

AvroraZ: Extending Avrora with an IEEE 802.15.4 Compliant Radio Chip Model

Rodolfo de Paz Alberola and Dirk Pesch

Centre for Adaptive Wireless Systems
Cork Institute of Technology
Rossa Avenue, Cork, Ireland
Phone +353 21 4326873
rodolfo.depaz@cit.ie; dirk.pesch@cit.ie

ABSTRACT

This paper presents AvroraZ, an extension of the Avrora emulator – The AVR Simulation and Analysis Framework – which allows the emulation of the Atmel AVR microcontroller based sensor node platforms with IEEE 802.15.4 compliant radio chips thus allowing emulation of sensor nodes such as Crossbow’s MicaZ. AvroraZ is based on design, implementation and verification of several extensions to Avrora: the address recognition algorithm, an indoor radio model, the clear channel assessment (CCA) and link quality indicator (LQI) of the IEEE 802.15.4 standard. We have AvroraZ and demonstrated its correctness in emulating MicaZ code.

Categories and Subject Descriptors

I.6.5 [Computing Methodologies]: Simulation and Modeling - Model Validation and Analysis. D.2.5 [Software Engineering]: Testing and Debugging;

General Terms

Algorithms, Design, Experimentation, Verification

Keywords

Wireless sensor networks, sensor node emulation, IEEE802.15.4, Avrora, Chipcon cc2420, micaZ.

1. INTRODUCTION

Over the past few years there has been a growing interest in sensor networks for a wide range of real world applications within the research community. However, programming these applications has been challenging for most of the developers due to the limitations of the hardware components used in current sensor platforms as well as the manifestation of these constraints in the software development environment. These restrictions have significantly influenced how the researchers evaluate their designs in real sensor networks test-beds. Instead of generating simulation models to study protocols and applications, sensor network researchers prefer to evaluate the implementations directly on the target platform in a physical test-bed, so that the operating constraints are appropriately reflected in the design of the software. The relatively low cost of sensor nodes and availability of free software is facilitating this approach.

Although this growing interest in the use of physical test-beds exists, these studies suffer from significant challenges including the lack of scalability and repeatability as well as the inability to represent a diverse set of operational scenarios. On the other hand, simulation based experiments can address problems of scalability and repeatability, which has made computer simulation an important step in the development of software for wireless sensor networks and has been used intensively in the last few years. Current sensor network simulators can be broadly divided into three categories: network simulators enhanced with sensor models, sensor emulators enhanced with network models and instruction cycle level emulators.

Originally work in sensor simulation focused on the first category, where sensor specific models were added to popular network simulators. This way of simulating is very attractive as it does not require migrating away from the existing and familiar network simulators. The most widely used network simulator, ns-2 [1], has been extended by SENSE [2] to include several features to simulate sensor networks like models of sensor channels, battery and power consumption. Several extensions for the well known OPNET simulator, such as a IEEE 802.15.4 Simulation Model [3], have also been released. Furthermore, for the OMNET++ [4] platform numerous WSN simulation models and tools, such as Castalia [5] have been developed. Many other publicly available network simulators such as J-Sim [6], GlomoSim [7] and its descendant Qualnet [8] include also sensor networks models. However, the limitation of all these network simulators is that they do not model the hardware of sensor platforms with sufficient accuracy. Furthermore, each new network protocol that wants to be tested has to be first implemented specifically for the simulation platform.

In sensor emulation with networking support the sensor code is compiled on a PC platform and the hardware API is abstracted with models. The most example of this category is TOSSIM [9] which is an emulator for TinyOS. Instead of compiling a TinyOS application for a mote, users compile it into the TOSSIM framework, which runs on a PC. This approach also suffers from various limitations. These emulators have their own implementation of the event scheduler which is typically not as efficient as network simulators. Also, such emulators do not offer flexibility in extending the frameworks to other sensor node operating systems, e.g. other than TinyOS.

The third approach is based on sensor node emulation at the instruction cycle granularity. Avrora [10], developed at the University of California at Los Angeles, is the best known example of this approach. For Avrora, the code is cross-compiled for the sensor node architecture and is executed on an emulated processor similar to execution of Java byte code on a

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

PM²HW²N'08, October 31, 2008, Vancouver, BC, Canada.
Copyright 2008 ACM 978-1-60558-239-9/08/10...\$5.00.

Java virtual machine. This provides the greatest measure of software modeling fidelity. Furthermore, this approach has the advantage that it can run the sensor code-base without any modifications either to programming syntax, compiler or the binaries from what would run on the real target hardware.

However, so far no model compliant with the IEEE 802.15.4 standard has been included in Avrora. This means that sensor node applications utilizing the radio communication standard cannot be tested on the emulator. For this reason, we have created AvroraZ, an extension to the Avrora emulator, which provides an emulation of the Texas Instruments Chipcon CC2420 radio chip [12], which is used in many sensor node platforms such as the Crossbow MicaZ nodes [13]. Our implementation enables precise emulation of IEEE 802.15.4 based protocols without any modifications in the code developed for the real hardware. In order to enable this emulation, the following extensions have been introduced in the emulator: the address recognition algorithm, an indoor radio model, clear channel assessment (CCA) and link quality indicator (LQI) support for the IEEE 802.15.4 standard.

The remainder of the paper is organised as follows: Section II explains the AVR-microcontroller emulator architecture and its fundamental characteristics. Section III describes the part of the IEEE 802.15.4 standard related to the physical layer which has been added to the emulator. Section IV explains the implementation of the new contributions introduced by AvroraZ. Section V evaluates those extensions by means of both test-bed experiments and computer emulations. Section VI revisits the main contributions of the paper and outlines possible areas in which AvroraZ may be used.

2. AVRORA: AVR-MICROCONTROLLER EMULATOR

Avrora is a cycle-accurate instruction-level simulator for the AVR microcontroller and for sensor networks built on the AVR microcontroller. Since Avrora is implemented in Java, the object-oriented design therefore lends itself to encapsulating the principal concepts in simulation such as instructions, devices, and state in an intuitive manner.

Avrora's architecture (see Figure 1) allows a model of each device to be built in software that emulates the device and allows the program to interact with it. Each device model is written in Java and connects to the main simulation through interfaces that separate it from the details of the simulator implementation.

The centerpiece of Avrora is the simulator class which consists of the interpreter and the event queue. The interpreter executes every assembly instruction of the program, while the event delta queue carries out the cycle-accurate instruction-level simulation and enables sleep optimization when the microcontroller is in idle mode. On-chip devices communicate with the simulator through the I/O register interface and the event queue interface. The microcontroller interface encapsulates an instance of the simulator class and supplies the on-chip devices such as the timer and the analog digital converter.

Finally, the platform interface encapsulates the microcontroller class and supplies the off-chip devices such as the radio and LEDs. The external devices communicate with the microcontroller through the pin interface.

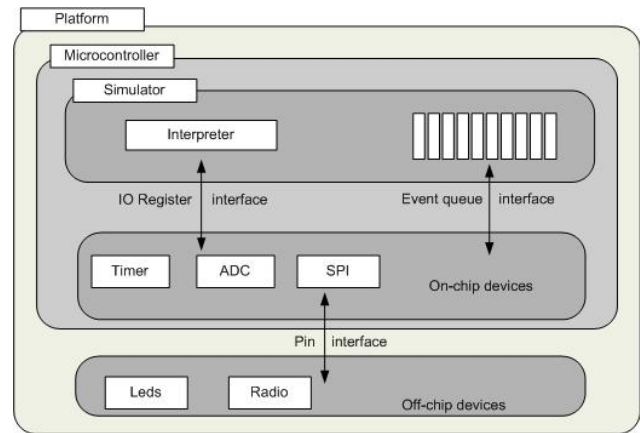


Figure 1 Avrora architecture

Avrora multi-node simulation is created by means of separate instances of the simulator class, each with its own local simulation time, and running in its own thread. The threads run in parallel and they periodically synchronize to a global clock (inserting periodic events into the queue) in order to manage the state of the environment –i.e. the radio communication channel.

In terms of scalability, since Avrora runs one thread per node, the number of nodes that can be simulated is only limited by the host operating system's limit in the number of threads per process. Those threads synchronize only when necessary to ensure that the global timing and order of radio communications are preserved during simulation. In [10], experiments showed that Avrora can handle up to 1750 nodes on a dual processor server. However, it achieves greater than real hardware system performance only for networks of less than 25 nodes.

Avrora offers three core instrumentation mechanisms: probes, watches, and events. This dynamic instrumentation allows adding arbitrary profiling behavior at any point in the program or in response to any type of event at any point in simulation time, without making any modifications to the simulator or influencing the program's behavior or timing. In the original Avrora design, propagation models are not explicitly supported, however the modular Java interface allows these to be added as we have done and describe in this paper.

3. IEEE 802.15.4 PHYSICAL LAYER

The Low Rate Wireless Personal Area Networks (IEEE 802.15.4/LR-WPAN) standard is intended to serve industrial, residential and medical applications, using very low power consumption with relaxed needs for data rates and QoS. This standard defines both physical (PHY) and medium access control (MAC) layers. The standard offers two PHY options based on the frequency bands of 868/915 Mhz or 2450 Mhz. The PHY option implemented in AvroraZ is the 2.4 Ghz band due to its worldwide availability. Signal encoding is based on direct sequence spread spectrum (DSSS) in which each symbol is mapped to a pseudo-random noise (PN) sequence. The data rate is 250kbps at 2.4GHz using offset Quadrature Phase Shift Keying (O-QPSK) for modulation of the bits.

The features of the PHY layer are activation and deactivation of the radio transceiver, energy detection (ED), link quality indication (LQI), channel selection, clear channel assessment (CCA) and transmitting as well as receiving packets across the physical medium.

1. *Activation and deactivation* of the radio transceiver. The radio should be turned ON, OFF or sleep depending on the request of the MAC sub layer. This allows support for energy efficient protocols.

2. *Energy Detection (ED)* within the current channel, which is an estimation of the received signal power within the bandwidth of one channel. The ED time should be equal to 8 symbol periods.

3. *Clear Channel Assessment (CCA)*, determines whether the channel is busy or idle by comparing the energy in the channel with a threshold. This threshold can be set according to three different methods:

- *Energy Detection Mode.* CCA reports a busy medium if the detected energy is above the ED threshold.
- *Carrier Sense Mode.* CCA reports a busy medium if a signal with the modulation and spreading characteristics of IEEE 802.15.4 is detected. This signal may be higher or lower than the ED threshold.
- *Carrier Sense with Energy Detection Mode.* This is a combination of both modes. The CCA reports a busy medium if it detects a signal with same characteristics of IEEE 802.15.4. with energy above the ED threshold.

4. *Link Quality Indicator (LQI)*, measures the strength/quality of the received signal. This measurement may be implemented using receiver ED, a signal to noise estimation or a combination of both methods. The LQI values should be distributed uniformly between the lowest and highest quality signals detectable by the receiver. LQI may be used by network and application layers, but it is not specified in the standard.

5. *Channel Frequency Selection.* The PHY layer should be able to tune its transceiver into a specific channel requested by upper layers.

Devices communicate in IEEE 802.15.4. networks using the packet format illustrated in Figure 2. This packet consists of a synchronization header (SHR) preamble, a physical layer header (PHR) and a data field.

SHR		PHR		PHY Payload
Preamble (4 bytes)	Start Frame Delimiter SFD (1 byte)	Frame length (7 bits)	Reserved (1 bit)	Data (up to 127 bytes)

Figure 2 Physical Protocol Data Unit (PPDU) format

4. AVRORAZ

AvroraZ adds support for IEEE 802.15.4 standard compliant radio chips such as the CC2420 from TI to the original Avrora implementation [10]. It includes an indoor radio model to support Received Signal Strength Indicator (RSSI) and correlation values of the CC2420 radio chip, which permits LQI measurements, ED detection and CCA necessary for the implementation of the CSMA-CA algorithm in the MAC layer. Additionally, address recognition and acknowledgement of frames have also been added since their hardware support is being carried out by standard compliant radio chips.

4.1 Address Recognition

Due to the nature of radio communications, a device with its receiver enabled will be able to receive and decode transmissions from all devices complying with the standard that are currently operating on the same channel and are in its receiving range, along with interference from other sources. The standard, therefore, defines in the MAC layer how to filter incoming frames in order to deliver only the frames that are of interest to the upper layers. Hardware support for this filtering is being carried out by standards compliant radio chips in the PHY layer. This filtering based on 802.15.4 standard requirements has been fully implemented in our Avrora extension - AvroraZ.

When any of the receiving requirements defined in the standard are not satisfied, the radio chip discards the incoming frame without processing it, and flushes the data from the received FIFO queue of the rejected frame. The radio chip will start searching new frames only after the rejected frame has been completely received to avoid detecting false SFDs. It is worth noting that some standard MAC attributes (i.e. PAN identifier, 16 and 64 bit address) have to be stored in the radio chip's RAM to perform the address recognition algorithm. Therefore, the implementation in AvroraZ emulates this RAM to compare the values stored in it to the ones in the radio chip registers.

4.2 Frame Acknowledgment

Supporting acknowledgment of frames is also important in order to accomplish 802.15.4 std. compliance. A frame transmitted with the acknowledgment request flag set to one shall be acknowledged by the recipient. If the intended recipient correctly receives the frame, meaning that address recognition is passed and resulted in a valid frame check, it shall generate and send an acknowledgment frame containing the same sequence number from the data that is being acknowledged.

In order to fulfill these procedures, AvroraZ emulates the way the compliant radio chips transmit acknowledgment frames when it is requested by the sender. In addition, the emulator supports automatically sending of acknowledgments when a frame is correctly received.

4.3 Indoor Radio Model

AvroraZ adds an indoor radio model in order to provide LQI and CCA real-time support as offered by the standard. The model implemented follows an indoor radio model developed for WiFi networks [14].

Transmitted power in each mote is programmed and controlled by a radio chip register called TXCTRL.PA_LEVEL register. In the CC2420 data sheet [12], only the output power for some settings is shown. However, in real hardware, all integer values ranging from 0-31 are supported by the radio chip register. The approach we have taken is to perform cubic spline interpolation in order to obtain values not specified in the data sheet. Figure 3 depicts the range of values obtained after interpolation, which are used in the radio model to set the transmitted power.

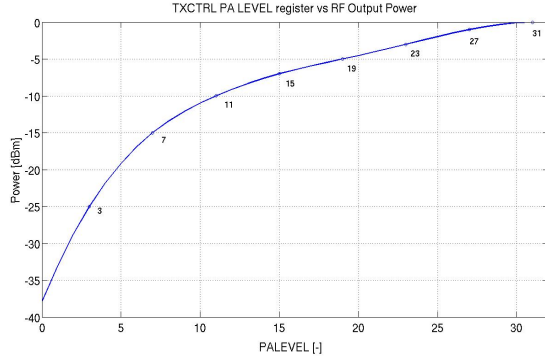


Figure 3 Radio chip output power values

Figure 4 draws the indoor radio model implemented in lineal scale. There, the path loss (L) has been computed following the log-distance path loss model

$$L(d) = PL(d_0) + 10 \log_{10}(d/d_0) \quad (1)$$

where $PL(d_0)$ is the path loss for a reference distance (d_0). This can be calculated empirically or theoretically by means of the free space Path Loss formula. Reference distance (d_0) usually represents the small distance in the far field and it is chosen typically to 1m in indoor scenarios. The path loss exponent η should be chosen to match the environment that a user wishes to model.

Fading (F) has been modeled following the indoor radio model depicted in Figure 4. This model has been adopted considering that motes are usually placed on ceilings, walls or sometimes even on desks to sense light, temperature, and humidity inside buildings. This typically corresponds with WLAN access point placement. Therefore, the adoption of a radio model developed for WiFi networks for radio channel modeling in wireless sensor networks makes sense. The chosen radio model considers that rapid and frequent transitions between line of sight (LOS) and non line of sight (NLOS) occur in typical node locations due to movement of obstacles in the radio path such as people. Fading is then defined as a combination of a coherent part represented by a Rice distribution and a diffuse part arising from multipath NLOS components which are represented by a mixture of Log-normal and Rayleigh distributions. This can be expressed as:

$$F = A \cdot P_{\text{RICE}}(k) + (1-A) \cdot P_{\text{LN}}(\sigma_s, \mu_s) P_{\text{RAYLEIGH}} \quad (2)$$

As we can see in the model depicted, transitions between these two states are controlled on a time shared (A) basis, which is defined as:

$$A = (1 - \rho)^{0.21} \quad (3)$$

Where ρ (m^{-2}) is the number of people or obstacles over an occupied area and l (m) is the length of the ray over that area.

When time sharing A indicates a bad channel, fading follows a combination of Rayleigh and log-normal distributions due to movement of obstacles. In this case, standard and mean deviations of the log-normal distribution are expressed as:

$$\sigma_s = \log_7(551\rho + 1) + 0.5 \quad (4)$$

$$\mu_s = (31\rho)^{0.7} \quad (5)$$

In good channel state fading is characterized by Ricean fading with the factor k representing the ratio between direct-path power and diffuse power. This ratio is expressed linearly, not in decibels.

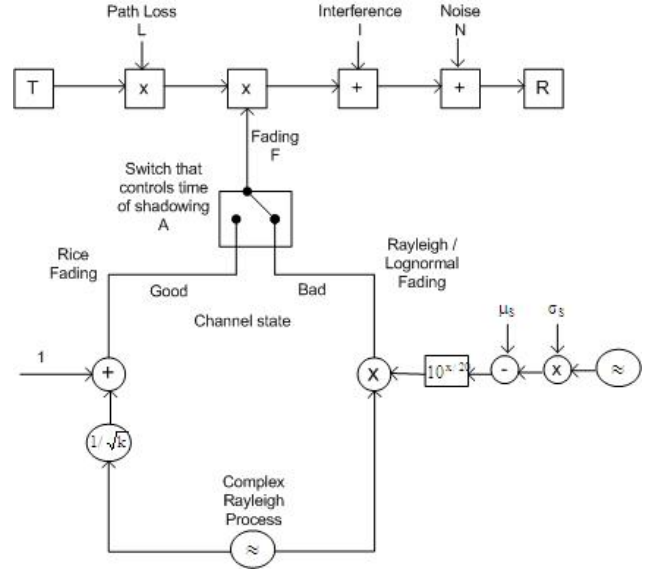


Figure 4 Indoor radio model for AvroraZ

The two radio channel parameters ρ , l can be set in the simulator by defining network topology and the density of people in the environment. Figure 5 shows the influence of these parameters in the radio model. It can be seen that the higher the number of obstacles (ρ) and/or the distance between motes (l) the higher is the influence of the shadowing in the model.

Interference from others sensor motes in the wireless sensor network is computed in AvroraZ following the simple approach of summing any power received, at the specific 3D position and time, to the receiver mote. Noise (N) can be created in the simulator by means of real measurements of a radio channel. For that, RSSISample application [15] from the Tinyos-2.x operative system is suggested to be used. This firmware generates a text file with readings from the RSSI register. Afterwards, the file can be read by the simulator to generate a Noise time trace that will be used by the radio model.

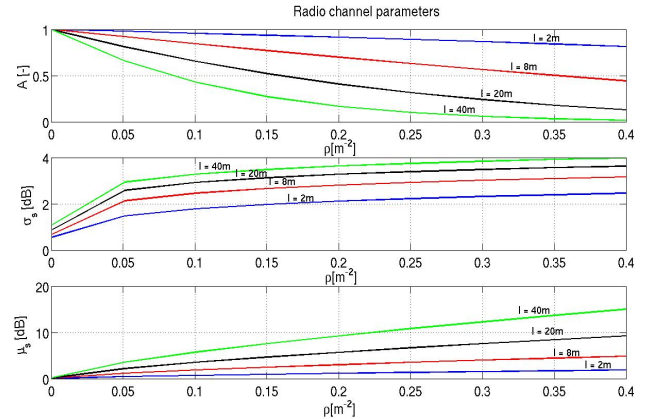


Figure 5 A , σ_s , μ_s for several settings of ρ , l

Finally, it is worth noting that the radio model has been implemented with an awareness of not decreasing the performance of Avrora. As explained before, the AVR emulator is based on an event queue that enables sleep optimization. This means that the emulator only needs to process events in the queue in order, skipping large amounts of idle simulation time. Following this approach, values in the radio model are updated

every symbol but only on a packet basis. This allows to maintain the same performance in Avrora since in receive mode Java threads can sleep while not receiving a packet.

4.4 Clear Channel Assessment (CCA)

The clear channel assessment signal is based on the measured RSSI value and a programmable threshold. This clear channel assessment value is used to implement the CSMA-CA functionality specified in the MAC layer of the IEEE802.15.4 standard [11]. All three CCA modes specified by the standard are implemented in the CC2420. However, as in AvroraZ all devices are considered to be 802.15.4 compliant the carrier sense mode is not really necessary. The other modes have been implemented as follows

$$\begin{aligned} \text{CCA} &= 1 \text{ when } \text{RSSI_VAL} < \text{CCA_THR} - \text{CCA_HYST} \\ \text{CCA} &= 0 \text{ when } \text{RSSI_VAL} \geq \text{CCA_THR} - \text{CCA_HYST} \end{aligned} \quad (6)$$

Where CCA_THR represents the carrier sense threshold level whose values must be set in steps of 1 dB, and CCA_HYST is a CCA hysteresis value defined for the CC2420 radio chip. Transmission will only start if the channel is clear. The radio chip status register may be used to detect the result of the CCA.

4.5 Link Quality Indicator (LQI)

The LQI measurement defined in the IEEE802.15.4 standard is a characterization of the strength and/or quality of a received packet. This measurement may be implemented using receiver RSSI, a signal-to-noise ratio estimation, or a combination of these methods. LQI result can be then used by the network or application layers to choose the best route between two motes, although its use is not specified in the standard.

Following the standard, LQI measurement shall be performed for each received packet, and the result shall be reported to the MAC sub layer as an integer ranging from 0 to 255, with at least 8 unique values. The minimum and maximum LQI values (0 and 255) should be associated with the lowest and highest quality compliant signals detectable by the receiver, and LQI values in between should be uniformly distributed between these two limits.

In order to comply with the IEEE802.15.4 standard, the CC2420 radio chip appends in each receive frame two values in order to calculate the LQI (see Figure 6). Assuming that the user is normally only interested in the correctness of the Frame Check Sequence (FCS) and not the FCS sequence itself. The two FCS bytes are replaced by the RSSI value and an average correlation value.

Afterwards, higher layers should be responsible for generating the appropriate scaling of the LQI value by means of these two values.

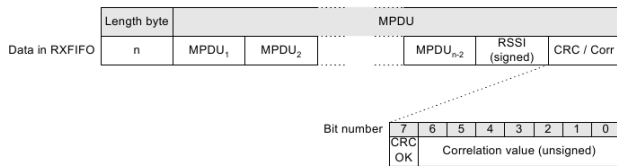


Figure 6 RSSI and correlation values in RXFIFO

4.5.1 Received Signal Strength (RSSI)

CC2420 has a built-in RSSI (Received Signal Strength Indicator) yielding a digital value that can be read from the 8 bit,

signed 2's complement RSSI.RSSI_VAL register. This RSSI value is always averaged over 8 symbol periods (128 μs), in accordance with the IEEE802.15.4 standard.

This RSSI register value RSSI.RSSI_VAL can be referred to as the power P received at the RF pins of the radio chip by using the following equation:

$$P = \text{RSSI_VAL} + \text{RSSI_OFFSET} [\text{dBm}] \quad (7)$$

Where the RSSI_OFFSET is approximately -45 in cc2420 radio chip, e.g. if reading a value of -30 from the RSSI register; the RF input power is approximately -75 dBm.

In the AvroraZ emulator, the RSSI register value RSSI.RSSI_VAL is calculated and continuously updated for each symbol, on a per packet basis, after RSSI has become valid using equation (7).

4.5.2 Correlation Value

The RSSI value described in the previous section may be used alone by the MAC software to produce the LQI value. However, using the RSSI value directly to calculate the LQI value has the disadvantage that a narrowband interferer inside the channel bandwidth could increase the LQI value although it actually reduces the true link quality. The radio chip therefore also provides an average correlation value for each incoming packet, based on the 8 first symbols following the SFD. This unsigned 7-bit value can be looked upon as a measurement of the "chip error rate" (CER) although cc2420 does not perform chip level decisions. Let us explain this; the 2.4 GHz physical layer of the IEEE 802.15.4 standard [11] uses 32 chips to represent 4 information bits. Each of the 16 possible symbols is represented by a predefined 32 chip sequence. The receiver is based on correlating the received chip sequence against all of the 16 possible symbols. The correlation value is computed as the sum of the 8 first correlation values for the selected symbols following the SFD. It is worth noting that this value does not go up to 127 (even if it is a 7 bit value). This is due to bandwidth limitations in both the transmitter and receiver that make the soft decision chips stay away from the "ideal" values

The process we follow to compute this correlation value in the simulator is depicted in Figure 7. This model considers that correlation value can be viewed as a CER, which will be related in same way to the final packet error rate (PER).

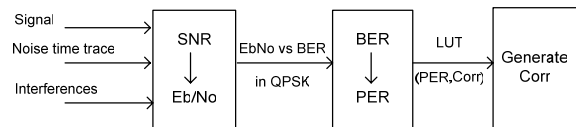


Figure 7 Process to compute correlation value

Firstly, SNR is computed as the ratio of the signal power (Ps) and the noise power (Pn) plus the interference from other sensor motes in the wireless network (Pi).

$$\text{SNR} = \frac{P_s}{P_n + \sum P_i} \quad (8)$$

Then, the ratio of Energy per Bit (E_b) to the Spectral Noise Density (N_o) can be calculated as the Signal to Noise Ratio normalized to the spectral efficiency (η):

$$\frac{E_b}{N_o} = \frac{SNR}{\eta} \quad (9)$$

Where the spectral efficiency can be computed for the IEEE802.15.4 PHY layer as:

$$\eta = \frac{R_b}{BW} = \frac{2500 \text{ kbps}}{2000 \text{ kHz}} = 1.25 \text{ (bit / s) / hz} \quad (10)$$

Since IEEE 802.15.4 physical layer at 2.4 GHz uses offset quadrature phase shift keying (OQPSK) modulation, the bit error rate (BER) in the AWGN channel can be related to the E_b/N_o as:

$$P_b = Q\left(\sqrt{\frac{2E_b}{N_o}}\right) \quad (11)$$

Where $Q(x)$ is the Gaussian error integral which can be approximated with an error of less than 3% as (see [16] for further info)

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} \exp\left(-\frac{x^2}{2}\right) dx \approx \frac{e^{-(x^2/2)}}{1.64x + \sqrt{0.76x^2 + 4}} \quad (12)$$

Figure 8 shows this relationship between the bit error rate and the E_b/N_o .

Assuming independent identically distributed errors, the Packet Error Rate (PER) can be computed from the BER as

$$PER = 1 - (1 - BER)^{N_{bits}} \quad (13)$$

Where N_{bits} is the number of bits per packet.

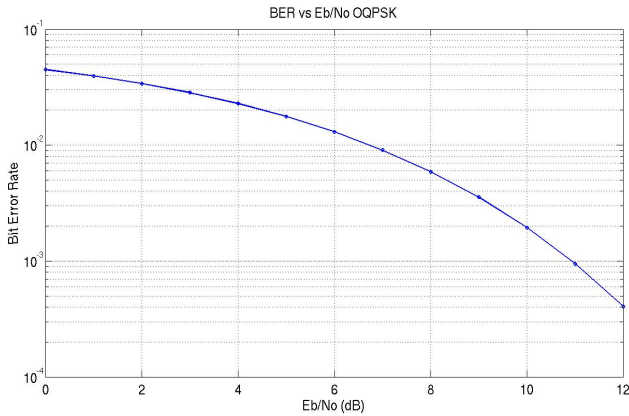


Figure 8 Theoretic bit error rate for OQPSK

Once we know the PER in the simulator, it is mapped to the correlation value through a Look Up Table (LUT). This LUT was computed by means of real measurements taken with the Ember Rangetest firmware installed on Tyndall motes [17]. This application gives directly PER and correlation values of each received packet. Hence, a mapping between both in different scenarios could be found. Measurements taken are depicted in Figure 9

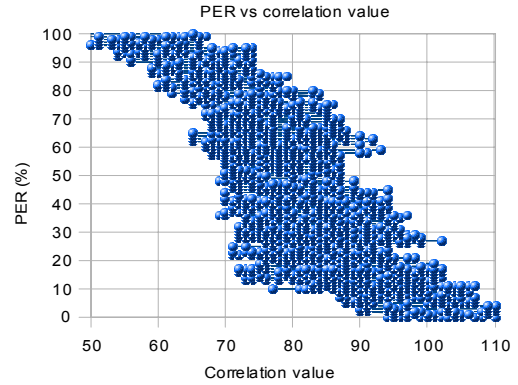


Figure 9 Packet error rate versus correlation value

Looking at this plot, the correlation value has very high variance over time, e.g. for a given $PER = 80\%$ correlation can be any value between 60 and 80. Hence, each correlation value cannot be determined for a single PER value. However, they truly have the association mentioned before, a low correlation means a bad channel (a high CER) and therefore a high PER. This suggests that correlation value can be calculated as a probabilistic distribution over the time, where statistical properties are taken from a LUT created with PER measurements.

5. EVALUATION

The implementation elements discussed in the previous sections have been evaluated by comparing results obtained in our AvrozaZ emulator extension with measurements from a test bed installed in our lab at Cork Institute of Technology, Ireland (see Figure 10). The evaluation process was carried out based on the following steps

1. Build test applications with TinyOS-2.x for micaZ platform.
2. Install same test application firmware on both, the real WSN (using the Re-Mote framework [18]) and an emulated WSN (AvrozaZ emulator)
3. Compare results between both. If any problems are found: modify AvrozaZ code, reinstall and compare again.

Table 1 shows environmental set up for all tests. Figure 11 shows the noise time trace used for tests. These RSSI readings were done at 1KHz sampling frequency inside the lab where motes were placed. It can be seen there that the noise floor for this radio chip fluctuates between -98 and -96. The noise peaks are probably due to the influence of other devices in the 2.4GHz ISM band, since there were WiFi networks active within the university that could influence the measurements.

The address recognition and frame acknowledgment implementation were tested via executing code in both AvrozaZ and Re-Mote framework unicast and broadcast communications between several devices. Results verified that both emulated and real hardware rejected frames when IEEE802.15.4 address recognition requirements were not accomplished.

Table 1 Environmental set up

Message Payload	20 bytes
Message rate	4 msg/s
Simulation time	1000 seconds
Transmitted power	0 dBm
Frequency channel	26
Scenario	Indoor (office)
Reference distance d_0	1m
Pathloss at d_0	55 dBm
Noise time trace	Figure 11
Rician K-factor	6
Pathloss exponent η	3

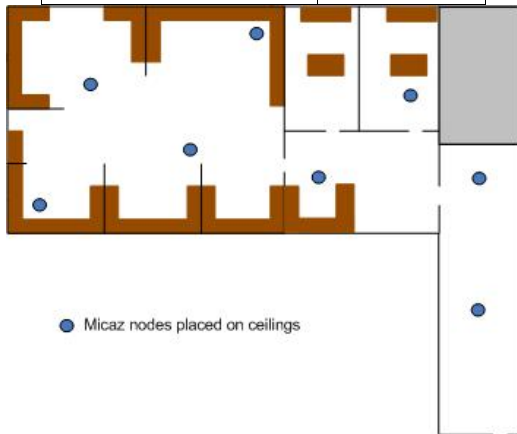


Figure 10 Test bed installed in the lab at CIT

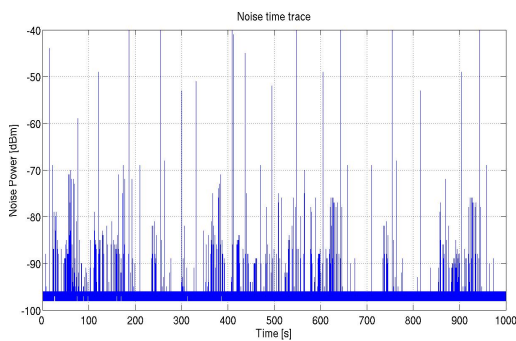


Figure 11 Noise time trace

In order to give an example, we refer to the testAck application from Tinyos-2.x. [19], this firmware can be installed in AvroraZ and in real motes to test the acknowledgement functionality of the radio chip. This application requests an acknowledgement when the receiver gets a correct packet. Therefore, a message exchange between motes 0,1 when are in range should produce the acknowledgement of packets.

However, when the application in mote 0 is modified to send messages to another mote id, different from 1 (e.g. id 3), mote 2

does not reply with acknowledgements although it continues in its range. This is because address recognition has failed since short destination address = 3 does not match: ShortAddress = 2 or broadcast address.

The same process as described before was followed to check that the other filtering requirements were accomplished.

We also launched several tests with the aim of evaluating the radio model and therefore the LQI, CCA implementation. Firstly, AvroraZ simulations with different settings of radio parameters ρ , l were done and the probability density function for the RSSI per packet was recorded as is shown in Figure 12. It can be seen there, that the power received decreases in a logarithmic manner with increasing distance as corresponding to equation (1). Furthermore, the influence of the parameter (ρ) is more noticeable when the distance between motes (l) is largest. This is clearly visible in the wider distribution for the received RSSI where l is higher, which indicates that the shadowing due to obstacles is having more importance in the radio model.

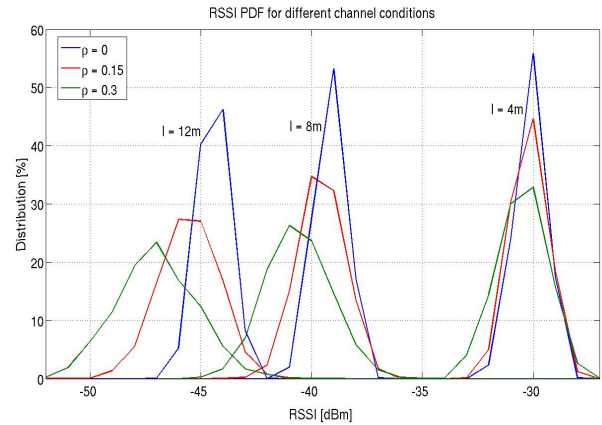


Figure 12 RSSI PDFs in AvroraZ for several ρ , l

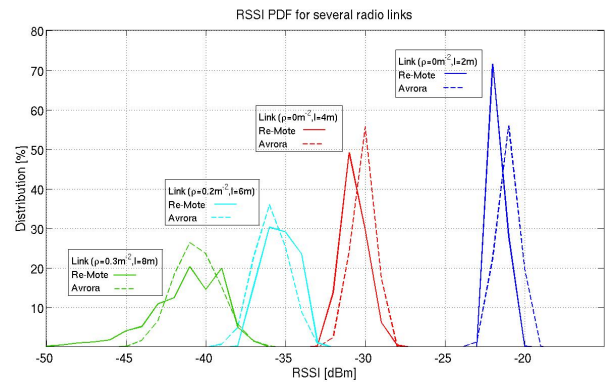


Figure 13 AvroraZ vs. test-bed results for RSSI

Taking then Figure 12 as a reference, hardware tests between several links of motes placed at in our lab were done to show how the radio model could be used to simulate a real wireless sensor network. Figure 13 shows AvroraZ and test bed results for some reference radio links that were simulated in AvroraZ. Outcomes depicted prove that the wireless sensor network can be modeled accurately in AvroraZ by setting appropriate radio parameters ρ , l for every link.

6. CONCLUSIONS

We have presented AvroraZ, which represents extensions to Avrora that enable the research community to validate time-dependent properties of sensor networks with IEEE802.15.4 compliant radio chips like the Texas Instruments Chipcon cc2420 chipset. Wireless sensor network deployments can now be carried out with more confidence after detailed instruction-level simulation has been performed. Furthermore, various interesting research issues can now be simulated with AvroraZ. We list some in the following:

- Cross Layer: RSSI and correlation 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.
- Low energy consumption: the implementation provides all 32 power levels provided by the cc2420 radio chip. Therefore, a power control algorithm for low power consumption can be simulated and tested with AvroraZ.
- Interference avoidance: a noise time trace recorded in real scenarios with interference devices in the ISM band can be introduced in the simulator. This characteristic, together with the 16 multi-channels provided and designed to protect the interferences between different channels in cc2420, can help researchers to perform multi-channel protocol simulations on AvroraZ to avoid interference in ISM band.
- IEEE 802.15.4 implementation: As the implementation 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 simulate 802.15.4 standard implementations that already work in real networks on AvroraZ.

The AvroraZ extension development branch can be tracked at <http://sourceforge.net/projects/avroraZ>

7. REFERENCES

- [1] S. McCanne and S. Floyd. Network simulator ns-2. <http://nslam.isi.edu/nslam/index.php/>
- [2] Chen, G., J. Branch, M. J. Pflug, L. Zhu and B. Szymanski SENSE: A Sensor Network Simulator and Emulator. <http://www.ita.cs.rpi.edu/sense/>
- [3] Peter Jurcik and Anis Koubaa, "IEEE 802.15.4 OPNET simulation model". Available at <http://www.open-zb.net/>
- [4] OMNET++ Discrete Event Simulation System. Available at <http://www.omnetpp.org/>
- [5] Castalia: a Wireless Sensor Network (WSN) simulator based on the OMNet++ <http://castalia.npc.nicta.com.au/>
- [6] J-Sim: A Simulation and Emulation Environment for Wireless Sensor Networks <http://www.j-sim.org/>
- [7] GloMoSim Global Mobile Information Systems Simulation Library. <http://pcl.cs.ucla.edu/projects/glomosisim/>
- [8] QualNet <http://www.scalable-networks.com/>
- [9] P. Levis, N.Lee, M. Welsh, and D.Culler, "TOSSIM: Accurate and scalable simulation for entire TinyOS applications" in Proceedings of SenSys'03, First ACM Conference of ENS Systems, 2003.
- [10] Ben L. Titzer, Daniel K.Lee and Jens Palsberg, "Avrora: Scalable Sensor Network Simulation with Precise Timing". Available at <http://compilers.cs.ucla.edu/avrora>.
- [11] IEEE Std 802.15.4-2006, Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs).
- [12] Texas Instruments CC2420 radio chip. Single-chip 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver. Available at <http://www.ti.com/lit/gpn/cc2420>
- [13] MicaZ platform data sheet. Available to download at www.xbow.com/products/product_pdf_files/wireless_pdf/6020-0060-01_a_micaz.pdf
- [14] Martin Klepal, Rajiv Mathur, Alan McGibney, Dirk Pesch. "Influence of people shadowing on optimal deployment of WLAN access points"
- [15] Lee,H. and Levis,P. (2007). RssiSample application for Tinyos-2.x.<http://tinyos.cvs.sourceforge.net/tinyos/tinyos-2.x-contrib/standford-sing/apps/RssiSample>
- [16] Nick Kingsbury "Approximation formulae for the Gaussian Error Integral Q(x)". <http://cnx.org/content/m11067/latest/>
- [17] Brendan O'Flynn, A. Lynch, K. Aherne, P. Angove, J. Barton, S. Harte, C. O'Mathuna Dermot Diamond, Fiona Regan "The Tyndall Mote. Enabling Wireless Research and Practical Sensor Application Development"
- [18] Jonas Fonseca and Rostislav Spinar. Re-Mote Testbed Framework. <http://code.google.com/p/remote-testbed/>
- [19] Tinyos-2.x Test Acks application. Available at www.tinyos.net/tinyos-2.x/apps/tests/cc2420/TestAcks/