Low-power sleep mode and out-of-band wake-up for indoor Access Points

Ivaylo Haratcherev[1], Michele Fiorito[1, 2], Carine Balageas[1]

[1] Alcatel-Lucent Bell Labs, Centre de Villarceaux, 91620 Nozay, France
[2] Eurécom, 06560 Sophia-Antipolis, France

The expected massive adoption of indoor Access Points requires specific solutions to meet requirements for eco-sustainability regarding consumed power and radio interferences. Since Access Points are unused the main part of the day, application of a sleep mode mechanism is a promising approach. However, its implementation represents a radical change in mobile networking paradigm and requires innovative mechanisms to wake up the Access Points and thus rapidly restore the connectivity.

Our approach uses an auxiliary low-power radio to carry out-of-band control information to maintain connectivity and wake up the Access Points when necessary. The paper details the software and hardware architecture as well as the prototyping results on Wi-Fi technology.

***Index Terms*— energy saving, low-power consumption, reverse paging, sensor networks, sleep mode, Wi-Fi**

I. INTRODUCTION

Within the past decade, global recognition of the need to protect the environment for future generations has emerged as a central issue that is now shaping behaviors in every industry. This study follows the trend to reduce the telecommunication networks footprint over energy consumption issues. Recent European directives [1] required to include eco-designed off-mode or standby-mode in the household and office equipment. By the end of 2012, the equipment will have to automatically switch to off/stand-by mode after a short period of inactivity. Furthermore, the energy consumption limits of those low-power modes are set to be challenging.

In this paper, we focus on indoor Wi-Fi Access Points in infrastructure mode and we implement a low-power sleep mode on actual equipment. Today, Wi-Fi Access Points (AP) are always consuming a relatively significant amount of energy as they have to periodically broadcast special beacons to signal their presence to the clients. This beaconing is mandatory in the 802.11 standard [8]. It means that the radio module is turned on as longer as the AP is on. Furthermore, even in idle state (i.e. mostly in receive mode except for beacons), AP consumes relatively high amount of energy. This is due to the low energy efficiency that is typical for the RF front end. Whenever there are no users under the coverage of an AP, at night for example, this can be seen as a waste of power, lack of security, and human exposition to useless radiation.

In this context, the technical problem addressed in this paper is how to save energy by switching entirely off the 802.11 interface of the AP, while still being able to serve a user request for a connection.

To tackle this issue, we use low-power properties of radios designed for sensor networks. In our case those radios carry out-of-band control information to maintain connectivity and wake up the AP when necessary. We have set-up a demonstrator of this low-consumption Wi-Fi system.

The rest of the paper is organized as follows. In Section II we will present the related work. Then in Section III and Section IV the design and the implementation of the hardware and software architectures of the low-power system are described respectively. The system performances are detailed in Section V giving energy saving amounts and wake-up delays results for our Wi-Fi prototype. The conclusions and future plans are discussed in the last Section (VI).

II. RELATED WORK

The low-power-wake-up concept has already been studied in the context of sensor networks [2][3][4] as the optimization of battery lifetime is a critical issue for sensor nodes.

For user mobile handsets, the similar idea of using a secondary out-of-band mechanism to remotely activate a user terminal using passive, low-power but short range RF ID tags was theoretically studied in [5]. The closest to our approach can be found in [6], where they use custom-designed low-power radio modules to wake up battery powered-devices like Personal Digital Assistants (PDAs). While using similar technique, the goal in that paper is completely different – saving power on mobile terminals. While important, we consider that goal well achievable by the standard means provided by the 802.11 standard, and thus, the higher price paid by introducing additional radio is not justified. On contrary, our method is aimed at power savings in Wi-Fi APs, which cannot be achieved without introducing the extra radio. Furthermore, we don't use custom made low-power radio module, but one that is well-established and that has huge software support. In this way, well known and tested complex MAC protocols for the low-power radio can be immediately put in place. Those protocols will be necessary for end-user industrial implementation of our approach.

For cellular access networks, Chiaraviglio et al. showed in [7] that a large amount of energy can be saved if an optimal dynamic radio coverage planning is used instead of a static one for an UMTS access network. In particular, they switched off some cells in urban areas during low-traffic periods. Unlike our paper, they deal with outdoor macro base station and provide analytical and simulated results, whereas we are focusing on indoor Wi-Fi Access Points. We have designed an actual implementation of a real system, using ultra-low-power radios traditionally employed in sensor networks.

III. OPERATION PRINCIPLES AND DESIGN

The basic idea of our solution is to add a second low-power radio link to wake-up or switch off the Access Point (AP).

With "waking up" we designate activating of the 802.11 radio at the AP – first electrically, and then logically.

The principle of the prototype is shown on Figure 1.

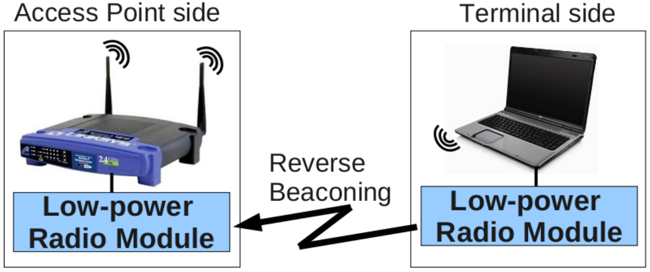

Fig. 1. Overall architecture of our proposal

At each side of the Wi-Fi link (Terminal and Access Point), there is a low-power radio module in addition to the Wi-Fi radio, with which extra radio out-of-band control information is carried out. This information is transmitted only from the terminal to the Access Point in a periodic fashion, further designated as "reverse beaconing". Note that the low-power radio operates at different frequency and uses different MAC protocol compared to the Wi-Fi.

At present we have two criteria for waking up the Access Point (AP) or for keeping it awake:

- "Reverse beaconing" concept: On the terminal side, every request by the kernel to send a packet over the Wi-Fi interface causes the low-power radio at the terminal to transmit periodic beacons to the AP to signal its presence. Whenever the low-power radio module connected to the AP receives those beacons, it checks identification information incorporated in them and wakes up the AP if the user can be authenticated. So, the AP is woken up every time a network connection is initiated by the terminal, and is kept awake as long as there is network activity registered at the terminal.
- Presence of active connections: On the AP side, the second keep-alive trigger is the presence of an active network connection belonging to the Wi-Fi interface. Normally this criterion is used only to maintain the AP on - it would not be used to wake up the AP once it is in sleep mode. The reason is that in most of the cases a Wi-Fi AP is also configured as NAT, so then the network connections are normally initiated only by the terminals, not by an external network node.

The AP turns off automatically when it does not hear anymore beacons sent by the low-power radio module at the terminal-side and when there is no more connection belonging to its Wi-Fi interface. Not all of the AP is turned off, only the Wi-Fi radio part, which consumes most of the energy supplied to the AP.

It should be noted that the criteria mentioned above are very flexible, and can be modified or augmented according to user needs or configuration without any problem (For example PC boot could trigger the terminal-side low-power radio module to perform a first handshaking procedure with the AP).

In the case of a commercial system and of an encrypted Wi-Fi connection (i.e. restricted access system) the low-power radio connection should be secured too. If not done, the system can be compromised by listening to the unencrypted messages sent over the low-power radio and by sending similar messages that will wake up the AP. However, the implications of such an attack are low: the AP can be kept awake by falsified messages, but this is the worst that can happen. So, the security of our Wi-Fi system will be the same as of an ordinary Wi-Fi system with the same security settings. Implementation of an encryption scheme for the low-power radio messages goes out of the scope of this paper. Encryption can be easily added later, and will not change the results discussed here.

IV. IMPLEMENTATION

We describe the components of the prototype below:

The Linksys WRT54G v3.1 has been chosen to be the Access Point in our testbed. The selection of this model has been based on decision criteria such as the capability to run open source firmware and the amount of available on-line documentation. Those criteria were established taking into account the need to modify the behavior of the device and to integrate the low-power radio module.

The terminal used in the simulations is the laptop Lenovo T400 Core2Duo P8400@2.26GHz, it incorporates an INTEL® Wi-Fi Link 5100 AGN wireless interface. Debian Linux runs on the terminal, and the driver used for the wireless card is IPW2200.

As low-power consumption module, we have chosen the TinyNode 184 [12] which is a state of the art ultra-low-power module that provides a simple and reliable way to add wireless communication to sensors, actuators, and controllers. It uses a Semtech SX1211 radio transceiver that can operate in the 868-870Mhz license-free ISM (Industry Scientific and Medical) frequency bands. This device fulfilled our requirements to have a transmission range similar to the coverage range of the Wi-Fi AP, and also permitted to have a wake-up signal that contains an identifier in order to avoid unnecessary wake-ups.

TABLE I
TINYNODE 184 – MAIN FEATURES

Power consumption	TX mode: 15/25 mA (0/10 dBm) RX mode: 3 mA Stand-by mode: 4 µA
Power supply	1.8 - 3.6 V
RF sensitivity	-104 dBm @ 25 kbps -100 dBm @ 66.7 kbps
Frequency range	868 ÷ 870 MHz
Transmission range	Outdoor 150 m @ 25 kbps +10dBm Indoor 50 m @ 25 kbps +10dBm
Microprocessor	Texas Instruments MSP430F2417
Interfaces	UART, LVTTL(3V) signal, SPI, IrDA supported, Molex 52465-3071
Firmware	TinyOS

Hardware

In order to shutdown the Wi-Fi interface on the AP, we exploit a feature of its Broadcom chipset. By issuing a special command the chipset can completely power down the Wi-Fi radio.

Figure 2 depicts our modification to the AP.

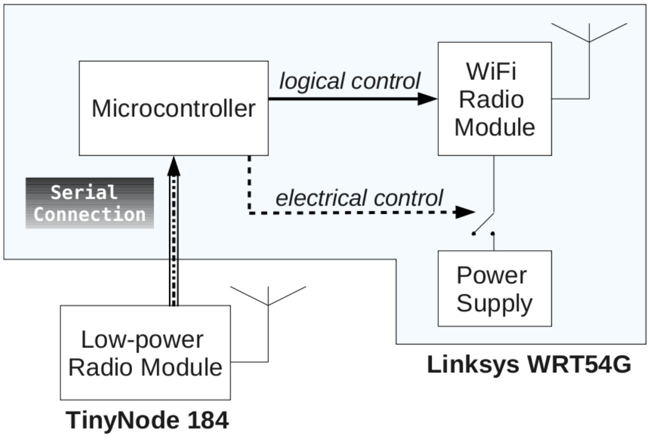

Fig. 2. Modifications to the Linksys WRT54G AP

We create a serial connection between the microcontroller and the low-power radio module. Indeed, the TinyNode has to communicate with the microcontroller of our Linksys AP in order to let it know whenever there is the need to turn on/off the Wi-Fi module. For that purpose we use the serial port that exists on this AP model. Since the electric signal levels are different on the AP and the TinyNode, we have put additionally a signal converter.

On the terminal side the TinyNode is placed on its Extension Board and is connected via a Serial-to-USB converter to the Lenovo laptop. At the machine, the Linux kernel sees logically the link to the low-power module as a serial connection.

Software

In order to implement the sleep mode and the wake-up process of the AP, we developed software modules for terminal, TinyNode low-power radio modules, and Access Point. Figure 3 summarizes the software end-to-end implementation of our reverse beaconing solution.

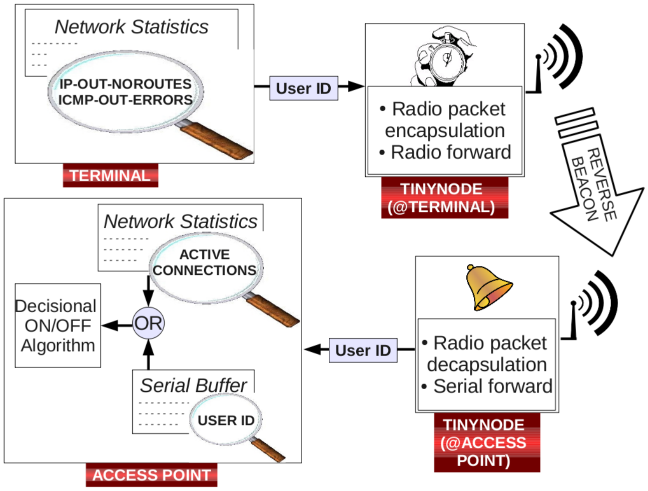

Fig. 3. Global interaction view of the software developed

The systems used in this simulation are Linux based and they share the open source philosophy.

The software running on the low-power modules has been developed under TinyOS in NesC language, please refer to [9] for further details.

Below follows a description of the software modules that were developed:

Terminal module

Whenever the user needs an Internet connection, the system has to wake up the access point it is associated with.

The user's applications that require an Internet connection, i.e. Browsers or VoIP clients, drive the OS to create IP packets. If there are no active Internet connections the kernel will generate other flags to signal that it has been problems while trying to send data over the net. This holds also when one launches a new application and there is no Internet connection.

Under a Debian environment, we have implemented an algorithm to signal the TinyNode about any packet that cannot be delivered by the kernel. Exploiting the virtual files which monitor the network status and track kernel events, our software transmits serial messages to the low-power radio when the kernel needs to send a packet over the 802.11 interface. More precisely, we use the IP-Out-No-Routes and Icmp-Out-Errors parameters which are obtained by executing the Nstat tool that allows the user to monitor kernel counters and statistics. At each error which generate one of the above alarm, the application will deliver a serial message containing user's identification information (labeled as User ID in Figure 3). Thanks to this mechanism, we do not have to care about synchronization issues among the kernel events, our terminal software, and the TinyNode application. Moreover, the IP-Out-No-Routes and Icmp-Out-Errors events are not activated by the kernel per single packet but they exploit a periodic timer. In this way, several errors are grouped in a single request.

”TinyNode at the Terminal” module

In this configuration the TinyNode sends reverse beacons in the case serial messages have been recently collected. After a periodic timer expiration, the software will check the serial buffer, pick the payload up and forward it on the radio interface. As mentioned above the serial message payload is labeled as User ID in Figure 3. TinyOS provides the Active Message (AM) layer to create AM radio packets which can be used to set a destination address and a data type. Since our purpose is to wake up a potentially unknown access point, we will deliver radio packets setting a broadcast address.

Figure 4 gives the packet format of the reverse beacon sent over the air.

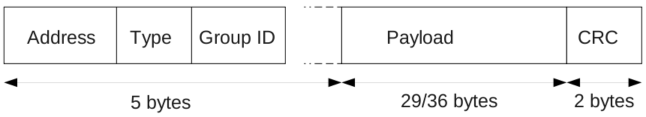

Fig. 4. AM radio packet format defined in the TinyOS AM Stack

”TinyNode at the Access Point” module

At the AP side, we exploit the event handler feature. Indeed, every time a radio message has been received, a task will deliver a serial message toward the AP. This can be seen as an asynchronous alarm thus, in Figure 3, we represented it as a ringing bell.

The procedure activated by this alarm will take out the radio packet payload, i.e. User ID, and forward it over the serial connection described in Figure 2 to the AP microcontroller. Serial messages are stream oriented, this means that we are not building a structured serial packet but we are sending streams of simple bytes.

Access Point module

We developed a module to handle signals coming from the TinyNode to the microcontroller and then drive the Wi-Fi interface. The software has been developed under OpenWrt [11]. Since it supports add-on packages, we have created a specific package which has to handle the auxiliary signals coming from the TinyNode, check the user activity, and consequently drive the Wi-Fi interface.

The algorithm keeps the AP’s Wi-Fi interface on if it receives serial messages from its TinyNode or if there are open connections with users under the Wi-Fi coverage. An open connection can be identified by exploiting the state machine within IP tables (see [10] for further readings). The application checks the virtual file in charge of tracking sessions: it filters the connections belonging to the Wi-Fi interface (defined under a different IP subnetwork). Those entries are related to a TTL (Time To Live): one can exploit this feature to give the user an inactivity margin. Therefore, even if the user is not using the 802.11 interface for a while, the software is not going to shut the AP down because of inactivity but it will wait a time period to confirm the inertia or change its evaluation.

If, after the inactivity margin there are no connections and no one is advertising its presence under the coverage, then the AP will shut down its Wi-Fi module. One can notice that this approach works properly in presence of one or more users under the coverage and also in a fast mobility scenario.

V. EXPERIMENTS AND RESULTS

The Figure below shows our measurement setup.

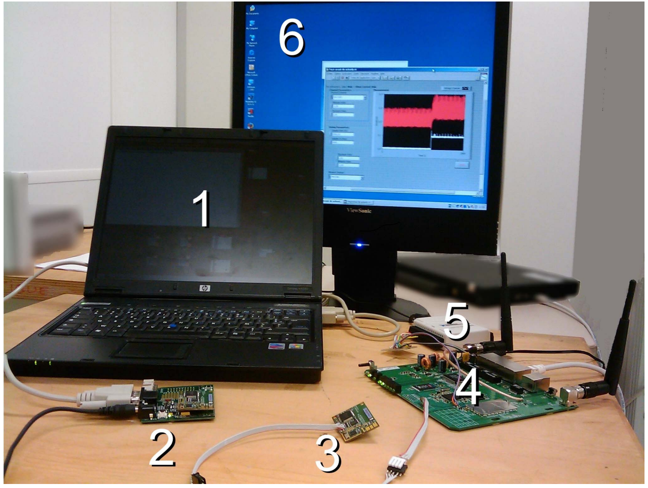

Fig. 5. Measurement setup

The terminal (1) is connected to a low-power radio module (2), which sends “reverse beacons” to another low-power radio module (3). The latter is connected to the Wi-Fi AP (4). The electric current consumed by the AP is measured by the acquisition module NI USB6008 (5). The measurement data is processed and logged by LabView on the acquisition station (6).

Fig. 6 depicts an overall schematic of an AP with emphasis on the electric energy distribution.

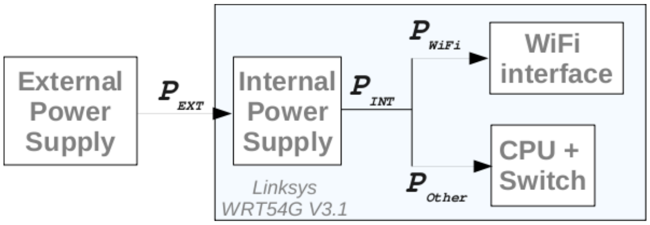

Fig. 6. Access Point logical block scheme presenting the system power flow

Power P_{EXT} is the one supplied by the external power adaptor to the AP and P_{INT} is the power provided by the internal power supply to the AP components. In Fig. 6 we also define P_{WiFi} and P_{Other}. The former is related to the energy consumed by the Wi-Fi RF chain and the baseband component, the latter is the power dissipated by the rest of the AP. When we shut down the Wi-Fi module, the total power consumed by the Access Point will decrease by P_{WiFi}.

Static experiments

In the first experiment we evaluate the average consumption P_{EXT} of the AP in the two cases: Wi-Fi radio on (P_{EXT_ON}), and Wi-Fi radio off (P_{EXT_OFF}). Here we measure the electric current with a multimeter attached between the external power adapter and the AP.

Table II summarizes the static results we have obtained.

TABLE II
LINKSYS WRT54G V3.1 – ELECTRICAL PARAMETERS

	Wi-Fi interface ON	Wi-Fi interface OFF
P_{EXT} [W]	2,94	2,1

We introduce a share factor S that represents the weight of the Wi-Fi power consumption in respect to the total power P_{INT} when the Wi-Fi interface is on.

$$S = P_{WiFi} / P_{INT\_ON} \qquad (1)$$

If we assume that the internal power supply unit of the AP has constant efficiency $\eta = P_{INT} / P_{EXT}$, we can then present S as a function only of P_{EXT_OFF} and P_{EXT_ON}:

$$P_{INT\_OFF} = P_{INT\_ON} - P_{WiFi},$$

so

$$S = (P_{INT\_ON} - P_{INT\_OFF}) / P_{INT\_ON} \qquad (2)$$

If we divide (2) by η, we get

$$S = (P_{EXT\_ON} - P_{EXT\_OFF}) / P_{EXT\_ON}$$

$$S = 1 - (P_{EXT\_OFF} / P_{EXT\_ON}) \qquad (3)$$

Now we can present on Fig. 7 the share of the Wi-Fi interface related to the whole (internal) Linksys WRT54G AP power consumption, by using (3).

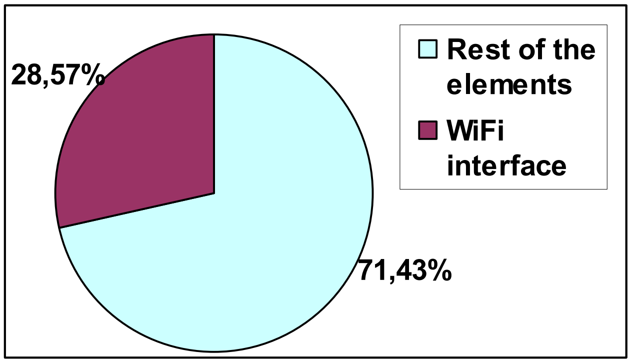

Fig. 7. Share representation of the power consumption in the Linksys WRT54G Wi-Fi Access Point

We introduce also the maximum power saving factor (Δ_{max}) for our setup that can be achieved in the hypothetical scenario when the AP has the Wi-Fi interface always switched off:

$$\Delta_{max} = (P_{EXT\_OFF} - P_{EXT\_ON}) / P_{EXT\_ON} \qquad (4)$$

from which it follows that

$$\Delta_{max} = -S \qquad (5)$$

So, it is clear from the data above that in the case of the WRT54G AP, the theoretical maximum reduction in the power consumption that can be achieved is -28.57 %.

Dynamic experiments

With the setup from Figure 5 we ran experiments to obtain traces of P_{EXT} over time. While measuring, the following usage pattern algorithm was followed:

1. Initial small period of inactivity
2. Using Internet (browsing) for about 1 minute
3. Period of inactivity for about 1.5 minute
4. If measurement time is not up, goto 2.

The measurement time was selected in a way that 3 or 4 full cycles of activity + inactivity were accomplished.

Table III shows an example of the time values related to important events for one cycle of the type described above.

TABLE III
SIMULATION TIMELINE

Event	Time [m:ss]
T0 (Start of measurement)	0:00
T1 (Open the browser)	0:19
T2 (Wi-Fi activated)	0:28
T3 (Close browser)	1:29
T4 (Wi-Fi shutdown)	2:02

On Figure 8 an important part of the power trace is shown for a period including T1 and T2. In particular, it emphasizes the power step and the response delay whenever a user needs to wake up the AP. From our simulation results we established that an average wake-up delay is around 10 seconds. In following improvements to our setup we will be able to lower this value since it mainly depends on the code developed for the terminal side.

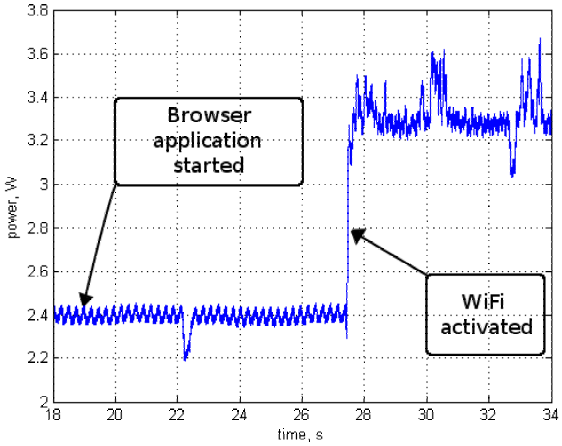

Fig. 8. Total power consumption (P_{EXT}) over time

After processing the power trace with MATLAB, we derived the following value for the maximum power saving factor Δ_{max} = -27.4 %. Comparing it with the value based on the static data obtained in the previous subsection, we see that the two values are coherent. This result confirms the validity of the assumptions made in the previous section.

Based on the activity pattern of the user, we can estimate what will be the real power saving (Δ) with our approach. If we assume that over a period of time t the user needs an Internet connection for time t_1, we can denote the usage factor (λ) as:

$$\lambda = t_1/t \quad (6)$$

With t_1 and t large enough, we can assume that during the period t_1 the AP consumption will be P_{EXT_ON}, while during the period $t-t_1$ the consumption will be P_{EXT_OFF}. Then, taking into account (4) and (6), we can derive:

$$\Delta = \Delta_{max}(1- \lambda) \quad (7)$$

So, if we take for example an average usage pattern of 4 hours of active Internet usage per day, the total real power savings based on the data from our measurements and (7) will be 22.9 %.

VI. CONCLUSIONS AND FUTURE WORK

Introducing a sleep mode into Access Points to fulfill eco-sustainability requirements, ensuring in the meanwhile a short wake-up delay, is a critical issue. In this paper, we have introduced a “reverse beaconing” concept, using ultra-low consumption properties of radios made for sensor network modules.

We have set-up a prototype with a popular Wi-Fi Access Point and measured the electrical consumption gain and wake-up delay.

The solution provides significant energy saving even with our particular prototype, where the AP has not been designed with energy-awareness in mind and the AP’s non-Wi-Fi components have a huge share of the overall power consumption. As shown in the previous Section, if the consumption of the CPU and other non-Wi-Fi components of the AP is minimized – a goal easy to achieve with the modern advances in electronics – the reduction in energy consumption of APs will be huge.

Other important advantage of this solution is the health argument, as there are no more unnecessary radiations when the equipment is not used actively. The security aspect is also a plus, as the AP is switched off when its usual user is away.

This solution has market perspective as we used standard 802.11 radio modules, with no need for standardization effort. The system components are simple and cheap. Access Point or Set Top Box manufacturers could easily insert this eco-designed functionality in their future products.

As future work we will continue on with perfecting different aspects of the prototype, such as minimizing the wake-up delay, and implementing encryption for the low-power radio communication. Furthermore, scenarios with multiple terminals will be investigated as well.

REFERENCES

[1]Commission Regulation (EC) No 1275/2008 of 17 December 2008 “Implementing Directive 2005/32/EC of the European Parliament and of the Council with regard to ecodesign requirements for standby and off mode electric power consumption of electrical and electronic household and office equipment”

[2]B. van der Doorn, W. Kavelaars and K. Langendoen, “A prototype low-cost wakeup radio for the 868 MHz band”, Int. J. Sensor Networks, Vol. 5, No. 1, pp.22-32, 2009G.

[3]J.L. da Silva Jr, J. Shamberger, M.J. Ammer, C. Guo, S. Li, R. Shah, T. Tuan, M. Sheets, J. M. Rabaey, B. Nikolic, A. Sangiovanni-Vincentelli and P. Wright “Design methodology for PicoRadio Networks”, Design, Automation and Test in Europe (DATE), Munich, Germany, pp.314-325, 2001.

[4]L. Gu and J. Stankovic, “Radio-triggered wake-up for wireless sensor networks”, Real Time Systems, Vol. 29, No. 2, pp.157-182, 2005.

[5]C. F. Chiasserini and R. Rao, “Combining Paging with Dynamic Power Management”, IEEE INFOCOM 2001, pp.12-19, April 2001.

[6]E. Shih, P. Bahl, and M. J. Sinclair, “Wake on wireless: An event driven energy saving strategy for battery operated devices,” Proceedings of the Eighth Annual ACM Conference on Mobile Computing and Networking, Altanta, Georgia, USA, Sept. 2004.

[7]L. Chiaraviglio, D. Ciullo, M. Meo, and M. A. Marsan “Energy-aware UMTS access networks”.WPMC’08

[8]IEEE 802.11-2007 IEEE standard for information technology - telecommunications and information exchange between systems - local and metropolitan area networks - specific requirement. Part 11: wireless LAN medium access control (MAC) and Physical layer (PHY) specifications.

[9]http://www.tinyos.net/

[10]O. Andreasson, “IPTables Tutorial 1.2.2”, Copyright © 2001-2006 Oskar Andreasson, GNU Free Documentation License.

[11]http://openwrt.org/

[12]H. Dubois-Ferriere, R. Meier, L. Fabre, and P. Metrailler. “TinyNode: A Comprehensive Platform for Wireless Sensor Network Applications”, Lausanne, Switzerland, 2006.