Error Correction in Single-Hop Wireless Sensor Networks - A Case Study

Daniel Schmidt, Matthias Berning, Norbert Wehn
Microelectronic Systems Design Research Group
University of Kaiserslautern
67663 Kaiserslautern, Germany
Email: {schmidt,berning,wehn}@eit.uni-kl.de

***Abstract*—Energy efficient communication is a key issue in wireless sensor networks. Common belief is that a multi-hop configuration is the only viable energy efficient technique. In this paper we show that the use of forward error correction techniques in combination with ARQ is a promising alternative. Exploiting the asymmetry between lightweight sensor nodes and a more powerful base station even advanced techniques known from cellular networks can be efficiently applied to sensor networks. Our investigations are based on realistic power models and real measurements and, thus, consider all side-effects. This is to the best of our knowledge the first investigation of advanced forward error correction techniques in sensor networks which is based on real experiments.**

I. Introduction

Wireless sensor networks (WSN) are the key enabling technology for many applications ranging from health, environment, or traffic monitoring to industrial automation or in military scenarios. These networks consist of nodes which are typically constrained in size and cost which, in turn, leads to a severe limitation of the available energy resources and computational power. The tasks of the individual nodes usually consist of periodic or event triggered transmission of sampled and preprocessed sensor data to a central node where the data is collected and further processed. In most cases, the wireless communication dominates the energy consumption of each wireless sensor node and, thus, limits the maintenance-free life time of individual nodes and the whole network. The optimization focus in these networks clearly lies on the optimization of the communication.

A similar problem exists in cellular networks (for example wireless LANs or cellular telephony), where the energy of mobile devices is constrained, too. *Forward error correction* (FEC) is a very efficient way to increase communication efficiency. By introducing redundancy they allow the receiver to correct errors that occur during transmission, making retransmissions obsolete. This so-called coding gain can be traded off to improve the bit error rate (BER) and frame error rate (FER) at a given transmission energy or to increase energy efficiency for a given FER and BER. Many error correction codes are characterized by their asynchronous complexity. While the encoding process is not computationally challenging, the decoding relies on complex techniques. Advanced error correction codes, such as the Turbo Codes [1] or many standard relevant LDPC Codes [2], use iterative schemes to compute the code word most likely transmitted based on likelihood information for every received bit, the so-called soft information. FEC introduces a computational overhead in both sender and receiver and thus can be seen as a way to trade-off between computation energy and transmission energy.

In WSNs longer distances between a sensor node and the receiving central processing node are normally spanned by using *multi hop routing*. The main argument is that transmission energy for a given distance d scales as d^{α}, where α is the so-called path loss exponent. Under this assumption routing over many very short hops is preferable, as by dividing the distance d in to n short hops of d/n, transmission energy will ideally only increase linearly with the distance between sender and receiver.

This first order theoretical analysis is, however, highly unrealistic as it assumes that all nodes from the sender to the receiver are lined up in equal distances on a linear path. It also completely neglects the energy consumption for reception and forwarding in intermediate nodes, as well as all parts of the energy consumption that do not scale with the path loss exponent, like bias power for example, which can be substantial.

Thus, *precise power models* for both, components and tasks, in a WSN are the most important basis of all energy related design decisions. The main weakness of many previously published studies is that they are based on theoretical models.

In this paper we present an investigation of the applicability and efficiency of forward error correction schemes in a WSN based on real measurements. We implemented and tested several codes (repetition codes with rate 1/3 and 1/6, a convolutional code (K=5, rate 1/3), as well as a UMTS like Turbo Code [3]). We show how soft information needed for advanced FEC schemes can be derived using simple radio front-ends.

In a controlled experiment we employed an automatic repeat request (ARQ) procedure to transmit messages from sensor nodes to a central node with high computational power. On top of that, we tested different coding schemes to protect radio transmission against channel noise. As retransmissions are extremely expensive, the use of proper FEC schemes improve reliability and energy efficiency in WSNs. At an average bit error rate of around 1% the use of a Turbo Code resulted in energy savings of more than 80% compared to the pure

978-3-9810801-5-5/DATE09 © 2009 EDAA

ARQ scheme. The main contribution of this paper is a proof of concept for the applicability of advanced FEC schemes in today's WSNs.

The rest of the paper is structured as follows: in Section II we will briefly summarize previous work on energy efficiency and FEC in WSNs. In Section III we describe the MICAz hardware and its energy consumption in different modes of operation. Section IV describes the channel codes under consideration and compares their capability of preventing frame errors. The controlled experiment is then presented in Section V, followed by a brief overview of the results in Section VI. Final conclusions and an outlook future work are given in Section VII.

II. Previous work

While a lot of research has focused on efficient multi hop routing schemes, there also have been a number of publications rising doubt about the benefits of using very short hops in wireless sensor networks.

Min et al. show that for several real-world radio transmitters the distance-independent overhead in energy consumption alone cancels out the benefits of multi hop transmissions, even in the ideal case [4]. Taking into account the receiver energy further penalizes multi hop schemes.

A precise investigation of the energy consumption of the MICAz, which is one of the most wide spread wireless sensor nodes today, is presented in [5]. The model is based on measurements and data sheets and consists of a static model, modeling the nodes energy consumption in different states, and a dynamic part modeling the energy consumption and timing behavior of recurring tasks. It shows that for this node the receiver energy is not only non-negligible, but also exceeds transmission energy, even at the highest output power (see also Table I). This identifies relaying and retransmissions due to lost frames as the dominant source of energy consumption in WSNs. As a conclusion frame losses and relaying have to be minimized to optimize energy efficiency.

Haenggi lists twelve general reasons not to use too short hops in wireless sensor networks [6], as for example end-to-end reliability, delay, and protocol overhead. In his later work [7] he even gives experimental proof in a setup with ten MICAz nodes that long hop routing can clearly perform better than short hop routing.

However, relatively little work has dealt with channel coding in wireless sensor networks Howard et al. have presented a theoretical analysis of the critical distance at which the transmission energy saved by encoding exceeds the energy overhead in the decoder [8].

In [9] Zhong et al. propose to use channel coding in wireless sensor networks with a star shaped single hop structure, similar to that of today's cellular telephony networks. They propose to have one central node with sufficient computational power and sufficient energy resources to perform the complex channel decoding for incoming packets. For powerful coding schemes, like Turbo Codes [1], the encoding process is simple enough to be done by a wireless sensor node even with its very limited computational power. This way, communication from the nodes to the central base station (the uplink) can be protected using strong error correcting codes while the downlink utilizes higher transmission power to avoid transmission errors. This so-called Single Hop Asymmetric Structure (SHAS) has a number of advantages over multi hop networks, as for example simpler time synchronization, low delay, possibility to use centralized media access control schemes, and many more [9]. In their theoretical analysis they compare a multi hop structure to a SHAS network using a convolutional code (K=7, 1/2 rate) and come to the conclusion that up to a range of 175 meters the SHAS network operates more efficiently than the multi hop structure. Their underlying model makes conservative assumptions and simplifications for the multi hop network, as collisions and RTS/CTS overhead are not taken into account and a perfect synchronization of nodes is assumed.

From all of these results it can be concluded that complex short hop structures are far from being optimal in WSNs. Empirical evaluation of the applicability and efficiency of FEC schemes is, however, lacking. Thus, the question remains open, whether or not SHAS are a feasible solution. This paper contributes to answer this question. In the following sections we investigate error correction capabilities and energy consumption of different codes in more detail.

III. MICAz hardware and restrictions

The MICAz MPR2400 by Crossbow Technology Inc. is a very popular sensor node. Figure 1 shows the main components on the board, the microprocessor Atmel ATMega128L and the wireless transceiver Chipcon CC2420. There is also a logger-flash Atmel AT45DB041B, a serial-number-chip Dallas DS2401, and three LEDs on the node, which are not needed and therefore switched off in all of our experiments to focus on the energy needed for communication and data processing. Via the 51-pin connector different sensor modules can be attached. This way, the platform can be used for various applications.

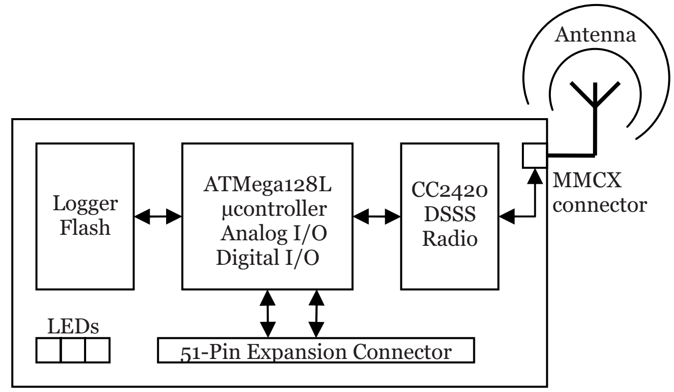

Fig. 1. MICAz sensor node

The 8bit Atmel ATMega128L risc-type microprocessor has 4kBytes of internal SRAM, 4kBytes of internal EEPROM and 128 kBytes of flash memory to store the program. It is clocked at a frequency of 7.37 MHz. The CC2420 transceiver operates

MICAz power state	current @3V [mA] total	μ-controller	CC2420
Standby	0.02	0.01	0.01
Active	8.01	8.0	0.01
Transmit [-25dBm]	16.50	8.0	8.50
Transmit [0dBm]	25.40	8.0	17.40
Receive	26.80	8.0	18.80

TABLE I
MICAZ POWER CONSUMPTION [5] IN DIFFERENT MODES OF OPERATION, SEPARATED FOR TRANSCEIVER AND μ-CONTROLLER.

in the 2.4 GHz band. The maximum data rate is specified with 250 kbit/s [10]. A separate 128-bytes buffer for transmit and receive operations is used for communication with the microcontroller. The lowest accessible layer offered by the CC2420 is the physical protocol data unit (PPDU), shown in Figure 2.

It has a maximum length of 133 bytes. 5 bytes are needed as Synchronization Header (SHR) for Preamble Sequence (PS) and Start of Frame Delimiter (SFD). The next byte is the Physical Header (PHR) storing the frame length of the following MAC Protocol Data Unit (MPDU) of up to 127 bytes. The Frame Check Sequence (FCS) contains a 2 byte CRC-Checksum, which is calculated by the CC2420 transceiver. Thus, payload data is limited to 125 bytes/frame.

On frame reception, the CRC checksum is also checked by the CC2420 transceiver. Even in case of a bad CRC the microcontroller can still access the received bit-sequence. Soft information is, however, not available, which is typical of the simple radio interfaces in today's sensor nodes. As the availability of soft information is crucial for advanced coding schemes this poses a severe limitation. We will show a pure software based method to derive soft information, which can also be applied to other sensor nodes.

The power consumption of the MICAz depends on its mode of operation. The most important modes are summarized in Table I along with the total current drawn from the 3V source and the current drawn by the μ-controller and the CC2420 transceiver, respectively. All values were taken from [5] and are based on measurements and data sheet information. The transmit power of the transceiver frontend can be adjusted in 8 steps from -25dBm to 0dBm. Even at highest output power the power consumption of the node is lower than during frame reception.

Prior to frame transmission or reception, the transceiver performs a calibration of 0.192 ms, which corresponds to the transmission time of 14 bytes of data. Before frame transmission a clear channel assessment (CCA) is done for 0.128 ms to avoid collisions. During these times the CC2420 transceiver is in receive mode, drawing a current of 18.8 mA from a 3V source.

IV. PERFORMANCE OF DIFFERENT CODES

To evaluate the performance of error correction codes in wireless sensor networks, we implemented the following codes on the MICAz, using standard C:

- Two repetition codes with rate 1/3 and rate 1/6 (Rep 1/3, Rep 1/6). Both encoding and decoding are very easy as simply every byte of the data is repeated 3 times or 6 times, respectively. Bit errors can then be recovered by a simple majority voter for every single bit. Due to its low complexity this code could even be used for communication between sensor nodes.
- The convolutional code (CC) under consideration has constraint length $K = 7$, rate 1/3, and does not use puncturing. It is part of many IEEE standards [11].
- A Turbo Code (TC) served as the third code under consideration. Some restrictions were made to reduce implementation complexity: input length was fixed to 20 bytes and a simple relative prime interleaver was used (starting index s=3; relative prime p = 23). The code rate is 1/3 and no puncturing is used.

To overcome the problem that using the low complexity transceiver the MICAz offers it is not possible to receive soft channel information for the Turbo Code decoder in the receiver, we used the following modified Turbo Code (TC*): the codeword generated by the Turbo encoder is repeated two times, decreasing the code rate to 1/6. Thus, at the receiver side hard information for every bit is received twice. By adding both values a certain degree of soft information can be gathered, allowing for the use of state of the art decoders. It is worth noting, that next-generation wireless sensor nodes might be equipped with transceivers allowing a direct extraction of soft channel information. This would render the proposed codeword repetition and the introduced energy overhead and latency obsolete. The advantage of the proposed TC* lies in its direct applicability in todays wireless sensor networks.

In a SHAS network, as proposed by Min et al. [9], the base station has enough computational resources to perform the decoding even of the more advanced of these codes. For the downlink or for communication between mobile nodes with lower computational power the repetition codes can be an alternative to uncoded communication.

Table II summarizes the characteristics of each of the implemented codes. The energy is calculated as the sum of the energy used for encoding 20 bytes of data using the specified code and frame transmission. This is sufficient for many practical applications (e.g. transmission of measured temperature, pressure or humidity values, and such). For 20 bytes net data a rate 1/3 code sends 60 bytes of payload data, a rate 1/6 code sends 120 bytes, which still fit into one PPDU. As a consequence of the high overhead for frame transmission (transceiver calibration, CCA, frame header), the energy consumption for sending one frame using the Rep 1/3 code, for example, is not three times as high as in the uncoded case. The additional energy spent for the coding is to be counterbalanced by the coding gain lowering the transmission error rate. In the last column the size of the encoder program on the MICAz is given. It can be seen that even the most complex of the codes under consideration easily fits in the ROM of the MICAz.

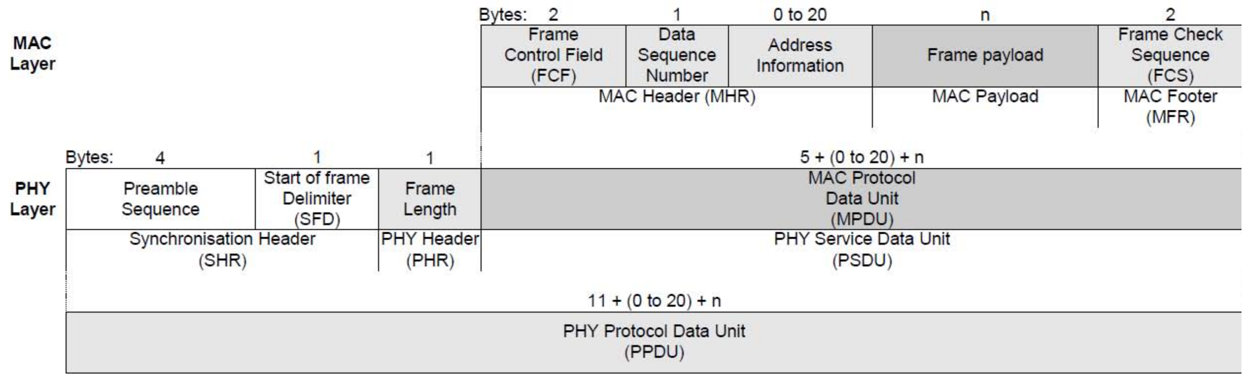

Fig. 2. Physical protocol data unit used by MICAz [10]

code	rate	energy[μJ]	ROM[B]
uncoded	1/1	104.73	4136
Rep 1/3	1/3	128.97	4245
Rep 1/6	1/6	233.18	4245
CC	1/3	254.88	4696
TC	1/3	282.79	4798
TC*	1/6	387.40	4798

TABLE II
COMPARISON OF IMPLEMENTATION COST AND TRANSMISSION ENERGY INCLUDING ENCODING AT SAME SIGNAL STRENGTH. LOWER RATE CODES OFFER HIGHER RELIABILITY.

Experimentally obtained performance curves for all of the above-mentioned codes are given in Figure 3. We measured the error correction capabilities of all the different codes in a lab environment using frames with known payload data. This way the base station could plot the frame error rate over the bit error rate in received packages after decoding. No ARQ was used in this test, such that every unrecoverable transmission error results in a frame error. Note that, without any coding every single bit error will result in an unrecoverable frame error.

While the rate 1/3 Turbo Code (TC), the convolutional code (CC), as well as the repetition codes (Rep 1/3 and Rep 1/6) do not provide a very strong protection against transmission errors, the Turbo Code with soft decision decoding (TC*) maintains almost error free frame reception even at bit error rates of up to 20%. Given the transmission cost per frame and the implementation complexity from Table II we chose the repetition code Rep 1/3 as the most simple code and the soft decision Turbo Code TC* because of its outstanding performance for a further experiment. Here, we employed an ARQ scheme with and without additional forward error correction to transmit messages in a noisy environment from energy limited mobile nodes to a powerful base station and measured the energy consumption and message error rate. This experiment is described in the next section.

V. RELIABILITY AND ENERGY CONSUMPTION: A CONTROLLED EXPERIMENT

Using three MICAz nodes set up in a lab environment we tested the energy efficiency of different error correcting schemes. All of the nodes were programmed to generate one package (with known content) per second and transmit it to a central station with high computational power and energy resources, the base station. After transmission the nodes wait for an acknowledgment frame (ACK) for a period of up to 250ms. If no ACK is received an automatic retransmission of the package is performed for at most three times (ARQ). If even after the fourth transmission no ACK is received the message is considered lost. As soon as an ACK is detected, the corresponding node goes to standby mode to save energy.

All three nodes use the described ARQ scheme. Their configuration is as follows:

- Node 1 relies solely on the ARQ scheme. No forward error correction is performed.
- Node 2 employs the Rep 1/3 repetition code to reduce the number of retransmissions.
- On top of the ARQ scheme, node 3 uses the TC* Turbo Code which offers the highest error correction capability of the codes under consideration.

The base station consists of a MICAz connected to a standard PC. The frames are received by the MICAz and the decoding is performed by the PC. In case of a correct decoding an ACK is sent via the radio interface of the MICAz. The PC was also used to keep statistics of BER, FER, and message error rate (MER), the rate of messages that could not be successfully received even after the fourth retransmission.

The base station was mains powered, the mobile nodes were powered by a rechargeable lithium polymer battery. According to measurements, the available capacity to the nodes is approximately 1200mAh, before the voltage drops too low for the node to continue its operation.

All nodes were positioned in a lab environment with significant channel disturbances in the 2.4 GHz band (mostly from WLANs) such that the average BER in received frames for all nodes was about 1% throughout the experiment. We left the nodes unattended for 120 hours and afterwards measured the left over capacity in the rechargeable batteries.

VI. RESULTS

Results are shown in Table III. As the node without any error correction had fully depleted its battery after 48.2 hours,

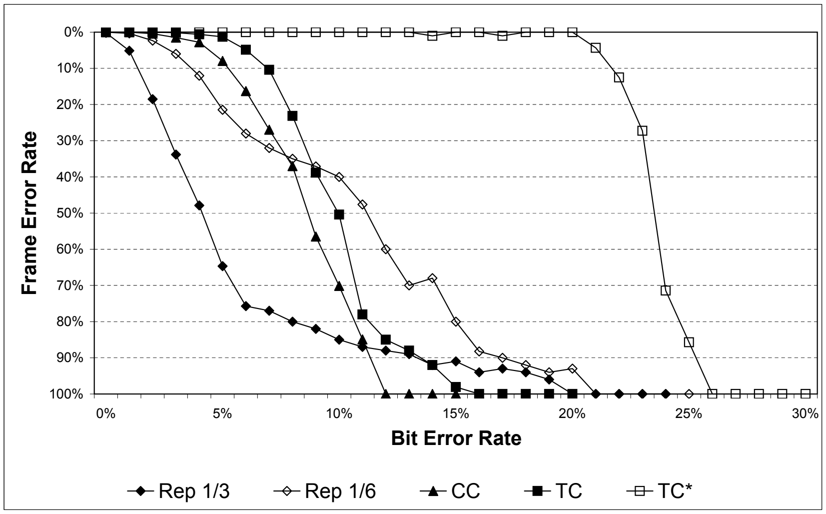

Fig. 3. Error correction capability for different codes. Rate of unrecoverable frame errors after decoding is shown over the percentage of bit errors in the received frame. Curves further to the right indicate better correction performance.

we linearly extrapolated the results for a runtime of 120 hours for comparison. It can be seen that the average BER of all nodes over the whole runtime was a little more than 1%. Due to channel variations the node using the TC* even had to deal with slightly worse conditions than the other two nodes. The FER for all nodes is quite high, which is attributed to the fact that neither the ACK nor the frame header are protected by any coding mechanism. If for example the frame length field in the PPDU is corrupted the frame is inevitably lost, necessitating a retransmission.

The message error rate decreases with increasing error correction capabilities of the used code. Only 21.6% of all messages could be succesfully transmitted without coding, while the node using the TC* can deliver 96.9% of all messages. This is also reflected in a very low number of ARQs. Most interesting of all is that the consumed energy during the 120 hour period is lowest for the node using the TC*, although the encoding and transmission of the encoded data introduces an overhead. Due to the very low number of necessary retransmissions this node can spend much more time in standby, thus effectively saving energy. Also the Rep 1/3 code performs very well at this bit error rate. Compared to the node using plain ARQ without FEC the repetition code and the Turbo Code consume roughly 80% and 85% less energy, respectively.

Taking into account the high number of messages that could not be delivered within four attempts and calculating the energy per successful message transmission the picture gets even worse for the pure ARQ scheme.

VII. Conclusions and Future Work

It is widely accepted that in current wireless sensor networks forward error correction is not applicable or beneficial as they require high computational power and cause a high energy overhead. In this paper we showed that in contradiction with common belief advanced forward error correction techniques are a promising alternative in star shaped single hop networks with one powerful base station. To the best of our knowledge it is the first such investigation which is based on real experiments and measurements. To achieve very high error correction capability it is necessary to employ decoding procedures based on soft channel information. As soft channel information was not directly accessible using the low complexity transceiver and had to be emulated by our proposed TC* scheme, the measured energy consumption of this code can be seen as an upper bound compared to a hardware based solution.

We identified retransmissions resulting from frame errors as the dominating source of energy consumption in wireless networks. Thus, even a simple repetition code, which can also be used in multi-hop structures, can be more efficient than an ARQ scheme without forward error correction capabilities. The optimal trade off between coding strength and coding overhead will vary strongly from one node to another depending on channel quality and distance to base station and has to be adapted at run time. This will be part of our next investigations.

Acknowledgment

This work was supported and partially funded by the Research Center for Ambient Systems (AmSys) at the University of Kaiserslautern, as well as by the BMBF (German Federal Ministry of Education and Research) within the framework of the Bilateral German-Hungarian Collaboration Project on Ambient Intelligence Systems (BelAmI).

References

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit error-correcting code: Turbo code," *Proceedings of the 1993 IEEE International Conference on Communications*, pp. 1064–1070, May 1993.

[2] R. Gallager, "Low-density parity-check codes," *IRE Transactions on Information Theory*, vol. 8, pp. 21–28, January 1962.

code	avg. BER	FER	MER	$\frac{Frames}{Message}$	consumed energy [J]	$\frac{Energy}{succ.Message}$ [μJ]
uncoded**	1.23%	89.80%	78.4%	2.34	32,239	346,049
rep. 1/3	1.26%	43.30%	9.7%	1.18	6,566	16,842
TC*	1.36%	21.93%	3.1%	1.12	4,936	11,798

TABLE III
ENERGY CONSUMPTION IN 120 HOURS (**EXTRAPOLATED, BATTERY DEPLETED AFTER 48.2H)

[3] E. T. S. Institute, "Universal mobile telecommunications system (UMTS): Multiplexing and channel coding (FDD), 3GPP TS 125.212 version 3.4.0," *Proceedings of the 1993 IEEE International Conference on Communications*, pp. 14–20, September 2000.

[4] R. Min and A. Chandrakasan, "Energy-efficient communication for high density networks," in *Ambient Intelligence: Impact on Embedded System Design*, T. Basten, M. Geilen, and H. de Groot, Eds. Kluwer Academic Publishers, 2003, pp. 295–314.

[5] F. Walther, "Energy modelling of MICAz: A low power wireless sensor node," Technical Report, University of Kaiserslautern, http://www.eit.uni-kl.de/wehn/files/reports/micaz_power_model.pdf, February 2006.

[6] M. Haenggi, "Twelve reasons not to route over many short hops," *IEEE VTC '04 Fall, Los Angelos, CA*, pp. 3130–3134, September 2004.

[7] M. Haenggi and D. Puccinelli, "Routing in ad hoc networks: A case for long hops," *IEEE Communications Magazine*, pp. 93–101, October 2005.

[8] S. L. Howard, C. Schlegel, and K. Iniewski, "Error control coding in low-power wireless sensor networks: When is ECC energy-efficient?" *EURASIP Journal on Wireless Communications and Networking*, pp. 1–14, 2006.

[9] L. C. Zhong, J. M. Rabaey, and A. Wolisz, "Does proper coding make single hop wireless sensor networks reality: The power consumption perspective," *IEEE Wireless Communications and Networking Conference*, vol. 2, pp. 664–669, March 2005.

[10] Texas Instruments, "2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver," Retrieved May 2008 from http://focus.ti.com/lit/ds/symlink/cc2420.pdf, March 2007.

[11] A. Batra *et al.*, "Multiband-OFDM physical layer proposal for IEEE 802.15 task group 3a," *IEEE P802.15-03/268r3*, March 2004.