Using Existing Infrastructure as Proxy Support for Sensor Networks
Licentiate Thesis Proposal

Jonas Neander
Department of Computer Science and Engineering
Mälardalens University
E-mail:jonas.neander@mdh.se

October 13, 2004

1 Introduction

With the growing interest in sensor networks, efficient communication infrastructures for such networks are becoming increasingly important. A sensor node is typically a tiny computer with limited computation resources and limited power supply, using on-board sensors to sense the surrounding environment, and using a wireless communication system to report to a network connection point (a network *sink*) [5, 7, 8].

Sensor networks are designed for many purposes. Among the interesting application areas are environmental surveillance and surveillance of equipment or persons in e.g. factories and hospitals. Common for all application areas are that sensor nodes are left unattended after deployment, that communication is wireless and the power supply is limited.

Having unattended sensor nodes with limited power supplies implies that one important feature of sensor networks is robust functionality in the face of network nodes dropping out of the network after some time of activity. Another implication is that, if the network is to survive a longer period of time, new nodes will have to be added to the existing network. The network topology is thus dynamic even if the sensor nodes not necessarily are mobile.

Some sensor nodes will not be able to directly communicate with the network sink. The traffic from these sensor nodes must be forwarded by other sensor nodes, hence routing schemes are necessary. Routing of traffic through other sensor nodes will however increase the power consumption of the forwarding sensor nodes. Therefore, routing decisions must be carefully evaluated in order to maximize network lifetime.

The main research focus in sensor networks has been on building networks consisting of sensor nodes only. These, peer-to-peer networks rely on energy draining and complex distributed algorithms to establish e.g. network topology and membership. In this licentiate thesis proposal, however, we are proposing a semi-centralized approach where existing, powerful, infrastructure can be used to off-load sensors and prolong network lifetime.

The outline of this licentiate thesis proposal, is as follows: In section 1.1 we present related work of sensor networks in general. In section 1.2 we motivate the use of a proxy backbone in our architecture and in section 1.3 we present related work done with TDMA in sensor networks. In section 2 we list some important problem issues in sensor networks and in section 3 we propose our asymmetric topology proxy backbone architecture for sensor nodes and address problems that need to be solved. In section 4.2 we decribe the proposed future work of the licentiate thesis and in section 4 we present the present achieved results and a future work time plan.

1.1 Related Work

This section briefly describes the sensor network field.

1.1.1 A brief history

In the early years of sensor networks, around 1980's, the sensors were as big as or larger than a shoebox. The sensors needed large batteries and could only survive for a couple of days at most. The sensors were communicating with each other with either Ethernet or with microwave radio. The sensors were few and placed by a vehicle, by hand or dropped out from a plane. The components in these sensors were custom built and expensive [7]. In the 1990's and up to now there has been an immense development of the hardware components to the sensors. The sensors have gone from big, expensive and custom built components to a single chip that is small, cheap and has off-the-shelves components; the smallest sensor today is only a cubic millimetre [22]. The sensors are communicating with each other using radio frequency (RF), laser or traditional wires. The most frequent used is RF even if it is the most energy consuming [17]. The sensors can be thrown out in thousands from aeroplanes, or deployed on the ground from vehicles or by hand, under water, on the body, inside buildings or cars etc. At the early years the main focus were done for the military defence industry. The main applications were focusing at surveillance, detecting movement of enemy forces etc. Today there are a huge amount of other applications such as; health care [19], environment and habitat monitoring [15], weather forecast [6], infrastructure security and property surveillance, industrial monitoring, marine monitoring, traffic surveillance and so on besides the military applications.

1.1.2 Main problems

- The main problem in sensor networks is that the sensors have *limited energy, limited amount of memory and computational capacity*. As said above the most energy consuming component is the radio communication. There is a lot of research going on in finding methods and algorithms that reduces the amount of radio transmission between the sensor nodes such as; data aggregation [12], data fusion [11], directed diffusion [13], scheduling of sensor nodes [20, 10], building clusters [10, 21], and combinations of the mentioned. These methods assume in most cases that the sensor network is homogenous and that the sensor nodes need to agree between them selves in order to organize and maintain the topology and routes.
- These unattended self-configuring systems need to *adapt to current environment* and the environment can change over the time. If the environment is changing it might be because a sensor node is moving form one place to another. Moving sensors change the topology and the topology changes frequently in sensor networks. This not only because of moving senor nodes but of failing sensor nodes, new senor nodes are added or old sensor nodes demise. The sensor nodes are often densely deployed and the disturbance between each other when sending messages is high. The environment the sensor nodes are located in might disturb the communication even more.
- To reduce the energy consumption at the sensor nodes the need to *reduce the communication* is one of the main issues that need to be handled. One way to do this is to let the sensor nodes exploit computation near data as much as possible [5]. This will reduce the amount of communication in the network. The communication is often broadcast based and the sensor nodes do not often have its own global identification (ID). Broadcast will, without smart algorithms, result in large amount of messages floating around in the network. Without knowledge about the destination the sensor nodes will just blindly broadcast the message. There exists solutions to

these problems but they are often application specific.

1.1.3 Important design factors

A sensor node needs to be inexpensive and consume little energy, the topology need to be scalable and fault tolerant.

The sensor nodes need to be cheap since sensor nodes are prone to error the often demise, or when monitoring out in the habitat some animal etc could step on it and destroy it. Another argument for cheap sensor nodes is that they often are deployed in dense sensor fields, up to several thousands or millions of sensor nodes [18]. Sometimes they are placed in hostile environment or in habitat not suitable for humans. To realize the densely deployed sensor network we need cheap sensor nodes.

It is not feasible to change the power resources of the sensor nodes because of the environment the sensor nodes often are deployed in [5]. This makes it important to reduce the power consumption as much as possible to prolong the lifetime of a sensor node. The power consumption should be reduced both in the hardware components and the software protocols and algorithms.

As stated above, the sensor nodes often are deployed in dense fields. Demand for scalable protocols is needed to handle the large amount of sensor nodes in the network. The area might vary from a few square meters to several thousands of square meters. The amount of sensor nodes is not always equally distributed in this area. There could be clusters of sensor nodes at different interesting locations that need to communicate or collaborate between each cluster.

Sensor nodes that are prone to error need to be fault tolerant [14]. Sensor nodes might fail, both in hardware and software, they might run out of power etc. The remaining sensor nodes should not be dependable on other specific sensor nodes to be able to perform their tasks. If a sensor node demises it should be alternative routes or other sensor nodes to collaborate with.

1.2 Arcitecture using proxies

In order to lower the risk of a sensor node draining its power resources by forwarding traffic from other nodes, we propose a hierarchical infrastructure where some nodes have more power resources and thus can assist the smaller nodes with communication and data processing. Since the more powerful nodes can offload the smaller nodes, we call the more powerful nodes *proxies*. Our sensor network architecture thus consists of a large number of sensor nodes, a smaller number of proxy nodes, and one or possibly more network sinks.

Often, the proxies can be situated in existing infrastructure. For instance, there are infrastructure networks built in hospitals and industrial factories that could be used to prolong the lifetime of the sensor networks. The infrastructure network can act as a fault tolerant proxy backbone for sensor nodes collecting data or monitoring patients. Industrial and hospital infrastructure networks are static and they do not have limited energy as sensor nodes have. Sensor nodes in the network connected to machines, medical equipment, patients etc. have a varying degree of mobility, however we will treat them as if they were mobile and as if the topology of the sensor network was frequently changed. The infrastructure network could be wired, wireless or a combination of both. Some sensitive hospital equipment could be disturbed by wireless transmissions so it may not be feasible to have strong-powered wireless proxies talking to each other. Some of the proxies thus need to be wired and have low-powered wireless transmitters that do not disturb sensitive equipment.

The advantage of using proxies as masters for a sensor *cluster* is that proxies have a lot of memory, high speed processors, "unlimited" energy etc. A proxy can always have the radio transmitter/receiver active to perform complex optimizations and routing for the sensor nodes. A proxy, in our architecture, has large radio coverage and can potentially accept all the sensor nodes that are receiving the signal to its cluster. To build clusters of sensor nodes to reduce the amount of traffic in the network is proposed in e.g. [10]. Some sensor nodes become cluster-heads and collect all traffic from/to their cluster. A cluster-head sends the collected traffic to a gateway in the cluster who will forward the traffic towards the sink.

The most power-consuming activity of a sensor node is typically communication [17]. Communication must hence be kept to a minimum. This applies to transmission, reception and listening for data. All activities involving communication are power-consuming and the most important way to save power is to turn the radio off as much as possible. We therefore propose the use of time-division multiplexed access (TDMA) schemes for sensor node communication.

1.3 TDMA scheduling for sensor networks

Several different TDMA schemes have been proposed for sensor networks and most of the schemes use sensor nodes to schedule the network.

In [20], methods for reducing energy consumption at all levels of the hierarchy is presented. The sensor nodes communicate with an adjacent basestation within ten meters from the sensor nodes. The sensor nodes send data directly to the basestation without involving other sensor nodes. A sensor node is assumed to synchronize its clock with the basestation several times per second when TDMA is used. When frequency-division multi access (FDMA) is used, the radio will be on for longer periods of time than with TDMA since transmission times are prolonged when using FDMA. FDMA on the other hand does not need to have the sensor nodes' clocks synchronized as TDMA does. The authors of [20] use a hybrid of TDMA and FDMA called TDM/FDM and they give an analytical formula to calculate the optimum number of channels to use in order to get the lowest power consumption.

LEACH is a TDMA cluster-based approach [10]. A node elects itself to be cluster-head by some probability. It broadcasts an advertisement message to the all the other nodes. A none-cluster-head node selects a cluster-head to join by the received signal strength. To be cluster-head is much more energy consuming than to be a non-cluster-head node. All nodes in the network are supposed to be cluster-head during some time period. The TDMA scheme starts every cycle with a set-up phase. After the set-up phase the steady-state phase begins for a certain amount of time. In the steady-state phase there are several frames where nodes have their slots periodically. Then after a certain amount of time the TDMA cycle ends and re-enters the set-up phase.

Dynamically changing the topology without global knowledge of the topology is energy consuming. It is impossible to do optimal route decisions without knowledge of the future topology. Further, several messages have to be exchanged between the sensor nodes to establish and maintain the topology.

In passive clustering [9], no extra messages for building the topology are needed. The first node sending a message will piggyback the sender state to the others. The nodes will form clusters by piggybacking two bits in the MAC layer. A node will need to store cluster-heads and gateways in its memory. If a cluster-head has been silent for a certain amount of time it is removed from the memory. When all the cluster-heads have been removed from the memory, the sensor node will set its state to the initial state and start over again. In [16], the authors extend the passive clustering with a low energy state. Sensor nodes below a certain amount of energy will put themselves in low energy state and will only participate in local collection of data. Still, sensor nodes will need to save the topology in memory and they will need to handle the changes. Also, a cluster-head or a gateway will remain in the same state until the energy falls below a certain threshold.

2 Problem Areas

Below we list some important issues.

- As already mentioned in Section 1, sensor nodes have scarce resources. A major part of their total energy is used by the wireless radio to send and receive data [17]. It is of great importance to reduce the traffic between sensor nodes in order to prolong their lifetime. Some sensor networks adjust the radio power to save energy. Some networks build clusters, fusion data etc., to reduce the amount of traffic in the network. To organize and distribute the clusters is costly and some sensor nodes will be more exposed than others. The need to reorganize the cluster to spread out the extra workload requires message exchange.
- Sensor nodes could be scheduled or schedule themselves to turn off their radio (sleep) for a specified amount of time. When scheduling themselves to sleep they have to inform the adjacent sensor nodes about this. Sending messages is costly and the energy saved by sleeping could be lost in messages scheduling sensor nodes to sleep.
- Sensor networks using the cluster-based approach could use carrier sense multiple access (CSMA), FDMA, TDMA etc., to schedule the sensor nodes. The radio needs to be turned on frequently when using CSMA. Otherwise it could miss messages from adjacent sensor nodes. Messages from sensor nodes could interfere with each other and result in retransmission of messages.
- Sensor nodes in a TDMA network need to have their clocks well synchronized. Since the clocks of sensor nodes with separate (local) time sources will drift in relation to each other and cause a *clock skew*, sensor node clocks must be resynchronized at regular intervals. If the clocks not are synchronized, scheduled messages could be missed or messages from one sensor node could collide with other messages, i.e., waste of energy. However, extra messages will have to be exchanged between sensor nodes to keep a global time.

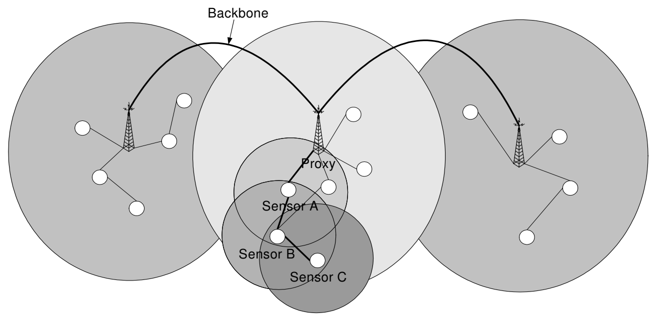

Figure 1: Overview of the architecture.

- Routes for messages from a sensor node to the sink will need to be established. Sensor nodes could be added, or disappear forcing new routes to be established. Building routes requires knowledge of the network or message exchange between sensor nodes. Building optimal routes for the packets in the network requires global knowledge of the network architecture. Global knowledge of the network requires a lot of memory, but sensor nodes have a limited amount of memory to their disposition. Using the greater part of the memory to store information about the topology drastically reduces the amount of work a sensor could perform.

  Sensor networks using messages to establish routes by flooding the network, omniscient multicast, advertising/requesting [12, 13] etc., will consume large amounts of energy to establish and maintain routes. Hence, the number of such maintenance messages needs to be minimized.
- To have sensor nodes with different quality of service (QoS) requirements in the same network will increase complexity, computations and radio uptime if managed locally at the sensor nodes. Some optimizations will not be cost-effective in a sensor network, i.e., it would cost more to calculate and distribute the optimization than what could be gained from the optimization itself.

3 Using Proxies in Sensor Networks

We propose to build our topology based on clusters with a proxy backbone that has “unlimited” energy and “enough” bandwidth in the backbone channels, see Figure 1. The proxy backbone could be e.g. regular computers, PDAs or small embedded systems. The proxies are connected to each other by wire, wirelessly or both. To be able to turn off the radio of the sensor nodes as much as possible, we propose to use TDMA to schedule the communication of the sensor nodes. Furthermore, we propose to build clusters of the sensor nodes where the proxy is the cluster-head. Using clusters will ease the scheduling of the sensor nodes. One proxy is used for each sensor cluster and the proxy is master for the sensor nodes in the cluster. The proxy can reach all the sensor nodes in the cluster directly and a similar TDMA scheme as in LEACH could be used in our topology.

Not all sensor nodes are assumed to be able to communicate directly with the proxy. Some sensor nodes need other sensor nodes to forward the traffic to the proxy. For example, regard Sensor B in Figure 1. It is located on the fringe area of the cluster and its radio power is not able to reach the proxy directly. Sensor B needs to use Sensor A to forward its traffic. Sensor B has in its turn to help Sensor C with forwarding of traffic. Thus, we propose an asymmetric topology where the proxy reaches all the sensor nodes in its cluster but the sensor nodes might not reach the proxy directly. This will result in a network hierarchy where proxies are at the top and sensor nodes are divided into different lower levels depending on the sensor nodes' task etc. Simulations and future experiments will show how to organize the best hierarchy.

The proxy will do the route decisions and manage topology changes for the sensor nodes. A proxy will make a TDMA schedule for its cluster and inform each sensor node about their assigned time slot. The proxy will look at other proxies' schedules and ensure that its sensor nodes do not interfere with other clusters. The sensor nodes only need to focus on their own tasks and thereby save energy that otherwise would be used to do extra computation and to exchange messages with other sensor nodes in order to maintain the network topology. The proxy will change existing routes to save highly exposed sensor nodes from draining their batteries. When a proxy receives a message from a new non-adjacent sensor node, it will compute the best proxy for that sensor node. The proxy will compute the best route for new sensor nodes and inform the concerned sensor nodes about the changes. It will also check if rearranging old routes to new ones would benefit the sensor nodes. No, or little, knowledge of the network is needed at the sensor nodes, and the memory can be used for data aggregation etc. Proxies can make optimizations that a pure sensor node network would not consider cost-effective by changing the relative cost of the optimization as work is moved from the sensor nodes to the proxy. Issues to solve include

- Mobility: Mobile sensor nodes will make the scheduling decisions worse.
- Energy: When is it worth to reroute the sensor nodes in order to save energy?
- Optimization: What is an optimization goal and when do we execute them?
- New sensor nodes/dead sensor nodes: When to do rerouting and optimizations when a new node enters the cluster or dies?

Depending on the TDMA scheme used, the maximum allowed clock skew will be known. From this, and from knowledge about the drift of the local clocks, the maximum time interval between synchronization events can be calculated. This in turn implies a maximum sleep time for the sensor nodes, i.e. how often they must listen to the radio in order to keep their clocks in synchronization with the TDMA schedule.

Some sensor nodes in the cluster could be scheduled for optimized energy saving, others could be scheduled for QoS. In our architecture we can handle sensor nodes with different demands without involving the whole sensor network for reorganization etc. Proxies will handle all extra workload and only the concerned sensor nodes will have to be reorganized. Depending on the application running on the sensor node, i.e. the requested QoS, the proxy will schedule the sensor nodes differently. A sensor node with low QoS demands could/would be scheduled to sleep during several TDMA cycles. Sensor nodes with higher demands could/would be scheduled every TDMA cycle or more often if necessary. Having sensor nodes with low QoS sleep during several TDMA cycles will increase the delay for topology changes and messages from the sensor nodes to the sink. Different QoS demands in the network imply high complexity not trivial to solve. We need to group sensor nodes within a cluster in a smart way to guarantee response time etc.

Sensor nodes in a cluster need to help new sensor nodes with connecting to the proxy. A new sensor node will try to contact the closest proxy but sometimes a message could be received by another proxy depending on which sensor node heard the message first. If the new sensor node was heard by an adjacent cluster they will forward the message to its proxy. The proxies then handle the possible handover. Timing issues for the sensor nodes are important to solve. How many cycles after the first request to join a cluster can a new sensor node be guaranteed to be in the cluster? Could a sensor node count on other sensor nodes forwarding the message to the proxy? These questions need to be solved.

The proxy backbone needs to be fault tolerant and if a proxy disappears, other proxies have to take over the orphan cluster. New proxies might enter the backbone and the clusters must be optimized to the new network structure. We need to solve how to handle the re-clustering of the clusters in the network if a proxy should be added, removed or disappear. We need to have solutions for the case if a proxy disappears and the remaining proxies in the network do not reach all the sensor nodes. Traditional sensor schemes could be one way to solve the problem with unreachable sensor nodes.

4 Publications

In this section, we describe the proposed contents of the licentiate thesis, as well as some work by the author outside the scope of the thesis.

4.1 Achieved Results

Below is a list of papers and courses, most of which are completed and all of which are started upon.

4.1.1 Published Papers

Paper A:
An Asymmetric Proxy Backbone Architecture for Sensor Nodes.
J. Neander, M. Nolin, M. Björkman.
MRTC report ISSN 1404-3041 ISRN MDH-MRTC-158/2004-1-SE, April, 2004.

Paper B:
Introducing Temporal Analyzability Late in the Lifecycle of Complex Real-Time Systems.
A. Wall, J. Andersson, J. Neander, C. Norström, M. Lembke.
In proceedings of RTCSA 03, February, 2003.

Paper C:
Introducing priorities for IPv4.
B. Çürüklü, J. Neander.
MRTC report ISSN 1404-3041 ISRN MDH-MRTC-76/2002-1-SE, November 2002.

Paper D:
Using Existing Infrastructure as Proxy Support for Sensor Networks.
J. Neander, M. Nolin, M. Björkman.
Work-in-Progress Session of the 16th Euromicro Conference on Real-Time Systems, Catania, Italy 2004

4.2 Future Work

In this section, we describe the proposed future work of the licentiate thesis.

4.2.1 Paper E

This paper will compare our proposed architecture to similar sensor architectures. The comparison is supposed to show that our proposed architecture will prolong the lifetime of the sensor network and outperform other existing architectures without proxy support. Further, the simulation will give us preliminary results of the behaviour of our proposed architecture. The simulated architecture is supposed to include the most important functionality, such as routing decisions, sensor hierarchy and proxies that handles the TDMA scheduling. This will give us a rough model of the architecture and highlight possible problems not known in advance.

We will implement the architecture in Network Simulator 2, NS2 [2], GloMoSim [1], OMNeT++ [3] or OPNET [4].

PACT[16], LEACH[10] and "A Hierarchical Routing Protocol for Networks of Heterogeneous Sensors"[21] are possible archtitectures for comparison with our architecture. These architectures are similar to ours. They all use a cluster based architecture but without proxy support.

4.2.2 Paper F

This paper will present a novel TDMA scheduler for our proposed architecture. After having simulated the proposed architecture and found the main problems that the proxy needs to handle we can present a TDMA scheduler. The TDMA scheduler should be able to decide when the sensors in the cluster are supposed to send and sleep. It will handle routing in the network as well as forwarding problems. The main focus of the proxy is to prolong the lifetime of the network.

This paper will not include mobile sensors, failing proxies, heterogeneous types of sensors or quality of service (QoS).

4.2.3 Survey

Write a survey of sensor networks and other important areas i.e. Proxies, TDMA etc. The survey is supposed to cover more aspects about the area than described here in the licentiate proposal.

5 The thesis

The licentiate thesis will contain an introductory part and a collection of articles. The papers that will be included in the collection are paper D, E and F. The introductory part will include the survey and an introduction to the thesis.

6 Research methods

The research methods we are using are common computer science methods. We have used the inductive method to formulate the hypothesis described in this document. We are planning to simulate and evaluate the hypothesis, this will become Paper E. To write paper F we need to use both the deductive method and the inductive. The deductive method is used to collect and use existing scheduling algorithms. The inductive is used to make a new hypothesis of the architecture after the simulation and to evaluate the scheduler.

7 Coursework

7.1 Completed courses

For the degree of licentiate, a minimum of 30 academic points is required. So far, these courses are completed:

Course	Points	End Date
Realtidssystem, fk	5p	2003-02-03
Forskningsmetodik inom dataområdet	5p	2003-02-03
Communication	3p	2003-02-05
Formella språk, automater och beräkningar	5p	2003-06-11

7.2 Started, but not completed courses

As for the remaining points, the plan is to include the following courses in the licentiate degree and to end them all before autumn 2004:

Course	Points	Status
Distributed Systems	3p	Waiting on the results
Real-Time and Embedded Systems	3p	Waiting on the results
Communication reading course	5p	Writing the summary
Forskningsprojektplaneringskurs	5p	Waiting on the results

8 Time Plan

The plan is to defend the Licentiate Thesis in the spring 2005.

Acivity	Ends No Later Than
Remaining Courses	Autumn 2004
Completion of Papers	January 2005
State of the Art Report	February 2005
Licentiate Thesis Writing	Mars 2005
Licentiate Thesis Defence	May 2005

References

[1] GloMoSim, Global Mobile Information Systems Simulation Library. World Wide Web, http://pcl.cs.ucla.edu/projects/glomosim/, 2004.

[2] NS 2, Network Simulator. World Wide Web, http://www.isi.edu/nsnam/ns/ns-build.html, 2004.

[3] OMNeT++, Discrete Event Simulation System. World Wide Web, http://www.omnetpp.org/, 2004.

[4] OPNET. World Wide Web, http://www.opnet.com/home.html, 2004.

[5] I. Akyildiz, W. Su, Y. Sankarasubramanian, and E. Cayirci. A Survey on Sensor Network. *IEEE Communications Magazines*, August 2002.

[6] A. Baptista, T. Leen, Y. Zhang, A. Chawla, D. Maier, W. Feng, W. Feng, J. Walpole, C. Silva, and J. Freire. Environmental Observation and Forecasting Systems: Vision, Challenges and Successes of a Prototype. *International Society for Environmental Information Sciences Annual Conference (ISEIS'2003)*, July 2003.

[7] C. Chong and S.P. Kumar. Sensor Networks: Evolution, Opportunities, and Challenges. *Proc IEEE*, August 2003.

[8] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar. Next Centuru Challenges: Scalable Coordination in Sensor Networks. *ACM MobiCOM '99*, 1999.

[9] M. Gerla, T.J. Kwon, and G. Pei. On Demand Routing in Large Ad Hoc Wireless Networks with Passive Clustering. *Procedings of IEEE WCNC 2000*, Sep 2000.

[10] W. Heinzelman. *Application-Specific Protocol Architectures for Wireless Networks*. PhD thesis, Massachusetts institute of technology, June 2000.

[11] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-Efficient Communication Protocol for Wireless Microsensor Networks. Maui, Hawaii, Jan 2000. In Proceedings of the Hawaii International Conference on System Sciences).

[12] W.R. Heinzelman, J. Kulik, and H. Balakrishnan. Adaptive Protocols for Information Dissemination in Wireless Sensor Networks. *ACM MobiCOM '99*, 1999.

[13] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks. *ACM MobiCOM '00*, pages 56–67, 2000.

[14] F. Koushanfar, M. Potkonjak, and A. Sangiovanni-Vincentell. Fault Tolerance Techniques for Wireless Ad Hoc Sensor Networks. volume 2, pages 1491–1496. Sensors, 2002, Procedings of IEEE, 2002.

[15] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Andersson. Wireless Sensor Networks for Habitat Monitoring. *WSNA'02*, September 2002.

[16] G. Pei and C. Chien. Low Power TDMA in Large Wireless Sensor Networks. *Military Communications Conference*, 1:347–351, 2001.

[17] G. J. Pottie and W. J. Kaiser. Wireless Integrated Network Sensors. *Communications of the ACM*, 43(5):51–58, May 2000.

[18] J. M. Rabaey, M. Josie Ammer, J. L.da Silva Jr., D. Patel, and S. Roundy. PicoRadio Supports Ad Hoc Ultra-Low Power Wireless Networking. *IEEE Computer*, pages 42–48, July 2000.

[19] L. Schwiebert, S. Gupta, and J. Weinmann. Research Challenges in Wireless Networks of Biomedical Sensors. *ACM Mobicomm 2001*, pages 151–165, 2001.

[20] E. Shih, S.H. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, and A. Chandrakasan. Physical Layer Driven Protocol and Algorithm Design for Energy-Efficient Wireless Sensor Networks. *ACM SIGMOBILE*, July 2001.

[21] S. Ming Tseng and R. Pandey. A Hierarchical Routing Protocol for Networks of Heterogeneous Sensors. Goteborg, August 2004. 10th International Conference on Real-Time and Embedded Computing Systems and Applications (RTCSA).

[22] B. Warneke, M. Last, B. Liebowitz, and K.S.J. Pister. Smart Dust - Communicating with a Cubic-Millimeter Computer. *IEEE Computer*, 34:44–51, Jan 2001.