Wake on Wireless: An Event Driven Energy Saving Strategy for Battery Operated Devices

Eugene Shih
Massachusetts Institute of Technology
Cambridge, MA 02139
eishih@mit.edu

Paramvir Bahl
Microsoft Research
One Microsoft Way
Redmond, WA 98052
bahl@microsoft.com

Michael J. Sinclair
Microsoft Research
One Microsoft Way
Redmond, WA 98052
sinclair@microsoft.com

ABSTRACT

The demand for an all-in-one phone with integrated personal information management and data access capabilities is beginning to accelerate. While personal digital assistants (PDAs) with built-in cellular, WiFi, and Voice-Over-IP technologies have the ability to serve these needs in a single package, the rate at which energy is consumed by PDA-based phones is very high. Thus, these devices can quickly drain their own batteries and become useless to their owner. In this paper, we introduce a technique to increase the battery lifetime of a PDA-based phone by reducing its *idle power*, the power a device consumes in a "standby" state. To reduce the idle power, we essentially shut down the device and its wireless network card when the device is not being used—the device is powered only when an incoming call is received. Using this technique, we can increase the battery lifetime by up to 115%. In this paper, we describe the design of our "wake-on-wireless" energy-saving strategy and the prototype device we implemented. To evaluate our technique, we compare it with alternative approaches. Our results show that our technique can provide a significant lifetime improvement over other technologies.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design

General Terms

Measurement, Performance, Design

Keywords

low-power radio, power consumption of wireless LANs, wake-on-wireless

1. INTRODUCTION

The popularity of cellular phones has continued to increase over the past few years. The EMC World Cellular Database predicts that by the year 2005, there will be over two billion subscribers to cellular services worldwide [6]. While the ability to talk with others is important, new users who subscribe also desire phones that include a larger set of functionalities. As a result, newer handsets that integrate phone technology with PDA-like data applications such as personal information managers and e-mail are beginning to emerge.

For the most part, these integrated devices or *universal communicators*, are cell phones augmented with computation and storage abilities. These devices operate over telecommunications infrastructure, such as GSM or CDMA. An alternative to these devices are emerging PDA devices that are already equipped with a rich set of data applications, but include built-in wireless LAN (WLAN) and IP telephony capabilities.

At a first glance, both categories of devices are capable of performing as universal communicators. However, in our opinion, the use of a PDA-based universal communicator is superior for the following reasons. First, as the Internet becomes ubiquitous, PDAs equipped with IEEE 802.11b cards will be able to obtain high-speed access to the Internet virtually anywhere. Second, because PDAs generally employ open protocols and standards, such as TCP/IP and IEEE 802.11, they are able to interoperate with a larger number of devices. A side benefit of this kind of implementation is that the end-user cost for communication can be far lower. This is in contrast to the cellular phone model where proprietary solutions are used to build the communication system and a premium is charged for service. Using a PDA-based phone also affords the user access to a larger number of mature, data and multimedia applications. Moreover, the user experience can be better since the device can take advantage of the higher bandwidth available over the wireless LAN and use better voice codecs. Finally, by using wireless LAN technology, PDA-based phones have the ability to form and participate in peer-to-peer ad hoc communication groups. Current cellular phones do not have these capabilities[1].

Another reason why PDA-based phones are important is that they are better suited for *indoor* communication. In our experience, cell phones often lose signal strength in buildings and consequently, stop functioning indoors. The problem arises due to the absorption of RF signals caused by certain kinds of building materials. Since people spend most of their time within buildings, reliable indoor communication is necessary. Wireless LANs can fill this need since they can be deployed within buildings at lower cost.

1.1 The Problem of Power Consumption

While PDA-based phones have many advantages over cellular phones, cellular phones outperform PDA-based phones in at least

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
MOBICOM '02, September 23–28, 2002, Atlanta, Georgia, USA
Copyright 2002 ACM 1-58113-486-X/02/0009 ...$5.00.

---

[1] While a few of these advantages disappear as 2.5G and 3G networks are deployed, most of them remain.

one area: power consumption. Current wireless PDAs do not manage their energy usage well and as such quickly drain their batteries. As the authors of [19] show, a large part of power drain can be attributed to the wireless LAN card. Using current WiFi technology, a PDA connected to an IEEE 802.11b wireless LAN [11] will quickly drain its battery after only a few hours. In contrast, cellular phones have standby lifetimes[2] of up to several days. This is illustrated in Figure 1 where we plot the standby time of a wirelessly-connected PDA (a Compaq iPAQ H3650 with an IEEE 802.11b card) and compare it to the standby time of a GSM cell phone (a Motorola V60t). For the PDA, we consider two cases. In the first case, the wireless card is continuously awake (CAM mode) and in the second, the wireless card is in power save mode (PS mode). In power save mode, the wireless card periodically switches between a low-power state and the active state to check for packets. In both cases, in order to keep the WiFi card operating, *the PDA remains on*, but it is in an idle, lower-power state. In general, we will refer to the power consumed by a PDA that is not being actively used, the *idle power*. From Figure 1, we see that with current technology, this PDA-based wireless LAN phone has three times less battery lifetime than that of a cell phone.

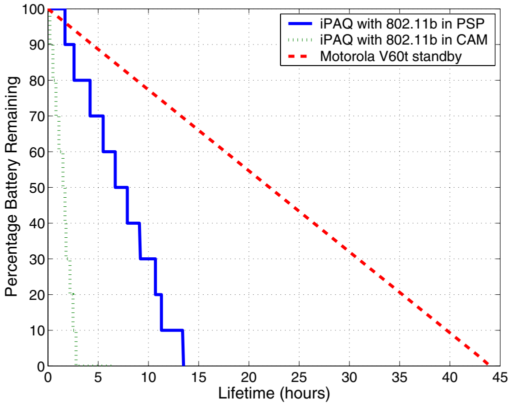


Figure 1: Standby lifetime of an iPAQ with an IEEE 802.11b card in PS/CAM modes compared to lifetime of a cell phone. The cell phone lifetime was computed assuming a constant power drain. The other curves were determined by monitoring the battery level.

1.2 Bridging the Energy Gap

Ideally, in order for a wireless LAN-enabled PDA to be successful as a universal communicator (UCoM), battery lifetimes must be comparable to that of cellular phones. Since current cellular phones have battery lifetimes of a day or more, an ideal UCoM device should have a battery lifetime of at least one day. One may argue that the problem of energy consumption is not significant since one can always recharge the battery. We argue that the problem *is* significant because a power source is not always available when the battery dies. Moreover, the goal of reducing power consumption is not merely to "save energy", but rather to reduce the load on the user's attention. In other words, if a user should happen to forget to recharge their device one day, they should not lose the ability to communicate with their PDA-phone.

To clarify the problem of power consumption, consider the following situation. To receive a phone call, a PDA-based UCoM device must have the ability to listen for and receive incoming packets. With current technology, both the WiFi card *and* the device must be powered in order to receive incoming packets. When the time interval between phone calls is large, energy stored in the battery is wasted since nothing useful is being done during this time. Eliminating this waste can improve the lifetime of the device. *We propose that when a UCoM device is not actively used that it and its wireless network card be powered off. The device is powered on only when there is an incoming or outgoing call or when the user needs to use the PDA for other purposes.*

There are different ways to achieve this objective. We will focus primarily on the design, implementation, and analysis of a simple technique that we call *wake-on-wireless*. The way this technique works is as follows. We physically separate the control channel from the data channel. We implement the control channel using a low-power radio operating on a frequency band that is different from the one used for the data channel. When the device is not actively in use, we shutdown both the device and its high-power wireless network card. To handle an incoming call, we use the low-power channel to send a *wakeup* message to the device. Once awake, the device accepts the call on its primary higher rate, higher power data channel. With our implementation of this technique, we are able to achieve an increase of 115% of standby time over a popular IEEE 802.11b-enabled PDA. Moreover, when we consider the daily talk pattern of a user who talks an average of 25 minutes a day, we achieve an increase of at least 40% over what is possible with current technology. In absolute terms, a PDA without wake-on-wireless dies in 10 hours with this use pattern, whereas our UCoM device has the potential to operate well over 14 hours.

While the particular energy savings achieved are large, there are more significant insights to be extracted from the results we present in this paper. First, if battery lifetime is a significant issue, forcing a radio that is not designed with low power consumption in mind to operate in a low power manner will likely result in only marginal gains. While it is true that varying the parameters and/or modifying a WLAN radio will result in better energy usage, using a special radio that is designed for low-power wakeup will make better use of available energy. In general, most radios cannot be designed to scale in terms of power, rate, range, size, and reconfigurability.

According to the IEEE 802.11b specification, it is designed to provide "wireless connectivity for fixed, portable, and moving stations within a local area." Essentially a wireless version of Ethernet, 802.11b must support multiple stations communicating simultaneously over a common medium. A radio that is used for infrequent short-range wakeup data messages clearly has a different goal. Therefore, such a radio does not need a complex MAC, a sophisticated front-end, or a high power amplifier.

A second, equally important point addresses the rest of the system. To effectively take advantage of the different power consumption levels of multiple radios, our systems and processors need to be able to scale and match the energy consumed by the lower power radios. There is no reason to scale the radio if the main unit and associated peripherals (including flash, RAM, and other I/O devices) do not scale. In our experiments, the iPAQ consumed about 112 mW when suspended! This is large when compared to the 2.0 mW consumed by our low power radio in standby mode. If power hungry devices are unable to scale appropriately, then a different low-power control processor should be used to control the wakeup radio while the main unit is shutdown to conserve energy.

2. RELATED WORK

The energy capacity of batteries has doubled roughly every 35 years [13]. While this trend has somewhat accelerated in recent

---

[2] In this paper, when we say lifetime, we mean *battery lifetime*.

years due to the needs of portable electronic devices, the rate of improvement is still fairly slow compared to Moore's Law. Since faster processors tend to require more power to operate, techniques to reduce and manage power consumption are necessary. Consequently, there has been much research into how to lower the energy consumption of laptops, PDAs, and other mobile devices. Researchers have focused on developing low-power techniques at all levels, from designing energy-efficient circuits [2] to managing power dynamically to adapting CPU frequencies [8] to enhancing and modifying network protocols [3, 21, 4].

Our work focuses on reducing the power consumed by the wireless network card for mobile devices. Because the power consumed by a wireless LAN card has a large impact on the lifetime of a battery-operated device, researchers have worked to understand the actual energy consumed by these devices. In [7], the authors investigate the per packet energy consumption of an IEEE 802.11b card in various modes. Packet-oriented energy measurements of the card in transmit, receive, broadcast, and idle modes are reported. Their data reveals how complexities introduced by the IEEE 802.11b protocol can impact the overall energy consumed by the card during its operation. In our work, we use a similar setup to measure the energy consumed by various wireless technologies.

In order to reduce the energy consumed by the wireless LAN card, researchers have introduced a number of techniques to minimize its impact on the system lifetime. The authors of [19] show that leaving the wireless LAN card in sleep mode whenever possible can dramatically reduce the power consumption of the device. This is basically the technique that the power save (PS) mode of IEEE 802.11b relies on. In PS mode, the card goes to sleep and periodically wakes up to check if data is available. As the authors of [20, 12] show, the PS mode of IEEE 802.11b as it is currently implemented does not reduce energy consumption in all cases. We will discuss PS mode in more detail in Section 3. Our technique is similar to that described in [19], however, instead of simply leaving the WLAN card in sleep mode we turn it off completely.

The concept of using a secondary low-power wakeup mechanism is not a new one, but to our knowledge, our work represents the first working implementation of this technique. Recently, it has come to our attention that the authors of [14] also suggest using a lower power radio in order to wakeup powered-down sensor nodes. However, they have not completed an implementation nor is there a complete analysis of the benefits that this radio could provide.

The idea of using a secondary out-of-band mechanism to wakeup mobile stations also recently appeared in [5]. Specifically, the authors describe the use of short-range, one-way, passive RF ID Tags to activate groups of nodes from their sleep states. Unlike our paper, they provide a theoretical study of wakeup, whereas we have designed and implemented an actual system. Moreover, we evaluate our system and show how our approach improves system lifetime. The PAMAS [17] protocol also uses a secondary signalling channel to reduce the energy consumption of communication between nodes in an ad hoc network. The authors report only analytical and simulated results. Moreover, the second channel is not used for wakeup and there is no discussion of the physical implementation of the second channel.

3. POWER SAVING IN 802.11

Before describing our wake-on-wireless technique in detail, we will quantify the power consumption of some popular IEEE 802.11b cards in various states of operation. The measured energy consumption in *power-save mode* will serve as a point of comparison for our approach.

To determine the power consumed in the different modes, we use a measurement technique similar to the one described in [7]. A Sycard PCCExtend 100 extender card is used to expose the connections of a Laptop PC Card slot, including the voltage supply pin V_{cc}. We insert the extender in a laptop and then insert the card to be measured into the extender. In [7], the current is measured indirectly; the authors monitor the voltage change $v_R(t)$ across a small resistor R placed between the supply and input voltages. If R is small, then the voltage into the circuit $v_{in}(t)$ is approximately equal to V_{cc} for all t. $i_R(t)$ can then be determined using Ohm's Law. Unfortunately, if R gets too large, then the current measured does not reflect the actual current sourced since the input voltage is no longer equal to the supply voltage. If the current is unknown and very small, choosing R is difficult. To avoid this problem, we directly measure the current using a Tektronix AM 503B current meter wrapped around a straight wire. A current meter determines the current flowing through a wire by measuring the magnetic flux generated by the current. Figure 2 shows the setup used to measure the energy consumed by various IEEE 802.11b cards.

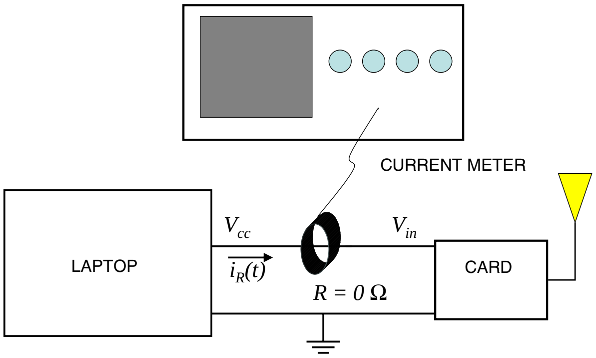


Figure 2: Measurement setup used to determine power consumption of various IEEE 802.11b cards in different modes.

Generally, according to the IEEE 802.11b specification [11], an 802.11b-based radio can be in one of three modes: *awake*, *doze*, and *off*. In the awake mode, the card itself can be in one of three substates: *transmit*, *receive*, and *idle*. In the doze or sleep state, the card is idle and unable to send or receive data. Using the measurement analysis techniques described, we determined the power consumed by an Cisco AIR-PCM350 and an ORiNOCO PC Gold Card in the various different modes described. The results are shown in Table 1. From the table, we can see that the power consumed by an IEEE 802.11b card in doze or sleep mode is an order of magnitude lower than the three other modes. The *power-save* (PS) mode of IEEE 802.11b tries to take advantage of this difference in order to reduce energy consumption. Consider an infrastructure network where wireless devices communicate with each other via an

Table 1: Average current, $\overline{i_r}$, of two IEEE 802.11b cards operating at 11 Mbps. $V_{cc} = 5.0$ V. The power is computed as $P = V_{cc}\overline{i_r}$.

Chipset	Sleep (mA)	Idle (mA)	Receive (mA)	Transmit (mA)
ORiNOCO PC Gold	12	161	190	280
Cisco AIR-PCM350	9	216	260	375

access point. When a wireless device goes into PS mode, it informs the AP of this fact. Once the AP receives this information, it buffers incoming data for the device. Every BeaconPeriod, the AP sends a beacon containing a traffic indication map (TIM) to indicate whether or not the device has pending data. At the same time, the wireless device wakes up every ListenInterval number of beacons and transmits a PS-Poll packet to receive buffered data if data is available. If no data is available, it can go to sleep right away. Figure 3 shows the power consumed during one cycle of PS mode as implemented on an ORiNOCO PC Gold Card with a ListenInterval of 100 ms or one BeaconPeriod. Since the ListenInterval parameter is a 16-bit value, theoretically, the mobile can be in sleep mode for up to 65535 BeaconPeriods. Since BeaconPeriod is typically set to around 100 ms, this value for ListenInterval can result in large sleep times and significantly lower energy consumption. Given Figure 3 and the values in Table 1, we can model the energy consumed during one complete PS cycle of the ORiNOCO card as

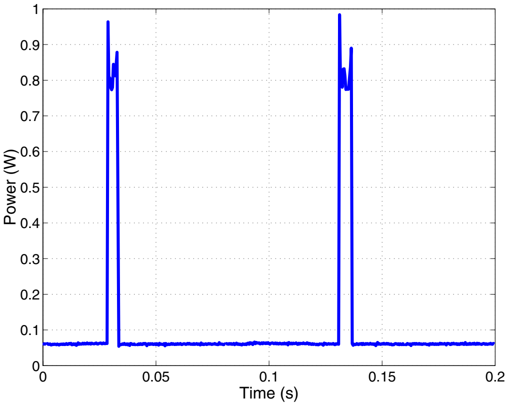


Figure 3: The power consumed by an ORiNOCO PC Gold card during power-save mode with a sleep duration of 100 ms.

$$E_{cycle}(n,t) = 0.060nt + 3300, 0 \le n \le 65535 \quad (1)$$

where t represents the duration of a BeaconPeriod and n is the value of ListenInterval. The constant represents the energy consumed during the beaconing (computed by integrating) in μJ while the coefficient represents the power consumed during doze mode in W.

Unfortunately, the time between beacons is limited by another factor. Since the beacons are also used for synchronization purposes, the duration between beacons is likely bounded. In fact, on the ORiNOCO PC Gold Card, we were only able to achieve a maximum sleep interval of 1 s. Using this value for nt, we can conclude that for one PS mode cycle, $E_{cycle} \approx 0.063$ J.

Before discussing our implementation, we would like to point out that different manufacturers have different implementations of power-save mode. Figure 4 shows a power trace of a Cisco AIR-PCM350 in Max-PSP mode. The Cisco card appears to go through several states even when data is not available. Also, during experiments with the card, it appeared that it would alternate between doze and awake states. If we ignore this anomaly and assume that the pattern in Figure 4 continues, then we can model the power save mode of the Cisco card as

$$E_{cycle}(n,t) = 0.045nt + 24200.$$

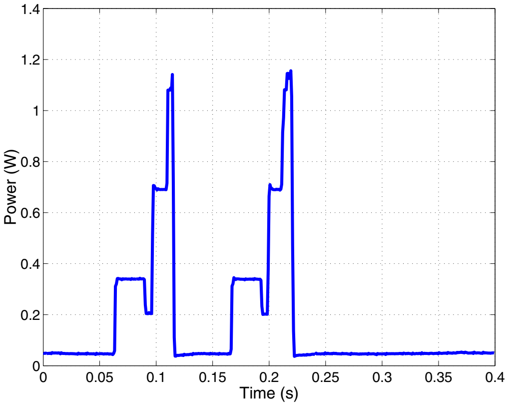


Figure 4: The power consumed by a Cisco AIR-PCM350 card during "Max-PSP" mode.

4. WAKE-ON-WIRELESS

The basic goal behind wake-on-wireless is to to eliminate the power consumed when a IEEE 802.11b-enabled UCoM device is idle. By adding a second, low-power channel, we are able to shut the rest of the system off and reduce the idle power. At the same time, out-of-band control information can be sent to maintain connectivity and wakeup the UCoM device when necessary. In the next few sections, we describe the design and implementation of the wake-on-wireless system we have built.

4.1 System Architecture and Components

The system we have built, called the *UCoM System*, can operate both in infrastructure mode, where clients are connected to each other through a infrastructure network and in ad hoc mode, where the clients are connected to each other directly. The techniques we have developed equally benefit both of these modes, but in this paper, we will focus primarily on the infrastructure mode.

Figure 5 illustrates the organization of the UCoM System. The three primary components of the system include: globally available *UCoM Servers*, WLAN-enabled *UCoM Clients* and wired or wirelessly connected *UCoM Proxies*. The UCoM Proxies and the UCoM Server are always connected to the Internet. UCoM Proxies communicate with UCoM Clients via the low-power channel. We describe the functions of these components below.

UCoM Server: The UCoM Server is a combination of a presence server (SIP Server) [18], a location server (WISH Server) [1], and a Brick Server. The Brick Server keeps track of all the registered UCoM Proxies and Clients in the system. A description of actual "Bricks" is provided in Section 4.2. Note that we make a distinction between the presence server, which keeps track of the location of clients at the IP level, from the location server, which keeps track of clients at the geographical level.

Since the UCoM Server contains a SIP Server, it performs all the tasks of a typical presence server. In addition, it manages relationships with UCoM Proxies, which communicate with UCoM Clients. The UCoM Server also acts as an intermediary between callers and callees. For example, when a connection request is received, the UCoM Server is responsible for confirming the caller's identity and protecting the callee from malicious callers who may send "wakeup" signals with the purpose of draining out the callee's battery.

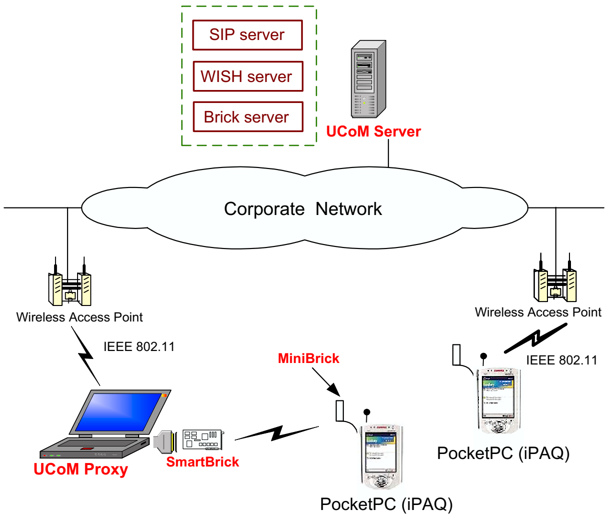


Figure 5: The UCoM System Components and Architecture.

UCoM Client: A UCoM client consists of a PDA device with both an IEEE 802.11b card and the secondary low-power radio. As we have explained, when the PDA device and IEEE 802.11b card are off, the secondary low-power radio enables the device to remain "in the system". When on, the UCoM client runs a telephony application (Talk) for communication purposes. The software includes an audio capture and playback module, an audio codec, a presence and location module and a socket-based networking module. The UCoM Client software registers with the server and queries the server to discover if its "buddies", defined as users the client is interested in knowing about, are registered. The client software is also responsible for initiating a call via the server, making the connection, and then managing the ensuing phone call.

UCoM Proxy: A UCoM Proxy runs a daemon process which may run on any machine connected to the Internet. It has the ability to communicate with UCoM Clients over the low-power control channel. Functionally, the UCoM Proxy does two things. First, it listens on the low-power control channel for registration requests from nearby UCoM Clients. On receiving such requests, the Proxy forwards the registrations to to the UCoM Server. Second, upon receiving a *wakeup* request from the UCoM Server for a UCoM Client, the UCoM Proxy sends a POWER_ON command to the UCoM Client over the control channel. Figure 6 illustrates an example of how an IP device can connect and talk to a UCoM Client that is powered off.

In the next two sections, we discuss the hardware and software implementation of our system.

4.2 Hardware Implementation

The UCoM Client device consists of a Compaq iPAQ H3650 PDA equipped with a single slot PC card expansion pack. In this slot, the UCoM device uses a Cisco AIR-PCM350 802.11b wireless networking card as its main communication channel. While the H3650 iPAQ is not the most energy-efficient PDA, we chose it because it is popular, easily programmable, and has excellent support for multimedia applications. To this platform, we added a *MiniBrick*, a piece of hardware which includes a secondary radio for low-power out-of-band signaling. It is this radio that enables the wake-on-wireless mechanism. Figure 7 shows a photo of the UCoM device.

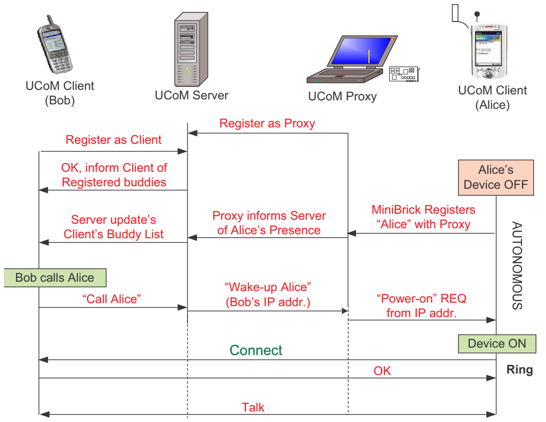


Figure 6: Call Setup

Figure 7: The Minibrick and COMPAQ iPAQ are integrated together into a single package to form the UCOM Client Device.

The UCoM Proxy consists of a networked computer equipped with a low-power radio capable of communicating with a UCoM Client via the MiniBrick. The new piece of hardware, called a *SmartBrick*, was designed to plug directly to the serial port of any networked computer. The serial port was chosen as the interface for the SmartBrick mainly because serial port interfaces are available on most computers. Using a serial port design would give us access to the Internet and other local services easily. Moreover, since networked computers are ubiquitous in our organization, UCoM proxies could be placed everywhere. Any device with a SmartBrick would be able to experience the benefits of wake-on-wireless.

4.2.1 The MiniBrick

The main components of the MiniBrick are a simple microcontroller, a low-power radio, and various sensors and actuators. Figure 8 shows an architectural overview of the MiniBrick.

The MiniBrick's microcontroller, a PIC 16LF877 running at 10 MHz, is used to control the radio and to send and receive messages to and from the radio when the iPAQ and the IEEE 802.11b card

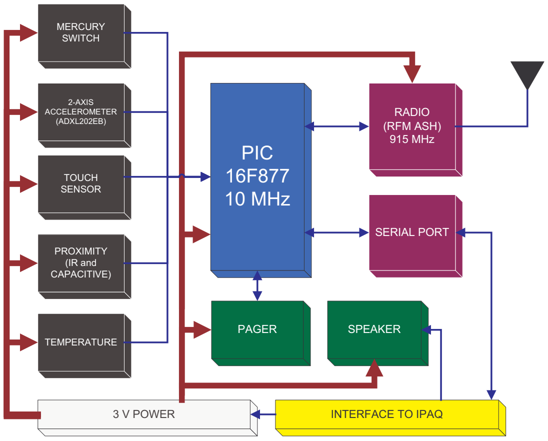
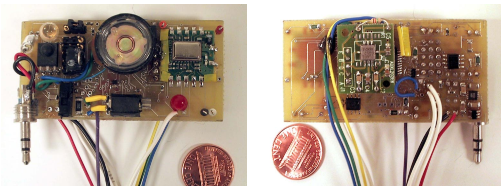


Figure 8: MiniBrick Architectural Overview

are off. The PIC also includes many different submodules to enable easy interfacing to peripherals, such as an analog-to-digital converter, timers, and a built-in UART. The radio, an RF Monolithics TR1000 ASH Transceiver, serves as the low-power communication channel between the MiniBrick and SmartBrick. The TR1000 uses the 915 MHz ISM band and modulates data using amplitude-shift keying. The maximum achievable data rate is 115.2 kbps. According to the ASH Transceiver Designer's Guide, in free space, the range of the radio is about 332 feet. When experimenting with the device in an office environment, we discovered that it is capable of transmitting at distances of about 30 feet. In our prototype implementation, we treat this radio as a UART and transmit data at 19.2 kbps. Finally, sensors are included to give the iPAQ a more phone-like feel. Figure 9 shows a MiniBrick PCB fully populated with all of its components. Note that power for the MiniBrick is derived directly from the 4.0 V Lithium Polymer battery contained in the iPAQ main unit. Communication between the MiniBrick and the iPAQ uses the built-in UART.



(a) Front (b) Back

Figure 9: The MiniBrick PCB. The front of the MiniBrick includes sensors, a pager, and the RFM TR1000. The back of the MiniBrick contains an accelerometer, the PIC processor (under accelerometer), and a temperature sensor.

It is important to note that while the TR1000 is a good radio to demonstrate the concept of wake-on-wireless, we are by no means tied to this radio. We prefer to use radios that are not only low-power, but also widely available, standardized, and usable worldwide without licensing issues. A single radio that can achieve a wide variety of data rates with extremely low-power is the ideal solution. Unfortunately, such a radio is not widely available today. Standards proposed by the IEEE 802.11 working group, while increasing data rates, have placed less emphasis on designing lower power systems. Meanwhile, the power consumption of IEEE 802.15.1 and Bluetooth remain relatively high [16].

4.2.2 *The SmartBrick*

To communicate with the MiniBrick, SmartBricks must be placed within the environment. As mentioned, our prototype SmartBricks are designed to plug into a single serial port of a PC. Two SmartBricks are shown in Figure 10. The SmartBrick is similar in design to the MiniBrick. Unlike the MiniBrick however, there are no sensors on board since its purpose is to relay information to and from the UCoM device using the TR1000. In addition, the SmartBrick does not require any additional power source. Power for the SmartBrick is derived from an unused control line on the serial port.

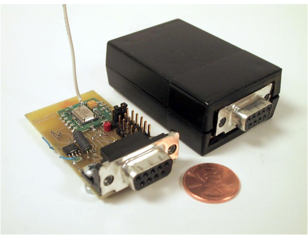


Figure 10: Two SmartBrick devices.

4.3 Software Implementation

In order to implement the wakeup mechanism for the UCoM device, the UCoM proxy must be able to communicate with mobile devices. A simple point-to-point protocol to communicate between a single MiniBrick and a single SmartBrick was designed for this purpose. Since it is conceivable that multiple MiniBricks may send messages to a single SmartBrick, a low-power MAC scheme to manage the communication channel is also necessary. However, the design of such a MAC scheme is not considered here.

4.3.1 *Packet Format*

To communicate between Bricks, we have designed a simple radio packet format. All communication between SmartBricks and MiniBricks uses this format. This packet format is shown in Figure 11. The PREAMBLE contains an alternating sequence of 1's and 0's to provide DC balancing for the receiver. The DEST_TYPE field specifies the device type of the intended recipient of the message. We allocate two bits for specifying the device type.

Figure 11: Packet Format - Low Rate Communications

The DEST_ID is a 16-bit field that contains the identity of the recipient Brick. The fields SRC_TYPE and SRC_ID are similar to the destination counterparts. DATA_SIZE specifies the length in bytes of the data. For ease of implementation and to reduce the number of packet errors during transmission, the length of the data field is limited to 48 bytes in our prototype system. Finally, the CRC field is the 16-bit CRC of everything in the packet except the preamble. Packets with a CRC error are currently dropped.

4.3.2 MiniBrick Modes of Operation

The MiniBrick is designed to function in two different modes: **Autonomous Mode** and **Command-Driven Mode**. In Autonomous Mode, the iPAQ portion of the UCoM device is turned off in order to reduce energy consumption. Since the iPAQ is unable to communicate using the wireless LAN, we utilize the low-power communication capabilities of the MiniBrick to register the UCoM device with the UCoM Server.

The MiniBrick transmits a short message containing its source type and identity. The destination type and identification are set to broadcast values. The DATA section of this packet is empty since only information about the UCoM device is required to register. We broadcast the 68-bit message ten times, which takes approximately 7 to 8 ms. We immediately switch the radio into a receive mode and wait for 20 ms for a response. If no WAKEUP message is heard within 20 ms, we go to sleep for 300 ms and then continue broadcasting our message. If a wakeup message is received, the MiniBrick turns on the iPAQ by toggling the Data Carrier Detect (DCD) line on the serial port. The MiniBrick then switches into Command-Driven Mode. Once in this mode, communication can proceed over the IEEE 802.11b channel and the application can turn off the MiniBrick or it can utilize the MiniBrick. If the iPAQ is turned off, the MiniBrick will return to Autonomous Mode. Figure 12 shows the states used by a MiniBrick in Autonomous Mode.

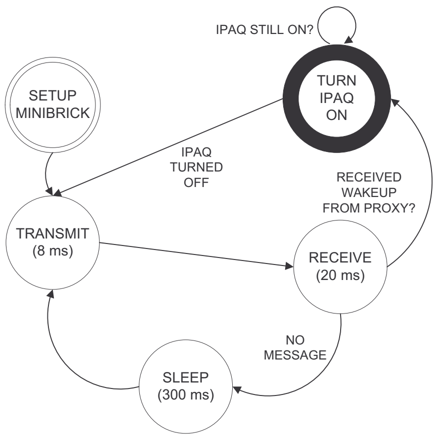


Figure 12: Autonomous Mode State Diagram

4.3.3 SmartBrick Modes of Operation

In our system, SmartBricks are always powered and perform operations based on function calls from the host. Our prototype system uses SmartBricks plugged into the serial port of a desktop computer. Access to the SmartBrick radio is performed using the SmartBrick API. The main functions of the Brick API include:

- **uSbrickGetId**: This function obtains the identity of the SmartBrick. A call to this function is a blocking call.
- **uSbrickTransmit**: uSbrickTransmit sends out a specified message (*msg*) on the low-power radio for *duration* milliseconds. A call to this function is non-blocking. The message cannot exceed 48 bytes.
- **uSbrickReceive**: This function puts the SmartBrick into a “listen” state. In the listen state the Smartbricks listens for messages on the RF link until *timeout* milliseconds or a message is received. Currently, a call to this function is a blocking call since our prototype does not take advantage of the interrupt capabilities of the microcontroller.

5. SYSTEM PERFORMANCE

Our justification for adding hardware to an existing iPAQ was to increase the operational lifetime of the device. In order to compare the lifetime that we can achieve with our new addition, we first examine the battery lifetime of the iPAQ when no additional hardware is present and when the iPAQ itself is in a suspended state.

Figure 13 illustrates the standby lifetime of an iPAQ without any additional hardware, i.e. without an IEEE 802.11 NIC and without the MiniBrick. The figure is based on actual measurements taken with an ammeter connected in series between a fixed voltage source and an H3650 iPAQ. Figure 13 indicates that without additional hardware, significant power drain occurs in the iPAQ in the suspend state. While the IEEE 802.11b card and backlight are off, 112 mW of power is still consumed since power is needed to refresh the system DRAM and for intercepting power-on interrupts. Interrupts are used by calendaring applications and for event notifications. *Since energy is used even when the iPAQ is in the suspend state, the upper bound on the lifetime of an iPAQ is 35 hours.*

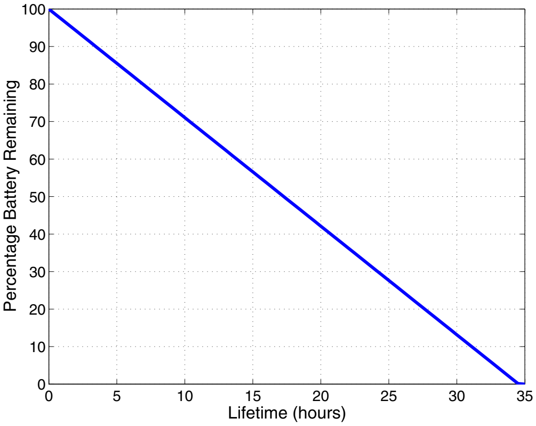


Figure 13: Standby lifetime of an iPAQ with expansion pack.

Initially, to evaluate how close our MiniBrick-enabled UCoM phone device approaches this lifetime bound, we hoped to use the battery monitoring features of the iPAQ to measure the power consumed during actual phone conversations. We were unable to do this for two reasons. Since the battery monitoring features of the iPAQ allows course tracking only, the power consumed would be difficult to monitor accurately. Moreover, because we had only two UCoM devices, the profiles we could generate would be limited. Instead of measuring the power consumption of real phone conversations, we decided to take a two step approach.

First, we measured the power consumption of the MiniBrick in various modes. The power consumed by the MiniBrick in sleep, receive, and transmit modes is shown in Table 2. We measured these values by capturing the power consumed by the MiniBrick while cycling through the states of Autonomous mode. The measurement technique used in Section 3 was also used here.

After measuring the MiniBrick power consumption, the power consumption of an iPAQ equipped with a Cisco AIR-PCM350 wireless card and MiniBrick was determined. We measured the power consumed while an actual conversation took place (ACTIVE state), while we attempted a call (ATTEMPT state) and when the

Table 2: The measured average power consumption of the MiniBrick. The numbers include the power consumed by the PIC and the radio.

	Sleep	Receive	Transmit
Current (mA)	0.6	2.2	2.4
Power (mW)	2.0	7	8

device was in a standby mode (STANDBY state) waiting for incoming calls. In the ACTIVE and ATTEMPT states, the MiniBrick is off, while in STANDBY state, the iPAQ itself is in a suspended state and, as described in Section 4.3, the MiniBrick is in Autonomous mode.

To measure the power consumed, we used an ammeter in series with the fixed voltage supply of the iPAQ. We recorded the current during the various modes. To put the device into the correct mode, we used the Talk application that we wrote for voice communication and setup a call via the UCoM Server. During each phase of the call, we recorded the average current. Since the IEEE 802.11b card can transition among idle, transmit, and receive modes during a call, we realize that this measurement technique may not accurately reflect the energy consumed. Nevertheless, using an average current will give a good estimate of the energy consumed. Table 3 shows the power consumption during the modes described.

Table 3: The power consumption of the UCoM device in three different modes. In standby mode, we take the power consumed by the iPAQ during standby mode and add the average power consumed by the MiniBrick during Autonomous Mode.

iPAQ mode	MiniBrick mode	Power consumed (W)
ACTIVE	Off	2.92
ATTEMPT	Off	2.92
STANDBY	Autonomous	0.114

Our second step was to obtain a realistic profile of phone usage. We acquired the cellular phone bills over a period of one month for two different users. The first user, Alice, had a total talk time of 798 minutes while the second user, Bob, had a talk time of 562 minutes. While both users likely keep their phone plugged into a power source whenever possible, for the purposes of assessing the lifetime of the UCoM device, we will assume that both users keep their phones *unplugged*. This assumption is reasonable since the goal of increasing the lifetime of the device is to avoid having to constantly recharge the device. It is also not always possible to access a continuous power supply.

Figure 14 shows the cellular phone usage of the two users on a representative day. To simplify our calculations, we have aggregated all incoming minutes made within an hour into a single incoming call. Likewise, we have aggregated all outgoing minutes made within the same hour into a single outgoing call. During the rest of the time, we assume that the device was in a standby mode, ready to receive phone calls.

Given the profile of the two different phone users and the power consumption of the UCoM device in various modes, we can now determine the lifetime of the UCoM device. As mentioned previously, the UCoM device uses energy from the iPAQ battery and the IEEE 802.11b card uses the battery contained in the expansion pack. *We will assume for the purposes of our analysis, that an aggregate amount of energy is available to both the iPAQ and the cards.* From the iPAQ 3600 specification sheets, we know that the capacity of the main Lithium-Polymer battery is 950 mAh. The expansion pack also contains a 950 mAh battery. At 4.0 V, this is equivalent to 7.4 Wh (or 26,640 J). Table 4 shows the average energy consumption of the UCoM device per day over a one month period given the usage patterns of Alice and Bob. After performing these energy consumption calculations for each day of the month, we discovered that Alice require an additional recharge on one day while Bob require no additional recharges. To better illustrate the rate of energy consumed, we took the representative profiles shown in Figure 14 and determined the total energy consumed by hour. Given the energy consumed by hour, we then graphed the percentage of battery energy remaining versus the hour of day. This is shown in Figure 15. We chose the representative profiles to demonstrate the rate of energy consumed.

Table 4: Total energy required per day assuming we use the iPAQ with the IEEE 802.11b wireless card and the MiniBrick.

Profile	Max (Wh)	Min (Wh)	Mean (Wh)	SD (Wh)
Alice	9.0	1.7	2.8	1.6
Bob	5.5	1.9	3.0	1.0

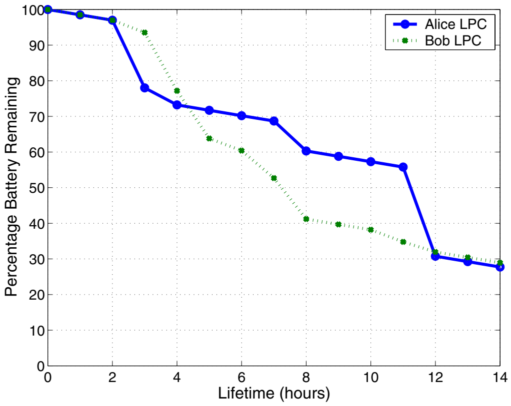
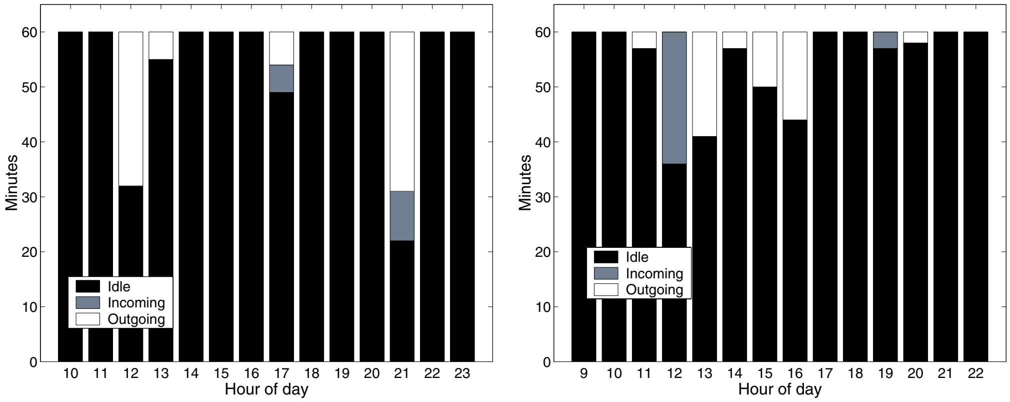


Figure 15: The lifetime of the UCoM Device (iPAQ, IEEE 802.11b card, and MiniBrick) for two different profiles.

In most cases, the lifetime achieved by adding the MiniBrick can push the lifetime of the UCoM device to a full day. In order to fully understand the gain in lifetime, however, we also examined the energy consumption of the iPAQ with only an IEEE 802.11b card given the cell phone usage profiles. We assume that the aggregate device (iPAQ with IEEE 802.11b card) can be placed into states that correspond to the UCoM Device/MiniBrick. *However, the iPAQ cannot be completely off during this time, otherwise it will not be able to accept incoming calls.* When the LCD is dimmed and the iPAQ is in transmit or receive mode, the power consumed was measured to be 2.9 W. When the IEEE 802.11b card and iPAQ are placed into a power save mode, the total average power consumed is approximately 0.45 W. This figure is determined by combining the power consumed by a standalone iPAQ in the lowest power-on state (340 mW) with the average power consumed by the AIR-PCM350 card in power save mode (110 mW). We assume that the Cisco AIR-PCM350 card behaves as shown in Figure 4 with a sleep interval of 300 ms and beacons of 50 ms.

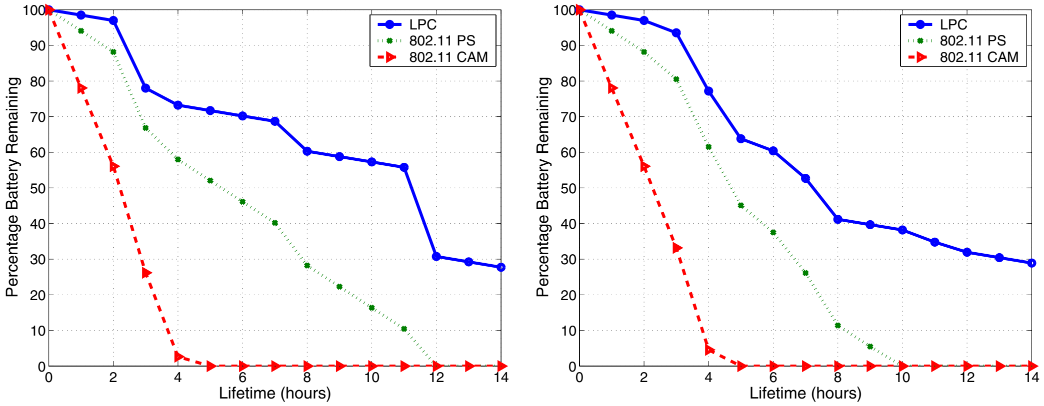
Using these numbers, we found that Bob would have to perform a midday recharge for 11 days in his profile. Alice would



(a) Alice

(b) Bob

Figure 14: Cellular phone usage profiles of two different users over a period of one day. From both of these profiles, one can see that the phone spends the majority of the time in standby mode. Alice spent a 82 minutes on the phone and Bob spent 80 minutes.



(a) Alice

(b) Bob

Figure 16: The lifetime of the UCoM device using a low-power channel (LPC) compared to the lifetime of the UCoM device with an IEEE 802.11b card only.

have to perform a recharge for 7 days in her profile. Figure 16 shows the lifetime of the UCoM device assuming that no MiniBrick is present. Figure 16 shows that using the MiniBrick can result in a considerable gain in lifetime. By using IEEE 802.11b alone in CAM or PS mode, the device has a greater probability of running out of energy before the end of the day. Table 5 summarizes the gain in lifetime that the UCoM device offers over the IEEE 802.11b-only iPAQ device.

5.1 Systematic Inaccuracies

In the analysis just discussed we made some assumptions that could impact the accuracy of the energy consumption calculations. Inaccuracies may result due to types of information that are *not* reported on a typical cell phone bill. For instance, cell phone bills typically round partial minutes to the next minute. Therefore, a 2 minute 1 second call would be recorded as a 3 minute call. This could result in an overestimate of the energy consumed during that call. Another inaccuracy is from what we will term "failed" calls. Failed calls are calls that occur that do not count towards actual "air time" and thus, do not appear on the cell phone bill. A failed call would occur when an incoming call goes to voice mail before

Table 5: The increase in lifetime of the UCoM device over the single IEEE 802.11b radio iPAQ phone device. In all four cases, a gain of $> x\%$ is given because the device is still usable after the end of the user's day has passed.

User	Gain over 802.11b PS	Gain over 802.11b CAM
Alice	$> 17\%$	$> 180\%$
Bob	$> 40\%$	$> 180\%$

the owner of the phone could answer it. A similar inaccuracy results from prematurely terminated outbound calls, i.e. hanging up before the callee answers. During both of these times, the phone operates in a higher energy mode. Because failed calls are omitted from phone bills, we cannot determine the energy consumed when those failed calls occurred. This may result in an underestimate of the energy used. These sources of inaccuracies may affect the absolute energy consumed. However, since the same profile is used in each of the different schemes, these inaccuracies will not affect the relative energy consumption.

5.2 Latency

One important parameter that should be considered is the latency of the UCoM system. A large latency is undesirable since callers are unwilling to wait a long time for the intended recipient of the call to pick up. Usually, callers will simply give up and hang up the phone. We did not determine the end-to-end latency of our system, however, we believe that on average, the wakeup took approximately 5 to 10 seconds. We believe that this latency can be reduced by optimizing each component in our system.

6. ALTERNATIVE STRATEGIES

In Section 5, we showed how we are able to increase the overall lifetime of an iPAQ-based phone by 1.8 times or more. While this performance improvement is large, there is one major drawback: one has to introduce a new piece of hardware to the client and new infrastructure into the working environment. On the surface, the addition of new infrastructure to the existing wireless LAN might make this solution less attractive. We will show, however, that this additional cost is necessary. To understand why this is the case, it is useful to examine some alternative single-radio approaches which require no additional infrastructure.

The first and most obvious approach is to continue using IEEE 802.11b WLAN. We have shown in Section 5 that using an unmodified IEEE 802.11b network leads to short battery lifetimes. Some may argue that with advances in microelectronic technology, IEEE 802.11b will decrease in power and be able to meet the low-power needs of wake-on-wireless. While these advances may benefit pure digital devices, it is not clear whether analog circuits will benefit. With this limitation in mind, one may propose to alter the way the IEEE 802.11b standard is implemented. Unfortunately, this could require reimplementation and redeployment of client cards and access points. Any significant changes to the standard would require end users to adopt new technology. Moreover, it not clear that these modifications would lead to any greater benefits.

6.1 Replacing IEEE 802.11b

Instead of investigating how one would alleviate the problems associated with IEEE 802.11b, one way to achieve lower power consumption would be to use another radio technology or to modify an existing one. For example, one could use the Bluetooth radio instead. Table 6 shows the measured power consumed by a Xircom CreditCard Bluetooth PC Card adapter in “discoverable” mode. For comparison purposes, we also show the power consumption of a Bluetooth chipset. By comparing these numbers with the TR1000 power consumption numbers, one can see that Bluetooth is much less power efficient than the TR1000.

Table 6: Power consumption of Bluetooth radios, n/a = not available.

	Xircom Credit Card Bluetooth Adapter	Silicon Wave SiW1502
Sleep (mW)	n/a	20
Transmit (mW)	250	140
Receive (mW)	263	160
Idle (mW)	140	n/a

The numbers shown in Table 6 are somewhat misleading since they are based on results obtained on evolving Bluetooth hardware. The increase in power consumption is likely due to the fact that Bluetooth radios have a sophisticated MAC protocol and supports a larger number of channels for communication. Also, as Bluetooth implementations continue to develop, the power consumption is expected to drop. Furthermore, since Bluetooth devices are capable of dropping into lower power modes after bonding with other Bluetooth devices (e.g. SNIFF, HOLD, or PARK), the average consumption could be extremely small.

While Bluetooth seems to be an attractive single-radio solution to support communication tasks for our UCoM device, there are several drawbacks. First, the 721 kbps maximum date rate is rarely achievable especially when other Bluetooth and IEEE 802.11b devices are also communicating. Adequate quality voice communication may be sporadic. Secondly, the range of Bluetooth is similar to that of the TR1000; most devices are designed primarily to support communication under 30 ft. Third, Bluetooth devices require synchronization before they are able to communicate with other devices. The latency required to synchronize can be up to 10 s. This can limit the mobility of the device. Finally, Bluetooth devices often require explicit connect and disconnect. This too can limit the mobility of the device.

While Bluetooth may not serve as a good single radio approach for mobile voice communication, it is conceivable that Bluetooth could be used as an alternative dual radio approach. At the time of this writing, we chose not to use Bluetooth because the hardware was difficult to obtain and programming a Bluetooth device was difficult. However, if Bluetooth power consumption continues to drop, there is no reason why it should not be used as a wakeup mechanism. In Figure 17, we compare the lifetime achievable by using a UCoM device equipped with a Xircom CreditCard Bluetooth radio instead of a TR1000 radio. Though the range of Bluetooth devices is still an issue, Bluetooth dual-radio devices can outperform the IEEE 802.11b single-radio approach.

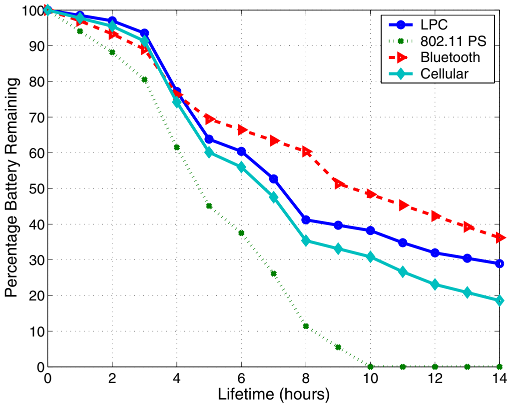


Figure 17: The lifetime of the UCoM device over a single day using various wireless technologies in lieu of a low-power channel for Bob. Using the low-power channel (LPC) results in the longest lifetime in *all* cases.

While none of the current “standard” radios are perfectly suited for use as a secondary wakeup channel, a new emerging standard appears promising. Members of the IEEE 802.15.4 working group are exploring the feasibility of designing an low-power, low-rate radio with a built-in MAC. According to early physical and MAC layer proposals, the power consumption of these radios will be ultra-low [9]. Early silicon shows that the power consumed by these radio is about half that of Bluetooth [15]. Perhaps this radio can be used instead of a proprietary solution.

6.2 Pagers

Another technology we considered using for wake-on-wireless was a pager radio. We measured the power consumption of a SkyTel Motorola Pager to be approximately 1.1 mW in standby mode. This low power consumption is achieved through careful power management of the pager radios. While the power consumption of papers is very low, the high latencies of paging make it unusable for wake-on-wireless. In our experience, the minimum paging latency is about 45 seconds. This period is too long for wake-on-wireless since most callers would likely give up on the unanswered call after 10 seconds. Another drawback of pagers is that they provide only one-way communication.

6.3 Cellular Phones

At the beginning of this paper, our main goal was to build a solution that would have a lifetime comparable to that of cell phones available in the market today. Figure 17 shows that the UCoM device with the MiniBrick uses a lower percentage of battery energy during the representative day than a typical cell phone. To determine the lifetime curve for a cell phone, we measured the power consumption of the Motorola V60t cell phone during talking, listening, standby, ringing, and during an attempt to call. We removed the battery of this cell phone and inserted an ammeter in series between the battery and the phone. Table 7 shows the results of our measurements. For comparison purposes, we also provide lifetime numbers advertised by manufacturers of various phones in Table 8. However, realize that the operation times are estimates only and they vary depending on transmitting power level, signal strength, operating mode and type of phone use.

Table 7: Measured power consumption of the Motorola V60t. Values are given in mW.

Mode	High	Low	Average
Standby (Weak Signal)	156	84	125
Standby (Strong Signal)	26	17	20
Ringing	1676	1440	1582
Talking	1612	1032	1254
Attempting call	884	704	696

Table 8: Advertised cell phone talk and standby times.

Model Name	Talk Time	Standby Time
Nokia 3360	1h to 3.5h	18h to 10.5d
Nokia 3285	40m to 2h55m	10h to 4.5d
Nokia 8860	35m to 2h40m	2.5d to 6.5d
Sprint TP 5250	70m to 150m	16h to 150h
Samsung SPH-I300	up to 4h	up to 100h
Samsung SPH-N200	up to 3.8h	up to 140h
Motorola V60t	up to 3h	up to 160h

While the lifetime of the cell phone is less than the lifetime of the UCoM device, there is one caveat to point out. Though the lifetime of the cellular phone is less than that of the UCoM device, the absolute energy consumed by the cellular phone was also less (2.6 Wh as opposed to 5.4 Wh for the UCoM device). The cellular phone cellular phone has a battery with a capacity of 3200 mWh (or 11,520 J) as opposed to the 7600 mWh available on the iPAQ.

7. DISCUSSION

In this paper, we have shown that the lifetime of an iPAQ used in phone applications can be significantly improved by utilizing a wake-on-wireless technique. Specifically, with this technique, we are able to achieve a standby time of 30 hours, which is an improvement of almost 115% over an unmodified iPAQ with a wireless card in power save mode. While lifetime has improved, the use of a second radio does introduce some new challeges.

One problem with using a low-power radio is its short range. Consequently, in order to successfully deploy a wake-on-wireless system, one would have to deploy a large number of infrastructure SmartBricks. In our opinion, the benefits afforded by the deployment of such infrastructure far outweighs this drawback. Moreover, because the radio is small and requires so little power, it can easily be integrated into other devices already in homes and offices. Finally, with the introduction of low-power IEEE 802.15.4 radios, the range problem may become a non-issue; these radios reportedly have ranges of 100 m [15].

Another issue that the wake-on-wireless technique must consider is the fact that IEEE 802.11b radios are getting lower in power. Newer IEEE 802.11b cards in the CompactFlash form factor require a voltage of only 3 V. Future chipsets may reduce this even lower. If this trend continues, why not simply use a single IEEE 802.11b radio? In our opinion, there is a limit to how low the IEEE 802.11b radio is able to go without changes to the standard. The power consumption of the TR1000 in all modes is already quite low. Moreover, having a second low-power radio can potentially help during the active operation of a UCoM device.

8. FUTURE WORK

While this paper has focused on how to extend the lifetime of a PDA-based phone using a low-power channel, there are other applications where it can also be beneficial.

Having a second control channel that does not interfere with the primary data channel allows us to revisit several other problems in wireless networking. For example, consider the problem of providing quality-of-service (QoS) over a wireless link. QoS can be implemented conveniently using a dual-radio system. Specifically, when implementing a centralized fair queueing algorithm, the wireless clients can use the control channel to send their state information (e.g. number of queued packets, priorities of each packet, their deadlines etc.) to the central scheduler, which may be in the wireless access point. By knowing the states of all nodes it is servicing, the AP could send a wakeup signal to each node individually when it is ready to let the node access the wireless channel. This saves power, since nodes do not have to constantly sense the channel for availability, and provides timeliness of service.

Another area that can benefit from a dual radio system is client handoffs. For example, in the IEEE 802.1x standard [10], the client is required to authenticate at every access point (a network port). Authorization requires time and consequently, if a handoff needs to be performed, packets will likely be dropped. Applications such as voice communications do not behave well under these conditions. If the wireless node has a control channel that is separate from its primary channel, it can initiate authorization with a neighboring AP before the actual handoff occurs on the primary channel. With the right design, this can result in zero-packet loss and low latency handoffs even when port-based security mechanisms are deployed in the wireless network.

There are other advantages to adding a secondary radio. In a system that has two radios, one of which is low power, it is possible to build an application-transparent power-aware communication system. Such a system can save power by using a low power radio for low data rate applications and the high power radio for high data rate applications.

9. CONCLUSION

Power consumption of battery-powered devices is a critical issue that requires major attention. In this paper, we have evaluated a concept, called wake-on-wireless, that addresses this issue. To show how this concept works, we have built a real system with a popular PDA and shown that its battery lifetime is significantly increased. While we have focused only on one specific aspect of power savings, namely idle power reduction, our architecture can more comprehensively solve the general power consumption problem.

Finally, while we have focused primarily on the battery lifetime gains achievable using wake-on-wireless, we do not necessarily advocate the implementation discussed here. Instead, we hope that this paper will compel wireless LAN equipment manufacturers to examine alternative ways to improve device longevity. Most current radios have focused on providing higher bandwidth at the cost of higher energy consumption. By adding a second, low-power channel of lower complexity and capability, lower energy consumption and longer lifetimes can be achieved. We believe that no existing single wireless technology can satisfy the need for high rate, low power, ubiquity, and high performance. In our opinion, a multi-radio solution is the most attractive approach to providing low-power, anytime, anywhere communication.

10. ACKNOWLEDGEMENTS

We would like to thank Atul Adya and Lili Qiu for their help in building our prototype application. We are also indebted to Geoffrey Nordlund and Jeff Chirico for helping with our experiments.

11. REFERENCES

[1] P. Bahl, A. Balachandran, A. Miu, W. Russell, G. Voelker, and Y.M. Wang. PAWNs: Satisfying the Need for Ubiquitous Connectivity and Location Services. *IEEE Wireless Communications*, 9(1):40–49, February 2002.

[2] A.P. Chandrakasan and R. Brodersen. Minimizing power consumption in digital CMOS circuits. *Proceedings of the IEEE*, 83(4):498–523, April 1995.

[3] Jae-Hwan Chang and Leandros Tassiulas. Energy Conserving Routing in Wireless Ad-hoc Networks. In *IEEE INFOCOM 2000*, pages 22–31, 2000.

[4] Benjie Chen et al. Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks. *ACM Wireless Networks*, 8(5), September 2002.

[5] Carla F. Chiasserini and Ramesh Rao. Combining Paging with Dynamic Power Management. In *IEEE INFOCOM 2001*, pages 12–19, April 2001.

[6] EMC. EMC's World Cellular Subscriber Forecasts. http://www.emc-database.com, October 2001.

[7] Laura Marie Feeney and Martin Nilsson. Investigating the Energy Consumption of a Wireless Network Interface in an Ad Hoc Networking Environment. In *IEEE INFOCOM 2001*, 2001.

[8] R. Gonzalez and M Horowitz. Energy dissipation in General Purpose Microprocessors. In *IEEE Journal of Solid State Circuits*, pages 1277–84, September 1996.

[9] Jose A. Guitierrez et al. IEEE 802.15.4: A Developing Standard for Low-Power Low-Cost Wireless Personal Area Networks. In *IEEE Network Magazine*, pages 12–19, September 2001.

[10] IEEE. IEEE 802.1x-2001 IEEE Standards for Local and Metropolitan Area Networks: Port-Based Network Access Control. 1999.

[11] IEEE802.11b/D3.0. Wireless LAN Medium Access Control (MAC) and Physical (PHY) Layer Specification: High Speed Physical Layer Extensions in the 2.4 GHz Band. 1999.

[12] Eun-Sun Jung and Nitin Vaidya. An Energy Efficient MAC Protocol for Wireless LANS. In *IEEE INFOCOM 2002*, June 2002.

[13] R. A. Powers. Batteries for Low Power Electronics. In *Proceedings of the IEEE*, pages 687–693, April 1995.

[14] J. Rabaey et al. PicoRadios for Wireless Sensor Networks: The Next Challenge in Ultra-Low-Power Design. In *Proceedings of the ISSCC*, February 2002.

[15] AMI Semiconductor. http://www.amis.com/pdf/ASTRX1_FS.pdf, July 2002.

[16] Tom Siep et al. Paving the way for Personal Area Network Standards: An Overview of the IEEE P802.15 Working Group for Wireless Personal Area Networks. In *IEEE Communications Magazine*, pages 37–43, February 2000.

[17] S. Singh and C. Raghavendra. PAMAS: Power Aware Multi-Access protocol with Signalling for Ad Hoc Networks. *ACM Computer Communications Review*, 28(3):5–26, July 1998.

[18] Henry Sinnreich and Alan B. Johnston. *Internet Communications Using SIP: Delivering VoIP and Multimedia Services with Session Initiation Protocol*. Wiley, 2001.

[19] Mark Stemm and Randy H. Katz. Measuring and Reducing Energy Consumption of Network Interfaces in Handheld Devices. In *IEICE Transactions on Fundamentals of Electronics, Communications, and Computer Science*, August 1997.

[20] H. Woesner et al. Power-saving mechanisms in emerging standards in wireless LANs: the MAC level perspectives. *IEEE Personal Communications*, 5(e):40–48, June 1998.

[21] Ya Xu, John S. Heidemann, and Deborah Estrin. Geography-informed energy conservation for Ad Hoc routing. In *ACM MOBICOM 2001*, pages 70–84, 2001.