FZepel: RF-level Power Consumption Measurement (RF-PM) for Zigbee Wireless Sensor Network-Towards Cross Layer Optimization

Md. Rezaul Hoque Khan, Roberto Passerone, David Macii
DISI - Dipartimento di Ingegneria e Scienza dell'Informazione
University of Trento, Via Sommarive, I-38100 Povo, ITALY
E-mail: [khan, roberto.passerone, david.macii]@disi.unitn.it

Abstract

*Energy consumption is one of the most crucial design issues in wireless sensor networks (WSN) and largely depends on energy-efficient communication protocols. In order to improve the lifetime of a WSN, the cross-layer optimization technique can potentially be used to jointly optimize the power consumption behavior between various layers of the protocol stack. In this paper we propose a **F**inite State Machine (FSM) based **Z**igB**e**e **p**ower mod**el** (**FZepel**) that can be used for cross-layer stack analysis.* **FZepel** *is a primitive execution-based power estimation method. To achieve the desired level of accuracy, we also present an RF-level power measurement (RF-PM) technique for Zigbee-based wireless sensor networks. Using a PICDEM and a CC2420 communication component, we show how to characterize the state-based machine model of the network coordinator using the RF-PM technique.*

1. Introduction

Motivated by theoretical and practical challenges, wireless sensor networks (WSN) have drawn the attention of the research community in the last few years. This growing interest can be largely attributed to new applications, such as environmental monitoring, home assistance services for elderly people and wireless video surveillance systems. Because of small footprint requirements, wireless video sensor nodes can only be equipped with a limited power source (e.g., in the order of 0.5 Ah, 3 V) [1]. Therefore, sensor node lifetime becomes strongly dependent on battery lifetime. In an ad-hoc sensor network any malfunctioning of a few nodes can cause significant topological changes and might require re-routing and reorganization of the network. Hence, power conservation and power management become a challenging issue. For these reasons researchers are currently focusing on the design of power-aware protocols for sensor networks [1].

The *ZigBee* standard is targeted specifically towards WSN. The ZigBee protocol is implemented on top of the IEEE 802.15.4 radio communication standard. To be specific, ZigBee is oriented to applications with low requirements for data transmission rates and devices with constrained energy sources. The design goals for the WSN have been driven by the need for transmitting small control packets and sensor data and a desire to keep the cost of sensor network to a minimum. Additionally, the ZigBee network possesses self-organizing capability so that little or no network setup is required [1]. The ZigBee wireless technology enables long battery lifetime and offers the opportunity to build up complex wireless sensor networks [3].

However, ZigBee protocol stacks, which are architected and implemented in a layered manner, might not function efficiently in mobile wireless environments [6]. In the last few years, a new design paradigm has arisen in the field of wireless sensor network research: the so-called cross-layer optimization. Cross-layer feedback in the protocol stack would be useful to improve its efficiency [7]. In fact, this paradigm implies the redefinition of the overall design strategies for this kind of systems as it breaks the classical model [8]. To design an energy efficient protocol, a detailed energy model is a prerequisite analytical tool. However, the aforementioned research related to cross layer analysis on WSN addresses layer dependent optimization only for specific layers, while the overall energy efficiency of a system is not analyzed due to the lack of a proper model. In this paper we propose a Finite State Machine (FSM) based ZigBee power model (FZepel) that can be used for cross-layer stack analysis. FZepel is a primitive execution-based power estimation method. To achieve the desired level of accuracy, we also present an RF-level power measurement (RF-PM) technique for Zigbee-based wireless sensor networks. Using a PICDEM and a CC2420 communication component, we show how to characterize the state-based machine model of the network coordinator using the RF-PM technique. The remainder of the paper is organized as follows: Section 2 gives an overview of the Zigbee protocol, while Section 3 discusses the related work; Section 4 presents our FSM-based model. Then, we describe our characterization procedure by presenting the system components (Section 5), the energy consumption measurement framework (Section 6) and the measurement procedure (Section 7). Experimental results and conclusions are given in Section 8 and 9, respectively.

2. ZigBee Overview

ZigBee is best described by referring to the 7-layer OSI model for layered communication systems. A non-

1-4244-1506-3/08/$25.00

profit industry consortium of semiconductor manufacturers, technology providers and other companies, all together designated as the ZigBee Alliance, manages the ZigBee specification. The ZigBee specification is designed to utilize the features supported by IEEE 802.15.4 [2]. A comparison of prevalent wireless technologies is presented in Table 1.

	ZigBee	Bluetooth	Wi-Fi
Standard	802.15.4	802.15.1	802.11b
Memory requirements	4-32KB	250KB+	1MB+
Battery life	Years	Days	Hours
Node per master	65,000+	7	32
Data Rate	250Kb/s	1-3 Mb/s	11Mb/s
Range	300m	10-100 m	100m

Table 1: Comparison of wireless technologies.

ZigBee is supported to become the global control/sensor network standard. It has been designed to provide the following features:

- Low power consumption.
- Low cost (device, installation, maintenance).
- High density of nodes per network.
- Simple protocol, global implementation.

A ZigBee system consists of several components. The most basic is the device. A device can be a full-function device (FFD) or a reduced-function device (RFD). A network shall include at least one FFD, operating as the personal area network (PAN) coordinator. An RFD is intended for applications that are extremely simple and do not need to send large amounts of data. The FFD can operate in three modes: PAN coordinator, a coordinator or a device. An FFD can talk to RFDs or FFDs while an RFD can only talk to an FFD. The ZigBee functional layer architecture is shown in fig. 1.

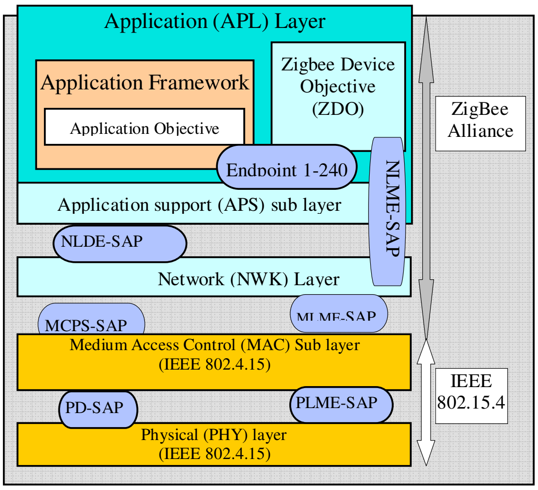

Figure 1. ZigBee functional layer architecture and protocol stack.

3. Related Work

Power consumption estimation models for wireless sensor network systems can be divided into two categories: measurement based techniques and simulation based techniques. Measurement based techniques use power measurement equipment while executing primitives on the target device. Simulation based methods use a simulator with an appropriate power model of the target device. An RF level power model measures power consumption while executing a program on the targeted RF board.

Our work is inspired by previous work in CPU profiling such as Morph [9]. The PowerScope, which is an updated tool for profiling energy usage by mobile applications, maps energy consumption to program structure [10]. Using PowerScope, it is possible to determine what fraction of the total energy consumed during a certain time period is due to specific processes in the system.

D. Shin et al. [11] present Seoul National University energy scanner (SES). SES estimates a target application's energy consumption in a hybrid fashion. SES is a highly integrated energy monitoring tool that collects power consumption data in a cycle-by-cycle resolution and associates the collected power data with C program and assembly language source code. SES directly *measures* the CPU core's energy consumption while it uses a memory-power model to *calculate* the memory system's energy consumption from collected memory traces in a cycle-by-cycle resolution. However, SES needs an extra profile acquisition module which consists of measurement circuit, profile controller and acquisition memory. Therefore, the techniques used in SES may not be applied to ordinary embedded systems which are not equipped with profile acquisition modules.

ePRO [12] is a tool for energy and performance profiling for embedded applications that can improve energy consumption and performance of a DSP application. Energy profiling is mainly based on the hardware instrumentation which measures current samples during execution of embedded applications with the support of a periodic kernel module. ePRO employs measure-based estimation techniques used in SES and PowerScope. However, ePRO is distinct from SES because ePRO does not need any extra hardware module such as profile acquisition module in SES. Therefore, ePRO can be used on ordinary embedded systems. The main advantage of ePRO over PowerScope lies in performance profiling. ePRO has a performance profiling module while PowerScope only profiles energy consumption.

Simple-Power [13], a cycle-accurate RT level energy estimation tool, uses transition sensitive energy models. Simple-Power also provides the energy consumed in the memory system and on-chip buses using analytical energy models.

Energy simulators such as Wattch [14] and SimplePower [13] estimate the energy consumption in reasonable time. However, their accuracy is not sufficient for an effective energy optimization.

Sensor network simulators, such as ATEMU [15], SensorSim [16], SENS [11], and TOSSIM [17] were developed for studying network protocols, but not for energy estimation. The ATEMU simulates the operation of sensors at the instruction level, while SensorSim, SENS, and TOSSIM simulate the operation of sensors in state transition instead of cycle-accurate simulation. The PowerTOSSIM [18] tool is based on TOSSIM, and is used to estimate the energy consumption of a sensor network. SensorSim, SENS, and PowerTOSSIM do not provide fine-grained power profiles since they are not based on an RF-level power model.

4. Contribution of this Work

Our proposed RF-level power measurement (RF-PM) for Zigbee wireless sensor network (**FZepel**) is a primitive-based power estimation method. In all previous traditional cross layer optimizations, hardware (RF module) power consumption is usually neglected and simply introduces a delay for a given routing, MAC, and PHY layer scheme. Though a number of papers emphasize energy/power efficient wireless sensor network protocols, a complete power model of the Zigbee wireless protocol is yet to be reported. The power model for wireless modules that we propose is based on finite state machines (FSM) and is a measurement–based estimation technique whose accuracy is much better than purely descriptive solutions. Appropriate sets of measured data are combined according to some predefined functions or rules in order to estimate the power consumption on each and every primitive. In this way a complete power consumption model can be established. We expect that a cross layer analysis based on this model would provide optimal results. The proposed measured procedure provides us with the power consumed while executing the individual primitives. This information can then be used to analyze the ZigBee stack in real applications. Also, the model can provide information on which operations and primitives are responsible for the highest energy consumption. We then could redefine the execution sequence and finally the whole protocol stack will be reestablished, namely cross layer protocol design.

We follow the methodology proposed by Macii et al. [19] to estimate the power consumption of Zigbee and propose a finite state machine (FSM) based ZigBee power model intended to analytically express the relationship between the sets of activities executed by ZigBee devices and the corresponding average current drain patterns. The reason for following Macii's approach is its accuracy vs. simplicity trade-off. The methodology is conceptually simple, yet the model produces estimates for the power consumption that differ at most by 5% from measured values for low duty–cycles.

4.1. Finite State Machine

Our model of the ZigBee protocol is a finite state machine (FSM) where each state corresponds to a particular operating condition. Transitions between the states are taken in response to the execution of actions or primitives that alter the state of the protocol. Several functions within the Zigbee stack require repeated calls before the function is completed. The FSM ensures that the program repeats a certain block of code within a state until such functions are complete, or jumps from state to state when a common state leads into several sub-states. The proposed FSM describing the transition of the states triggered by the primitives of a generic Zigbee device is shown in fig. 2. This FSM is split into a Zigbee coordinator (FFD) and a Zigbee node (RFD) side because a quite different amount of current is drawn by different macro states of the FSM. This is also measured by following experimental results. This proposed FSM is inherently a cross layer optimization approach because it considers the complete layer specifications and primitives. For example, while changing from Coordinator Response (CR) state to Active Coordinator (AC) state, the FSM incorporates NWK, MAC and PHY layer primitives like NLDE-DATA.request/ confirm/ .indication (NDT), MCPS-DATA.request / .confirm / .indication (MDT), PD-DATA.request/ .confirm / .indication (PDT). So this FSM model is suitable for a cross layer approach.

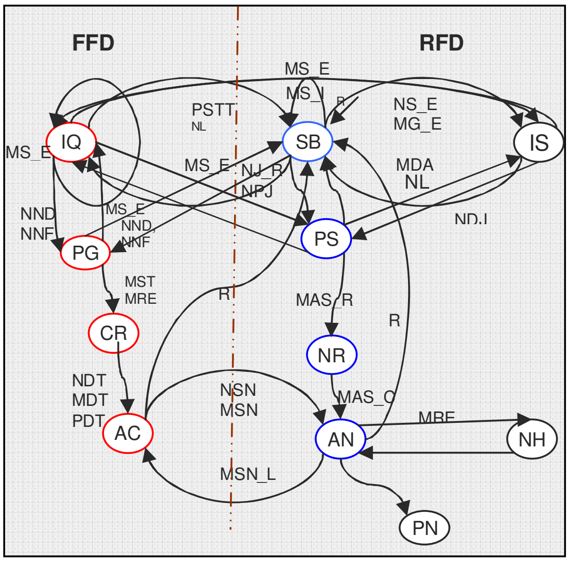

Figure 2. Finite State Machine (FSM) of the proposed Zigbee power model (FZepel).

From a functional point of view, the states of the FSM can be roughly grouped into five categories, i.e. *Disconnected states*, namely states in which either a node is inactive (*Standby* (SB)) or it performs a certain amount of activity to discover (*Inquiry* (IQ)) or to be discovered by other nearby devices (*Inquiry Scan* (IS)) *Transient states for connection*, i.e. transient states that

are used to establish and/or to configure a new link (*Page* (PG) or Page scan (PS).
In *Response states*, Coordinator Response (CR) or Node Response (NR) is used according to the connection request.
Active connected states, in which a node is fully active and able to perform data transfers between the RFD and a FFD of a wireless network over an established connection (i.e. Active Coordinator (AC), Active Node (AN)).
Low-power connected states. In Hold mode (*Hold node* (HN) and *park node* (PN)) data transfers are suspended for a fixed time interval, hence freeing channel capacity to do other operations like scanning, paging, inquiring or entering a deep sleep mode.

A short list of acronyms of the states is illustrated in table 2. To perform the characterizations we have developed a prototype system that consists of 2 PICDEM Z development boards wirelessly linked in a P2P topology. Boards are equipped with temperature sensors. One of the nodes functions as the ZigBee coordinator (FFD) and another as Zigbee node (RFD), and both are connected to a PC interface.

State Name	Primitives
Active Coordinator (AC)	NLDE-DATA.request MCPS-DATA.request MLME-SYNC-LOSS.indication(MSYNL_I) PD-DATA.request
Active Node (AN)	NLME-SYNC.request (NSYN) MLME-SYNC.request (MSYN) MLME-ASSOCIATE.confirm(MAS_C) MLME-POLL.request (MPL)
Coordinator Response (CR)	MLME-START.request (MST) MLME-RX-ENABLE.request (MRE)
Hold Node (HN)	MLME-RX-ENABLE.request (MRE)
Inquiry (IQ)	MLME-SCAN.request (MS_E)
Inquiry Scan (IS)	NLME-SET.request (NS_E) MLME-GET.request (MG_E) MLME-SCAN.request (MS_I) MLME-DISASSOCIATE.request (MDA)
Node Response(NR)	MLME-ASSOCIATE.request(MAS_R)
Page (PG)	NLME-NETWORK-DISCOVERY.request(NND) NLME-NETWORK-FORMATION.request NNF) MLME-SCAN.request (MS)
Page scan (PS)	NLME-JOIN.request (NJ_R) NLME-PERMIT-JOINING.request (NPJ) MLME-SCAN.request (MS_E)
Standby (SB)	NLME-RESET.request (R) NLME-LEAVE.request (NL) MLME-RESET.request (R) MLME-SCAN.request (MS_I) PLME-SET-TRX-STATE.request (PSTT)

Table 2. Acronyms of the states of FZepel.

We perform a general binding of coordinator with a node. The purpose of binding the coordinator to the node is to allow the coordinator to toggle an LED on the node. Based on our proposed FSM model, the transition of the states triggered by the primitives of a generic ZigBee coordinator device are shown in fig 3.

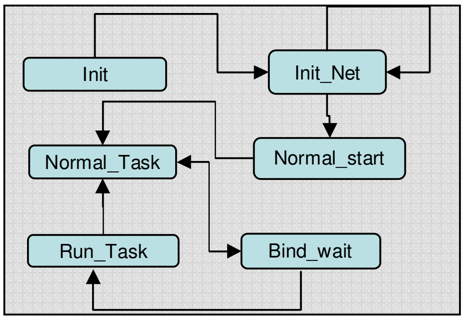

Figure 3. The coordinator's FSM.

With a physical switch (RB5 pin) of the µP, the interrupt is activated low and the coordinator enters the light toggle state. The coordinator initializes its timeout and number of operations before terminating the toggle request, upon entering the RUN_TASK state. While the coordinator performs these steps, the node continually polls the APL to see if a message has arrived from the coordinator. From this point, the node uses the toggle data sent from the coordinator and switches the LED pin, which is RA1 on the µP. At the same time, the node makes an internal MAC acknowledgment to the coordinator that the message is received. The coordinator awaits this MAC acknowledgment before exiting the RUN_TASK state and returning to the RUN_TASK state.

4.2. Zigbee Conceptual Model:

The RF primitives execution system developed for the RF system consists of a microcontroller PIC18F4620 and CC2420 Transceiver. The PICDEM Z board from Microchip is used to download the code onto the PIC microcontroller. The CC2420 starter kit from Microchip is used to convert digital data from the PIC microcontroller into RF format. Fig. 4 below shows the Zigbee RF measurement framework system with components and interface.

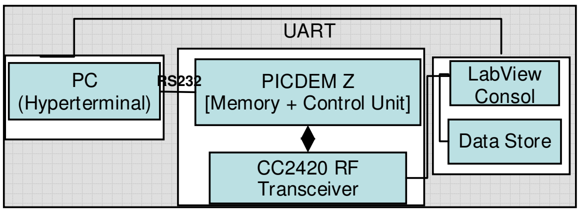

Figure 4. Conceptual Model of the Zigbee power measurement system.

As part of the measurement process, we connect the RS-232 connector on each test board to a serial port on a PC, and use the HyperTerminal serial interface software to communicate with the PICDEM Z board. We used MPLAB 8.0 and LabVIEW 8.0 to program this Zigbee power meter as simulator and virtual instrument.

4.3. Code Sequence

The complete code sequence for the measurement procedure includes initialization of several devices and state machine execution. The code execution steps are shown in fig 5. At the beginning of the measurement process all the hardware and devices connected with the main module are initialized. After that, the state machine will execute depending on the requested operations.

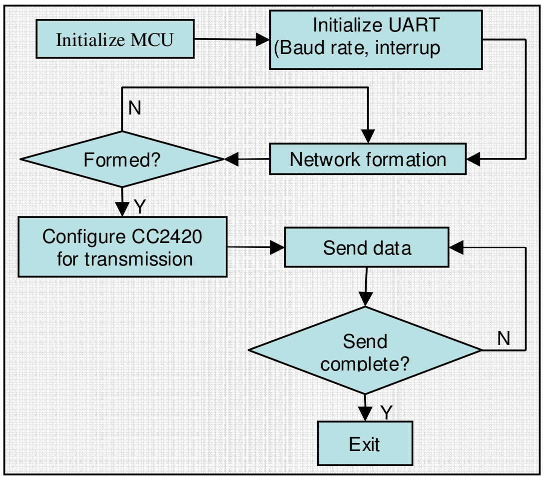

Figure 5. Flow chart explaining ZigBee primitives execution process.

5. Components of RF Zigbee Primitives Execution System

This section takes a closer look at the Microchip PIC18F4620, CC2420 Transceiver module and other components.

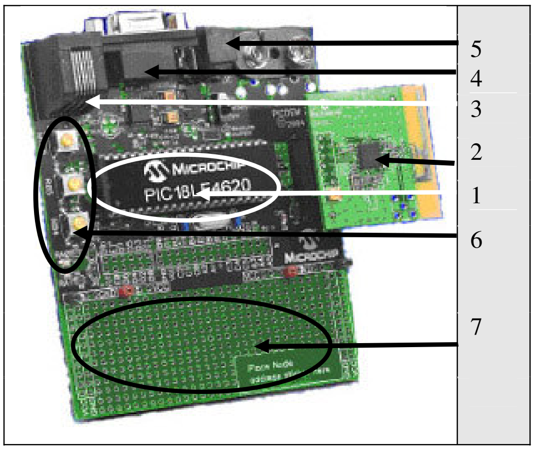

Figure 6. Microchip, PICDEM Z development board (1) PIC18F4620 (2) Chipcon CC2420 Microchip's Zigbee-capable uP (3) ICD2 debugger connector (4) 9-pin RS-232 serial port (5) AC adapter input jack (6)Switches (7) Breadboard.

5.1. PICDEM Z boards

PICDEM Z boards are comprised of several devices. Fig. 6 shows the PICDEM Z demonstration board that consists of a pre-built breadboard containing the PIC18F4620 Microchip's Zigbee-capable µP, a Chipcon CC2420 2.4GHz transceiver, the TC77 Thermal Silicon Sensor, LEDs and switches, and a 9-pin RS-232 serial port to connect to a PC via the HyperTerminal application.

5.2. PIC 18F4620

PIC is a member of Microchip's popular PICMicro (TM) line of microcontrollers. The PIC18F4620 is selected as microcontroller to develop the embedded RF Zigbee Wireless system because of the built-in USART support (RS 232) and A/D converter. Built-in USART and A/D converter modules execute in parallel with CPU, thus saving a lot of CPU time. Also, the PIC18F series has a large number of I/O ports which come handy for debugging and other applications. PIC-DEM2 is PIC development board from Microchip to debug and program PIC18F4620. PIC18F4620 MCU can also be seen in fig 7. A USB programming connector (ICD2 Header) is used to download C code onto the PIC MCU and a UART connector is used to access its UART pins.

5.3. PIC Microcontroller and C Program

The PIC18F4620 is a general purpose MCU from Microchip. It has several convenient built-in modules such as A/D, UART, etc., which reduce development time. In particular, an onchip Analog to Digital Converter (ADC) eliminates the need to design separate ADC circuitry. Also, an on-chip Universal Asynchronous Receive Transmit (UART) module makes asynchronous communication with the CC 2420 Transceiver Module relatively simple. ADC and UART modules operate in parallel with the microcontroller's CPU. Operation of A/D and UART modules in parallel with CPU saves CPU instruction cycles. In other words, once ADC and UART are configured properly, they operate almost independently with the CPU inside PIC Microcontroller.

5.5. CC 2420 Starter Kit

The **CC 2420** starter kit from Microchip is shown in Figure 7. This is the development board from Microchip on which the **CC 2420** Transceiver is soldered. The **CC 2420** Transceiver on this starter kit handles the wireless communication. This transceiver is the heart of the board. We will refer to this board as the RF board.

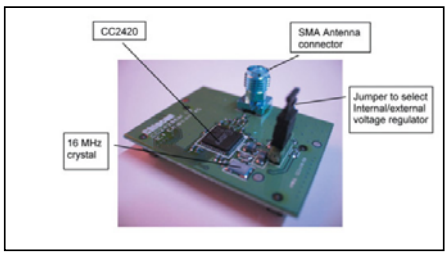

Figure 7. CC 2420 Starter Kit from Microchip[1].

[1] www.microchip.com

6. Energy Consumption Measurement Framework

Once a wireless connection is established with a proper configuration, data received by the evaluation board from the transmitting station is sent through a serial cable to the computer. The LabVIEW graphical user interface code (GUI) is used to display data collected from the sensor on the computer's screen. Since LabVIEW makes it easy to collect data and display data on graph, LabVIEW is used for developing the final user interface. The LabVIEW front panel on receiving side is shown in fig. 8.

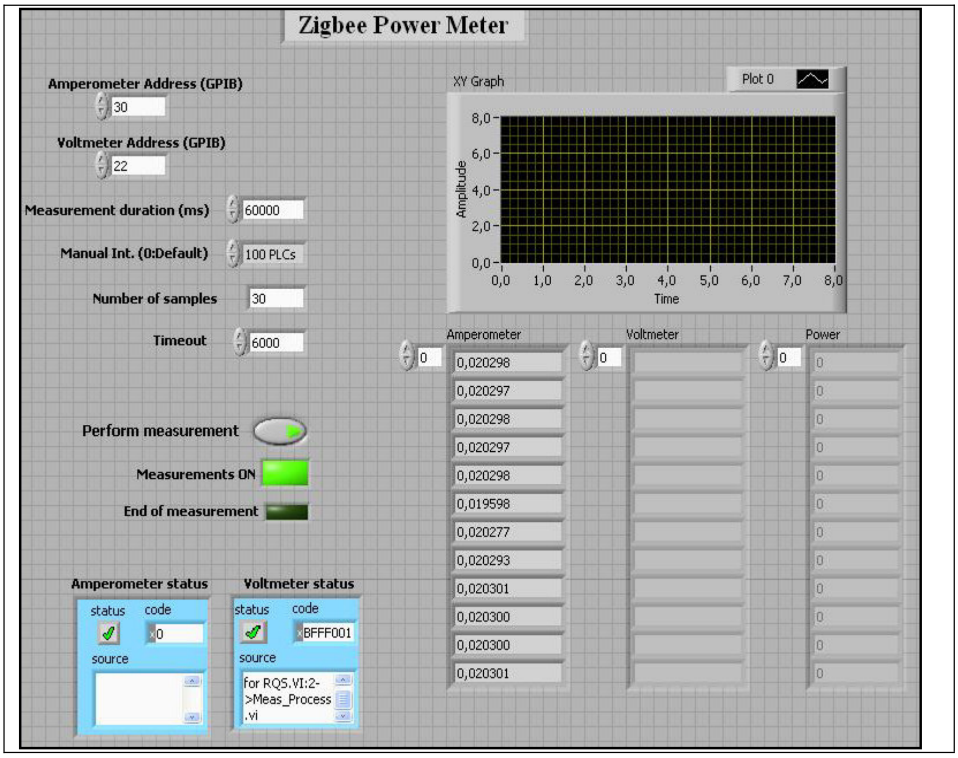

Figure 8. LabVIEW front panel.

The waveform graph is used to display a single set of samples received from the computer RS232 buffer. As soon as a complete message packet is received without errors, it is displayed on the waveform graph. The section "Serial Port settings" on the front panel allows the user to change the serial port number (COM1, COM2), baud rate, data bit size and parity bits. The port number is used to change the serial port which accesses received data. Baud rate is used to select the serial transmission baud rate. All the serial port settings are set by default to the parameters at which the CC 2420 board works so that the user need not worry about serial port settings. LabVIEW enables easy access to a computer's serial port through the use of built-in Virtual Instruments (VIs) [20]. VISA Configure Serial Port.vi initiates connection to a serial port. Baud Rate, Parity are set using VISA Configure Serial Port.vi. VISA Write.vi is used to write data to a serial port. VISA bytes from serial port.vi determine the number of bytes in the serial port buffer and VISA Read.vi data can be read from a serial port (shown in fig. 9). VISA Close.vi closes connection to a serial port.

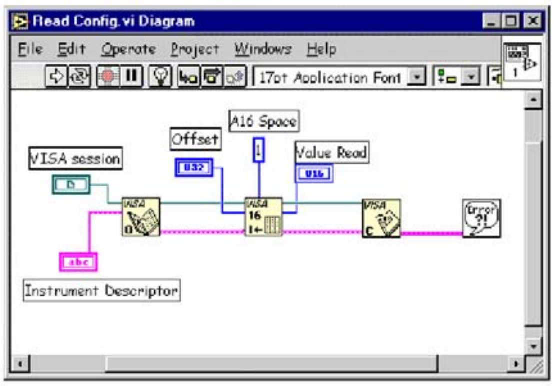

Figure 9. LabVIEW read.vi control diagram

If the serial port connection is not closed, memory leakage occurs. Remove Header.vi and Packet Decoder.vi separates data from the RF packet received through the serial port. Queues are used to buffer data received from the serial port until complete data packet is received into LabVIEW. Queues are used in

LabVIEW to synchronize data reception from the serial port and display on the front panel.

7. Measurement Procedure on Zigbee Power Meter

This section presents our proposed measurement procedure and steps

7.1. User-Interface (UI)

Our Zigbee system utilizes the RS-232 serial connection present on the PICDEM Z boards to interface with a PC running the Hyperterminal communications application. The connection is established with 19200 bits per second, 8 data bits, no parity, one stop bit and hardware flow control. This application is capable of running on the coordinator as well as the node, however the coordinator's interface is the main focus for the UI since the node should not be connected to a PC during runtime. The hardware structure is shown in fig. 10.

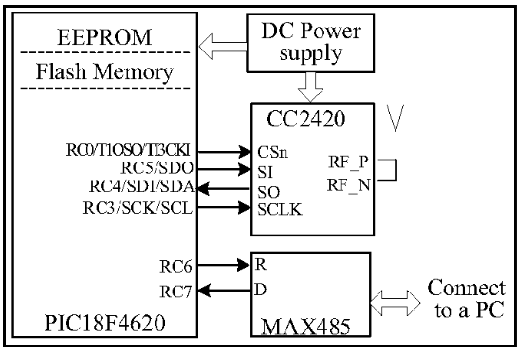

Figure 10. Hardware structure of coordinator node.

7.2. Initialization

Two initializations must occur for the µP to run properly. Board initialization is the first step, which occurs immediately after power-on to set the directions and functions of the ports and registers for the pins on the PIC. It also includes obtaining the board's built in media access control (MAC) address for networking. This is followed by the stack initialization and formation of a new network. To initialize the stack, the coordinator must call the following main stack functions:

- APLInit, which initializes the stack,
- APLEnable, which enables the Zigbee stack to operate,
- APLNetworkInit, which initializes the network,
- APLTask, which performs all of the network tasks,
- APLNetworkForm, which establishes the network and
- APLPermitAssociation, which allows nodes to join the network.

Once initialization has occurred and a network is formed, the coordinator cycles through its RUN_TASK state (figure 3) which polls for user input. The node cycles in and out of a sleep state function between periodic polls for any commands from the coordinator.

7.3. Binding Procedure

Before being able to send and receive messages from the node, the coordinator and the node must perform endpoint binding. Endpoint binding allows any two boards to bind switches, lights or other ports to one another to enable control over these devices or to establish a message link between the boards. This process is dynamic as opposed to the hard-coded static custom binding, and is also Zigbee specification compliant, meaning that other Zigbee enabled devices may potentially join the network, provided that binding is possible between the devices.

Binding implementation within the prototype occurs in the Bind Wait state in the main FSM. Any of the two boards may initiate a binding request, which involves pushing a physical switch on the board. The switch is connected to the RB4 pin on the PIC µP and throws an interrupt on the processor. The processor services the interrupt by identifying the binding switch being pressed and proceeds to enter the binding state in the FSM. In the same manner, the opposite board must respond within a given amount of time to respond to the request. For the coordinator, the node must respond within six seconds before a timeout occurs. On the other side, the coordinator must respond within six iterations of the node's binding state before a timeout occurs. This feature is due to the node going to sleep and being unable to respond to any bind requests during its sleep mode.

8. Some Experiment Results

We have performed experimental tests using the proposed method to measure the average current drawn by the radio chip CC2420, thus estimating the average power (under the assumption that the voltage supply is constant) [21]. Measures have been carried out with an oscilloscope (Agilent TDS 220) and shown in table 3.

Service	Primitives	Power consumed (mW)
Network formation	NLME_NETWORK_FORMATION_confirm	0.130
Network Join	NLME_JOIN_indication	0.022
Device binding	NLME_PERMIT_JOINING_confirm	0.029
Message Send	APSDE_DATA_indication	0.019
Leave network	NLME_LEAVE_indication	0.025
Network Reset	NLME_RESET_confirm	0.034

Table 3: Power consumption for different service primitives.

By referring the proposed **FZepel** (figure 2) it is possible to show that a different amount of power will be drawn at different states of the operation mode. For instance, the page (PG) and page scan (PS) state have been analyzed by measuring the average power values dissipated by our Zigbee module PICDEM Z board with CC2420. All measurements are performed automatically using our proposed **FZepel.** The power consumption associated with above state configuration is:

$$\mathbf{W_{PG}} = [0.279] \text{ mW for standard PG;}$$
$$\mathbf{W_{PS}} = [0.071] \text{ mW for standard PS;}$$

Moreover, all the transitions between states are considered to be instantaneous, i.e., they dissipate no power. In fact, the maximum transition times are always several orders of magnitude smaller than the time usually spent in any of the states of the model.

We believe the models used by the sensor network community currently do not provide this level of insight, but instead only provide an overall system power consumption estimation idea.

8. Conclusion

In this paper, we investigated a cross layer energy consumption model with primitives measured RF characteristic and described the modeling of Zigbee power consumption. This model takes into account energy dissipation during the execution of primitives. Following the methodology we would achieve a detail power consumption scenario of the Zigbee node. In our approach, the traditional Zigbee model is broken down by the power consumption profiling. We concentrate our focus more on state services rather than on individual layer services. For a specific state service, potentially several layer services would execute, thus achieving cross layer interaction. The detail mapping of the measured parameters provides valuable hints for cross layer protocol analysis for optimized performance. Thus, a detail energy expenditure analysis of different layers provides a basic for developing an energy efficient wireless sensor network protocol.

Reference

[1]. Ian F. Akyildiz et al., "Wireless sensor networks: a survey", Computer Networks, Elsevier Science, Volume 38, Issue 4, 15 March 2002, Pages 393-422.

[2]. www.Zigbee.org.

[3]. P. Valdastri et al., "An implantable Zigbee ready telemetric platform for in vivo monitoring of physiological parameters" Sensors and Actuators A: Physical, Elsevier Science 2007.

[4]. G Pekhteryev et al., "Image Transmission over IEEE 802.15.4 and Zigbee Networks" IEEE International Symposium on Circuits and Systems (ISCAS), Vol. 4, pp 3539-3542, May 2005.

[5]. W. C. Jakes, Microwave mobile communications, Ed., New York, John Wiley and Sons, 1974.

[6]. Vijay T. Raisinghani et al., "Cross-layer design optimizations in wireless protocol stacks", Computer Communications**,** Elsevier Science Volume 27, Issue 8, May 2004, Pages 720-724.

[7]. G. Xylomenos, G.C. Polyzos, "Internet protocol performance over networks with wireless links", IEEE Network 13 (4) (1999) pp 55–63.

[8]. Luis Alonso and Ramón Agustí, "Optimization of wireless communication systems using cross-layer information", Signal Processing, Volume 86, Issue 8, August 2006, Pages 1755-1772.

[9]. X. Zhang, Z. Wang, N. Gloy, J. B. Chen, and M. D. Smith. System Support for Automated Profiling and Optimization. In *Proceedings of the 16th ACM Symposium on Operating Systems and Principles*, Saint-Malo, France, October 1997.

[10]. J. Flinn et al., "PowerScope: A Tool for Profiling the Energy Usage of Mobile Applications," *Proc. the Second IEEE Workshop on Mobile Computer Systems and Applications*, Feb. 1999, pp.2-10.

[11]. S. Sundresh, W.-Y. Kim, and G. Agha. "SENS: A Sensor, Environment, and Network Simulator," *Proc. 37th Annual Simulation Symposium,* 2004, pp.221-228.

[12]. W.K. Baek, Y.J. Kim, and J.H. Kim, "ePRO: A Tool for Energy and Performance Profiling for Embedded Applications," *Proc. Of International SoC Design Conference* (*ISOCC*), Seoul, Korea, Oct. 2004, pp. 372-375.

[13]. W. Ye et al., "The Design and Use of SimplePower: A Cycle Accurate Energy Estimation Tool," *Proc. 37th Design Automation Conference* (*DAC*), 2000, pp.340-345.

[14]. D. Brooks et al., "Wattch: A Framework for Architectural-Level Power Analysis and Optimizations," *Proc. International Symposium on Computer Architecture* (*ISCA*), 2000, pp.83-94.

[15]. J. Polley et al., "ATEMU: A Fine-Grained Sensor Network Simulator," *Proc. First IEEE Int'l Conference on Sensor and Ad Hoc Communication Networks* (*SECON*)*,* Santa Clara, CA, Oct. 2004, pp.145-152.

[16]. S. Park, A. Savvides, and M.B. Srivastava. "SensorSim: A Simulation Framework for Sensor Networks," *Proc. MSWIM*, Boston, MA, Aug. 2000, pp.104-111.

[17]. P. Levis, N. Lee, M. Welsh, and D. Culler. "TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications," *Proc. the First ACM Conf. Embedded Networked Sensor Systems* (*SenSys*), Nov. 2003, pp.126-137.

[18]. V. Shnayder, M. Hempstead, B.R. Chen, G. Werner-Allen, and M. Welsh, "Simulating the Power Consumption of Large Scale Sensor Network Applications," *Proc. the Second ACM Conference on Embedded Networked Sensor Systems* (*SenSys*), Baltimore, MD, Nov. 2004, pp.188-200.

[19]. D. Macii,et al., "A Measurement-Based Power Consumption Estimation Technique for Bluetooth Modules", I2MTC 2008 - International Instrumentation and Measurement Technology Conference, Vancouver Island, Canada, May 12–15, 2008.

[20]. http://www.ni.com/support/visa/vintro.pdf

[21]. D. Macii, D. Petri, "An Effective Power Consumption Measurement Procedure for Bluetooth Wireless Modules," *IEEE Transactions on Instrumentation and Measurement*, Vol. 56, pp. 1355-1364, Aug. 2007.