Efficient and Reliable High Level Communication in Randomly Deployed Wireless Sensor Networks*

Carme Àlvarez
Josep Díaz
Jordi Petit
Departament de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya
Campus Nord
08034 Barcelona, Spain

José Rolim
Centre Universitaire d'Informatique
University of Geneva
24, Rue Géneral Dufour
1211 Genève 4, Switzerland

Maria Serna
Departament de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya
Campus Nord
08034 Barcelona, Spain

ABSTRACT

In this paper we present a way to establish a reliable and efficient high level communication system in a randomly deployed network of sensors equipped with directional antennas. Such high level communication system will enable the programming of the sensor network using high level communication functionalities without the burden of taking care of their physical capacities (low range, unidirectional links, single frequency, presence of collisions...). The high level communication functionalities we offer include point-to-point communication, point-to-area communication, and one-to-all communication. The basic idea to implement this system is to simulate a virtual network that emerges from the ad-hoc network using self-organization, self-discovery and collaborative methods. The analysis of the protocols we present shows their reasonable efficiency, scalability and robustness.

Categories and Subject Descriptors

C.2 [**Computer-communication networks**]; C.2.2 [**Network Protocols**]; C.4 [**Performance of systems**]

General Terms

Algorithms, Design

Keywords

Middleware, sensor networks, ad hoc networks, random graphs.

*This research was partially supported by the EU within the 6th Framework Programme under contract 001907 and by the Spanish CICYT project TIC2002-04498-C05-03.

MobiWac'04, October 1, 2004, Philadelphia, Pennsylvania, USA.

1. INTRODUCTION

Current development in micro-electrical-mechanical systems and wireless communication have opened a fast development of small, low-power, low-cost sensor devices that integrate sensing, computation and communication. Sensor devices are very small but contain a computer, radio, sensors and a power supply, and they are designed to transmit data about location, temperature, vibration, light, sound or airflow. A very large number of these devices can be spread to form *sensor networks*. Applications of sensor networks include fire or flood detection, disaster relief, smart environments, battlefield monitoring, health control, etc. Sensor devices are supposed to be deployed in hostile environments, and therefore must use *self-organization* and *collaborative methods* to form in an *ad-hoc* way a sensor network. Realizing such an infrastructure in an efficient and robust way is, from a technical and algorithmic point, a challenging task. In recent years, many papers have been devoted to network of sensors, see for example the surveys [5, 1, 14].

Radio frequency is the most widespread choice of communication for sensor networks. Both omni-directional antennas or directional antennas have been considered. Directional antennas have a focused emission beam which spans a sector of α degrees (currently the beam angle can be as narrow as 5 or 10 degrees [15]). In networks of sensors, directional antennas may have multiple advantages over omni-directional antennas [4, 13, 16]: less energy consumption, less fading area and, since the transmission area is smaller, the channel interference may have less influence. However, directional antennas involve unidirectional links, which complicate the design of communication protocols.

In this paper we consider the general setting of sensor networks in which a large number of sensor devices are scattered on a terrain. We assume that such devices are immobile and that they are thrown at random on a square. Our goal is to present a way to establish a reliable and efficient high level communication system in a network of sensors. Such a system will enable the programming of sensor networks using high level communication functionalities without the burden of taking care of their physical capacities, including low range, unidirectional links, use of a single frequency and presence of collisions among others. Among the high level communication functionalities we could offer, in this paper we consider *point-to-point* communication (device u sends a message to device v for arbitrary u and v), *point-to-area* communication (device u sends a message to one or all devices in some area), and *one-to-all* (device u sends a message to all other devices).

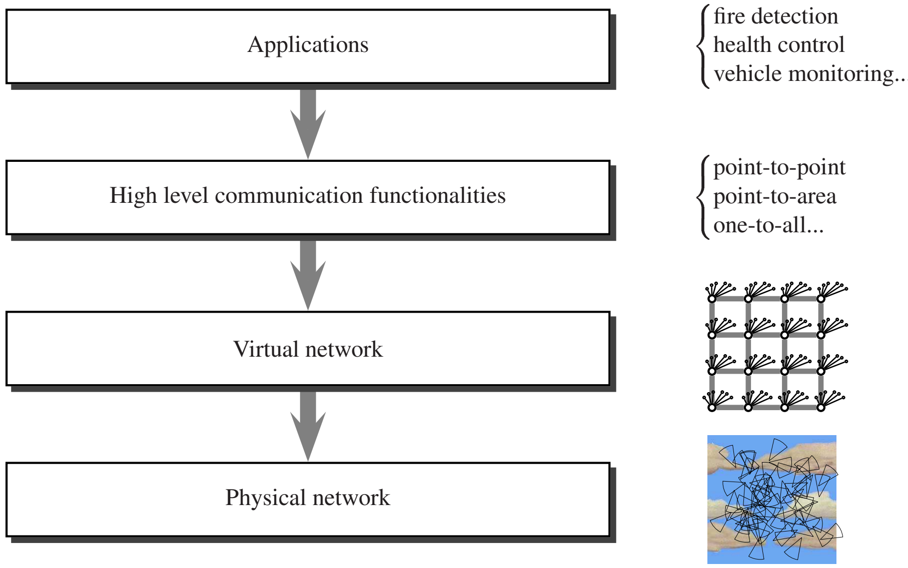

Figure 1: 4 levels architecture

Because the network of sensors is an ad-hoc and randomly deployed system, such high level communication system cannot be precomputed and encoded in the devices, it must rather emerge from the system itself. Our design for such system is made up of four layers (see Figure 1): On the upper layer lie the applications, which use the high level communication functionalities. These high level communication functionalities are implemented on the top of a virtual network that is simulated by the physical network. The configuration of this virtual network involves an initialization phase where the devices perform self-discovery and auto-organization tasks and compute routing and scheduling tables.

The paper has the following organization: First, in Section 2, we present the formal model in which we study networks of sensors. In Section 3, we present a protocol for initialization, that is, to establish a virtual network. This protocol involves electing a leader and computing routing and scheduling tables. Then, in Section 4 we describe how to implement several high level communication functionalities. Finally, in Section 4 we move the applications level and discuss how some applications can use our functionalities for their own goals. We close the paper with a summary of our conclusions and discuss possible extensions.

2. THE MODEL AND SOME PROPERTIES

In this section we present the specification of the model of the sensor networks we consider all through the paper. This model was introduced in [7] for optical communication networks and exploited in [2]. We also present the hypotheses of our model as well as some terminology.

In the following, we denote by $S_x = (x, r, \alpha, \beta)$ the sector centered at position x with radius r, amplitude α and elevation β, see Figure 2.

We assume that we have n devices that are scattered uniformly at random on a square area. For sake of simplicity, we assume that this area is the unit square $[0,1]^2$. The position of a device u is represented by $x[u]$. Each device is equipped with a radio frequency system with a directional antenna that can emit in a sector of α degrees up to a distance r_n. When a device is deployed, the shift angle between the basis of its sector and the horizontal axis is considered to be uniformly random. We denote by $\beta[u]$ such angle for device u. This description leads to a directed graph modelization:

Definition 1. Assume that α is a fixed angle. Let $(x_i)_{i\geqslant 1}$ be a sequence of independently and uniformly distributed (i.u.d.) random variables giving the coordinates of points in $[0,1]^2$, let $(b_i)_{i\geqslant 1}$ be a sequence of i.u.d. angles and let $(r_i)_{i\geqslant 1}$ be a sequence of numbers in $[0,1]$. For any natural n, we write $X_n = (x_1,\ldots,x_n)$ and $B_n = (b_1,\ldots,b_n)$. We call $\mathcal{G}_\alpha(X_n, B_n, r_n)$ the *random sector graph* on n nodes $\{1,\ldots,n\}$, where there is an arc from i to j, if and only if $x_j \in S_{x_i}$.

In the following, to diminish boundary effects, we focus our attention to *interior devices*, that is, devices whose distance from the boundaries of the terrain is greater than r_n. We also assume that the range is of the form

$$r_n = \sqrt{\frac{a_n}{n}} \quad \text{with } a_n = \omega(\log n) \text{ and } r_n = o(1).$$

This selection ensures that the subjacent random sector graph is sparse but yet strongly connected when restricted to the interior devices (see [7] for the details).

We consider that devices always communicate in a synchronous way using send and receive primitives that take one unit of time

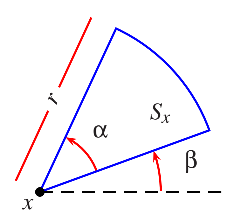

Figure 2: Sector associated to a device located at x and angle β with respect to the horizontal axis. Its transceiver can reach the sector S_x, centered at x, with angle α and range r.

(pulse). Any device v that falls inside the sector $S_{x[u]}$ of a device u can potentially receive the signals emitted by u. However, due to the use of a single frequency, devices can only receive messages if they get no collision, that is, if only one device is emitting over their coordinate. In terms of graphs, this means that if two devices u and v are emitting at the same time and that there is a vertex t so that (u,t) and (v,t) are arcs in the corresponding sector graph, then t does not get any message.

We dissect the unit square in a grid of $\kappa_n \times \kappa_n$ cells, each of size $s_n = r_n/\sqrt{8}$, so that $\kappa_n = 1/s_n$. Cells in the grid are numbered from top to bottom and from left to right starting from 0. Two cells c_1 and c_2 are called *mutually dangerous* if the distance between their centers is smaller or equal than $3r_n$. This means that a message originated by some device in c_1 could potentially be received by some device in c_2.

This grid serves as reference position for the devices, so that the data they report refers to the cell in the grid that contains them. In fact, in this paper we assume that each device knows in which cell it belongs. One could do this by installing a GPS on each device, but this is sometimes considered to be too expensive in terms of size and power consumption. Several localization algorithms have been proposed, but this problem is difficult [3]. In any case, we take a weaker assumption as devices just know an approximation of their coordinates.

Let us now present some results that will be useful in the rest of the paper. The first result provides sharp bounds on the number of devices that fall in a cell; its proof is based on the use of Chernoff's bounds and Boole's inequality.

Lemma 1. Let $\varepsilon > 0$ and let $M_n^+ = M_n = \frac{1}{8}(1+\varepsilon)a_n$ and $M_n^- = \frac{1}{8}(1-\varepsilon)a_n$. Then, with high probability, no cell contains more than M_n^+ devices or less than M_n^- devices.

The following result, provable with standard probabilistic tools, is a variation of the birthday paradox:

Lemma 2. Consider that we have m balls thrown to m^5 bins. Then, the probability that some bin contains two or more balls is $o(1/m^3)$.

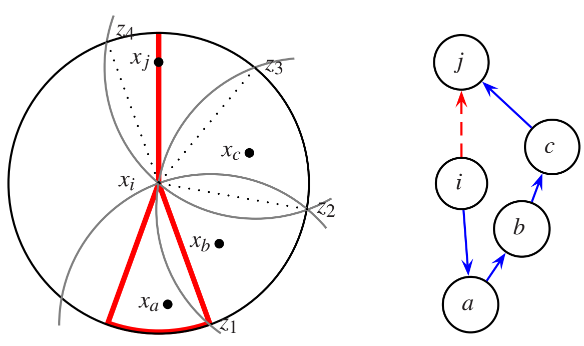

Figure 3: A simulating path.

Our final result is about connectivity properties in random sector graphs:

Lemma 3. Consider a random sector graph under the hypotheses stated in this section. Then, with high probability, for any two interior operative devices i and j with $d(x[i],x[j]) \leqslant r$, there is a directed path of length at most 4 from i to j.

Figure 3 sketches the proof given in [7]. The basic idea is that the graph is sufficiently dense to always have some antenna directed to the suitable direction. The directed path of length at most 4 between devices i and j is called a *simulating path* for i and j.

3. VIRTUAL NETWORK ESTABLISHMENT

In this section we present an initialization protocol to establish a virtual network on the top of the physical network.

3.1 Basic idea

Our basic idea consists in establishing a virtual network whose topology consists on a mesh of stars (see Figure 4). In this virtual network, we have a node of the mesh for each cell. The stars are centered around this node and include all the devices in its cell (we call peripheral these not central devices). The links in the virtual network are bidirectional. This network is virtual in the sense that no all its links exist in the physical network, they are rather simulated by paths of real links by the underlying logical system. In order to simulate the virtual links appropriately without collisions, some medium access protocol is needed. To do so, the system has to configure an efficient and reliable routing and scheduling.

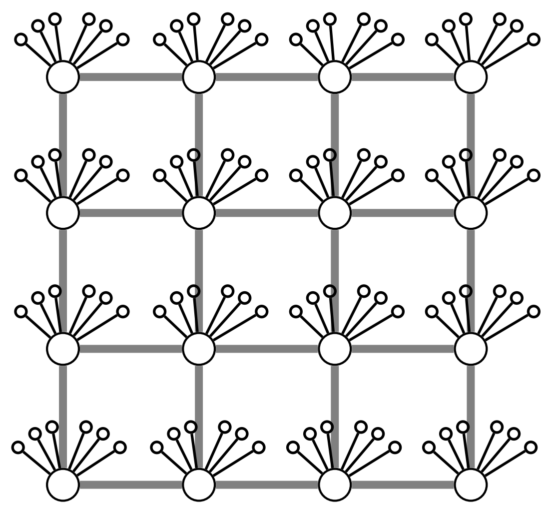

Figure 4: The virtual network has a mesh of stars topology.

3.2 Safe medium access control

In order to present our initialization algorithm we need a reliable (even if slow) method to guaranty that any device can send information without interferences from others.

Our first task is to assign to each device a unique identifier in its cell. We do so with the following randomized algorithm: Assign to each device u a random integer *rid*[u] uniformly selected between 0 and $M_n^5 - 1$. According to Lemma 2, with high probability, all devices in the same cell will have distinct identifiers.

Our second task is to coordinate the emission of messages between different cells. To do so, let $cx[u]$ and $cy[u]$ be the column and the row of the cell to which a device u belongs; then, define $s[u] := 7*(cx[u]\%7) + cy[u]\%7$. We multiplex time in a-phases. Each a-phase is made of 49 pulses. During the execution of our initialization protocol, at the t-th pulse, only devices u satisfying $s[u] = t\%49$ will be allowed to send information. Observe that because of the definition of $s[u]$ and the dimensions of the cells, this scheme allows the communication in one cell while avoids the communication in its $48 = (3+1+3)^2 - 1$ dangerous cells in a round robin way.

Since the previous scheme does still allow collisions inside a cell, our third task is to coordinate the emission of messages in a given cell. To do so, we further multiplex time in b-phases. Each b-phase is made of M_n^5 a-phases (and hence of $49M_n^5$ pulses). During the execution of our initialization protocol, at the t-th b-phase, only devices u satisfying $t\%M_n^5 = rid[u]$ will be allowed to send information.

As, by Lemma 1, M_n is an upper bound on the number of devices per cell and identifiers are unique, this time division multiplexing scheme guarantees that only one device can emit a message at the same pulse in a cell, and that no two devices can emit a message at the same pulse in mutually dangerous cells. Therefore, this scheduling excludes collisions and allows every device to send a message per round. Consequently, we describe now our protocol in terms of these safe primitives, which follow the indicated schedule.

3.3 Virtual network self-organization

In order to self-organize the system, we will ensure that each device knows the topology of the directed graph restricted to its own cell and its eighth neighboring cells. Notice that this information is not that big: it contains $9M_n = \Theta(a_n)$ vertices and $O(a_n^2)$ arcs.

By Lemma 3, there is a path of length 4 between any pair of devices in neighboring cells. Therefore, in order to have a complete knowledge of the directed graph restricted to nine boxes, it suffices to repeat 7 times the broadcasting of all the acquired information, that is, the arcs that u discovers by receiving from v. At each emission, the acquired information is incremented by the new obtained information. Note that 7 steps suffice: by pointer jumping, after time 1, all devices know the arcs at distance 1, after time 2, all devices know the arcs at distance 2, after time 3, all devices know the arcs at distance 4; then 4 time steps are necessary to ensure this information is broadcast to the rest of the devices in its cell and its neighboring cells. Observe that no two simultaneous emissions can cause collisions using the above mentioned scheduling.

Once each device has the knowledge of the local connections, each cell chooses the device with the lowest identifier as the central node of its corresponding star. As the central node of a cell also knows all the identifiers from its adjacent cells, it can also know their central nodes. Then identifiers of the rest of devices in each are ranked. In this way, identifiers will be in the $[0..M_n - 1]$ range rather than in the $[0..M_n^5 - 1]$ range. At this point, to improve communication speed, we redefine the b-phases of the communication schedule so that they contain M_n a-phases rather than M_n^5.

Next, the system must set the simulating paths. To do so, each device randomly chooses a simulating path of minimal length to communicate with its central node. Central nodes do the same to communicate with the central nodes of their four adjacent boxes. Observe that these paths have at most length 4 and can use devices from neighboring cells.

Each device proceeds now to inform other devices about its path. This is done by a round of $4M_n$ a-phases, started in turn according to the ranked identifiers. Each 4 phases started by a device consist in forwarding its message to the central node. Many devices can hear the messages, but only the unique device that must follow the path will forward it. In this way, the system learns the routing and the scheduling to simulate the virtual links. As a result, we have obtained a way to send information from each non central node to its central node and from each central nodes to its four adjacent central nodes. From this point on, we consider the initialization protocol finished and work with the virtual links using the resulting routing and scheduling.

3.4 Efficiency analysis

The initialization protocol we have described needs $7*49*M_n^5 + 4M_n = \Theta(a_n^5)$ pulses to finalize. This time is sub-linear with respect to n, but possibly not optimal. However, taking into account that this initialization protocol is just executed once, this is not an important concern. In fact, the expected number of times a device must do an emission of a message is just 11, which shows its energetical efficiency.

4. FUNCTIONALITIES

In this section we briefly present how the high level communication functionalities can be implemented on the top of the virtual network.

We describe communication on the virtual network implicitly assuming that we are using the emulating paths and the emission schedules described in the previous section. Communication on the virtual network is achieved through the repetition of the following three phases:

- Phase 1: Peripheral nodes in the stars send to their central node.
- Phase 2: Repeat k times: Central nodes send to their four neighbors in the mesh (left, right, up and down in turn).
- Phase 3: Central nodes send to their peripheral nodes.

Of course, no emission is done if there is nothing to transmit and the radio receiver can turned off when the scheduling indicates there is nothing no receive.

Observe that phases 1 and 3 need $\Theta(M_n)$ pulses but phase 2 only needs $k \cdot \Theta(1)$ pulses. Therefore, the value of the k parameter gives the potential to tune the priority one wants to give between the communication inside cells and the communication between cells. For instance, taking $k = 1$ gives to peripheral nodes the possibility to achieve a high throughput, while long distance communication speed is reduced, whilst taking $k = M_n$ fixes a low throughput to peripheral nodes but speeds up the long distance communication.

4.1 Point-to-point communication

Point-to-point communication in the network of sensors is easy to achieve on the top of the virtual network: We can send a message from device u to device v as follows: First u sends the message to the device in the center of the star of its cell, then this device forwards it to node in the center of the star of the cell of u using the mesh links, finally this node forwards it to the destination node v using the star.

Forwarding in the mesh is relatively easy, but in the case that congestion could be a matter, full of literature on the problem exists. However, we don't think this would be the case: almost all the devices will be sleeping most of the time. Also, one could increase k to reduce congestion.

Of course, the implementation of this point-to-point functionality must take into account that several different point-to-point transmissions can coexist at the same time in the system (other high level communication functionalities can also coexist). This means that devices must aggregate messages that follow the same direction. Also devices must separate messages that stop following the same direction. In the case that the size of the messages becomes too large, devices will have to queue them (again, we think this would be a rare situation in applications of sensor networks).

4.2 Point-to-area communication

We foresee that point-to-area communication can exist in two variations: a device wishes to send a message to all the devices in some area or a device wishes to send a message to a single but arbitrary device in some area. Both variations have similar solutions: The message, which contains some header with its destination area is sent from the originator device to its central node. Then, it is forwarded through the mesh to the central nodes or a central node of the reception area, from which it it is forwarded to all or one of the peripheral nodes. Since each device has knowledge on the coordinates of its cell, this is easy to do.

4.3 One-to-all communication

One-to-all communication is just the same as point-to-area communication when the area coincides with the full square. However, being a particular case, chances may be that the process can be slightly optimized when implemented.

5. APPLICATIONS

One of the most common scenarios for randomly deployed sensor networks involve the continuous monitoring of an extended geographic area at relatively low data rates [1]. The sensors may either be set up to report at regular intervals on some quantity of interest, such as temperature, humidity or concentration of some chemical substance, or be tuned to respond to some sporadic events of interest occurring within their observation range [10, 9, 12, 1]. The main task in those scenarios is to route all the relevant information to a base station for further processing. The collecting points can be located in a particular position [6], some randomly selected network points [7], or a unique designed node [8]. Our high level communication functionalities provide the means to easily solve the above mentioned problems using them directly, applying standard mesh communication protocols [11] or using other specific protocols for meshes of sensors [8].

6. CONCLUSIONS

In this paper we have considered how to organize a randomly deployed sensor network in order to offer high level communication functionalities. These functionalities make the programming of such networks much easier, as one concentrates on the logical properties of the devices rather than they rather limited physical properties. In such sense, we have provided a useful middle-ware level between the applications, the hardware and the transmission medium.

Our system is efficient and reliable in the sense that the resulting routing and scheduling permit to the devices to turn down their receiver or transmitter when not used. Moreover, the initialization protocol we have described runs in sub-linear time. Our system is also tunable, in the sense that one can adjust the balance between the throughput at which sensors emit messages and the speed these are transmitted through the network.

There still exist some limitations to our system: On one hand, fault-tolerance is not considered because we perform a self-organization that remains static. We think that periodical re-organizations of the system could cope with these failures. These re-organizations would also help to alleviate the higher power consumption of the central nodes. On the other hand, our model does not capture mobility at all. We leave these issues as future work.

7. REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramanian, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38:393–422, 2002.

[2] G. Arzhantseva, J. Díaz, J. Petit, J. Rolim, and M. Serna. Broadcasting on networks of sensors communicating through directional antennas. In P. Spirakis, A. Kameas, and S. Nikoletseas, editors, *International Workshop on Ambient Intelligence Computing*, pages 1–12, CTI Press, 2003. Ellinika Grammata.

[3] J. Aspnes, D. Goldengerg, and Y.R.Yang. On the computational complexity of sensor network localization. In *Proceedings of the First International Workshop on Algorithmic Aspects of Wireless Sensor Networks*, 2004. To appear.

[4] L. Bao and J. Garcia-Luna. Transmission scheduling in ad hoc networks with directional antennas. In *ACM Int. Conference on Mobile Computing and Networking*, pages 48–58, Atlanta, 2002.

[5] A. Bharathidasas and V. Anand. Sensor networks: An overview. Technical report, Dept. of Computer Science, University of California at Davis, 2002.

[6] I. Chatzigiannakis, S. Nikoletseas, and P. Spirakis. Smart dust local detection and propagation protocols. In *Proceedings of the second ACM international workshop on Principles of mobile computing*, pages 9–16, 2002.

[7] J. Díaz, J. Petit, and M. Serna. A random graph model for optical smart dust networks. *IEEE Transactions on Mobile Computing*, 2(3):186–196, 2003.

[8] M. Enachescu, A. Goel, R. Govindan, and R. Motwani. Scale free aggregation in sensor networks. In *Proceedings of the First International Workshop on Algorithmic Aspects of Wireless Sensor Networks*, 2004. To appear.

[9] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar. Next century challenges: Scalable coordination in sensor networks. In *ACM/IEEE Int. Conf. on Mobile Computing and Networking*, pages 263–270, Seattle, 1999.

[10] J. M. Kahn, R. H. Katz, and K. S. J. Pister. Emerging challenges: Mobile networking for smart dust. *Journal of Communications and Networks*, 2(3):188–196, 2000.

[11] F. T. Leighton. *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes*. Morgan Kaufmann, San Mateo, CA, 1993.

[12] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA 2002)*, pages 88–89, 2002.

[13] J. Razavilar, F. Rashisd-Farrokhi, and K. Liu. Software radio architecture with smarts antennas: a tutorial on algorithms and complexity. *IEEE Journal on Selected Areas in Communications*, 6:662–676, April, 1999.

[14] S. Tilak, N. B. Abu-Ghazaleh, and W. Heinzelman. A taxonomy of sensor network communication models. *Mobile Computing and Communication Review*, 6:28–36, 2002.

[15] V. Voipio and P. Vainikainen. Narrowbeam antenna array with sparse antenna spacing. In *IEEE Vehicular Technology Conference*, volume 1, pages 465–469, Ottawa, 1998.

[16] J. Winters. Smart antennas for wireless systems. *IEEE Personal Communications*, 5:23–27, 1998.