A SURVERY ON NETWORK PROTOCOLS FOR WIRELESS SENSOR NETWORKS

Ahmed A. Ahmed, Hongchi Shi, and Yi Shang
Department of Computer Engineering & Computer Science
University of Missouri-Columbia, Columbia, MO 65211, USA
aaab6f@mizzou.edu, ShiH@missouri.edu, ShangY@missouri.edu

Abstract—**Recent advances in MEMS (micro-electro-mechanical systems), processor, radio, and memory technologies have dramatically enabled development of wireless sensor networks. A sensor network is a large network of small sensor nodes, capable of sensing, communication, and computation. It can be deployed to sense some physical phenomenon for a wide variety of applications. During recent years, research in wireless sensor networks has become more and more active. Network protocols developed for sensor networks are of great importance to meet specific design goals of sensor networks. In this paper, we present a survey of recent work addressing network protocols, including routing and information dissemination algorithms, for wireless sensor networks. We evaluate them in terms of design goals, assumptions, operation models, energy models, and performance metrics.**

Keywords— sensor network; energy-efficient protocol; routing; information dissemination

I. INTRODUCTION

Recent advances in MEMS (micro-electro-mechanical systems), processor, radio, and memory technologies made it possible to manufacture microsensor nodes. Being characterized by their low-power, small size, and cheap price, these nodes are capable of sensing, wireless communication, and computation [18, 19, 7, 3]. If hundreds or thousands of these nodes are deployed in a large geographical area, they will form a dense wireless ad-hoc sensor network (a typical example of networked systems of embedded computers (EmNets) [17]). Such a network is capable of collecting and disseminating environmental data. Whereas each single node has limited sensing capabilities, this network is capable of monitoring an area by performing high-level distributed sensing of environmental phenomena [3, 8, 13] (e.g., light, temperature, sound, etc.).

The applications of such networks include deploying the sensor nodes in harsh inhospitable physical environments such as remote geographic regions or toxic urban locations or in benign environments such as large industrial plants and aircraft interiors [7, 14]. This allows collecting seismic, audio, or other type of data in such environments. Other applications may be environmental control in office buildings, robot control and guidance in automatic manufacturing environments, interactive toys, high-security smart homes, and identification and personalization [8].

Network protocols for sensor networks are of great importance to meet specific design goals. Due to the unique constraints of wireless sensor networks, the network protocols needed are different from conventional protocols. In this paper, we present a thorough review of recent work on network protocols for wireless sensor networks, including their advantages and drawbacks, and highlight some guidelines for improvement.

The rest of the paper is organized as following. In Section II, we present the dimensions of protocol survey, including typical design goals of sensor networks, general assumptions, different operation models used in the literature, energy models developed, and performance metrics used to measure the performance of a network protocol for sensor networks. In Section III, we discuss several protocols in some details and a brief comparison of them. We conclude in Section IV.

II. SURVEY DIMENSIONS

A sensor network consists of hundreds or thousands of sensor nodes deployed in a geographical region. These nodes are able cooperate together to form a high-level description of the event being sensed. This description is then sent to a distant base station (BS), through which an end-user can get the information. Design of routing protocols for networks is affected by several factors in different dimensions. In this section, we review several important dimensions.

A. Design Goals

Several design goals need to be met. First, the protocols need to be *scalable*. How to manage communication among many nodes and propagate an image of what is happening in the sensing field to the BS is a basic requirement. Second, sensor networks are *unattended*. Consequently, self-configurable algorithms need to be developed. Third, sensor networks are *resource-constrained*. A sensor node has a limited memory, limited communication bandwidth, limited energy, and limited computation capabilities. Fourth, a sensor network is required to work for *long time*. Therefore, protocols need to keep the network alive for as much time as possible. Finally, *fault-tolerance* is design goal. A network protocol should make use of the redundant data and redundant nodes to compensate for any dead nodes due to lack of energy.

B. Assumptions

A sensor node is expected to be of a matchbox size [7]. It has a processor clocked at several hundred MHz, a small memory, a radio modem, an ADC, sensors, and batteries [7, 3]. Nodes run some version of OS like the tiny operating system (TinyOS) developed at UC Berkeley [11, 12]. Radios can expend the minimum energy to reach the intended recipients by controlling the radio range [15]. Nodes are fixed and can be at tens of feet of each other [7]. The BS is fixed at a far distance from the nodes. Beside these assumptions, each

0-7803-7724-9/03/$17.00

protocol may add its own assumptions. Any specific assumption will be mentioned in Section III.

C. *Operation Models*

There are two common scenarios of operation: the continuously-operating model and the query-response model. In the former, it is required to acquire the sensed data *all the time*. Therefore, without any request, the sensor nodes need to be active and send the sensed data periodically to the BS. In the query-response scenario, a human operator poses a query to any node in the network. Then, this query is transferred to nodes in region R using some routing mechanism. It is worth mentioning that there exists the property of energy-awareness. A protocol is energy-aware if each node is aware of the amount of energy dissipated and the amount remaining.

D. *Energy Models*

A popular energy model [1,2,9,13,16] is as following. To *transmit* an l-bit message over a distance d, the radio expends

$$E_{Tx}(l,d) = E_{Tx-elec}(l) + E_{Tx-amp}(l,d)$$

$$E_{Tx}(l,d) = \begin{cases} lE_{elec} + l\varepsilon_{friss-amp} d^2 & if\ d < d_{crossover} \\ lE_{elec} + l\varepsilon_{two-ray-amp} d^4 & if\ d \ge d_{crossover} \end{cases}.$$

To receive an l-bit message, the receiver expends

$$E_{Rx}(l) = E_{Rx-elec}(l)$$

$$E_{Rx}(l) = lE_{elec},$$

where $E_{Tx\text{-}elec} = E_{Rx\text{-}elec} = E_{elec}$ is the energy required to run the transmitter or the receiver circuitry with a typical value of 50 nJ/bit for a 1 Mbps transceiver, $d_{crossover}$ is the cross-over distance with a typical value of 86.2 m, $\varepsilon_{friss\text{-}amp}$ is the energy needed for the transmitter amplifier when d is less than $d_{crossover}$ and has a typical value of 10 pJ/bit/m^2, and $\varepsilon_{two\text{-}ray\text{-}amp}$ is the energy needed for the transmitter amplifier when d is greater than $d_{crossover}$ with a typical value of 0.0013 pJ/bit/m^4. Computational energy for beamforming can be set to a typical value of 5 nJ/bit/signal [1, 2, 16].

E. *Performance Metrics*

Performance metrics include: energy consumption, lifetime, latency, quality, and amount of data disseminated per unit of energy. Energy consumption may be measured as the total consumed energy over a period of time. Another aspect of energy consumption is the network lifetime. It may be defined as the time between the moment the network operates and the moment the first node dies [1, 8]. Latency is the time from the moment a sensor node sends its sensed data out to the time the BS receives the data. The quality of the sensed data measures the accuracy with which the result of the sensor network matches what is actually occurring in the environment. Usually, the more data the base station receives, the more accurate its view of the remote environment is [16]. In data dissemination protocols, e.g., SPIN [10], the amount of data disseminated per unit of energy is another performance metric.

III. SURVEY OF PROTOCOLS

In this section, we evaluate a few well-cited routing protocols in terms of design goals, assumptions, operation models, energy models, and performance.

A. *LEACH*

The LEACH (Low-Energy Adaptive Clustering Hierarchy) algorithm [1, 2, 16] is non-energy-aware and assumes a continuously-operating model. Unlike many other routing protocols, LEACH does not follow a hop-by-hop routing.

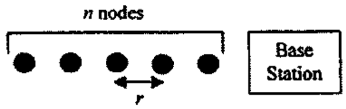


Figure 1. Simple linear network [1]

For a simple linear network, like the one shown in Fig. 1, the authors of LEACH argue that direct communication requires less energy than hop-by-hop routing if and only if

$$E_{elec} / \varepsilon_{amp} > r^2 n / 2, \tag{1}$$

where n is the number of nodes, r is the distance between nodes, E_{elec} is the energy required to run the transmitter or the receiver circuitry, and ε_{amp} is the energy used by the amplifier.

In LEACH, the time span is divided into fixed-length rounds. The duration of a round is pre-determined for a network with specific parameters. A round contains two phases: setup phase and steady-state phase. During the setup phase, a number of clusters (*almost* the same each round) are formed dynamically. A *cluster head* in each cluster is selected. The cluster head schedules the nodes in its cluster in a TDMA. During the steady-state phase, cluster heads receive data packets from their cluster nodes through direct communication. Then, the cluster head fuses the received data and sends it to the BS through direct communication. Fig. 2 shows the dynamic formation of clusters. Through simulation results, the authors of LEACH found that the energy dissipation is minimized if 5% of the nodes are cluster heads in each round. LEACH achieves a factor of 8 improvement compared to direct transmissions.

LEACH is an efficient and self-organized algorithm. However, it suffers from some problems. First, assuming a bandwidth of 1Mbps, the authors of LEACH used 50 nJ/bit for E_{elec} and 100 pJ/bit/m2 for ε_{amp}. Thus, (1) will be reduced to $r^2 n < 1000$. If 10 ft < r < 30 ft (3 and 9.1 m), we will have $12 < n < 111$. Therefore, n cannot exceed 111 nodes in the best case for the given parameters. Simulation in [1] was done using 100-node network. For networks bigger than 111 nodes, the direct communication may not be preferred to hop-by-hop routing, whereas it is typical for a sensor network to have hundreds of nodes. Second, LEACH results in a long latency for the BS to receive the sensed data. Moreover, the larger the sensor network is, the longer the latency will be. Finally, the number of clusters may not be fixed every round. At each round, a node n selects a random number k between 0 and 1. If this number is less that a threshold $T(n)$ defined as in (2), the node becomes a cluster head.

$$T(n) = \begin{cases} \dfrac{P}{1 - P * (r \bmod (1/p))} & if\ n \in G \\ 0 & otherwise \end{cases}, \tag{2}$$

where P is the desired percentage of cluster-heads, r is the current round, and G is the set of nodes that have not been cluster heads in the last $1/P$ rounds. Due to the selection of k, number of cluster heads may not be fixed.

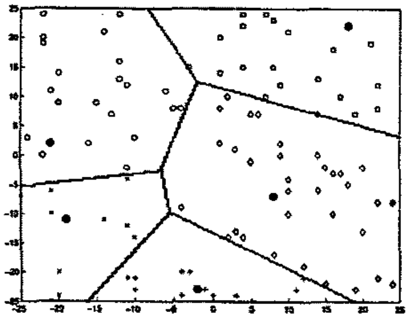
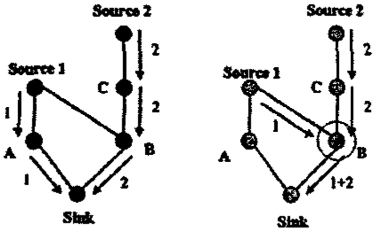


Figure 2. Dynamic clustering in LEACH [1]

B. *Directed Diffusion*

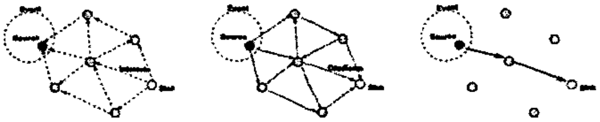
Directed diffusion [7] is a communication paradigm for information dissemination in sensor networks based on *data-centric* routing. In data-centric routing, all the interest is in the data, not the location of the node. "Where are the nodes whose temperature recently exceeded 30 degrees?" is an example of a data-centric request. Hence, data-centric routing is to find routes from multiple sources to a single destination that allows in-network consolidation, as in Fig. 3.



a) AC Routing **b) DC Routing**

Figure 3. Illustration of (a) address-centric versus (b) data-centric routing [4]

Directed diffusion utilizes the query-response operation model. A query (*interest*), like: "Give me periodic reports about animal location in region R every *I* seconds for the next T seconds", is posed at any node (*sink*). An interest may take the following form: *type* = four-legged animal, *interval* = 20 ms, *duration* = 10 min, *rect* = [-100,100,200,200], where rect is a rectangular area. Periodically, the sink broadcasts the interest to all its neighbors but with lower data rate than specified. When the interest reaches the nodes within rect, each node has an interest cache to store the interest in. The interest entry has a timestamp field and several fields for gradients. A gradient specifies the required data rate and the direction to the *interested* node. When a neighbor receives the interest, it checks if it exists in its cache. If no entry is found, one is created. When a node in region rect senses an event, it sends out the response to all the interested nodes. After receiving the initial data, the sink reinforces one of its neighbors by re-sending the interest but with a higher data rate. Then, reinforcement propagates till it reaches the source. Simulation results showed that directed diffusion performs better than both flooding and omniscient multicast in terms of energy dissipated [7]. Directed diffusion is robust and fault-tolerant. However, the low-data rate paths and the periodic broadcast of the interest reduce network lifetime. The few nodes that are within the radio range of the BS may die quickly, reducing network lifetime too.



(a) Interest propagation **(b) Initial gradients set up** **(c) Data delivery along reinforced path**

Figure 4. A schematic for directed diffusion [7]

C. *PEGASIS*

The PEGASIS (Power-Efficient GAthering in Sensor Information Systems) protocol [13] allows only one node to transmit to the BS in each round and makes nodes transmit only to close neighbors. The scenario and radio model in PEGASIS are the same as in LEACH. PEGASIS is centered on two ideas: chaining and data fusion. To construct a chain, nodes employ the greedy algorithm starting with the farthest node from the BS. In each round, a node is chosen randomly to be a *leader*. This leader node initiates a control token to start data transmission from the ends of the chain. Each node *fuses* its neighbor's data packet with its own to generate a single packet of the same length and then transmits that to its other neighbor. This is repeated till all the sensed data are collected at the leader node, which then transmits one data packet to the BS through direct communication. If nodes can communicate only with neighbors, the leader node can start a multi-hop routing to the BS. PEGASIS assumes that nodes have location information about all other nodes. Simulation results showed that PEGASIS performs better than LEACH energy-wise by about 100 to 300% when 1, 20, 50, and 100% of nodes die for different network sizes and topologies.

The main problem with PEGASIS is the long latency, which is at the order of N, where N is the number of nodes. This may be solved using multi-level chaining. Moreover, every node needs to have location information about all the nodes in the network. Also, a node may need to expend extra energy to find its closest neighbor. Finally, the quality of the sensed data may not be that good.

D. *SPIN*

The SPIN (Sensor Protocols for Information via Negotiation) [10] is a family of protocols used to disseminate individual sensor observations to all the nodes in the network. SPIN tries to solve three problems associated with classic flooding: implosion, overlap, and resource-blindness. The implosion problem occurs when a node D receives two copies of the same data from two neighbors B and C, as shown in Fig. 5. Since sensors cover overlapping geographical areas, they gather overlapped pieces of sensed data, as illustrated in Fig. 5. Finally, in classic flooding is resource-blind, i.e., nodes do not modify their activities based on the available energy. SPIN uses application-specific *meta-data* to name the sensed data. SPIN uses three types of data: ADV for advertising new data, REQ for requesting data, and DATA for a data message. SPIN-1 starts when a node has new data. It sends out an ADV message, containing meta-data about the new data, to all its neighbors. If a neighbor is interested in these data, it sends out an REQ message to the broadcasting node, which then sends the DATA to the interesting node. This process is repeated at every node that gets new data. This is shown in Fig. 6. SPIN-2 is simply SPIN-1 with a low-energy threshold. In SPIN-2, if a node gets new data or receives an ADV message, it will not participate in the protocol if it does not have enough energy. In classic flooding, a node sends out its data to all its neighbors. When a node receives new data, it sends it to all its neighbors except the one it received it from. Simulation results showed that the general performance of SPIN is better than both flooding and gossiping, as in Table 1.

Figure 5. The implosion and overlap problems associated with classic flooding [10]

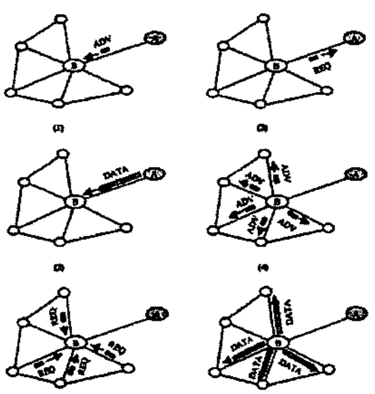


Figure 6. Illustration of the SPIN protocol [10]

SPIN solves the implosion, overlap, and resource-blindness problems. Each node needs to know about its neighbors only and it consumes little energy in computation. However, we think that network lifetime should have been studied as a performance metric. One factor that my affect network lifetime is that high-degree nodes may consume more energy than others, which may reduce network lifetime. Also, the energy model could be more sophisticated.

E. GEAR

The Geographical and Energy-Aware Routing (GEAR) algorithm [6] follows the query-response model. It assumes each node knows its location, energy level, and its neighbors' locations and energy levels. During the first phase, a query is routed to region R using energy-aware and geographically-informed neighbor selection heuristics. In the second phase, Recursive Geographic Forwarding or Restricted Flooding algorithm is used to disseminate the packet inside R. Formally, each node N maintains its *learned cost* to region R $h(N, \mathrm{R})$, which is infrequently updated from its neighbors. Moreover, it has a learned cost, $h(N_i, \mathrm{R})$, of its neighbor N_i. If a node does not have $h(N_i, \mathrm{R})$, it computes the estimated cost $c(N_i, \mathrm{R})$ as a default value for $h(N_i, \mathrm{R})$ as follows: $c(N_i, \mathrm{R}) = \alpha d(N_i, \mathrm{R}) + (1-\alpha)e(N_i)$, where α is a tunable weight, $d(N_i, \mathrm{R})$ is the distance from N_i to the centroid D of R normalized by the largest such distance among all neighbors of N, and $e(N_i)$ is the largest consumed energy at node N_i normalized by the largest consumed energy among neighbors of N. After a node picks a next-hop neighbor N_{min}, it sets its own $h(N, \mathrm{R})$ to $h(N_{min}, \mathrm{R}) + C(N, N_{min})$, where the latter term is the cost of transmitting a packet from N to N_{min}. Once the query packet is inside R, a Recursive Geographic Forwarding approach is used to disseminate the packet, as shown in Fig. 7.

Simulation results showed that for non-uniform traffic, GEAR delivers 70% - 80% more packets than the Greedy Perimeter Stateless Routing (GPSR) algorithm and 25%-35% more for uniform traffic.

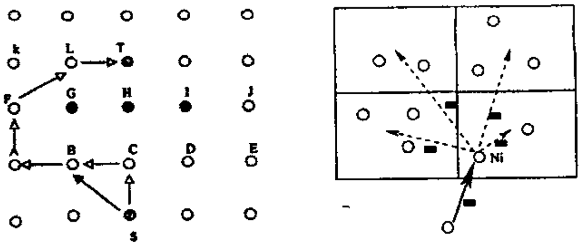


Figure 7. Illustration of the GEAR algorithm [6]

TABLE 1. A BRIEF COMPARISON OF THE PROTOCOLS DISCUSSED

Protocol	**Energy model**	**Performance Metrics**		**Routing scheme**		**Simulation**[b]	
		Lifetime	***Others***	***Address-centric***	***Data-centric***	***Size (nodes)***	***Compare to***
LEACH	As in Section II.D	√	Total energy dissipated	√		100	Direct communication, min. transmission energy routing, static clustering
Directed diffusion	660 mW in transmit 395 mW in receive 35 mW in idle		Average dissipated energy		√	50-250	Omniscient multicast, flooding
PEGASIS	As in Section II.D	√		√		100	LEACH
SPIN	600 mW in transmit and 200 mW in receive.		Data disseminated per sec, energy consumed per sec, data disseminated per unit energy		√	25	Flooding and gossiping
GEAR	1 energy unit to transmit or receive	√			√	400-4800	GPSR

Protocol	Problem	Design goals	Assumptions[a]		Operation model		
			Node knows:	*Global IDs*	*Continuously -operating*	*Query-response*	*Energy -aware*
LEACH	Collecting sensed data from nodes and sending it to the BS.	Min. energy and max lifetime		√	√		
Directed diffusion	Routing a query from a sink node to a region R and routing the sensed data back to the sink.	Min. energy				√	
PEGASIS	Collecting sensed data from nodes and sending it to the BS.	Min. energy and max lifetime	locations of all nodes	√	√		
SPIN	Dissemination of individual sensor observations to all the nodes in the network. This includes solutions to the implosion, overlap, and resource-blindness problems.	Max. data disseminated in unit time and in unit energy	its energy level	√	√	√	√
GEAR	Routing a query from a sink node to nodes in a region R.	Max lifetime.	locations & energy			√	√

[a] The assumptions in Section II.B are applied to all the five protocols.

[b] All simulation was done using the ns-2 simulator.

F. Brief Comparison of Surveyed Protocols

We now briefly compare the protocols discussed in this section. Table I summaries the comparison by describing the problem each protocol tries to solve, its design goals, assumptions, operation and energy models, performance metrics, and simulation.

IV. CONCLUSION

Wireless sensor networks have become popular due to the recent advances in sensing, communication, and computation. To make wireless sensor networks practically useful, we need to develop network protocols for them that meet several unique requirements and constraints. Among those come the low power consumption, small size, fault-tolerance, long lifetime, adaptivity, scalability, robustness, and low latency. We have surveyed some of the recent work on network protocols for sensor networks. We have covered design goals, assumptions, operation models, energy models, and performance metrics used in these protocols in some depth. Moreover, we have highlighted the advantages and drawbacks of each protocol and pointed out possible improvement. Future work may include a general mathematical framework that can be used to measure the total performance of a network protocol for sensor networks.

ACKNOWLEDGMENT

This work was supported in part by the U.S. National Science Foundation under Grant EIA-0086230.

REFERENCES

[1] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocols for wireless microsensor networks," *Proceedings of the 33rd International Conference on System Sciences (HICSS '00)*, January 2000.

[2] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, No. 4, pp. 660-670, October 2002.

[3] Q. Li, J. Aslam, and D. Rus, "Hierarchical power-aware routing in sensor networks," *DIMACS Workshop on Pervasive Networking*, May 2001.

[4] B. Krishnamachari, D. Estrin, and S. Wicker, "Modelling data-centric routing in wireless sensor networks," *In Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'02)*, June 2002.

[5] S. Ratnasamy et al., "Data-centric storage in sensornets," *The First Workshop on Sensor Networks and Applications (WSNA)*, September 2002.

[6] Y. Yu, R. Govindan, and D. Estrin, "Geographical and energy-aware routing: a recursive data dissemination protocol for wireless sensor networks," *In Proceedings of the Seventh Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'01)*, July 2001.

[7] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," *In Proceedings of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'00)*, August 2000.

[8] R. C. Shah and J. Rabaey, "Energy aware routing for low energy ad hoc sensor networks," *IEEE Wireless Communications and Networking Conference (WCNC)*, March 2002.

[9] S. Lindsey, C. Raghavendra and K. Sivalingam, "Data gathering in sensor networks using energy*delay metric," *International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing*, April 2001.

[10] W. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive protocols for information dissemination in wireless sensor networks," *In Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'99)*, August 1999.

[11] J. Hill et al., "System architecture directions for networked sensors," *In Proceedings of Ninth International Conference on Architectural Support for Programming Languages and Operating Systems*, November 2000.

[12] P. Buonodonna, J. Hill, and D. Culler, "Active message communication for tiny networked sensors," *In Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'01)*, April 2001.

[13] S. Lindsey and C. S. Raghavendra, "PEGASIS: Power-efficient gathering in sensor information systems," *In Proceedings of the IEEE Aerospace Conference*, March 2002.

[14] The Wireless Integrated Network Sensors project (WINS), http:/www.janet.ucla.edu/WINS

[15] R. Ramanathan and R. Rosales-Hain, "Topology control of multihop wireless networks using transmit power adjustment," *In Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'00)*, March 2000.

[16] W. Heinzelman, "Application-specific protocol architectures for wireless networks," *PhD thesis, Massachusetts Institute of Technology*, June 2000.

[17] Committee on Networked Systems of Embedded Computers, Computer Science and Telecommunications Board, Division on Engineering and Physical Sciences, and National Research Council, *Embedded everywhere*. National Academy Press, 2001.

[18] Crossbow. www.xbow.com

[19] G. J. Pottie and W. J. Kaiser, "Wireless integrated network sensors," *Communications of the ACM*, vol. 43, No. 5, pp. 51-58, May 2000.