A Power Saving Architecture for Web Access from Mobile Computers

Giuseppe Anastasi[1], Marco Conti[2], Enrico Gregori[2], and Andrea Passarella[1]

[1]University of Pisa, Dept. of Information Engineering
Via Diotisalvi 2 - 56126 Pisa, Italy
{g.anastasi, a.passarella}@iet.unipi.it,

[2]CNR - CNUCE Institute
Via G. Moruzzi, 1 56124 PISA—Italy
{marco.conti, enrico.gregori}@cnuce.cnr.it,

Abstract. This work proposes new power-saving strategies for mobile access to the Web. User mobility is a key factor in the evolution of Web services. Unfortunately, the legacy approach for Web access is very inefficient when applied to mobile users. One of the critical issues is the inefficient usage of energetic resources when adopting the legacy TCP/IP architecture for Web access from mobile devices. In this paper we address this problem by proposing a new architecture, namely *PS-Web*, which works at the transport layer and exploits some knowledge about the application behavior. PS-Web is transparent with respect to the application and independent from the sub-network technology. We implemented a prototype of PS-Web. Experimental results provided by this prototype have shown that a relevant energy savings (about 70% on average) can be achieved with respect to the legacy TCP/IP approach. Furthermore, power saving is obtained without a significant degradation in the QoS perceived by the users. Specifically, PS-Web introduces almost neglible additional delays (with respect to the legacy approach) in the downloading of a Web page.

1   Introduction

The *Mobile Internet* is emerging as one of the most promising fields in the area of computer networking. The Internet explosion in the last years has demonstrated that accessing information of some interest in the same moment they are needed is a valuable opportunity. In this context, the concept of *mobility* adds a new dimension: information is carried directly to the user at *any time* and *any place*. However, integrating mobile computers in the legacy Internet scenario is still a challenging problem for a number of reasons. Internet protocols and applications were designed with the implicit assumption that links are wired and hosts do not change their position in time. Mobile computers have less computation and storage resources with respect to desktop computers. Furthermore, they usually connects through wireless links that are characterized by lower bandwidth and greater bit error rate with respect to wired links. Finally, mobile computers have a limited energy autonomy since they are battery-fed. Hence, the use of legacy solutions causes a non-optimal usage of the

system that heavily limits the growth of the mobile Internet. In particular, the scarcity of energy resources is a very limiting factor [7, 11, 15].

In principle, energy-related problems could be solved by either increasing the battery capacity or reducing the energy consumption. Projections on progresses in battery technology show that only small improvements in the battery capacity are expected in next future [20]. Hence, it is vital to manage energy efficiently.

Strategies for energy saving have been investigated at several layers including the physical-layer transmissions, the operating system, the network protocols and the application level [8,10,14,16,18,19,22,23,24]. In this paper we focus on strategies aiming at reducing the energy consumed due to networking. Although this component only accounts for about 10% of the total consumption in current notebooks [13], it increases to approximately 50% in hand-held computers (palm top, PDA, etc.) [12]. Hence, it becomes very important to design a power efficient networking subsystem.

Based on experimental measurements, [21] and [13] conclude that the only way to actually reduce the networking component of the energy consumption consists in switching the wireless network interface off during inactivity periods. Works in [12] and [13] show that the legacy TCP/IP network architecture may have a negative impact on the energy consumption and propose to exploit the *Indirect-TCP* approach [2]. A further improvement could be achieved by exploiting some knowledge about the application behavior. According to this evidence, power management should be controlled at higher layers, potentially even at the application layer [12,13,21].

In this paper we propose new energy-saving strategies implemented in software network protocols. Specifically, we operate at the transport and application layers and use an application-dependent approach in the sense that envisaged strategies exploit some characteristics of the application. However, the proposed solutions do not require any modification to the application itself.

 We focus on Web services but our design is *modular* and could be easily adapted to any other network application. Web choice is justified by several reasons. First, it is today the most widely used Internet application and is seriously candidate to become the killer application for the mobile Internet too. Furthermore, Web users are typically sensitive to delays. Hence, achieving a significant reduction in the energy consumption while maintaining an acceptable Quality of Service (QoS) level is a very challenging task.

We defined a new architecture, throughout referred to as *PS-Web* (Power Saving Web), which allows mobile users to exploit Internet Web services with a QoS similar to the one provided by the legacy network architecture based on the TCP/IP protocol stack, but with a significant reduction in the energy consumption. The PS-Web architecture is based on the Indirect-TCP model [2], i.e., the TCP connection between the browser and the Web server is split into two connections: one between the browser (on the mobile computer) and an Access Point (at the border between the wireless and wired networks), and the other one between the Access Point and the Web server. Unlike the solution proposed in [12], however, a simplified transport protocol is used between the mobile host and the Access Point. Furthermore, inactivity timeouts and sleeping times used to switch off and on the network interface are not fixed – as in [21] and [12] – but are adjusted dynamically based both on information about the past history collected on-line and on statistical models of Web traffic pattern available in the literature. The Access Point works as a Power Saving Proxy Web, i.e., a Proxy Web with power saving support for mobile users. Specifically, it implements a pre-fetching mechanism.

Experimental results obtained on a prototype implementation of the PS-Web architecture based on a IEEE 802.11 WLAN [9] have shown that the PS-Web allows to save 70% of power with respect to the legacy TCP-based architecture. Furthermore, this is not obtained at the cost of a significant degradation in the Quality of Service (QoS). The additional delay introduced by the PS-Web in transferring a Web page is always lower than 1.5 s.

The paper is organized as follows. Section 2 sketches the characteristics of Web traffic. Section 3 is devoted to the definition of the PS-Web architecture. Section 4 reports some experimental results obtained by using the prototype implementation. Finally, Section 5 concludes the paper.

2   Web Traffic Characterization

The power saving strategies implemented in our system are based on the characteristics of the application. Hence, as a preliminary step, it is necessary to understand the traffic profile generated by Web browsing. Many papers in literature provides mathematical characterizations of Web traffic [1,3,4,5,6].

Fig. 1 shows the typical ON/OFF profile of the network traffic generated by an individual Web user [5]. As is well known a Web page consists of a main file and zero or more embedded files (e.g., figures). All files composing a Web page are transferred during the *Active Time* interval while in the *Inactive Time* (or *Think Time*) interval the user reads the content of the downloaded Web page. Within an Active Time, *ON Times* correspond to actual file transfers while during *Active OFF Times* the browser parses a piece of the main file and sends the request for the next embedded file.
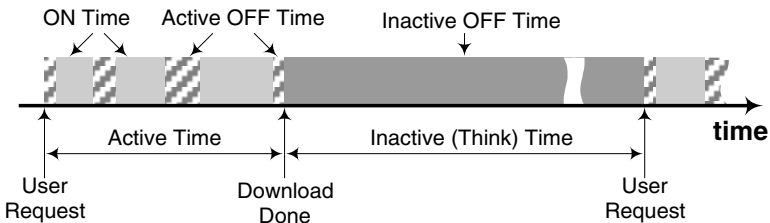
Fig. 1. Typical phases of a page transfer as observed by the user.

Fig. 1 suggests us the following hints. During *Inactive OFF Times* the network interface can be switched off. On the other hand, *Active OFF Times* are often too short (less than 1s) to turn the interface off with some profit (recall that the interface has a transient in going on, during which it consumes energy but is not available for data transfer). However, one could manage the transfer of Web page in such a way that all files in a page are transferred on the wireless link in a single burst. By following this approach different Active OFF Times are concentrated in an single large *OFF Time*, and this gives more chance to turn the interface off, actually saving some energy.

It may be worthwhile to point out that the ON-OFF behavior of Web traffic generated by individual users is related to the self-similarity that is a *structural*

property of (aggregate) Web traffic [5]. This means that the ON-OFF behavior is independent of the specific access pattern followed by the user and the type of files available in the Web server.

3   PS-Web Architecture and Protocols

A typical mobile scenario is depicted in Fig. 2. The communication between a mobile host and a machine connected to the Internet (*Fixed Host*) is made possible by a third entity (*Access Point*), which provides Internet connectivity to the mobile host through a wireless link.

Although very simple and costless, a legacy TCP-based solution is prone to various drawbacks that heavily impacts the energy consumption at the mobile host.

1. The TCP congestion control wrongly interprets losses in the wireless link as congestion signals. Hence, the overall throughput is usually low and the wireless network interface at the mobile host remains idle for most of the time.
2. Congestions in the wired networks limits the throughput in the wireless link as well. The overall effect is the same as in 1.
3. The ON/OFF behavior of Web traffic forces the wireless network interface to be inactive for long time intervals.
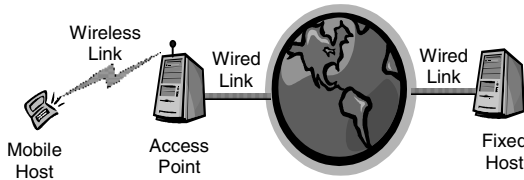

Fig. 2. A typical mobile environment

To overcome these problems we exploited a network architecture based on the *Indirect-TCP* model [2]. The transport connection between the client at the mobile host and the Web server is split into two parts: the first one between the mobile host and the Access Point and the second one between the Access Point and the Web server. At the Access Point an agent (*I-TCP daemon*) relays data from one connection to another. A *Simplified Transport Protocol* (STP in Fig. 3), instead of the legacy TCP protocol, is used to transfer data on the wireless link.

The *Indirect-TCP* model eliminates problems related to point 1 above. However, bottlenecks in the Internet might still cause a low transfer rate in the wireless link. To overcome this second problem, we use pre-fetching of Web pages at the Access Point. Embedded files – if any – are requested to the remote server even without an explicit request from the user and will be transferred to the mobile host, on request from the mobile host itself. This approach allows to transfer embedded files on the wireless link at full speed, irrespective of the throughput available in the wired connection. At the mobile host side, pre-fetching is managed by the PSP (*Power Saving Protocol*) module. At the Access Point side, it is handled by the *PS-Daemon* (see Fig. 3). This is the *I-TCP Daemon* enriched with pre-fetching and power management mechanisms.
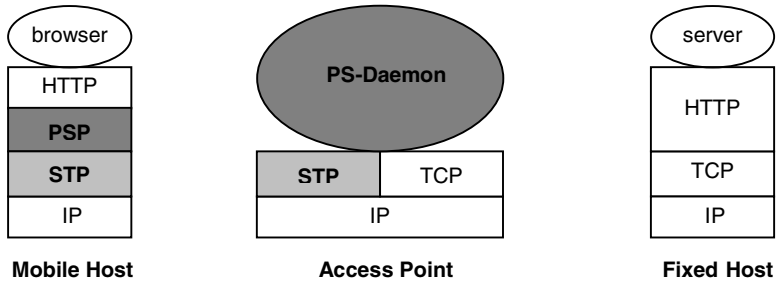
Fig. 3. Overall PS-Web network architecture; evidence on added components

Finally, with reference to point 3 above, it can be observed that, by grouping the transfer of the embedded files on the wireless link in a single burst, Active OFF Times can be compacted in an unique long OFF Time. This reduces significantly the time during which the network interface must be on. At the mobile host, the PSP layer is responsible for identifying the beginning of the Inactive OFF Times, and turning the network interface off until a new request from the browser arrives.

3.1   Power Saving Protocols

The PS-Daemon can be seen as made up of two components. The upper level component interacts with the HTTP modules at the mobile host and fixed server, and implements the same functionalities of a Proxy Web. The lower level component implements power management by interacting with the PSP module at the mobile host via the *Power Saving Protocol*. Therefore, the PS-Daemon can be regarded as a Proxy-Web with power saving support.

Since we are interested in power management, in the following we shall focus on the PSP protocol. Fig. 4 and Fig. 5 show the actions performed at the mobile host and the Access Point, respectively.

Upon receiving the main file from the remote server, the PS-Daemon forwards it to the mobile host, together with an estimate of the residual transfer time (see below), i.e., the time needed to fetch the embedded files from the server (lines 2-3 and 18-22). Upon receiving such an estimate the mobile host turns the network interface off for the corresponding time interval (lines 4-7). Possible requests for embedded files generate by the browser in the meanwhile will be blocked by the PSP layer until the network interface is turned on again (lines 8-12).

When the time interval has elapsed, the PSP module at the mobile host turns the network interface on and sends requests for embedded files, if any, to the PS-Daemon (lines 13-15). The PS-Daemon has already fetched these files from the server and can thus send them back to the browser (lines 23-25).

When the Web page is completely available at the mobile host the PSP module turns the network interface off (lines 16-17) until a new request arrives from the user.

```
 1   OnNewPageRequested(httpRequest)
 2     resumeInterface()
 3     send httpRequest to Access Point
 4     receive (mainFile, estimate) from access point
 5     if(estimate ≥ MIN_USEFUL_TIME)
 6       suspendInterface()
 7     setTimer(estimate)

 8   OnRequestFromBrowser(httpRequest)
 9     if(interface is ON)
10       send httpRequest to Access Point
11     else
12       insert httpRequest into pendingRequests

13   OnTimerExpired()
14     foreach httpRequest in pendingRequests
15       send httpRequest to Access Point

16   OnPageTransferFinished()
17     suspendInterface()
```

Fig. 4. PSP protocol: actions performed at the mobile host.

```
18   OnNewPageRequested(httpRequest)
19     Send httpRequest to server
20     receive mainFile form server
21     estimate = evaluate_time(mainFile)
22     Send (mainFile, estimate) to mobile host

23   OnRequestForEmbedded(httpRequest)
24     File = identifyFile(httpRequest)
25     Send file to mobile host
```

Fig. 5. PSP protocol: actions performed at the Access Point.

The architecture depicted in Fig. 3 is completely transparent to the application and the HTTP protocol, respectively. Like any other Web proxy, the PS-Daemon do not introduce any modification either at the client or server side of the application. In particular, the PSP module at the mobile host presents a socket-like interface to the application layer.

The PS-Web architecture relies upon estimates of the file transfer times. These estimates are performed by the PS-Daemon at the Access Point and communicated to the mobile host (see [17] for details). As it clearly appears from the protocol description, the accuracy of the estimates is a key factor to achieve a significant power saving at the mobile host. The above architecture can be easily modified to include optimizations like handling of inaccuracies in the estimates supplied by the PS-Daemon (line 22, 4), isolation of the application-dependent functionalities to achieve an higher modularity and reusability, and so on. Details on such optimizations can be found in [17].

4   Experimental Results

The objective of our system is twofold. First, it should achieve a good power saving with respect to the legacy approach. At the same time, the reduction in the power consumption should not occur at the cost of an unacceptable degradation in the QoS. To evaluate the performance of our system we considered a *Power Saving Index* (I_ps) and a *Page Delay Index* (I_pd). The Power Saving Index is defined as

$$I\_ps = \frac{network\ interface\ consumption\ in\ PS\ \text{-}\ Web}{network\ interface\ consumption\ in\ TCP\ architecture} \qquad (1)$$

I_ps gives an immediate indication of how much energy can be saved by using the PS-Web approach instead of the legacy solution. The Page Delay Index measures the additional delay introduced by the PS-Web to transfer a Web page with respect to the legacy architecture, i.e.,

$$I\_pd = (page\ transfer\ time\ in\ PS\text{-}Web) - (page\ transfer\ time\ in\ TCP\ architecture) \quad (2)$$

To assess our architecture we performed an extensive measurements campaign. In our measurements, to simulate the application level, we used SURGE, a web traffic generator designed by Barford and Crovella [3]. SURGE can simulate an individual user by generating ON/OFF traffic with the same statistical properties of real Web traffic (see Section 2). So, it allows to evaluate our system in realistic conditions.

To significantly simulate a Web session, each experiment stopped after 150 files have been transferred from the web server to the client[1], when the whole page "in flight" arrives at the client. In each experiment, two different instances of a Web client request the same set of pages from the same server. One client uses the PS-Web, while the other one uses the legacy architecture. We take care that the path conditions between the client and the server are the same in both cases by executing the two instances in parallel.

We performed a set of experiments where each set spanned an entire day and was carried out during italian business time to test our architecture when the Internet is heavily loaded. To increase estimates' reliability, each set of experiments was replicated on several days.

To take into account the influence of the Internet client-server route, we chose to perform separate sets of trials along different paths. Specifically, the mobile host was always located in Pisa (Italy), at the CNUCE-CNR Institute, while the server was located either at EPFL in Lausanne (Switzerland) or at the University of Texas at Arlington.

---

[1] With SURGE, files requested to the web server are 93% extracted from body and 7% from the tail. Hence, in our experiments, we have about 10 files taken from the tail of the distribution.

4.1 Power Saving Analysis

As shown in [13] and [21], while the network interface is in the on state, it drains a nearly constant power from the battery source, whether it receives or transmits data, or remains idle. The energy consumed by the wireless interface is thus proportional to the time it remains in the on state. Therefore, according to equation (2), I_ps can be closely approximated by measuring, both in the legacy and PS-Web architectures, the overall time the network interface is in the on state. This index is easier to compute.

Fig. 6(a) shows the overall time the network interface remains in the on state in the PS-Web and in the legacy architecture, respectively. The available service rate (i.e., the downloading speed in Kbps), as observed by the Web client in the legacy architecture, is also reported. It appears that the PS-Web is almost independent from the service rate. This is due to the joint effect of using the indirect-TCP model and data pre-fetching mechanism: the mobile host turns its network interface on only when the Web page is available at the Access Point. Therefore, the energy consumption does not depend on the state of the connection between the Access Point and the Web server, i.e., the service rate. The small variability that can be observed in Fig. 6(a) is caused by the transfer time estimator (see Section 3).
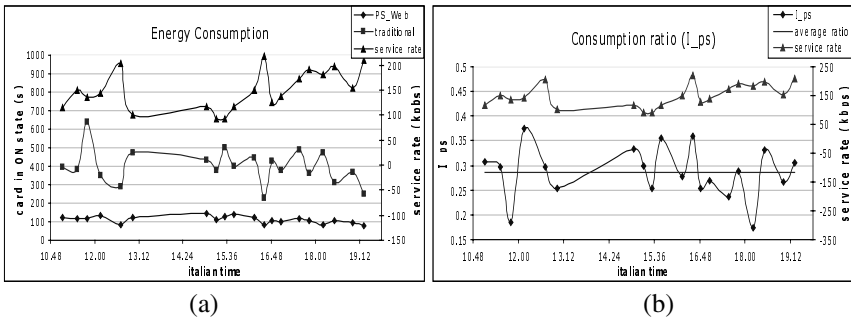

(a)   (b)

Fig. 6. Energy consumption and I_ps as a funcion of time (Web server in Texas).

The above results show that the wireless link is protected from the Internet congestions and, thus, the Indirect-TCP model and data pre-fetching work correctly. This conclusion is corroborated by Fig. 6(b) where the Power Saving Index I_ps is reported as a function of the day time. Fig. 6(b) also shows the I_ps index averaged on the whole day (average ratio in the figure) and the service rate. The power consumption in the PS-Web is always less than 40% the one in the legacy system, its average value on the whole day is below 30%, with values down to 20%. This means that, by using the PS-Web, we can save always more than 60% of battery, on average more than 70%, with saving peaks over 80%.

The above results refer to the case when the Web server is located in Texas. Analogous experiments have been performed with the Web server in Switzerland, obtaining similar results. Indeed, when the server is in Switzerland the service rate is only slightly higher than the Texas's one (175 Kbps vs. 150 Kbps) and the power saving is slightly lower (68% vs. 71%). The similarity can be justified by observing that the two paths share the initial part, which goes from Pisa to London. Furthermore,

in the European business time, when the experiments were performed, this part is more congested than the rest of the path, and thus determine the overall service rate in both cases.

4.2   Delay Analysis

Fig. 7(a) shows the additional delay experienced by Web files, averaged on each experiment and on the whole set of experiments in a day, respectively. The service rate is also included for reader's convenience. Fig. 7(b) shows the same indices but with respect to Web pages instead of Web files (recall that a Web page consists of a main file and possible embedded files).
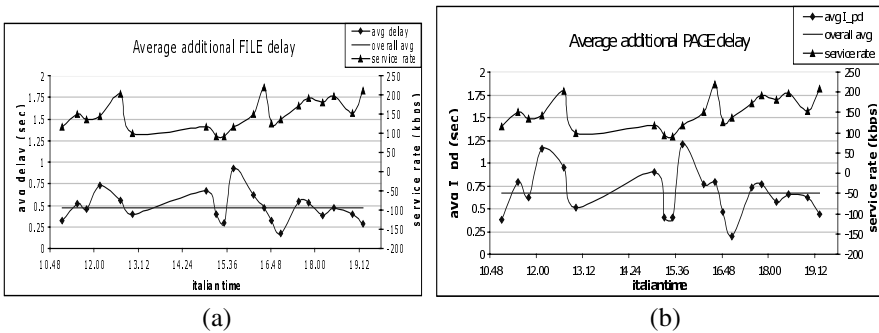

(a)   (b)

Fig. 7. Average additional file and page delay (the Web server is located in Texas).

From Fig. 7(a) it appears that the PS-Web introduces very small additional delays in transferring Web files: the average daily value of the additional delay is less than 0.5 s. Similarly, the additional delay (averaged in a day) related to Web pages is in the order of 0.7 s. In general, the additional page delay is only slightly greater than the additional file delay. This is a very important result since it proves that the pre-fetching mechanism at the Access Point works properly. In fact, the estimator of pages' transfer times forces the mobile host to keep the network interface off until the Web page is completely available at the Access Point. This means that the first file of a page experiments a certain additional delay, while successive files experiment a smaller one. Therefore, the additional delay related to the Web page is mainly determined by the first file.

Fig. 8 shows the tail of the distribution of the additional delay for files and pages, respectively. The additional delay is never greater than 1s for individual files and never greater than 1.2 s for pages.

Results discussed above were obtained with the Web server located in Texas. However, as above, experiments with the Web server in Switzerland provided very similar results which are omitted for the sake of space. Based on these results we can conclude that the power saving achieved in the PS-Web architecture is not paid with an unacceptable degradation of the QoS preceived by the user.
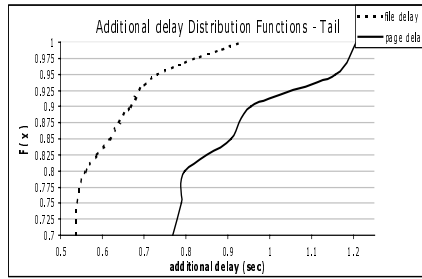
Fig. 8 . Tails of the additional delay distributions, for single files and whole pages

5   Conclusions

In this work we have proposed and experimented new strategies for reducing the power consumption while accessing a Web service in a mobile wireless environment.
 To overcome drawbacks caused by the TCP-based legacy architecture we have designed a new architecture – referred to as PS-Web – that implements novel strategies to reduce the power consumption while accessing Web services. Specifically, we operate at the transport layer and we fully exploit information about the application behavior. So, we have an application-dependent approach but our system have been designed in a modular way that can be easily adapted to other network applications, as well. Furthermore, our system requires no modification to the Web application.

The PS-Web architecture is based on the Indirect-TCP model and, as such, it isolates the wireless network from the wired network protecting the former from possible congestions in the latter. Furthermore, it makes a large use of pre-fetching to improve the performance: Web pages are first stored at the Access Point and then downloaded to the mobile host at the maximum rate allowed by the wireless link. Finally, it allows the mobile host to maintain the wireless network interface in the off state during inactivity periods (e.g., user think times).

The experimental results obtained on a prototype implementation of the system have shown that the PS-Web is able to save, on average, the 70% of energy with respect to the legacy TCP-based approach, with saving peaks over 80%. More important, this energy saving is not obtained at the cost of a degradation in the QoS perceived by the user. In fact, experimental results have shown that the additional URT delay in transferring a Web document introduced by the PS-Web with respect to the legacy TCP based approach is, on average, below 1.5 s.

We are currently working on PS-Web in order to refine the estimation of the file transfer time. This is very important since a better estimation allow to minimize the number of times the mobile host's network interface is unnecessarily turned on. Also, we are comparing the performance of our application-dependent approach with those of application-independent solutions that operate without any preliminary information about the application behavior.

References

1. M.Arlitt, C.Williamson, "Internet Web Servers: Workload Characterization and Performance Implication", *IEEE/ACM Transactions on Networking*, Vol.5, No.5, pp.631-645, Ottobre 1997.
2. A.Bakre, B.R.Badrinath, "Implementation and Performance Evaluation of Indirect TCP", *IEEE Transactions on Computers*, Vol.46, No.3, Marzo 1997.
3. P.Barford e M.Crovella, "Generating Representative Web Workloads for Network and Server Performance Evaluation", *Proceedings of ACM SIGMETRICS '98*, Madison, WI, pp. 151-160, June 1998.
4. P.Barford, A.Bestavros, A.Bradley e M.Crovella, "Changes in Web Client Access Patterns", to appear in *World Wide Web Journal*, Special Issue on Characterization and Performance Evaluation, 1999.
5. M.Crovella e A.Bestavros, "Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes", *IEEE/ACM Transaction on Networking*, Vol.5, No.6, pp.835-846, December 1997.
6. C.Cunha, A.Bestavros e M.Crovella, "Characteristics of WWW Client-Based Traces", Technical Report TR-95-010, Boston Univeristy Department of Computer Science, April 1995.
7. G. H. Forman, J. Zahorjan, "The Challenges of Mobile Computing", *Tecnical Report*, University of Wachington, March 1994.
8. D.P. Helmbold, D.E. Long, B. Sherrod "A Dynamic Disk Spin-down Technique for Mobile Computing", Proceedings of the Second Annual *ACM International Conference on Mobile Computing and Networking*, NY, pp. 130 - 142, November 1996
9. IEEE standard for Wireless LAN- Medium Access Control and Physical Layer Specification, P802.11, November 1997.
10. T. Imielinski, S. Vishwanathan, B.R. Badrinath "Power Efficient Filtering of Data on air", *Proc. of the EDBT*, Cambridge, England, March 1994.
11. T. Imielinscki B.R. Badrinath "Wireless Computing", *Communication of the ACM*, Vol. 37, No. 10, October 1994.
12. R.Kravets e P.Krishnan, "Power Management Techniques for Mobile Communication", *Proceedings of the Fourth Annual ACME/IEEE International Conference on Mobile Computing and Networking* (Mobicom'98).
13. G. Anastasi, M. Conti, W. Lapenna, "Power Saving Policies for Wireless Access to TCP/IP Networks", Proceedings of the *8-th IFIP Workshop on Performance Modelling and Evaluation of ATM and IP Networks (IFIP ATM&IP2000),* Ilkley (UK), July 17-19, 2000.
14. J.R. Lorch, A.J. Smith, "Scheduling Techniques for Reducing Processor Energy Use in MacOS", *ACM/Baltzer Wireless Networks*, 1997, pp.311-324.
15. J.R.Lorch e A.J.Smith, "Software Strategies for Portable Computer Energy Management", *IEEE Personal Communication – June 1998*, pp.60-73.
16. 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.
17. A. Passarella, "Un'architettura power saving per l'accesso al Web da computer mobile", *Laurea Thesis*, University of Pisa, October 2001 (in italian).

18. 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.
19. M.Rulnick e N.Bambos, "Mobile Power Management for Wireless Communication Networks", *ACM/Baltzer Wireless Networks*, Vol.3, No.1, Marzo 1996.
20. S. Sheng, A. Chandrakasan, R.W. Brodersen, "A Portable Multimedia Terminal", *IEEE Communications Magazine*, December 1992.
21. M.Stemm e R.H.Katz, "Measuring and Reducing Energy Consumption of Network Interfaces in Hand-Held Devices", *Proc. 3° International Workshop on Mobile Multimedia Communication*, Princeton, NJ, Settembre 1996.
22. 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.
23. M.Zorzi e R.R.Rao, "ARQ Error Control on Fading Mobile Radio Channels", accepted for pubblication in *IEEE Trans. Veh. Tech.*, Also in *Proc. IEEE ICUPC '95*, pp.211-215, Novembre
24. M.Zorzi e R.R.Rao, "Energy Constrained Error Control for Wireless Channels", *Proceeding of IEEE GLOBECOM '96*, pp.1411-1416, 1996.