Adaptive Channel Hopping for Interference Robust Wireless Sensor Networks

Suk-Un Yoon, Robert Murawski, Eylem Ekici
Department of Electrical and Computer Engineering
The Ohio State University, Columbus, OH
Email: {yoons, murawskr, ekici}@ece.osu.edu

Sangjoon Park, Zeeshan Hameed Mir
Electronics and Telecommunications Research Institute
Daejeon, South Korea
Email: {sangjoon, zhmir}@etri.re.kr

***Abstract*—In this work, an *Adaptive Channel Hopping* (ACH) mechanism for sensor networks is proposed to avoid interference from other sources and narrow-band jamming. Under unfavorable channel conditions, ACH lets sensors switch to a new operating channel. ACH reduces the channel scanning and selection latency by ordering available channels using link quality indicator measurements and weights. The proposed ACH scheme is evaluated through simulations and a hardware implementation, which suggest low latency and high channel selection quality even in very adverse conditions.**

***Index Terms*—Wireless Sensor Networks; Interference; Adaptive channel hopping.**

I. INTRODUCTION

With the increased usage of ISM band, contention among different networks becomes an important problem. Wireless Sensor Networks (WSN) must be able to compensate for such contention cases to sustain high performance. Many ISM-based solutions implement their own interference avoidance methods. In the 2.4 GHz ISM band, Wi-Fi, WirelessUSB, and 802.15.4 [1] use direct-sequence spread spectrum (DSSS) while Bluetooth uses frequency-hopping spread spectrum (FHSS) [2]. Wi-Fi utilizes (CSMA) to reduce the probability of collisions within as well as across networks. Bluetooth v.1.2 defines an adaptive frequency hopping algorithm which allows Bluetooth devices to coarsely classify channels and allows using only good channels. WirelessUSB devices allow change of the operating channel if the link quality becomes unacceptable. ZigBee provides a collision-avoidance algorithm similar to Wi-Fi [2], [3]. To minimize data loss caused by collisions, ZigBee relies on its low duty cycle and collision-avoidance algorithms, but does not change channels, hence cannot avoid heavy interference or narrow-band jamming.

When considering the coexistence of WSNs and other ISM-based solutions, one of the main concerns is performance degradation due to local interference. This issue has been researched for large scale 802.15.4 multi-hop sensor networks. In [4], the authors propose an adaptive channel allocation scheme, which allows nodes experiencing significant interference to switch to new frequency channels with less congestion. To minimize the effect of Wi-Fi interference in 802.15.4 WSNs, an interference detection and avoidance mechanism in proposed which selects the radio channel that is least likely to have interference [5]. To minimize interference of 802.11b/g, frequency hopping schemes that utilize the four guard channels of IEEE 802.11b/g have been proposed [6]. In this paper, we propose *Adaptive Channel Hopping*, a simple and reliable channel hopping algorithm that can be applied to existing WSN implementations. The algorithm has been tested with simulations and implemented on a hardware platform.

II. ARCHITECTURE

A pre-determined sequence of channels in a given portion of the unlicensed ISM band is utilized. All nodes have a single communication interface and run-time channel selection capability via software configuration. All nodes are of similar hardware and configuration. The WSN consists of multiple devices organized into groups. Each group contains a single parent and one or more child nodes. The parent node is the coordinator for all frequency hopping events. Our algorithm is applicable to MAC layer protocols that have the following properties: time synchronization, neighbor discovery, and the the ability to organize users into separate timeslots for channel hop synchronization. Such protocols include 802.15.4 [1] in a single-hop topology and Sectored Antenna MAC (SAMAC) [12]. Each group becomes active in a given time slot. Combining all $N_{timeslot}$ time slots within a network forms a superframe that is repeated every $N_{timeslot} \times t_{timeslot}$ seconds, where $t_{timeslot}$ is the duration of each time slot. Time slot schedules in which a group of nodes become active are assumed to be available in all nodes.

III. ADAPTIVE CHANNEL HOPPING

A. Hopping Sequence Generation

The proposed ACH algorithm utilizes a channel hopping sequence for coordination between the parent and child nodes of each group. The channel hopping sequence is periodically generated every $T_{LQIreport}$ seconds by the parent node and distributed to all children. Parents generate this sequence first by gathering a Link Quality Information (LQI) map from each child which describes the current state of all N available channels. The complete LQI map gathered at the parent contains LQI values represented as LQI_i^k, where i and k are the channel and node IDs, respectively. For the current operating frequency, the LQI information can be estimated based on the Signal-to-Noise (SNR) ratio for recently received packets. For the remaining $N-1$ channels, nodes can act as

978-1-4244-6404-3/10/$26.00 ©2010 IEEE

an energy detectors and estimate the current noise floor. To improve the accuracy of these readings, children scan the N channels m times every $T_{LQIreport}$ seconds. An additional weighting factor, W^k, can be defined for each node k. If all nodes have the same priority, the weight can be set as $W^k = 1, \forall k$. W^k can also be chosen proportional to the traffic carried by a node k, i.e., $W^k =$ number of leaf nodes. Nodes with a large W^k will have a greater effect on the aggregate LQI (ALQI) calculation. The ALQI for each channel can be calculated as shown in Equation 1.

$$ALQI_i = \frac{\sum_{k=1}^{K} LQI_i^k W^k}{\sum_{k=1}^{K} W^k}, \tag{1}$$

where $i \in N$, K is the total number of group members. The channel hopping set is a list of available channels, sorted by the calculated ALQI values. The channel with the highest ALQI value is first in the channel hopping sequence.

B. Channel Hopping Algorithms

Parents utilizing ACH are assumed to have complete knowledge of their children. Coordination between the parent and children is handled through a set of control messages as defined below:

1) *Channel Hopping Command (CHC)* used by parents to:
    - Announce their existence on the operating channel.
    - Send directives to change the operating channel.
    - Send directives to confirm a clear operating channel.
2) *Channel Hopping Reply (CHR)* used by children to:
    - Request channel hopping.
    - Respond to the hopping command with the channel availability for the group.

When the parent detects interference sufficient to cause link quality degradation, loses communication with its child nodes for a pre-determined amount of time, or receives channel hopping requests from children, the sensor group tries to change its operating channel from the current operating frequency (f_c) to a newly selected frequency (f_n). The channel hopping scheme is Three-Way Handshake which includes the channel hopping command, reply of channel availability, and confirmation of channel use. The channel hopping algorithms for parent and children are as follows:

The channel hopping algorithm for parents is shown in Figure 1. When the operating channel is no longer available due to severe interference for more than the threshold time ($t_{LQI_i^k<LQI_{LB}} \geq t_{threshold}$, $LB : Lower\ Bound$) or the parent loses communication with its children for threshold time, the parent broadcasts CHC on the current channel, f_c, to indicate a channel hop to the next channel f_n in the channel hopping sequence. CHC is used to coordinate the channel hopping for all nodes within a group and is broadcast b (default 3) times every T_M seconds. Upon hearing CHC, all children reply with a CHR on f_n. All control messages are transmitted via a CSMA/CA mechanism. If the parent receives CHRs from all its children, it broadcasts confirmation CHCs and then continues normal data communication. If some children fail to respond within T_W seconds, it repeats the channel hopping procedure with the next frequency channel in the pre-defined channel hopping sequence.

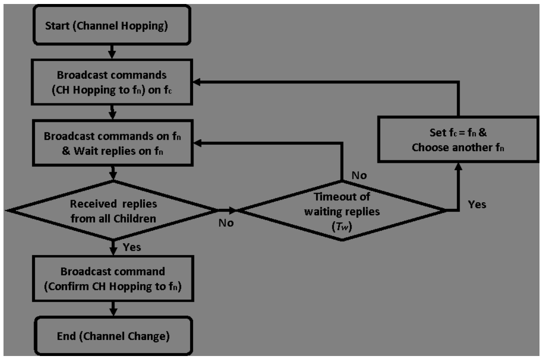


Fig. 1. Channel Hopping Algorithm for Parent

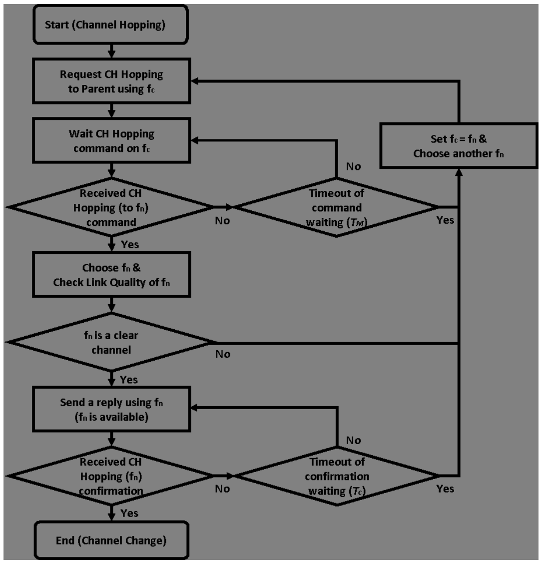


Fig. 2. Channel Hopping Algorithm for Children

The channel hopping algorithm for children is shown in Figure 2. When a child detects severe interference for threshold time ($t_{LQI_i^k<LQI_{LB}} \geq t_{threshold}$), it transmits a CHR to the parent. After transmitting this message, it waits for a predefined timeout period, $T_{Mb} = T_M \times b$. If a CHC is not received within this period, it continues scanning channels following the channel hopping sequence described previously. When a child receives a CHC from its parent, and the LQI of the received packet is sufficient ($LQI_i^k \geq LQI_{LB}$, $UB : Upper\ Bound$), the child node responds to the parent with a CHR. If a confirmation message is not received within T_C seconds, the channel hopping continues. Due to the spatial distance between wireless nodes, it is possible that a child experiences severe interference, whereas others in the same group do not. Due to the circular nature of the channel hopping sequence, we can ensure that the parent and child will converge to either a new frequency channel that is clear of interference,

or the same channel if the interference was temporary. The convergence of a group onto a single channel is handled by carefully selecting the timeout values for channel switching. The timeout values are chosen as follows:

$$\begin{aligned} &T_C \geq (T_M + t_S) \times N, \\ &T_W \geq T_C \times 2 \geq (T_M + t_S) \times N \times 2, \end{aligned} \tag{2}$$

where t_S is channel switching time, and N is the number of channels to scan. For example, after a channel selection of a parent, the parent periodically (T_M) broadcasts CHCs in the pre-determined time slot of the superframe. The children circularly and incrementally scan the channels until they receive a CHC. After receiving a CHC and sending CHR for the channel availability, children wait for a confirmation message from the parent. During the confirmation message wait time (T_C), the parent can collect replies from all its children. If the parent does not collect replies from all children within waiting reply timeout (T_W), it then switches to the next channel in the sequence. The *channel hopping latency* is defined as time from sending the first CHC until all children receive a confirmation message. If the parent and children converged on the K^{th} channel of the sequence, then the channel hopping latency is bounded as follows:

$$\begin{aligned} K \cdot T_W + T_{Mb} \geq L_{hopping} &\geq (K-1) \cdot T_W + T_{Mb} \\ &\geq (K-1) \times T_C \times 2 + T_{Mb} \\ &\geq (K-1) \times (T_M + t_S) \times N \times 2 + T_{Mb}. \end{aligned} \tag{3}$$

C. Packet Model and Channel Hopping Latency

The packet model of IEEE 802.15.4 is proposed and mutual interference between IEEE 802.15.4 and IEEE 802.11b is analyzed in [7]. An average packet transmission delay of WSNs is defined as the time from placement of a packet in the sender queue to the time of ACK reception. Then, the average transmission delay is obtained as $E[T] = t_f \sum_{i=0}^{\infty} iP^i(1-P) + t_s = \frac{P}{1-P} t_f + t_s$, where P is the packet error rate, t_f and t_s are the time required for unsuccessful and successful transmission [7]. The channel hopping latency, L, at the K^{th} channel selection of the parent can be expressed as

$$\begin{aligned} L = (K-1) \times T_W + (n-1) \times T_M + T_B \times 2 \\ + T_{PR} + E[T] \times C + T_{PC} + T_{Mb}, \end{aligned} \tag{4}$$

where n is the number of CHCs when the last child receives a command, C is the number of children receiving the CHC at the n^{th} CHC broadcast, T_B is an average packet broadcasting delay, T_{PR} is CHR processing time, and T_{PC} is processing time of a confirmation message.

IV. WSN SIMULATION AND RESULTS

To simulate the Adaptive Channel Hopping algorithm, we developed a custom network model using the WSN simulator Prowler [8]. We simulated a single group consisting of one parent and 19 children using 802.15.4. All nodes are within each other's communication range. The processing times, T_{PR} and T_{PC}, were set to 192 μs, the channel switching time, t_S, to 500 μs, and the command interval T_M to 10 ms. Data and control packets are 120 and 40 bytes in length, respectively. Reported results reflect averages of 20 simulation runs.

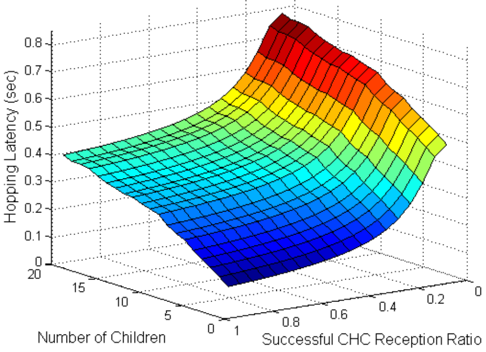


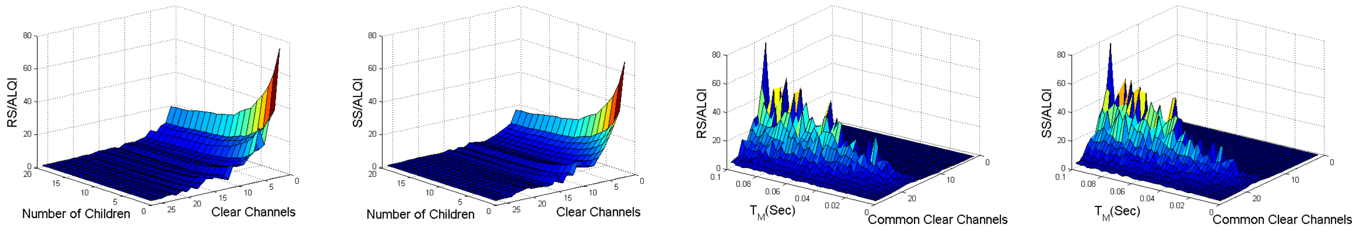
Fig. 3. Channel Hopping Latency with Successful CHC Reception Ratio

Under ACH, children may receive a command message at different times due to their different timeouts or transmission errors. To see the effect of the number of commands in hopping latency, we vary successful CHC message reception ratio of children P_C between 0.05 and 0.95 and observe the channel hopping latency. The number of channels, N, is fixed at 27 which is half of the maximum number of command broadcasts in the proposed system. In Figure 3, the channel hopping latency ranges between 0.084 and 0.7625 sec. As P_C decreases from 1 to 0.4, the latency increases at a low rate. However, there is rapid latency increase below 0.4 of P_C due to low probability of CHC reception. If the parent does not receive any reply message from the last group member after 54 command broadcasts ($T_W = (T_M + t_S) \times N \times 2 = 567ms$), the parent selects a new channel and broadcasts CHC messages again. For every failure of channel locking, the latency increases by 0.567 seconds.

The channel hopping sequence generation algorithm, ALQI, is proposed to order channels to reduce channel hopping latency. To compare our proposed scheme with, we choose two competitor channel selection algorithms:

- Random Sequence (RS): The channel hopping sequence is randomly generated from the target channels.
- Sequential Sequence (SS): The channel hopping sequence is generated in sequentially and circularly increasing order from the current channel.

In the first comparison, the total number of channels is 27, where one is the current channel and the remaining ones can be used for channel hopping. Based on the correlation between LQI and PRR (packet reception ratio) [9], the channels with high LQI value are configured to have high PRR. In the comparison, all nodes in the sensor group have the same clear channels with high LQI values ($\geq LQI_{UB} = 100$ with $PRR \geq 0.95$) and their number ranges from 2 to 27. The RS and SS are configured to have the same timeout mechanism as ALQI. In comparison of RS/ALQI and SS/ALQI, the ratios greater than 1 mean that ALQI has superior performance. In comparison results in Figure 4, the ratios are greater than 1 in all ranges. We note that ALQI requires channel information



(a) Comparison of RS/ALQI

(b) Comparison of SS/ALQI

(c) RS/ALQI with T_M

(d) SS/ALQI with T_M

Fig. 4. Channel Hopping Latency Comparison

sharing and sorting in normal situation which is not used in RS and SS. Despite the overhead of information sharing during regular communication, the latency is significantly reduced by the prompt channel locking at the channel switching. Without shared channel information, RS and SS consume large amounts of time for channel locking and the largest latency of RS and SS is 14.772 sec ($T_W \times 26 + T_{Mb}$) if there is at least one clear channel in the channel selection. The results in Figure 4(a) and 4(b) show that ALQI has superior channel hopping latency for 1-20 clear channels. This means the sensor group can have shorter channel hopping latency with ALQI when more than 25% of channels are used. One channel of Wi-Fi consumes about 25% of the channels in 2.4GHz ISM bands. Thus, our proposed method will have great benefits for WSNs coexisting with Wi-Fi. In case the number of clear channels is more than 20, ALQI has relatively small benefits because of high channel locking probability of RS and SS. For less than 20 clear channels, the comparison ratios converge to 1 because there are enough clear channels. In the comparison with the number of children, ALQI significantly outperforms RS and SS as the number of children decreases.

In real environments, sensor nodes might receiver different levels of interference depending on their location. To see the effects of interference on on ACH performance, we configure each child to have a different interference level on each channel. Each child is configured to have the same number of clear channels though the clear channels might be in different frequency bands. When there is no common clear channel for the group members, the latency is configured to be 14.742 sec, which is 26 rounds of channel selection and the comparison ratios converge to 1. In this comparison, instead of the number of children, we use another parameter, the command message interval (T_M), which is an important parameter in the proposed system to minimize channel hopping latency. In the comparison, the number of children is fixed at 20. In comparison results with command message interval in Figures 4(c) and 4(d), the ratios are greater than 1 as long as the sensor group has at least one common clear channel among target channels. In the range of short command message interval, ALQI has relatively small benefits on channel hopping latency. Because the interval of the command messages decreases, the latencies of RS, SS, and ALQI decrease together. However, there are limitations to the reduction of the interval such as high spectrum usage and power consumption. Furthermore, the interval should be greater than channel switching time ($T_M \geq t_S$) and the average packet transmission delay of sensor nodes ($T_M \geq E[T]$) for a reliable channel locking. In [7], average transmission delay of the IEEE 802.15.4 under IEEE 802.11b interference is greater than 6 ms, in which range proposed ALQI has great performance benefits (short channel hopping latency). The maximum benefits are located around 17 common clear channels. The location of maximum benefit is dependent on the number of children due to the possibility of existence of common clear channels among children. When the number of children decreases, the peak region shifts to smaller values than 17 whereas when the number of children grows, the peak shifts to larger values than 17. Overall, RS and SS show similar performance in latency comparisons.

Finally, we compare the average rate of successful message delivery over communication channels. To compare overall throughputs ρ, we use 27 channels, of which 17 are clear, 5 children, 10ms command message intervals, a PRR of 0.95 or 0.35 before switching, and a PRR between 0.95 or 1 in the clear channel after switching. Table I shows the throughput and average channel hopping latency comparisons. Before channel switching, the throughputs are 35.12 Kbps with 0.95 PRR and 12.59 Kbps with 0.35 PRR. After the channel hopping is complete, each channel hopping mechanism converges to a channel with a PRR greater than 0.95. Due to the improved channel hopping sequence of the ALQI method, less time is required to find a new channel with both similar and different levels of interference.

V. HARDWARE IMPLEMENTATION

To verify the functionality of our frequency hopping mechanism, we implemented ACH on a custom hardware (Figure 5) platform developed by Electronics and Telecommunications Research Institute (ETRI). All nodes utilize a TI MSP430 [10] series processor for computation and a TI CC2420 [11] transceiver for communications. The nodes utilize the SAMAC [12] protocol, which is a TD-CSMA MAC protocol that separates users into TDMA groups that compete

TABLE I
THROUGHPUT AND HOPPING LATENCY COMPARISONS

Throughput (Kbps)	**Before Switching**		**Same levels of interference**		**Different levels of interference**	
	PRR 0.95	PRR 0.35	Latency (Sec)	After switch	Latency (Sec)	After switch
ALQI	35.12	12.59	0.191	35.99	2.627	35.99
RS			0.436	35.98	7.251	35.81
SS			0.433	35.97	7.143	35.81

for resources via CSMA. Each group consists of a parent and one or more children. This structure conforms to the structure of the ACH mechanism in that frequency hopping events occur within a given group and are directly controlled by the group parent. The nodes can communicate on one of 16 frequency channels available in the ISM band.

Fig. 5. Sensor Node Hardware

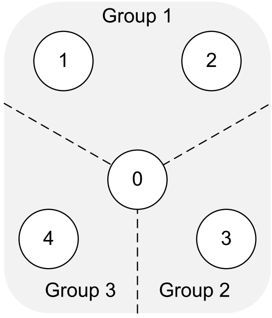


Fig. 6. Testbed Topology

To test ACH, we incorporated the ability to drop a frequency channel in software by turning off the CC2420 transceiver for selected channels. When the currently used channel is disabled, communication between all nodes is cut off. According to ACH, the loss of communication triggers a frequency hopping event.

In our test bench evaluations, we utilize a topology of five nodes and three groups. Sink Node 0 is the parent for each of the three groups and each group is separated into different sectors of the sink nodes's three-sectored antenna, as shown in Figure 6. Each of the three groups is assigned a different TDMA time slot of 50 ms, resulting in a TDMA superframe of 150 ms. Each of the four sensor nodes generate a 110-byte data packet every 100 ms. After 100 superframes, the manual channel hopping is triggered and all nodes disable the default frequency channel, channel 15. The loss of the primary channel results in the loss of all data and control packets between parent and children. This loss of communication triggers the frequency hopping event as described in section III after 10 superframes (or 1.5 seconds) of inactivity. As all channels are relatively equal in this scenario due to the limited interference in our lab environment, the default sequential list of channels is used. After the frequency hopping event is triggered, the sink node broadcasts in five consecutive superframes the CHC message that indicates to available children that a channel hopping event is taking place. For each of the tests, each group converged to the next available channel (16). After converging to the next frequency channel, the sink node indicates in the next five consecutive superframes that a new channel has been selected for communication. In each test run, the entire channel hopping event required 22 superframes to complete, where children converge to channel 16 in the two additional superframes. In total, the frequency hopping event introduced 3.3 seconds of latency to the system. This value is increased from the simulation results due to the increased channel switching times for the underlying SAMAC protocol.

VI. CONCLUSION

In this paper, a simple and reliable adaptive channel hopping scheme with standard compatibility is introduced for interference robust WSNs. The proposed ACH method is designed to avoid the interference from other networks in the ISM band based on link quality estimations. Simulation results confirm superior performance of ACH to other channel search methods under all simulated scenarios. We also implemented a proof-of-concept of the proposed ACH scheme in a testbed. Our future work will concentrate on the implementation and testing of the proposed method under real operating environments.

ACKNOWLEDGMENT

This research was supported by the Dual Use Technology Program, South Korea (06-II-LC-01, Surveillance and Reconnaissance Sensor Network Development).

REFERENCES

[1] Institute of Electrical and Electronic Engineers, IEEE STD 802.15.4, 7 June 2006.
[2] Ryan Winfield Woodings and Mark Gerrior, Avoiding Interference in the 2.4-GHz ISM Band, http://www.wirelessnetdesignline.com/howto/60401206.
[3] ZigBee Alliance, ZigBee and Wireless Frequency Coexistances, ZigBee White Paper, June 2007.
[4] Chulho Won, Jong-Hoon Youn, Hesham Ali, Hamid Sharif, and Jitender Deogun, Adaptive Radio Channel Allocation for Supporting Coexistence of 802.15.4 and 802.11b, IEEE Vehicular Technology Conference Vol.4 pp.2522-2526, Sep. 2005.
[5] Razvan Musaloiu-E. and Andreas Terzis, Minimising the effect of WiFi interference in 802.15.4 wireless sensor networks, Int. J. Sensor Networks, Vol. 3, No. 1, 2008.
[6] Aamir Mahmood, Frequency hopping in wireless sensor networks, MS Thesis, HELSINKI UNIVERSITY OF TECHNOLOGY, March 2009.
[7] Soo Young Shin, Hong Seong Park, and Wook Hyun Kwon, Mutual interference analysis of IEEE 802.15.4 and IEEE 802.11b, Computer Networks Volume 51, Issue 12, 22 August 2007.
[8] G. Simon, P. Volgyesi, M. Maroti, and A. Ledeczi, Simulation-based optimization of communication protocols for large-scale wireless sensor networks, IEEE Aerospace Conference, Vol3 pp.1339-1346, March 2003.
[9] Kannan Srinivasan and Philip Levis, RSSI Is Under-Appreciated, In Proceedings of the Third Workshop on Embedded Networked Sensors (EmNets), 2006.
[10] Texas Instruments, MSP430 Mixed Signal Microcontroller, July 2001.
[11] Chipcon AS SmartRF CC2420 Preliminary Datasheet (rev 1.2), 2004-06-09.
[12] E. Felemban, S. Vural, R. Murawski, E. Ekici, K. Lee, Y. Moon, and S. Park, "Samac: A cross-layer communication protocol for sensor networks with sectored antennas," to appear in IEEE Trans. on Mobile Computing.