [7, 20] and for the hard-disk management [3]. At the application-level, among the other, have been investigated policies that exploit the application semantic (e.g. 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) or profit of tasks remote execution (user’s jobs are transferred from a mobile host to a fixed host to reduce power consumption by the mobile-host CPU) [5, 6, 8, 11].

In this paper we investigate the relationship between power saving and the data transfer from/to mobile hosts. Data transfer contributes to the power management problem as it uses significant power when data is sent and received.

Communication hardware and software, involved in data transfer, can be partitioned in two classes: subnetwork protocols and internetwork protocols. Subnetwork 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. Internetwork protocols exploit the subnetwork services to provide a data transfer between all the computers that have a network access.

The impact of network technologies on power consumption has been investigated in depth in [17]. The power saving features of the IEEE 802.11 wireless LANs have been analyzed in [10, 19]. 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 [26, 27]. Power-saving strategies at the subnetwork level are strongly dependent on the network technology and no general solution can be developed. On the other hand, at the internetwork level the TCP/IP protocol stack is a de-facto standard, 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. Specifically, we focus on transport layer protocols and we investigate the impact of the TCP protocol when a mobile host access a fixed host, see Figure 1. In our work the mobile host is connected to the Internet by an IEEE 802.11 wireless LAN.

To evaluate the impact of the TCP protocol on the consumption of the battery power we first define a

---

Marco Conti and Willy Lapenna are with the Italian National Research Council (CNR), Istituto CNUCE Via Alfieri 1, 56010 Ghezzano Pisa, Italy. e-mail: Marco.Conti@cnuce.cnr.it, Willy.Lapenna@guest.cnuce.cnr.it.

Giuseppe Anastasi is with the University of Pisa, Department of Information Engineering, Via Diotisalvi 2, 56126 Pisa, Italy, e-mail: anastasi@iet.unipi.it.

metric for measuring the battery power consumption of a protocol, then we apply this metric to estimate the paper consumption of the TCP protocol. From this analysis we identify that the congestion inside the Internet, by maintaining, for most of the time, the IEEE 802.11 network interface in the idle status is one of the main element in causing an inefficient utilization of the battery power. For this reason we considered the Indirect TCP model [1] as an interesting direction for power saving. Specifically, according to the Indirect model, we defined a new 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 and the fixed hosts is subdivided (at the border between the wireless and the wired network) in two transport connections: one between the mobile host and the (fixed-network) access point and one from the access point and 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.

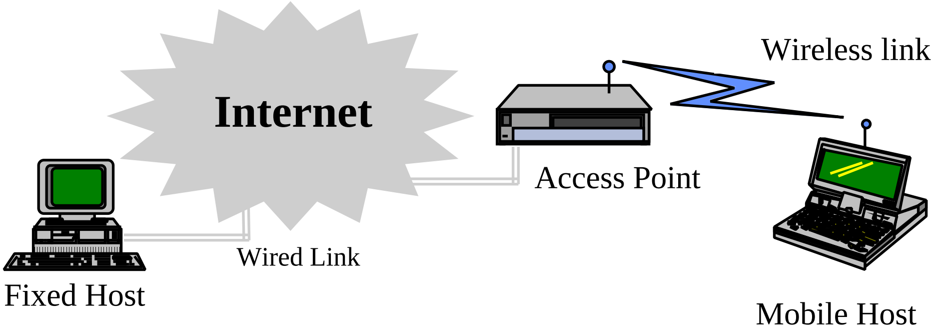

Figure 1: Scenario

Experimental results show that, by adopting our PSNA architecture to integrate a portable computer into Internet, 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 status.

The idea to use the indirect TCP model for power saving is not new [28]. Our work provides original contributions as it extends previous results and provides an implementation of the Indirect TCP architecture to evaluate the performance of the indirect approach. In [28] a similar scenario was considered, i.e., a mobile host that accesses Internet through a wireless access. However, in that work the measurements were performed by connecting the portable computer to Internet through a serial (voice grade) line. More precisely, a 3.52 KBps modem was adopted thus obtaining an access line that represents a slow access line to 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. On the other hand, in our work we use a real wireless access by interconnecting the portable computer to the fixed network through an IEEE 802.11 WLAN.

Another original contribution of our work is the definition, implementation and (experimental) evaluation of the Power-Saving Network Architecture.

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 impact of the network interface on the battery consumption. 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 ($number\_of\_bytes$) transmitted or received by the network interface divided by the amount of the battery power ($battery\_consumption$) drained by the network interface to manage this amount of data:

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

The target of a power saving strategy is to maximize I_{Power_saving}. By dividing the numerator and the denominator by the time unit *t*, 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. In the following we first investigate separately the two quantities, then we analyze the I_{Power_saving} index that captures the interdependencies between these two quantities.

In our experiments we utilize a portable computer with Linux operating system connected via an IEEE 802.11 WLAN to an access point connected 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 utilizes group of Lithium batteries (BTP-31 Sony) which provides a 10.8 Volt power and has a capacity of 4050 mAh.

II.1 NETWORK INTERFACE POWER CONSUMPTION

The first set of experiments we performed studies the impact of network interface on the battery lifetime. 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 2, when the network interface is ON the battery lifetime reduces of about 22 minutes corresponding to about 11.5% of the total battery lifetime. 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.

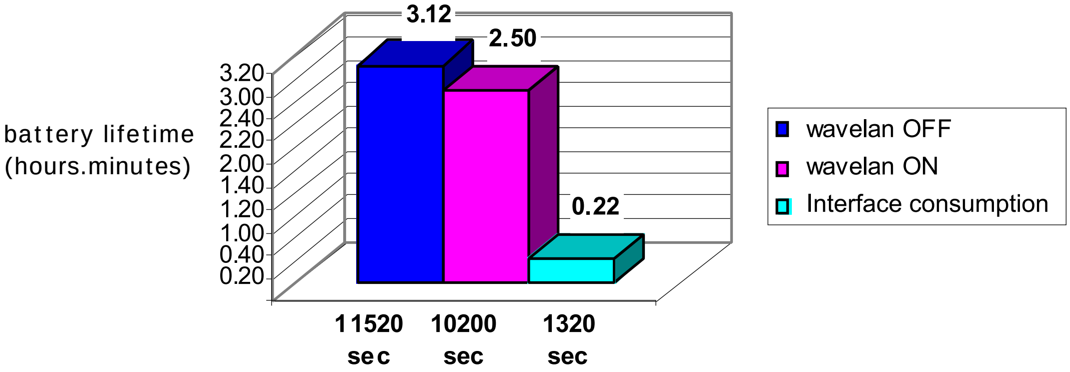

Figure 2: Battery power consumption when the portable computer is in the *stand_by* state

II.2 THROUGHPUT ANALYSIS

The second index we investigate is the throughput of the network interface when we use a TCP connection to transfer the data to/from the fixed network. We run an extensive set of experiments to determine the impact of the Internet TCP protocol on the throughput. In our experiments, the mobile terminal located in Pisa accesses the Internet via a 2Mbps IEEE 802.11 WLAN. The fixed host was located at the Curtin University of Technology (Perth, Western Australia) in the School of Computing. To verify the protocols' behavior under the various network conditions we perform our experiments during a wide time interval. Each type of experiment was performed several times during each selected time interval, and the statistics that we present are averaged on all the performed experiments.

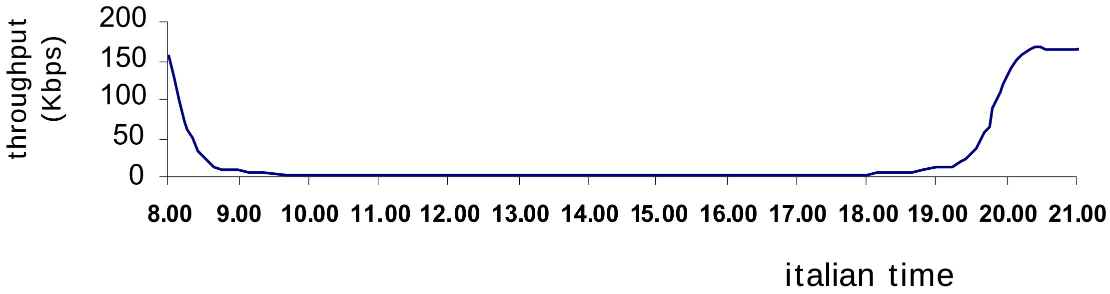

Figure 3: data transfer from the fixed host to the mobile host

Figure 3 presents the estimated throughput when data is transferred from the fixed host towards the mobile host. To better highlight the throughput during the Italian business hours Figure 4 must be analyzed. For these results we can observe that the throughput is always very far from the access line speed, and during business hours the access link (and hence the network interface) is usually in the idle state.

We performed a second set of experiments to analyze the throughput in the reverse condition.

Specifically, in these experiments we evaluated the data transfer is from the mobile to the fixed host. As in the previous experiments, we observe that in all situations the throughput is very low with respect to access 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 during weekends the throughput is constant and it is generally higher than during working days. It must also be remarked that the throughput of a data transfer from the mobile to the fixed host (Figure 5) is much higher than that throughput experienced by the mobile host to download the data from the fixed host. To understand this difference, we have studied, with the `traceroute` command, the traffic path between Pisa and Perth (and vice versa) and then we have analyzed the traffic statistics collected by Italian research network for this path. The difference is caused by asymmetric congestion conditions in the international links.

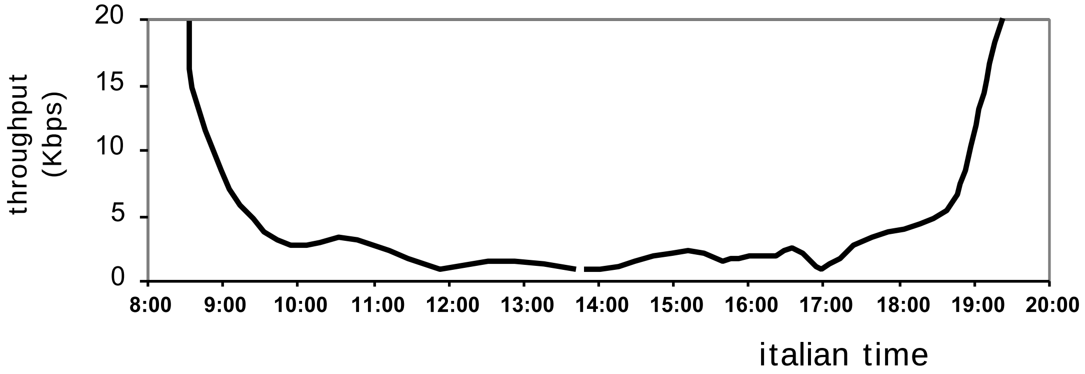

Figure 4: TCP Throughput during peak working hours

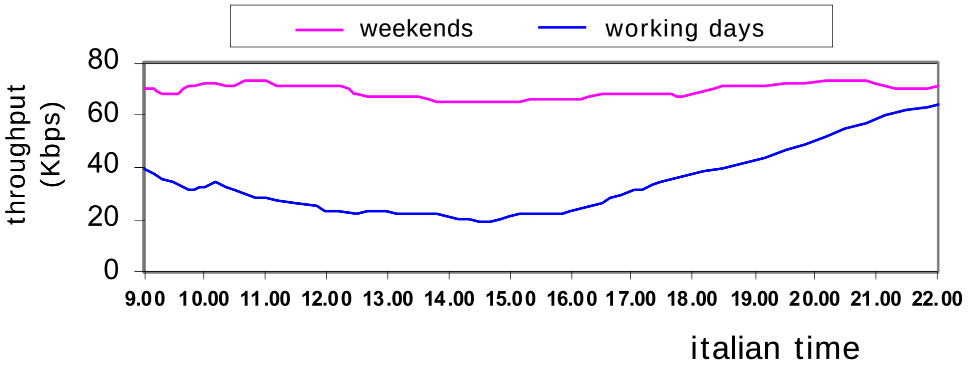

Figure 5: data transfer from the mobile host to the host fixed

II.3 EVALUATION OF I_{Power_saving}

We run 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. To this end we implemented a simple application program that generates data with a speed that always guarantees a not empty buffer at the sender side.

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 run an experiment until the battery dies. To reduce the inaccuracies induced by our measurement approach we run each experiment several times.

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 does not show meaningful differences (from the battery characteristics' standpoint) between the first and last experiments we performed.

As before, the mobile host is located in Pisa and the fixed host is located at the Curtin University of technology (Perth, Western Australia). We performed two sets of experiments. In the type A experiments the mobile host is the source of 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 am (Italian time).

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.2), during type A experiments a larger quantity of data is managed on the mobile host and this implies an increase of the power consumption in all the components of the mobile host (disk, CPU, network interface, etc.)

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

Type of experiment	Battery lifetime (seconds)	Delivered data (Mbytes)	Throughput (Kbps)
A	10005	24.85	19.87
B	10153	2.46	1.82

Some interesting observations on the relationship between the battery lifetime and the status of the network interface can be derived by comparing Table 1 results with those presented in Section II.1. By comparing these results 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 is 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 of about 10200 seconds, and 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 we can approximate that 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 Table 1 results, we also derive the power saving index for type A and type B experiments. Specifically, by applying Equation (1) we have

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

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

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. 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 Internet produces an under utilization of the access link, and hence an increase of the time the network interface remains in the idle state. Since the power consumption of the network interface linearly depends 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 link from the fixed host (in Perth) to the mobile host located in Pisa. This is the cause because in type B experiments we experienced a lower value for the I_{Power_saving} index with respect to type A experiments.

The major outcome deriving from the analysis of 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 access-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 [1].

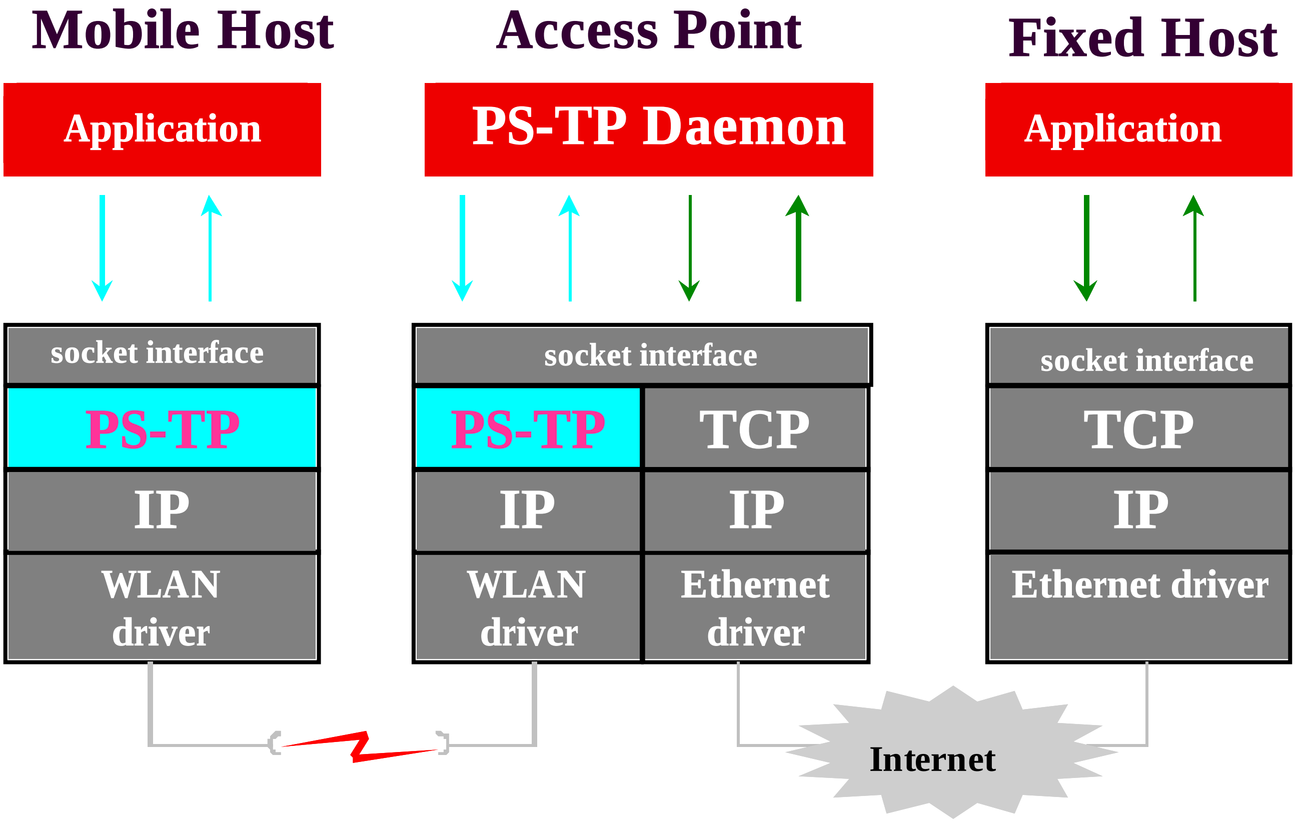

Figure 6: Power-Saving Network Architecture

By exploiting the basic idea of the Indirect TCP approach, we have designed a new network architecture for integrating a mobile host in Internet by maintaining (from the application standpoint) the TCP semantics 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 [18] 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 TCP protocol can be adopted for providing a reliable transport connection between the mobile host and the base station. 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 TCP 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 will be clarified below, to transfer in a power saving way a given amount of data between the mobile host and the fixed host we adopt a paradigm in which the data are transferred between the mobile host and the access point during time intervals that are separated by periods during which no network connection exists between the mobile host and the access point. 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.

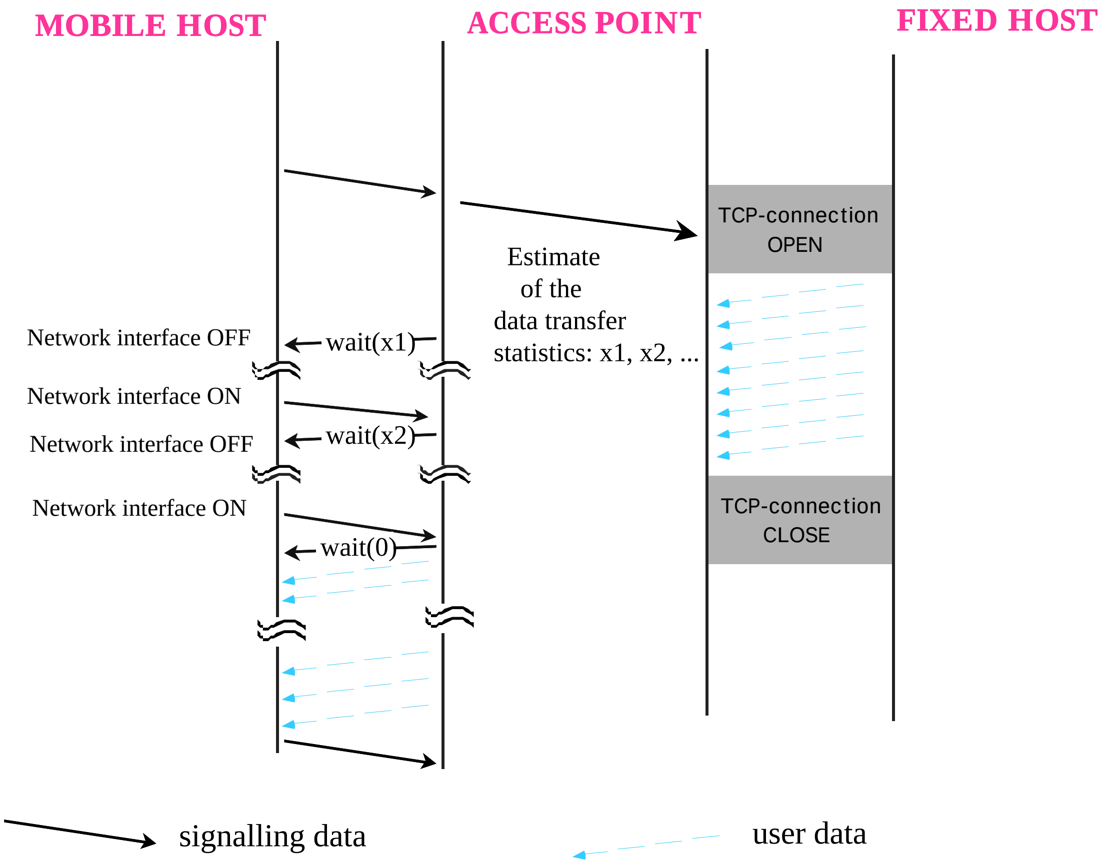

Figure 7. Read-access operations.

Taking into account that the round trip time between the mobile host and the access point is almost negligible, we have included in the PS-TP a simple ARQ mechanism based on the Stop and Wait paradigm [29] to guarantee a reliable data transfer.

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. In the former case the mobile host downloads some information from the fixed host, e.g. a web server, while in the latter case it sends data to a fixed host.

Read Access. The operations performed by the mobile host are performed in (at least) two steps. In the first step the mobile host delivers (by exploiting the services of the daemon operating on the access point) to the end system the receive (download) request. 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 to the remote host to deliver its data (through 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. When the mobile host sends the request (to the access point) to open a connection with the remote host it does not immediately turn its network interface in the OFF state. Rather it remains in the receive state waiting for some information on the estimated time to complete the data transfer. Specifically, as soon as the access point has computed a meaningful estimate of the *Transfer_delay* returns this information to the mobile host. In Figure 7 this is represented by the message *wait*(*x1*), where *x1* is the expected value of the *Transfer_delay*. By exploiting this information the mobile host decides the amount of time, for example *x1*, its network interface should remain OFF. As the statistics related to the *Transfer_delay* 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 data transfer completion time (see wait(x2) in Figure 7). Finally, after some polls the mobile host turns its interface ON when all data are ready in the buffer of the access point and can be delivered in a continuous way on the access link to the mobile host.

The presentation given above provide 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 the data are already in the buffer of the PS-TP daemon. This choice maximizes the utilization of the access link during the data transfer phase, however it 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.

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 8.

The mobile host sends a signalling packet to the access point in which it requests that a TCP connection is opened between the access point and the remote host. As soon as the mobile host receives the reply from the access point, it starts sending (on the access link) all its data packets to the access point. When the mobile host has no more data to send the state of its network interface is set to OFF. Meanwhile the access point transmits on a TCP connection the mobile-host data to the remote host. In the figure we have assumed that the network interface of the mobile host remains OFF for an interval of time that it is enough to complete the remote data transfer. Hence, when the mobile host makes its network interface active and polls the access point, it immediately receives a message from the access point indicating that all data have been correctly delivered to the remote host. At this point the data transfer is completed and the mobile host puts its network interface in the OFF state. It may happen that the mobile host polls the access point too early, while the remote data transfer (i.e. from the access point to the fixed host) is still ongoing. In this case the access point returns to the mobile host an indication of the time at which it expects that the data transfer will be completed. Among the tasks of the access point is to compute at run-time the statistics of the time it takes to complete the remote data transfer. This information is used to help the mobile host to determine the length of the time interval during which its network interface should remain in the OFF state.

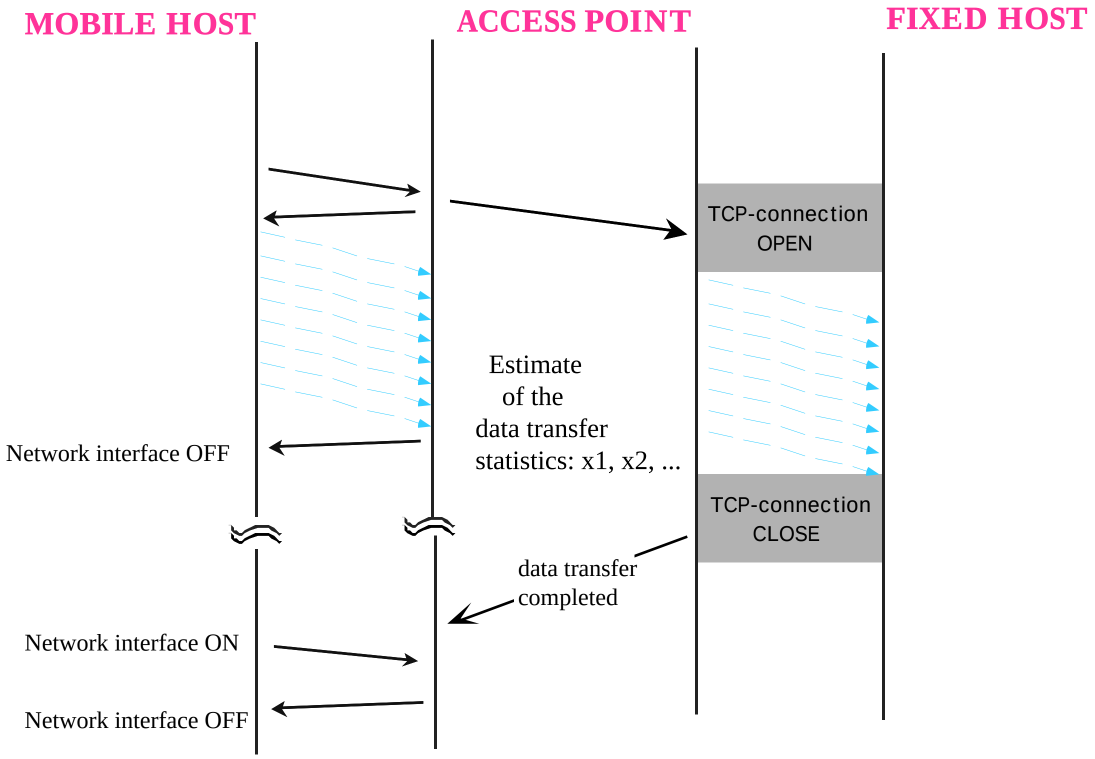

Figure 8: Send-access operations

IV. EVALUATION OF THE *PSNA* ARCHITECTURE

To evaluate the effectiveness of the Power-Saving Network Architecture we have developed a prototype implementation of its components. Specifically, we have 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 [30].

By exploiting the PSNA implementation we have evaluated the I_{Power_saving} index to estimate the performance from the power-saving standpoint of the indirect TCP approach. Specifically, we have 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, in the type A (B) experiments we evaluate the power consumption of PSNA architecture when the mobile host sends (or receives) the same amount of data that the mobile host can send (receive) with the legacy TCP approach by exploiting all the battery capacity (see Section II.3). The results are summarized in Table 2.

Table 2: Comparison of the legacy and indirect TCP approaches

Type	Architecture	Battery lifetime (sec.)	Battery consumption (sec)	Delivered data (Mbytes)	Throughput (Kbps)	length of the data transfer (sec)
A	legacy TCP	10005	1515	24.85	19.87	10005
A	PS_TP	11501	19	24.85	1525.48	130
B	legacy TCP	10153	1367	2.46	1.82	10153
B	PS_TP	11512	8	2.46	1525.48	13

Furthermore, by remembering that the battery lifetime when the network interface is OFF corresponds to 11520 seconds we have that, by adopting the PSNA architecture, the $battery\_consumption$ for type A and type B is 19 and 8 seconds, respectively. Hence, by applying Equation (1) we can estimate the power saving index

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

$$I_{Power\_saving}(\text{type B, PSNA}) = \frac{2519}{8} = 314.8 \text{ Kbytes/seconds\_of\_battery\_lifetim} \tag{3}$$

In both cases we have a marked improvement with respect to the corresponding experiments performed with the legacy TCP approach. Specifically, 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 experiments A and B, respectively. In addition we have that the battery consumption due to the network interface reduces more than 98%.

V. 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. It is also worth noting that the indirect approach for integrating a mobile host in Internet, beyond power saving, present additional advantages such as:

i) the maximization of the throughput on the wireless channel thus minimizing the time the mobile host needs to be connected to the fixed network;
ii) the possibility to use an ad-hoc transport protocol on the wireless link;
iii) the 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 channel, negatively affect the TCP connection in the fixed network.

Acknowledgements. The authors wish to express their gratitude to Dr. Mohan Kumar (Curtin University of Technology - School of Computing) for his support to the experimental study.

REFERENCES

[1] Ajay Bakre, B.R.Badrinath, “Implementation and Performance Evaluation of Indirect TCP”, IEEE Transactions on Computers, Vol. 46, No. 3, March 1997.
[2] G. H. Forman, J. Zahorjan, "The Challenges of Mobile Computing", Tecnical Report, University of Wachington, Mar. 1994.
[3] 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, Nov. 1996
[4] IEEE standard for Wireless LAN- Medium Access Control and Physical Layer Specification, P802.11, November 1997.
[5] T. Imielinski, S. Vishwanathan, B.R. Badrinath "Power Efficient Filtering of Data on air", Proc. Of the EDBT, Cambridge, England, Mar. 1994.
[6] T. Imielinscki B.R. Badrinath "Wireless Computing" Communication of the ACM, Vol. 37, No. 10, Oct. 1994
[7] J.R. Lorch, A.J. Smith, “Scheduling Techniques fo Reducing Processor Energy Use in MacOS”, Wireless Networks, 1997, pp.311-324.
[8] 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.

[9] W.R.Stevens, “TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms”, Request for Comments: 2001.

[10] C. Rohl, H. Woesner, A. Wolisz, "A Short Look on Power Saving Mechanisms in the wireless LAN Standard Draft IEEE 802.11", University Berlin, www-site: http://www-tkn.ee.tu-berlin.de/bibl/ours/powersaveletter.ps.gz.

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

[12] M Rulnick, N. Bambos "Performance Evaluation of Power-managed Mobile Comunication Devices", Proceedings of IEEE International Communication Conference (ICC), Dallas, Texas, 1996.

[13] M.Rulnick, N. Bambos "Mobile Power Management for Maximum Battery Life in Wireless Communication Networks", Proceedings of IEEE Infocom, 1996.

[14] M.Rulnick, N. Bambos "Mobile Power Management for Wireless Communication Networks", Wireless Networks, Vol. 3, No. 1, Mar. 1996.

[15] M.Rulnick, N. Bambos "Power Control and Time Division: TDMA versus CDMA Question", Proceedings of Infocom '97 The Joint Conference of the IEEE Communications and Computer Societies, Kobe, Japan, Mar. 1997.

[16] S. Sheng, A. Chandrakasan, R.W. Brodersen, “A Portable Multimedia Terminal”, IEEE Communications Magazine, Decembre 1992.

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

[18] W.R.Stevens, *TCP/IP Illustrated*, Vol 1, Addison Wesley, 1994.

[19] J. Weinmiller, M. Schlager, A. Festag, A. Wolisz, "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 (1997) pp. 55-67.

[20] Mark Weiser, Brent Welch, Alan Demers Scott Shenker. "Scheduling for Reducing CPU Energy", USENIX Association, First Symposium on Operating System Design and Implementation Monterey, CA, Nov. 1994.

[21] M. Zorzi, R. R. Rao "Energy Management in Wireless Communication", 6th WINLAB Workshop on Third Generation Wireless Information Networks, Mar. 1997.

[22] M. Zorzi, R. R.Rao "The Effect of Correlated Errors on the Performance of TCP", Conference on Information Science and Systems (CISS), Baltimore, MD, 19-21 Mar. 1997.

[23] M. Zorzi, R. R. Rao "ARQ Error Control on Fading Mobile Radio Channels", accapted for pubblication in IEEE Trans. Veh. Tech., Also in Proc. IEEE ICUPC'95, pp. 211 - 215, nov 1995.

[24] M. Zorzi, R. R. Rao "ARQ Error Control for Delay-Constrained Communications on Short-Range Burst-Error Channels", VTC'97, Phoenix, AZ, 5-7 May. 1997

[25] M. Zorzi, R. R. Rao "Energy Constrained Error Control for Wireless Channels", Proceeding IEEE GLOBECOM '96, pp.1411 - 1416, 1996.

[26] L. Bononi, M. Conti, L. Donatiello, “A Distributed Mechanism for Power Saving in IEEE 802.11 Wireless LANs” *ACM/Baltzer Mobile Networks and Applications Journal*, (to appear).

[27] L. Bononi, M. Conti, L. Donatiello, “A Distributed Contention Control Mechanism for Power Saving in random-access Ad-Hoc Wireless Local Area Networks”, Proc. *MoMuC'99* Sixth International Workshop on Mobile Multimedia Communication, San Diego, CA, November 15-18, 1999.

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

[29] D. Bertsekas, R. Gallager, "Data Networks" *Prentice Hall*, 1992.

[30] W. Lapenna, "Politiche di Power Saving per l’accesso a server Web da computer mobili" *Laurea Thesis*, Pisa ,May 2000 (in Italian).