New RF Models of the TinyOS Simulator for IEEE 802.15.4 Standard

Changsu Suh, Jung-Eun Joung and Young-Bae Ko†
R & D Departments, Hanback Electronics Company, Daejeon, Republic of Korea
†College of Information & Communication, Ajou University, Suwon, Republic of Korea
{scs, jin}@hanback.co.kr, youngko@ajou.ac.kr†

Abstract— **Recently, wireless sensor networks have gained increasing attention from the industry as well as academia. Various research issues related with sensor networks are intensively proposed, and they are evaluated by some network simulators or real sensor platforms. One of the well-known simulators for wireless sensor networks is called TOSSIM. It can simulate with TinyOS source codes on the real testbed without any significant modifications. Although TOSSIM's architecture and interfaces are well designed for wireless sensor networks based on IEEE 802.15.4 standards, its current RF model is too simple to support main features of the PHY stack of the IEEE 802.15.4. In order to enhance the accuracy of wireless simulation results and implement IEEE 802.15.4 standard, we design a new wireless propagation model and RF physical stack based on the two-ray ground path loss model and CC2420 RF transceiver. Our work contributes on the performance evaluation areas of wireless sensor networks and IEEE 802.15.4 WPAN standard using simulations.**

I. Introduction

Advancements of the wireless communication and micro electronic technologies have enabled the development of smart sensors which have various functions like sensing, processing and networking. A wireless sensor network consists of a number of smart sensors, each of which has a limited battery life and short-range radio communication. The wireless sensor networks can be well-suited for variety of ubiquitous applications such as detecting fire, animal habitat monitoring, detecting enemy zones, smart home and human healthcare etc. There have been numerous researches [1] done in developing efficient hardware and software platforms for the smart sensors. However, performance evaluations on the real sensor boards are not easy and sometimes financially infeasible. Alternative way to study sensor networks through simulations can provide more comfortable developing environments and researchers can easily analyze the behavior of algorithms between numerous sensor nodes. Various simulation tools have been suggested and developed for this purpose. One of the popular simulation tools for wireless communication is *ns2* [2]. Its architecture consists of two parts: operation core part (C++ language) and configuration part (Tcl language). Many protocols designed for wireless ad-hoc protocols (such as AODV, DSR, and 802.11) and sensor networks (such as Diffusion and S-MAC) have been implemented using the ns2 simulator package. SensorSim [3], Qualnet [4] and Opnet [5] are also generally utilized for an evaluation of wireless sensor networks. However, these simulators have a common problem that their implemented source codes are specific to their simulator only and are infeasible to reuse it directly on the embedded sensor boards. Therefore, implementating source codes on the simulators are limited for its utilization. Another problem is that they mostly focus on the network stacks and wireless communication only, despite a good simulator for embedded sensor nodes should include data communication as well as operating system overheads. The process scheduling, sensor controlling, data processing and interrupts handling should be influenced to performances of the sensor networks simulations.

Comparing with other simulators, TOSSIM [6] has solutions in aspects of the above problems. It can simulate with real source codes on the sensor boards and it is performed with operating system overheads. TOSSIM is built on the TinyOS [7] which is the most popular operating system for the sensor networks and runs on various custom hardware platforms [8, 9]. TinyOS provides the PC based tool-chain which can directly change the source codes on the sensor platforms to TOSSIM source codes without any modification. However, the wireless RF model design of TOSSIM is very simple and does not have any wireless propagation characteristics. Moreover, its physical stack was designed based on the old RF transceiver (CC1000 [10]) which does not support 802.15.4 standard.

In this paper, in order to improve accuracy of wireless simulation results of TOSSIM, we design a new wireless propagation model based on the two-ray ground path loss model [13] and implement the new wireless physical stack based on the CC2420 transceiver[1] for IEEE 802.15.4. Since our RF models are affected by a transmission power, distance, antenna gain and height, reality of simulation results can be increased. Additionally, the new physical stack can help to implement or research the IEEE 802.15.4 standard on the TOSSIM. By using our proposed RF models, we can implement IEEE 802.15.4 full-standard and we believe this is the major contribution of the paper. The rest of paper is organized into four sections. In the section 2, we describe the TinyOS and TOSSIM in more detail. Section 3 introduces the new wireless propagation model and new physical stack. Section 4 presents various performance evaluations of our proposed RF models and implemented 802.15.4 standard. Finally, we conclude this paper and describe future works in section 5.

[1] Most of the recent sensor hardware platforms utilize the CC2420 [11] as the RF transceiver because it can support the physical characteristics of IEEE 802.15.4 standard [12].

†This work was in part supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Advancement), (IITA-2006-C1090-0602-0011) and (IITA-2006-C1090-0603-0015).

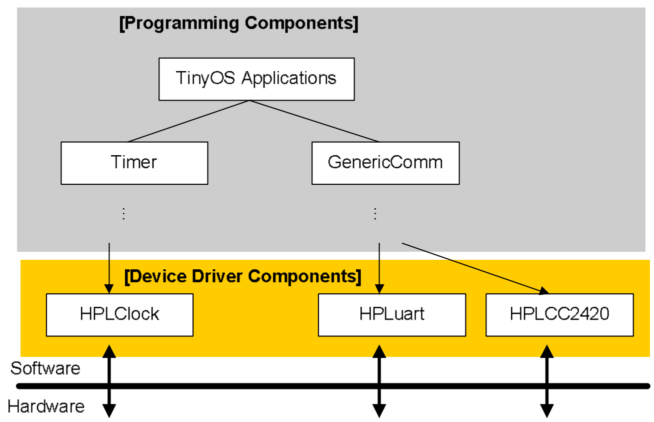


Fig.1. Two component types in the TinyOS

II. Background

A. TinyOS

TinyOS is the platform specifically designed for the wireless sensor networks. Its architecture is based on components and lightweight threads. TinyOS is different from other traditional embedded operating systems. It is not separated to a kernel or user levels, and does not support a dynamic memory allocation mechanism in order to reduce its code sizes. For optimizing an operating system upon the embedded sensor hardware platform, it builds a set of components that can be utilized for its specific application. Any component in the TinyOS represents an independent computational entity, and generally consists of several lower-level components. Through the component architecture, TinyOS can abstract the complex hardware controlling or a communication protocol to the application developers. A component has three interrelated parts: commands, events and tasks. In general, a command is used to perform some operations to control hardware, and an event is a function to be called when received signals or interrupts from hardware platforms. A task is often utilized to computing functions without any hardware control.

Figure 1 shows two component types of TinyOS, i.e., *a programming component* and *a device driver component*. The programming component, independently performing with hardware platforms, includes a number of sub-components such as the one for checking a timer, for controlling sensors and for toggling LED as well as the one of wireless routing and MAC protocols. The device driver component becomes different according to hardware. Application developers are required to know the interface functions of device driver components to perform hardware chips. These two type component models allow TinyOS to easily work on various target platforms. Hardware developers can easily install the TinyOS on their platforms by modifying the device driver components only.

B. TOSSIM

TOSSIM is the sensor network simulator coupling with TinyOS. By using TOSSIM, users can easily analyze operations and interactions among more than hundreds of sensor nodes. TOSSIM is executed based on the event-driven model and its simulator clock unit is 4MHz, an instruction clock cycle of Mica [8]. Any interrupts and signals are stored into the event queue of TOSSIM and managed based on the global clock unit. To avoid artificial synchronization, each node is randomly started at the simulation initial step.

As mentioned previously, one of the merits of TOSSIM is that the whole structures and system source codes for the real sensor platforms can be directly utilized to simulation. In the pervious subsection, we described two type component models (i.e., programming and device driver components) in TinyOS. This separation can allow source codes to be easily implemented to various hardware platforms by modifying a small number of device driver components. This feature can also be applied to the TOSSIM simulator. TOSSIM replaces the device driver components as simulation components for virtually performing numerous sensor nodes in PC. As the interfaces of simulation components are same with device driver components, other programming components can be performed transparent on the TOSSIM. Therefore, users can easily test and debug their applications, routing and MAC protocols before porting to real sensor platforms. TinyOS uses NesC language [14] for a component based programming. It utilizes the nesC compiler (NCC) for compilation of their source codes to hardware platforms or simulation.

III. New RF Models for TOSSIM

A. Wireless Propagation Model

In the traditional RF model of TOSSIM, a wireless link among sensor nodes is represented by a link connection probability. Thus, each link connectivity can be specified by some probability with a random value that is picked up by a programmer. However, because such a simple model cannot express wireless characteristics properly, it is not enough to simulate the wireless communication algorithm. To get more valid results with TOSSIM, a wireless model should be designed considering the wireless characteristics such as a path loss, a transmission power, a receive sensitivity and an antenna gain. To enhance the accuracy of simulation results, we designed a new wireless propagation model based on the two-ray ground path loss model [13]. Our proposed wireless model can catch the receiving signal strength based on the transmitted signal power and a distance between a sender and a receiver. Two-ray ground path loss equation is following:

$$P_r = \frac{P_t \times G_t \times G_r \times h_t \times h_r}{d^4 \times L} \qquad (1)$$

Table 1 lists variables used in the equation (1). By substituting values in the equation (1), we can calculate the receiving signal strength at each receiver node. If we define the receiving signal strength threshold (*RX_Threshold*), each node can know whether a wireless link is connected or not. If a receiving signal strength at some node i is bigger than its *RX_threshold*, the node can successfully receive a data packet from a sender. That is, the equation can be expressed as:

$$P_r(i) > RX\_Threshold(i) \qquad (2)$$

TABLE I VARIABLES USED IN THE EQUATION (1)

Parameters	Descriptions	Default Value
G_t	*Transmitter antenna gain*	1.0
G_r	*Receiver antenna gain*	1.0
h_t	*Transmitter antenna height*	0.3
h_r	*Receiver antenna height*	0.3
L	*System loss factor*	1.0 (no loss)

Algorithm 1: The algorithm for deciding wireless link connectivity.

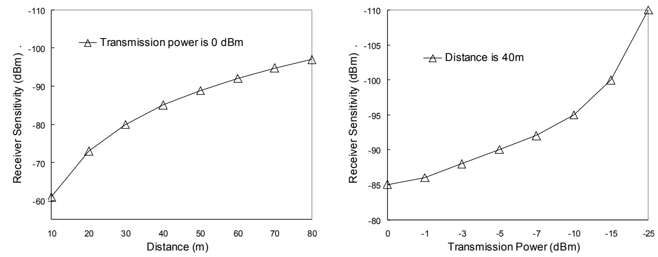
```
IF catch data(p) signal THEN
    Pr ← tworay_rfmodel(moteID)
    IF Pr > CS_Threshold THEN
        IF Pr > RX_Threshold THEN
            Receive (p)
        ELSE
            Carrier_signal (p)
        ENDIF
    ELSE
        Throw (p)
    ENDIF
ENDIF
```

Transmission power and receiver sensitivity are generally expressed by decibel (dBm) unit. The power unit (W) can be changed to dBm by using the following equation.

$$dBm = 10 \times Log(P / 0.001) \qquad (3)$$

By using equation (1), (2) and (3), we design our new wireless propagation model which can be influenced by the transmission power, the distance between two nodes, the antenna gain and height. Algorithm 1 shows the decision of wireless connection based on our wireless propagation model. In the algorithm, *CS_Threshold* means the physical carrier sensing strength which can disturb receiving other information, and *carrier_signal(p)* should inform that the node is included in the carrier sensing range of some sender.

To show the influence of our proposed RF model, we analyze the receiver sensitivity in the TOSSIM according to the distance and transmission power. For the analysis, we utilize values in Table I above and refer to CC2420 RF transceiver [11]. The minimum receiver sensitivity of CC2420 is -90 dBm and its transmission power can be changed from 0 dBm to -25 dBm. When a transmission power is fixed as 0 dBm and a distance between sender and receiver nodes is varied, a RF transceiver may cover nearly 60 m where the receiver sensitivity is -90 dBm as shown in Figure 2(a). Since this distance measured using a real sensor platform is known as about 55 m in [15], our proposed RF model can be said to be valid. When a distance between sender and receiver is fixed as 40m and a transmission power is changed from 0 dBm to -25 dBm, the receiving sensitivity at the receiver node is shown in Figure 2(b). These variations of the signal strength according to the transmission power mean that our proposed model performs well and express wireless features properly. Moreover, they can be utilized to MAC and routing layers as the cross layer concepts.



(a) (b)

Fig. 2. Receiver sensitivity according to the distance and transmission power

TABLE II SPECIFICATIONS OF THE CC1000[10] AND CC2420[11]

	CC1000	CC2420
Data rate	76.8 kbps	250 kbps
Frequency	300-1000MHz	2.5GHz
Transmission Power	10 dBm	0 dBm
Receiver Sensitivity	-110 dBm	-90 dBm
Supply voltage	2.1 V	1.8 V
IEEE 802.15.4 PHY	Nothing	RSSI and LQI 16 channels in 2.4GHz CRC check

B. A New Physical Stack for IEEE 802.15.4 Standard

The IEEE 802.15.4 standard [12] addresses a simple, low-cost and low-rate communication network that allows a wireless connectivity between devices with a limited power. Recently, most of sensor platforms equip the specific RF chip which can provide the IEEE 802.15.4 physical characteristics. CC2420 RF chip is one of these RF transceivers that can be utilized for a number of sensor hardware platforms. Table II shows some specifications of CC1000 and CC2420 RF chips.

As traditional RF physical stack was designed based on the CC1000 RF chip, TOSSIM cannot test and simulate IEEE 802.15.4 standard. Therefore, we design a new RF physical stack referring to CC2420 RF chip for implementing IEEE 802.15.4 standard in TOSSIM. Our physical stack transmits data at 250 kbps and can detect CCA (Clear Channel Assessment), RSSI (Receive Signal Strength Indicator) and LQI (Link Quality Indication) based on our wireless propagation model. Moreover, a user can change the transmission power from 0 dBm to -25 dBm and select 0 to 16 radio channels. These characteristics of our RF physical stack can help to test and implementation of IEEE 802.15.4 standard in TinyOS.

We believe, by our proposed wireless propagation model and new physical stack, various research issues can be simulated on the TOSSIM. Following lists show those issues.

- **Power control:** Our physical stack provides 8 different power levels. Therefore, a power control algorithm for low energy consumption and topology can be simulated and tested on the TOSSIM.

- **Multichannel:** 16 multi-channels are provided and designed to protect the interferences between different channels. This characteristic can help to perform multi-channel protocol simulations on the TOSSIM.

- **Cross Layer:** RSSI and LQI can indicate the quality of each channel. They can be utilized to enhance routing metrics in a network layer and the positioning systems in different applications.

- **Carrier Sensing Range:** Receiver signal strength based on the wireless propagation model may help to perform researches for the carrier sensing and interference range.

- **IEEE 802.15.4 implementation:** As our physical stack is designed referring to the CC2420 RF chip, it has characteristic of 802.15.4 physical such as CCA, RSSI, LQI, multi-channel and CRC etc. Therefore, developers can implement and simulate 802.15.4 MAC on the TOSSIM.

IV. Performance Evaluations

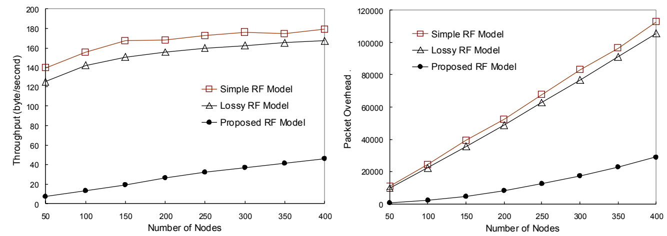
A. *Results of proposed RF models on the TOSSIM*

In this section, we evaluate the performances of TOSSIM based on our proposed RF models. For the evaluations, we make a simple communication program which periodically generates a data packet, and broadcasts it to the wireless medium based on the CSMA/CA mechanism. All simulations are performed in 100 seconds. We compare our proposed models with traditional RF models (*simple* and *lossy*) in the TOSSIM. The simple RF model means that all nodes should be received any transmitted data frame by a sender. Whereas in the lossy RF model, each link between nodes has some probability, and the link connectivity is decided by a random number. We set an average loss rate as 5 % for the lossy RF model.

First, we evaluate the performance of the three RF models by varing the number of nodes. We set a packet generation period as ten seconds, a transmission power as -25 dBm and the receiver sensitivity as -90 dBm. Figure 3(a) shows the throughput of each RF model. With simple and lossy RF models, data packets are transmitted to all nodes or most of nodes. But with our model, they are transmitted to a relatively small number of nodes, i.e., mostly neighbors located near from a sender. Therefore, the amount of receiving data per every second could be larger for those traditional RF models than our proposed RF model. However, it should be pointed out that such a high throughput of them comes with the cost of high packet overhead as shown in Figure 3(b). The figure represents a number of transmission packets as the packet overhead during the simulation time. The packet overhead of simple and lossy RF models are very large because of their unrealistic wireless connections. Since our proposed model makes wireless connections based on the equation (1), the packet overhead is very lower than the other two RF models.

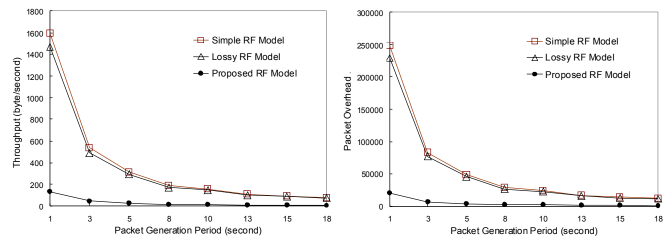
Next, we evaluate the performance of three RF models by varing the packet generation period. We set the number of nodes as 100, the transmission power as -25 dBm and the receiver sensitivity as -90 dBm. Figure 4(a) shows the throughput of three RF models. The packet generation period affects the total amount of data traffic generated by a source. Simple and lossy RF models are exponentially influenced by the data traffic because of their numerous wireless connections. Otherwise, since our proposed RF model reasonably manages the number of wireless connections to be smaller, throughput of it slowly increases according to the data traffic. Figure 4(b) shows the packet overhead of three RF models. As shown in the figure, simple and lossy RF models are continuously worse because lots of data traffic affects negatively against the packet overhead aspects. However, in our reasonable wireless connections, packet overheads are less influenced to the wireless communications.

Finally, we evaluate the performance of the proposed RF model by varing a transmission power. We set the packet generation period as 10 seconds, the number of nodes as 100 and receiver sensitivity as -90 dBm. Actually, a transmission power affects a communication distance between two nodes in general scenarios of wireless communications. However, since simple and lossy RF models do not take into account a transmission power, their performances remain unchanged. Otherwise, the performance of the proposed RF model is mainly influenced by the parameter of a transmission power. Since the decreasing the transmission power means the decreasing number of connections, both the throughput and packet overhead also decrease as shown in Figure 5(a) and (b).



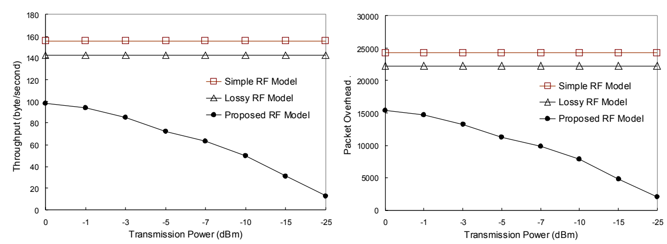
(a) Throughput (b) Packet overhead

Fig. 3. Throughput and packet overhead according to *the number of nodes*: the packet generation period is 10 second, transmission power is -25 dBm and receiver sensitivity is -90 dBm



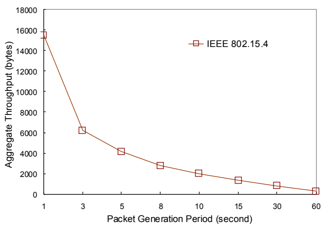
(a) Throughput (b) Packet overhead

Fig. 4. Throughput and packet overhead according to *packet generation period*: number of node is 100, transmission power is -25 dBm and receiver sensitivity is -90 dBm

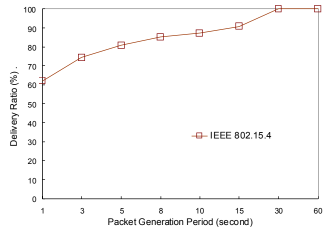


(a) Throughput (b) Packet overhead

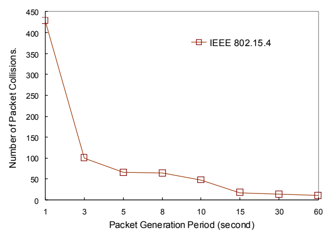
Fig. 5. Throughput and packet overhead according to *the transmission power*: the packet generation period is 10 second, number of node is 100 and receiver sensitivity is -90 dBm



(a) Aggregate Throughput



(b) Delivery ratio



(c) Packet collision

Fig. 6. Aggregate throughput, delivery ratio and collision according to the packet generation period: number of node is 10, transmission power is -25 dBm, receiver sensitivity is -90 dBm, SO is 3 and BO is 8

B. *Results of IEEE 802.15.4 Standard on the TOSSIM*

In this section, we evaluate the performance of IEEE 802.15.4 standard based on our proposed RF model. On traditional RF models of TOSSIM, the IEEE 802.15.4 protocol stack has not been implemented. However, by using our proposed new RF models, we can implement its full stack with 51 primitives on the TOSSIM. The following list shows our implemented interfaces for providing 802.15.4 standard services in TinyOS:

- MA_SAP interface: 3 primitives, to support data transmission services between 802.2 LLC and 802.15.4 MCPS.
- MCPS_SAP interface: 5 primitives, to support the transport of SSCS protocol data units to MAC sub-layer.
- MLME_SAP interface: 30 primitives, to allow the transport of manage- ment commands between SSCS and MLME.
- PD_SAP interface: 3 primitives, to support the transport of MPDUs between peer MAC sub-layer entities.
- PLME_SAP interface: 10 primitives, to allow the transport of manage- ment command between MLME and PLME.

In our simulation, we evaluate performance of the standard in the star topology with 10 devices. One device is chosen as the coordinator while other devices act as the associated devices. They operate as the Beacon-enable mode [12] and values for SO and BO are set to 3 and 8, respectively[2]. The data frame size is 30 bytes. The total simulation time is 100 seconds, and the energy consumption model is referred by CC2420 datasheet [11]. Within the initial 5 seconds of simulation, each general device tries to associate with a coordinator and then nodes (including a coordinator) start their data transmission based on the packet generation time factor. We evaluate the implemented standard with the various data traffic load by varying a packet generation period.

[2] SO and BO define the superframe duration and beacon interval [12]

Figure 6(a) shows the aggregate throughput of the IEEE 802.15.4 standard. It seems to show a good performance in terms of throughput, but it does not process all data traffics in the high data traffic as shown in Figure 6(b). IEEE 802.15.4 can transmit data in the active duration only because it maintains a periodic active/inactive scheduling for efficient energy consumptions. This characteristic also influences packet collisions. Figure 6(c) shows that packet collisions are also not good in high data traffic because an active duration is limited in Beacon-enable mode. In order to increase efficiency of data throughput, delivery ratio and packet collisions, the standard needs some algorithm to adjust active duration based on the data traffic information.

V. CONCLUSION AND FUTURE WORKS

In this paper, we design and implement the new RF models on the TOSSIM to increase the accuracy of wireless simulation results. To establish more efficient wireless connection, we develop a new wireless propagation model based on the two-ray ground pass loss model. Moreover, we design a new physical stack referring to CC2420 RF chip for applying various research issues to the TOSSIM and implement IEEE 802.15.4. Simulation results show the efficiency of proposed RF model compared with simple and lossy RF models. Based on our proposed RF models, we can implement the IEEE 802.15.4 full standard in TOSSIM, and this is one of the contributions in this paper. As a part of our future works, we plan to evaluate the performance of IEEE 802.15.4 on real hardware platforms, and to study more adaptive active duration algorithm in order to increase efficiency of data throughput, delivery ratio and packet collisions in the IEEE 802.15.4 Beacon-enable mode.

REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, "Wireless Sensor Networks: A Survey," *in Elsevier Computer Networks*, 2002
[2] The CMU Monarch Project's Extensions to NS.
[3] S. Park, A. Savvides, and M. B. Srivastava, "SensorSim: A Simulation Framework for Sensor Networks," *in ACM MSWiM*, 2000.
[4] Qualnet simulator: www.scalablenetworks.com/
[5] Opnet simulator: www.opnet.com/
[6] P. Levis, N. Lee, M. Welsh, and D. Culler, "TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications," *in ACM SenSys*, 2003.
[7] TinyOS, http://webs.cs.berkeley.edu/tos/
[8] Mica: http://www.xbow.com/
[9] ZigbeX: http://hanback.co.kr
[10] CC1000: "Datasheet for Chipcon (TI) CC1000 315, 433, 868, 915 MHz ultra-low-power RF Transceiver",
[11] CC2420: "Datasheet for Chipcon (TI) CC2420 2.4 GHz IEEE 802.15.4/ZigBee RF Transceiver",
[12] IEEE Standard 802.15.4, Part 15.4: Wireless Medium Access Control and Physical Layer specification for Low Rate Wireless Personal Area Networks. IEEE Std. 802.15.4-2003, Dec. 2003
[13] K. Pahlavan and P. Krishnamurthy, "Principles of Wireless Networks," *in Prentice Hall PTR*, 2002.
[14] D. Gay, P. Levis, R. von Behren, M. Welsh, E. Brewer, and D. Culler "The nesC Language: A Holistic Approach to Networked Embedded Systems," *in ACM PLDI*, June 2003.
[15] G. Anastasi, M. Conti, E. Gregori, A Falchi, A. Passarella, "Performance Measurements of Mote Sensor Networks," *in ACM MSWiM*, Oct. 2004.