A Realistic MAC and Energy Model for 802.15.4

Vishwesh Rege
Robert Bosch Centre for Cyber Physical Systems
Indian Institute of Science (IISc)
Bangalore, India
vishwesh.rege@cps.iisc.ernet.in

Tommaso Pecorella
Dpt. of Information Engineering (DINFO)
Università di Firenze
Firenze, Italy
tommaso.pecorella@unifi.it

ABSTRACT

The IEEE 802.15.4 standard defines the physical and media access control layers for LR-WPANs (Low-Rate Wireless Personal Area Networks), and is one of the enabling technologies for Wireless Sensor Networks (WSNs) as well as the emerging Internet of Things (IoT) applications. The ns-3 network simulator offers support for simulating LR-WPANs as specified by the IEEE standard 802.15.4 (2006). However, only the ad-hoc mode is currently supported and many important features of the MAC such as radio duty cycle management are missing from the implementation.

Moreover, at the moment ns-3 does not support simulating the energy consumption of LR-WPAN devices. Since energy efficiency is an important consideration for WSN and IoT applications, support for accurate energy modeling is highly desirable in order to develop energy-aware protocols for such applications. In this paper, we present the models developed for simulating the energy consumption of nodes in LR-WPANs. Further we implement the ContikiMAC radio duty cycling protocol in order to provide a realistic 802.15.4 compliant MAC layer which supports sleep/wake mechanisms.

CCS Concepts

• **Computing methodologies** → **Model development and analysis;** Discrete-event simulation; **Networks** → *Network simulations*

Keywords

ns-3, energy consumption, IEEE 802.15.4, IoT, WSN, MAC, duty cycling

1. INTRODUCTION

Wireless sensor networks (WSNs) consist of a large number of battery-powered wireless sensor nodes required to operate for years without any human intervention. The Internet of Things (IoT) is the future of wireless sensor network applications where any device can be connected to and operated through the Internet. Since the wireless network nodes in WSN and IoT applications are typically battery-powered, they have a limited amount of energy available. As a result, efficient utilization of the available energy is essential for continuous operation over long periods of time. Wireless sensor network research attempts to design network protocols to meet these energy constraints and increase network lifetime.

IEEE 802.15.4 is an important standard for wireless sensor network and Internet of Things applications. The standard specifies the physical layer and media access control for low-rate wireless personal area networks (LR-WPANs), and is the underlying protocol used in a majority of sensor network deployments with applications such as remote sensing, surveillance and monitoring. The standard provides an energy efficient and cost effective option for low latency and high accuracy communication required in these scenarios, with the ability to survive on battery or harvested power for extended periods of time.

A number of custom and industry standard networking protocols have been developed using the services provided by 802.15.4. It is the basis for standards like ZigBee, WirelessHART, and the IPv6 adaptation protocol 6LoWPAN [4] which further extend it by specifying the upper layers which are not defined. When combined with 6LoWPAN, it can be used alongside the standard Internet protocols to build a wireless embedded Internet and facilitate the Internet of Things vision.

The 802.15.4 standard is the basis for the LRWPAN model in ns-3. The model allows simulation of WSN scenarios, though it is severely limited in its current capabilities. There is no support for simulating the energy consumption of LRWPAN nodes in the model. Since, energy efficiency is one of the major concerns for researchers looking to test their WSN protocols, this is a serious limitation that needs to be addressed.

Furthermore, the MAC (Media Access Control) layer doesn't incorporate sleep/wake mechanisms. As a result, the radio is always ON, which doesn't represent the actual deployment situation. Most WSN nodes when deployed, use some sort of radio duty cycling mechanism in order to turn off the radio when not in use. As the radio is the main source of energy consumption on a node, this is one of the primary methods used to increase energy efficiency and consequently the network lifetime. Lack of radio duty cycling therefore prevents users from running realistic simulations. On one side this increases the performance (delay, throughput, etc.), and on the other side overestimates the energy consumption for receiving packets and underestimates the transmission energy.

A simulator model that supports modeling energy consumption of 802.15.4 devices will allow studying the energy consumption of nodes and thus the energy efficiency and network lifetime of 802.15.4 based protocols. A MAC model that incorporates radio duty cycling will allow researchers to test their network protocols under realistic conditions. Thus the developed models are expected to greatly simplify the future development and evaluation of protocols for WSN/IoT applications.

In this paper, we present our contributions to ns-3 which are as follows:

- PHY layer support for modeling energy consumption of 802.15.4 radios
- LRWPAN device energy model that represents the energy consumption of 802.15.4 devices
- A MAC model that incorporates radio duty cycling, in order to realistically simulate WSN scenarios in ns-3

The rest of the paper is organized as follows: Section 2 gives an overview of the IEEE 802.15.4 standard and the existing implementation in ns-3. Section 3 presents the in detail design and implementation of the proposed ns-3 energy and MAC models. Section 4 includes the validation of the developed models. Section 5 discusses possible future work and Section 6 concludes the paper.

2. IEEE 802.15.4 AND LRWPAN SIMULATION MODEL IN NS-3

In this section, we give a brief overview of IEEE 802.15.4 and introduce the relevant features of the standard. We also discuss the LRWPAN simulation model in ns-3 which is based on the standard.

2.1 PHY Layer

The 802.15.4 PHY layer provides two services:

1. The PHY data service, which enables the transmission and reception of PHY protocol data units (PPDUs) across the physical medium.
2. The PHY management service interfacing to the physical layer management entity (PLME) service access point (SAP) (known as PLME-SAP).

The features of the 802.15.4 PHY are activation and deactivation of the radio transceiver, transmission and reception of packets across the wireless channel, as well as performing additional tasks that may be required by the upper layers, such as Energy Detection (ED), Clear Channel Assessment (CCA) and Link Quality Indicator (LQI) measurement for received packets. ED and CCA operations are required in the Carrier Sense Multiple Access Collision Avoidance (CSMA-CA) functionality of the MAC layer.

In ns-3, the Phy model is based on the specification described in section 6 of IEEE Std 802.15.4-2006. It models the PHY service specifications, PPDU formats, PHY constants and PIB attributes.

2.2 MAC Layer

The medium access control (MAC) layer enables transmission of frames through the physical channel.

The 802.15.4 MAC layer provides two services:

1. The MAC data service, which enables the transmission and reception of MAC protocol data units (MPDUs) across the PHY data service.
2. The MAC management service interfacing to the MAC layer management entity (MLME) service access point (SAP) (known as MLME-SAP).

The features of the 802.15.4 MAC layer are beacon management, channel access, GTS management, frame validation, acknowledged frame delivery, association, and disassociation.

The MAC model in ns-3 is based on the specification described in section 7 of IEEE Std 802.15.4-2006. The standard allows three different MAC modes: beacon-enabled, non beacon-enabled (star and cluster tree modes) and beaconless (ad-hoc mode). However, the current MAC implementation lacks some features foreseen by the standard. In particular, there is no active support for coordinators, association, disassociation and beacon management, required for beacon-enabled and non beacon-enabled modes of operation. Only channel access (using unslotted CSMA/CA) and the basic data transfer API along with acknowledged frame delivery is currently implemented. Thus, the only mode supported is ad-hoc mode (i.e., beaconless).

Further, the current ns-3 MAC does not use low power features, forcing the radio always in an ON state. As a result, it is impossible to simulate realistic scenarios in which the nodes are assumed to be duty-cycled spending most of the time with their transceivers switched off. It must be stressed that the standard specifies only the sleep management for beacon-enabled MAC, while for the beaconless case the standard does not give any strict rule.

Due to the lack of beacons and centralized timing in the beaconless case, the main challenge is to achieve a local synchronization between two different nodes, because the local clocks drift does not allow to keep the synchronization if not for small periods. As a consequence, it is not guaranteed to know when the receiver node will be awake.

In order to save energy in beaconless mode, many alternative 802.15.4 compliant MAC protocols such as S-MAC, B-MAC, X-MAC, WiseMAC [9] have been proposed in the literature. Any of these duty cycling schemes can be used in optimizing the radio power consumption as required by the application. In our proposed model, we implement ContikiMAC [10], which is the default radio duty cycling mechanism in the Contiki OS [11].

2.3 ns-3 Energy Framework

The current ns-3 simulator (since ns-3.9) provides a basic framework for modeling energy consumption in wireless networks [5]. The model consists of 2 major components:

1. The Energy source, and
2. The Device energy model

The Energy Source represents the energy supply of a node in the simulation, while the Device energy model is used to represent the energy consumption characteristics of a specific device, such as an 802.11 radio on a node.

The energy framework focuses on modeling radio energy consumption because the radio is assumed to consume the most power in a wireless node. The framework also assumes a state-based model, i.e., the radio is assumed to be in one of several states

(Receive, Transmit, Idle, etc.) with a corresponding load current associated with each state. The energy model and energy source attached to each node keep track of the time spent in a particular state and the total energy consumed by the node respectively.

The most difficult part in using the energy model is to define the device states and the transitions, in order to mimic the real device energy characteristics.

3. DESIGN AND IMPLEMENTATION

3.1 LRWPAN Energy Model

The LRWPAN energy model extends the basic energy framework in order to model energy consumption of LRWPAN nodes. As noted in the previous section, it follows a state based approach, modeling the nodes' energy consumption as a function of the states of the radio transceiver. We base our model on the AT86RF231 [14], which is a typical 802.15.4 compliant radio transceiver.
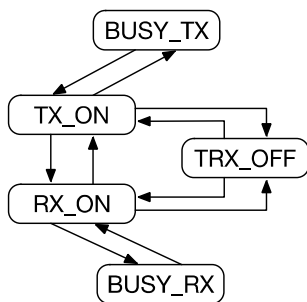
Figure 1: State transition diagram.

According to the model, the radio can be in one of the following states:

- **TRX_OFF** - Transceiver disabled
- **TX_ON** - Transmitter enabled
- **RX_ON** - Receiver enabled
- **BUSY_TX** - Transmitting
- **BUSY_RX** – Receiving

However, there is no difference between the (TX/RX) ON and corresponding (TX/RX) BUSY states with respect to the radio transceiver circuitry. The energy consumed in these states is approximately the same. Hence, the node's energy consumption can be modeled as a function of only three states TRX_OFF, TX_ON and RX_ON.

Most transceiver chips also have an additional shutdown (sleep in case of AT89RF231) state in which the chip is completely

deactivated. This state can only be entered through external interrupts and isn't modeled. The rest of the states can be entered by sending appropriate commands to the PHY layer through the PLME. BUSY_RX and BUSY_TX are entered automatically during frame reception and transmission respectively. Figure 1 illustrates the corresponding radio transceiver states and the allowed transitions between them. All the transitions are triggered by MAC events, except for the BUSY states, which correspond to actual PHY layer events (e.g. preamble detection causes the transition from RX_ON to BUSY_RX).

A PHY Listener is registered with the LrWpan PHY in order to notify concerned objects of every PHY state transition.

Energy update algorithm:
The PHY, on each successful transition notifies the Energy model through the PHY Listener. Correspondingly, the LrWpan-RadioEnergyModel notifies the EnergySource object to update its energy. The EnergySource object queries the energy model for the current draw of the state and uses that to calculate the energy consumed using the formula:

$$stateDuration \times stateCurrent \times supplyVoltage$$

where, *stateDuration* represents the time spent in the state, *stateCurrent* represents the state's current draw, and *supplyVoltage* is the node's attached energy source's supply voltage.

The energy source is then updated with the new value of remaining energy. When the energy is completely depleted, the LrWpanRadioEnergyModel is informed by the EnergySource, using the EnergyDepletionCallback (defined in LrWpanRadio-EnergyModel).

The node's attached energy source is also updated periodically to keep track of the energy consumed even when there are no radio state transitions.

In most cases, the state transitions in radio transceivers aren't immediate. There is a finite time difference between the PHY receiving a request to change the state, and then issuing a confirm primitive indicating that the state change is accepted. This transceiver switching time not only affects the MAC operations, but also has a significant impact on the total energy consumption as a result of the very low radio duty cycle in wireless sensor networks [7]. Hence, it is important to also take into account the transition time and energy between the states. In the Three States Model described, six different transitions are possible:

- TRX_OFF -> TX_ON and vice versa.
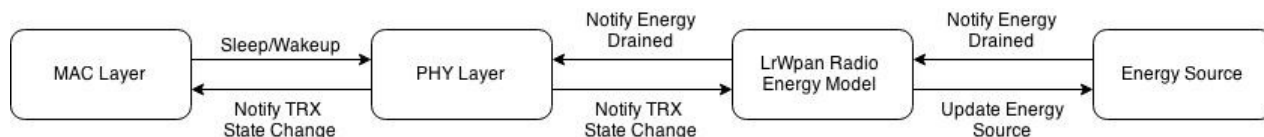- TRX_OFF -> RX_ON and vice versa.
- TX_ON -> RX_ON and vice versa.

Figure 2: Energy model flow diagram.

Table 1: State transition timing.

Transition	Transition Time
TRX_OFF → TX_ON	110 us
TX_ON → TRX_OFF	Immediate
TRX_OFF → RX_ON	110 us
RX_ON → TRX_OFF	Immediate
TX_ON → RX_ON	8/12 symbol periods* (128/192 us)
RX_ON → TX_ON	8/12 symbol periods* (128/192 us)

* defined by the standard

The transition times have been modeled according to the values given in the datasheet. Table 1 shows the transition timings of the AT89RF231 radio transceiver.

When a state change is requested by the MAC through the PLME, the state machine is put in the transition state. On receiving confirmation from the PHY that the target state is operative, the state machine is finally updated with the new state. The state transition energy is calculated by multiplying the transition time by the power in the target state. This is a worst-case assumption since the transition energy has been shown to depend on the power in the state the transceiver is switching *to*, multiplied by a parametric constant (less than 1) [8].

As noted before, the current *LrWpanEnergyModel* is modeled based on Atmel's AT89RF231 transceiver. However, other 802.15.4 compliant transceivers such as TI CC2420 [15] can be simulated by changing the current draw to appropriate values. For the CC2420, TRX_OFF, TX_ON and RX_ON correspond to the chip's Idle, Transmit and Receive states respectively.

The PHY enumeration values defined in *LrWpanPhyEnumeration* are used to represent the different states of the radio transceiver, with IEEE_802_15_4_PHY_UNSPECIFIED being used for the TRANSITION state.

3.2 LRWPAN ContikiMAC Model

ContikiMAC is the default duty cycling mechanism in the Contiki OS. It is based on the concept of periodic wake-ups in which nodes sleep most of the time and wake up at regular intervals to check for radio activity. It uses an asynchronous sender-initiated radio duty cycling mechanism, i.e. it doesn't depend on any a priori synchronization between nodes, and the sender initiates communication by repeatedly sending the same packet until a link layer acknowledgment is received. It is simple to implement, since no signaling messages or additional packet headers are used. All

ContikiMAC packets are ordinary link layer messages already being used by the existing 802.15.4 standard.

The primary features of ContikiMAC can be summarized as follows:

- *Periodic sleep/wake mechanism* defined by the channel check rate.
- *Stable wake-up interval* unaffected by node's radio activity.
- *Two consecutive CCAs* after wake-up to detect packet - stay awake to receive the packet if energy is detected on the channel, else sleep if channel is found idle.
- *Link layer acknowledgment* followed by sleep on packet reception (transmitter repeatedly sends the frame until an ACK is received, after which it too returns to sleep).

- *Broadcast*: Continuous transmission of the data packet by the transmitter for full length of sleep period – since the transmission must wake up and deliver the packet to all the node's neighbors.
- *Fast sleep optimization* – To optimize the power at the receiver side by letting potential receivers go to sleep earlier if the CCA caused a node to stay awake due to spurious radio noise instead of an actual packet transmission.
- *Transmission Phase-Lock* - sender side optimization to reduce the power at the transmitter by minimizing the number of retransmissions.

Note that the retransmissions by the ContikiMAC Radio Duty Cycle (RDC) is independent of the MAC layer retransmissions which refers to the number of times a frame is given to the RDC layer for transmission.

The MAC and RDC mechanisms in the ContikiMAC model could be split into two separate classes, like in Contiki OS. However, in order to support multiple MAC protocols and reuse the code, the ns-3 implementation follows a different approach.

The first choice has been to split the MAC functionalities into a base, abstract class (LrWpanMac), implementing only the APIs defined by the 802.15.4 standard and some common elements. The particular MAC behavior is implemented in two subclasses: LrWpanNullMac and LrWpanContikiMac. LrWpanNullMac implements the current behavior, as is the always-on radio with no sleep or duty cycle. The LrWpanContikiMac is responsible for duty cycle, sleep wake, and node synchronization. All ContikiMAC features, with the exception of the Fast sleep optimization, have been implemented and tested in the proposed model. Both MAC layers use the LrWpanCsmaCa module to detect any activity on the channel.

It should be remarked that the new modular architecture enables an easy integration of new MAC protocols. This is an important feature, as the future IoT systems could be hindered by the use of 802.15.4-compliant (but mutually incompatible) protocols. As a matter of fact, even if two protocols are formally following the standard, different radio duty cycle management policies can make them very inefficient, leading to high energy consumption.

The LrWpanContikiMac API is identical to LrWpanMac. Additionally, the user can specify attributes such as the sleep interval, the interval between the two CCAs, the interval between each packet retransmission, and the maximum number of retries by the RDC.

4. VALIDATION AND RESULTS

In this section, validation results of the proposed LRWPAN energy and MAC models are presented. First, the energy consumed by the LRWPAN energy model is compared against expected values obtained through manual calculations. Next, in order to evaluate the ContikiMAC model, the energy consumption values of simulated nodes are compared with those obtained from running the same scenario in the Cooja simulator.

Energy model evaluation

In order to verify the correctness of the energy model, a simple scenario involving 2 nodes is considered. Node 1 sends a packet of size 50 bytes to Node 2 and receives an acknowledgement. The

simulation is run for 10s with both the nodes having their radios turned on during the first 5s and turned off for the next 5s.

The time spent by both the nodes in each state is shown in Table 2. The currents in TX, RX and OFF states are taken to be 19.5mA, 21.8mA and 1800μA respectively. For a supply voltage of 3.3V, the energy consumption comes out to be 0.3894 Joules for Node 1 and 0.3895 Joules for Node 2 which agrees with the simulation results.

Table 2: State timings for simulation scenario.

States	Node 1 (sender)	Node 2 (receiver)
TX_ON/BUSY_TX	0.002400	0.000544
*to TX_ON**	0.000192	0.000192
RX_ON/BUSY_RX	4.997106	4.998962
*to RX_ON**	0.000302	0.000302
TRX_OFF	5.000000	5.000000
*to TRX_OFF**	0.000000	0.000000

* from any arbitrary state

ns-3 vs. Cooja Comparison

Cooja [12] is a sensor network simulator for the Contiki OS. The Cooja energy estimation mechanism consists of keeping track of the time the radio spends in the different transceiver states and multiplying these times with corresponding power levels to obtain a rough estimate of the energy consumed.

Cooja also separately keeps track of the CPU power consumption. Apart from the regular CPU mode used during communication, the nodes can also operate in a Low Power Mode (LPM) during periods of inactivity. The total consumed energy is then calculated as the sum of the total time spent in all the states – similar to how the energy is calculated in our developed model in ns-3. The state transition energy however isn't taken into account. Thus the energy consumed by a node at the end of the simulation is given by:

$E = P_{rx}T_{rx} + P_{tx}T_{tx} + P_{slp}T_{slp} + P_{lpm}T_{lpm} = I_{rx}V_{rx}T_{rx} + I_{tx}V_{tx}T_{tx} + I_{slp}V_{slp}T_{slp} + I_{lpm}V_{lpm}T_{lpm}$

where *rx*, *tx*, *slp* and *lpm* refer to the transmit, receive, sleep and LPM states respectively.

The amount of time spent in each state can be tracked using the Powertrace [13] tool provided with Cooja. Powertrace outputs four values: *cpu*, *lpm*, *transmit*, and *listen*. These values correspond to the time spent in each of the four states. *transmit* and *listen* correspond to the "transmitter enabled" and "receiver enabled" states in our model respectively. The *cpu* state includes the time spent in *transmit* and *listen* states as well as the "transceiver off" state. Therefore, the time when TRX is off, but CPU is in its normal mode is given by *cpu* – (*transmit* + *listen*). *lpm* refers to the state when TRX is off and the node is in its low power mode. Since, our model only takes into account energy consumed by the transceiver, we consider *cpu* with transceiver off and *lpm* as one state by using the same current draw for both states.

For the comparison, we consider a two node scenario with Node 1 sending a packet to Node 2 periodically every 2.5 seconds over a 60 second interval. The channel check rate is set to be 8 Hz, which corresponds to a sleep interval of 125 ms. The energy consumption of both the sender and receiver nodes using the original ContikiMAC implementation and the ns-3 ContikiMAC

implementation is compared in Table 3, while Table 4 compares the respective NullMAC implementations.

Table 3: ContikiMAC energy comparison (in Joules).

Protocol	Contiki OS ContikiMAC	ns-3 ContikiMAC
Sender	0.47945	0.44100
Receiver	0.37694	0.38750

Table 4: NullMAC energy comparison (in Joules).

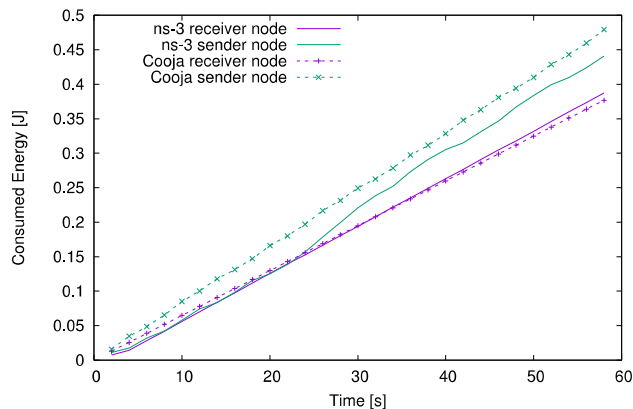| Protocol	Contiki OS NullMAC	ns-3 NullMAC
Sender	4.04262	4.28037
Receiver	4.17223	4.28071

Figure 3: ContikiMAC energy comparison.

In Figure 3 it is evident that the energy consumed by the receiver nodes in both the implementations is in close agreement, while there is a slight disagreement in the energy consumption values of the sender nodes. This can be attributed to the difference in time both the nodes spend in transmit mode which varies due to the CSMA-CA random backoff algorithm.

5. FUTURE WORK

In this section, we discuss possible extensions to the proposed models.

Local Clocks Drift Simulation

One major improvement toward the model precision is to be done as part of a different ns-3 development: per-node local clocks. LR-WPAN energy efficiency is highly dependent on how nodes can keep (or lose) a mutual synchronization. But ns-3 clock is global, preventing a real evaluation of clock drift effects.

CPU Energy Consumption

Currently, the LRWPAN energy model only considers energy consumed by the radio transceiver and doesn't take into account the energy consumption of the CPU. As described in [5], states representing computation tasks available on the node can be used to incorporate CPU energy consumption information into the simulations.

Another possible development goal is to upgrade the current model to 802.15.4-2011 standard.

6. CONCLUSION

In this paper, we presented an energy model for 802.15.4 radio transceivers, which will enable users to measure the energy consumption of 802.15.4 nodes in the network and thus allow them to develop energy-efficient protocols for WSNs using this information. We also developed a modular and easily extensible MAC model and implemented the ContikiMAC radio duty cycling protocol in order to enable realistic simulations of WSN scenarios. The code is currently under review to be merged into ns-3.

7. ACKNOWLEDGMENT

8. REFERENCES

[1] GSOC2015LrWpanMac. 2015. http://www.nsnam.org/wiki/GSOC2015LrWpanMac

[2] ns-3 lr-wpan. http://www.nsnam.org/wiki/index.php/Lr-wpan

[3] IEEE Std 802.15.4-2006, Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs).

[4] J. Hui and P. Thubert. Compression format for IPv6 datagrams over IEEE 802.15.4-Based Networks. RFC6282, IETF Sep. 2011.

[5] H. Wu, S. Nabar, and R. Poovendran. 2011. An energy framework for the network simulator 3 (ns-3). In Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques (SIMUTools '11). Brussels, Belgium, March 2011.

[6] A. Dunkels, F. Osterlind, N. Tsiftes, and Z. He. 2007. Software-based on-line energy estimation for sensor nodes. In Proceedings of the 4th workshop on Embedded networked sensors (EmNets '07). New York, USA, June 2007.

[7] B. Bougard, F. Catthoor, D.C. Daly, A. Chandrakasan, W. Dehaene. 2005. Energy efficiency of the IEEE 802.15.4 standard in dense wireless microsensor networks: modeling and improvement perspectives. In Proceedings of the conference on Design, Automation, and Test in Europe (DATE '05). Munich, Germany, March 2005.

[8] P. Hurni, B. Nyffenegger, T. Braun, and A. Hergenroeder. 2011. On the accuracy of software-based energy estimation techniques. In Proceedings of the 8th European conference on Wireless sensor networks (EWSN'11). Bonn, Germany, February 2011.

[9] A. Bachir, M. Dohler, T. Watteyne, and K. K. Leung. 2010. MAC Essentials for Wireless Sensor Networks. *Commun. Surveys Tuts.*

[10] A. Dunkels. The ContikiMAC radio duty cycling protocol. Technical Report T2011:13 , Swedish Institute of Computer Science Decembers 2011.

[11] A. Dunkels, B. Gronvall, and T. Voigt. 2004. Contiki - A Lightweight and Flexible Operating System for Tiny Networked Sensors. In Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks (LCN '04). Tampa, USA, November 2004.

[12] F. Osterlind. A sensor network simulator for the Contiki OS. Swedish Institute of Computer Science (SICS), Tech. Rep. T2006-05, Feb. 2006.

[13] A. Dunkels, J. Eriksson, N. Finne, and N. Tsiftes. Powertrace: Network-level power profiling for low-power wireless networks. Technical Report T2011:05, Swedish Institute of Computer Science, Mar. 2011.

[14] AT86RF231 Datasheet. http://www.atmel.com/images/doc8111.pdf

[15] CC2420 Datasheet. http://www.ti.com/lit/ds/symlink/cc2420.pdf