Self-configuring Node Clusters, Data Aggregation, and Security in Microsensor Networks*

Maleq Khan[1], Bharat Bhargava[1], Sarika Agarwal[1], Leszek Lilien[1], and Pankaj[2]

[1] Department of Computer Sciences and
Center for Education and Research in Information Assurance and Security (CERIAS)
Purdue University, West Lafayette, IN 47907
{mmkhan, bb, sagarwal, llilien}@cs.purdue.edu

[2] Department of Management Information Systems
Krannert Graduate School of Management
Purdue University, West Lafayette, IN 47907
pankaj@mgmt.purdue.edu

Abstract Microsensors operate under severe energy constraints and should be deployed in large numbers without any pre-configuration. We construct a generalized self-clustering algorithm, called Low-energy Localized Clustering (LLC). It integrates the ideas of two self-configuring clustering algorithms: the Localized algorithm and the Low Energy Adaptive Clustering Hierarchy algorithm. LLC covers a range of behaviors from the better-clustering performance of the Localized method to the energy-efficient operation of the LEACH method. The main advantage of LLC is that it can be energy-efficient while maintaining localization. Data aggregation techniques such as summarization, finding representative data items, and pattern matching are proposed. Data aggregation is a necessity in microsensor networks, since transmitting huge volumes of raw data is an energy-intensive operation. Finally, security issues are discussed and an energy-efficient Randomized Data Authentication algorithm is designed specifically for microsensor applications.

Keywords Sensor Networks, Microsensors, Self-configuring Clusters, Data Aggregation, Security in Microsensor Networks.

* This research was supported by CERIAS, and NSF Grants CCR-0001788 and EIA-0103676.

1 Introduction

Advances in integrated circuit technology have enabled mass production of tiny, cost-effective, and energy-efficient wireless sensor devices with on-board processing capabilities. The emergence of mobile and pervasive computing has created new applications for them. Sensor-based applications span a wide range of areas, including remote monitoring of seismic activities, environmental factors (e.g., air, water, soil, wind, chemicals), condition-based maintenance, smart spaces, military surveillance, precision agriculture, transportation, factory instrumentation, and inventory tracking [Bulu01, Hein00b].

A *microsensor* is a device which is equipped with a sensor module (e.g., an acoustic, a seismic, or an image sensor) capable of sensing some entity in the environment, a digital unit for processing the signals from the sensors and performing network protocol functions, a radio module for communication, and a battery to provide energy for its operation [Hein00b]. Currently, microsensors typically consist of 8-bit 4-MHz processors (80% of all microprocessors shipped in 2000 were 8-bit [Tenn00]), with slow 10-Kbps communication, an 8-Kbyte read-only program memory, and a 512-byte RAM [Perr01]. These parameters ensure limited weight, size, and cost. We use the term *sensor* to refer to a microsensor.

When deployed in large numbers and embedded deeply within large-scale physical systems, sensors gain the ability to measure aspects of the physical environment in unprecedented detail [Bulu01]. Networking these sensors with the ability to coordinate amongst themselves in a large sensing task will revolutionize information gathering and processing. Large scale, dynamically-changing, and robust sensor colonies can be deployed in inhospitable physical environments such as remote geographic regions or toxic urban locations. They will enable low-maintenance sensing in more benign but less accessible environments such as large industrial plants, enemy terrain, aircraft interiors, etc. [Estr99].

In this paper, we use a cluster-based hierarchical architecture for sensor networks to achieve and support scalability. An architecture, and description of its components and their functionalities is given in Section 2.

The large number of sensor nodes deployed for an application precludes manual configuration, and the environmental dynamics preclude design-time pre-configuration [Cerp02]. Nodes will have to self-configure to establish a topology that enables communication and sensing coverage under stringent energy constraints. Two existing self-configuring clustering algorithms: the Localized algorithm [Estr99] and the Low-Energy Adaptive Clustering Hierarchy (LEACH) [Hein00] are analyzed in Section 3. While the Localized algorithm forms good quality clusters, LEACH focuses on using lower energy consumption in forming clusters. Integrating the ideas of these two algorithms, we build a generalized scheme called Low-energy Localized Clustering (LLC).

Data aggregation is a good paradigm for wireless routing in sensor networks [Heid01, Itan00]. The idea is to combine the data coming from different sources and routes. This eliminates redundancy, minimizes the number of transmissions and thus saves energy [Kris02]. Beamforming [Oppe78, YaoH98] and functional decomposition [H00b] are two ways of aggregating sensor data. Their limitations are identified and a few other data aggregation methods are proposed in Section 4.

The data security requirements related to sensor networks such as confidentiality, authentication, integrity, and freshness [Perr01] are presented in Section 5. We also propose an energy-efficient Randomized Data Authentication algorithm.

2 Cluster-based Architecture for a Sensor Network

Sensor networks are large-scale data-intensive systems that manage parallel and real-time communications in dynamic environments. To support scalability we use a cluster-based hierarchical structure (see Fig. 1). As the number of basic sensors is increased, more clusters can be formed without increasing the processing or communication loads on individual cluster heads. The three levels in the hierarchical design of this architecture consist of a base station at the top level, cluster heads at the middle level, and basic sensors at the leaf level.

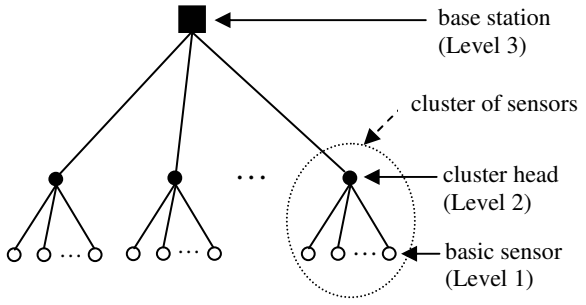

Fig. 1. A cluster-based hierarchical architecture for sensor networks.

The base station is a machine capable of analyzing the data collected from the cluster heads and displaying a global view of events being monitored. It is responsible for initiating and managing the network and is ultimately the gateway of the sensor network to the Internet or some other network.

Basic sensors are deployed in large numbers across an area of observation. Their primary function is to collect data from their surroundings. A direct communication between basic sensors occurs only at the time of cluster formation or cluster reconfiguration. Otherwise, the main stream of communication consists of conveying

data results to the corresponding cluster head. Before deployment, each basic sensor is given an *id* that uniquely identifies the sensor. Similarly a s*ecurity code*, which could be implemented as a hardware-embedded signature on the microsensor chip, is assigned to each sensor. It is used to authenticate data sent by the sensor node.

Cluster heads are selected from among the basic sensors by a self-configuring mechanism as detailed below. The role of a cluster head is to collect data from the basic sensors and manage them. Basic sensors in a particular cluster register themselves with their respective cluster head. The registration database is maintained by each cluster head to keep track of the range and type of information it is receiving. This database may also be used for other maintenance purposes such as determining the number of basic sensors supported and location lookup of sensors. The cluster heads become immediate points of contact for basic sensors for communication and reporting purposes. The heads collect data from the basic sensors, aggregate it and send to the base station.

3 Self–configuring Sensor Clusters

3.1 The Need for Clustering

As mentioned in the introduction, sensors must be able to self-configure into clusters. Clustering allows sensors to efficiently coordinate their local interactions in order to achieve global goals. Localized clustering can contribute to a more scalable behavior. As the number of nodes increases, it leads to improved robustness and more efficient resource utilization for many distributed sensor coordination tasks [Estr99].

Localization saves transmission energy since it allows communicating with a closer local coordinator instead of a more distant base station [Hein00a]. To transmit a signal over a distance d, the required energy E is proportional to d^m where m is 2 in free space and ranges up to 4 in environments with multiple-path interference local noise [Deli00].

An advantage of using clusters is data aggregation at the cluster heads, in which the collected data from the basic sensors is aggregated and sent to the base station. In this way, the amount of energy that would have been required to transmit huge volumes of data is reduced.

3.2 Analysis of Two Algorithms for Sensor Clustering

We introduce a self-configuring clustering algorithm, which is a generalization of the Localized and LEACH algorithms. Before presenting the new algorithm, some details of these two need to be explained.

The *Localized algorithm* to form self-configuring clusters of the sensor nodes is presented in [Estr99]. All sensors start by sending advertisements to sensors within a pre-specified radius defined in terms of physical hops. Sensors wait after setting their *wait timer* to values proportional to their radius. This allows advertisements from various sensors to reach each other. At the end of the wait period, sensors start a *promotion timer* which is set to be inversely proportional to the sensor's remaining energy and the number of other sensors from whom the advertisement were received. That is, the sensors in the dense regions and with higher energy have smaller timeout values. When a sensor's promotion timer expires, it promotes itself to Level 2 (a cluster head), and advertises itself as a cluster head by broadcasting the list of its potential child (basic) sensors. The list consists of the basic sensors whose advertisements it previously received. If a basic sensor appears in the lists of potential children of several cluster heads, it chooses the closest one as its cluster head. Now it cancels its own promotion timer (if it is still running), and thus drops out of the election process. If a cluster head does not have any children, or if its energy level drops below a certain threshold at some point, it demotes itself to a basic sensor. The process is repeated periodically. Thus, any change in network conditions or in sensor energy levels results in re-clustering.

The Low Energy Adaptive Clustering Hierarchy (LEACH) algorithm for self configuring clusters is proposed in [Hein00a]. Periodically, every basic sensor elects itself a cluster head with a certain probability. The probability for node n in round r is defined as:

$$T(n)=\begin{cases}\dfrac{P}{1-P\left(r \bmod \dfrac{1}{P}\right)} & \text{if } n \in G \\ 0 & \text{otherwise}\end{cases}$$

where P is the predefined percentage of sensors that should become cluster heads, and G is the set of nodes that have not been cluster heads in the last $1/P$ rounds. In every $1/P$ rounds each node is elected a cluster head once. Thus, the energy-intensive tasks of cluster heads are evenly distributed among the sensor nodes. The elected cluster heads broadcast an advertisement message to the rest of the nodes. A basic sensor selects its cluster head based on the strength of the received advertisement signal.

The main disadvantage of the Localized algorithm is that every node needs to broadcast messages and manage wait and promotion timers in each round of a cluster head election process, which requires a significant amount of energy. The LEACH algorithm is energy-efficient but the expected number of clusters is predefined. The authors presented an experimental result showing that the optimal number of cluster heads to minimize energy dissipation in data communication is approximately 5%. This value has been used to determine the predefined number of clusters. Unfortunately, when the sensors are highly dispersed, the percentage of sensors might not be sufficient to cover the whole area of sensor deployment. Even if a full coverage can be accomplished, the area covered by a cluster could increase to a point where long-range communication, and thus higher energy, is required. In contrast, since the Localized algorithm sends advertisements only to the sensors within a specified radius, the maximum area of a cluster is bounded, and there is no long-range communication that exceeds this radius.

Since the LEACH algorithm selects cluster heads randomly, in some instances all selected cluster heads can group into one end of the region. The sensors at the other end might not hear any cluster heads, and hence remain isolated from any cluster. In the Localized algorithm there can be no isolated groups of sensors. Every sensor has a promotion timer, which expires at some time and, if it does not hear from any other cluster head, it promotes itself to a cluster head.

3.3 The Low-energy Localized Clustering (LLC) Algorithm

We propose an algorithm called Low-energy Localized Clustering (LLC) that integrates the ideas of the algorithms discussed above to reduce the required energy in an election process (to improve upon the Localized algorithm), and to reduce the chance of having isolated sensors and to keep the number of cluster heads variable (to improve upon LEACH). The algorithm works in two phases: (a) a specified percentage of the nodes are randomly selected to be candidates for being cluster heads; (b) only the selected candidates compete to become cluster heads. Details of these two phases are given below.

Candidate selection Every node selects itself to be a candidate for a cluster head with a probability p. The probability p is proportional to the remaining energy of the node. Thus a sensor with higher energy has a greater chance to become a candidate.

Let t be the estimated lifetime of the system, which is estimated before deploying the sensors, and t_p be the time passed since deployment of the sensors. The estimated total energy (the sum of the remaining energies of all sensors) remaining in the system becomes:

$$e_{tr} = \frac{ne_i\left(t - t_p\right)}{t}$$

where e_i is the initial energy of each sensor node and n is the number of sensors. If the desired number of candidates is x% of the total number of sensor nodes,

$$p = \frac{e_r}{e_{tr}} \times n \times \frac{x}{100}$$

where e_r is the remaining energy of the sensor node.

Cluster head election The cluster heads are elected from the pool of candidates following the Localized algorithm. There are two exceptions: (a) only the candidate sensors compete while the remaining sensors sleep, and thus conserve energy, until the election process is completed; and (b) after a promotion, a node declares itself a cluster head but does not publish any potential children list. The other sensors (both former candidates and non-candidates) select their cluster

heads based on the strength of the signal of these declaration messages.

LLC overcomes the shortcomings of the two analyzed algorithms. Suppose that 20% of sensors are selected as candidates. Then, 80% of the sensors do not participate in the election process thus saving 80% of energy that is required to broadcast advertisements in the election process in the Localized algorithm. In LLC there is a slight chance of having an isolated group of sensors. If LEACH optimally selects 5% of the nodes as cluster heads, then in LLC the probability of selecting a candidate from an isolated group increases fourfold (for the candidate ratio set to 20%). If any sensor in the isolated group is selected as a candidate, the group will have a cluster head. The cluster radius is bound by radius hops as in the Localized algorithm.

LLC is an adaptive generalization of the Localization and the LEACH algorithms, with the candidate ratio being the control parameter. When the ratio is 100%, LLC behaves exactly like the Localized method, since all of the sensors are competing to become a cluster head. When the ratio is 5%, the algorithm operates nearly identically to the LEACH method, since for a low number of candidates nearly all will become cluster heads.

4 Sensor Data Aggregation

4.1 Motivation and Methods

Data aggregation is a paradigm for wireless routing in sensor networks [Heid01, Itan00]. The idea is to combine the data coming from different sources and routes. This eliminates redundancy, minimizes the number of transmissions, and saves energy [Kris02]. Automated methods of combining or aggregating the data into a small set of meaningful information are required [Hein00b].

Sensor data is different from data associated with traditional wireless networks since it is not the data itself that is important. Instead, it is the analysis of data, which allows an end-user to determine something about the monitored environment, which is the important result derived from a sensor network [Hein00b]. For example, if sensors are monitoring temperature, the measurements from all sensors in a cluster need not be transmitted. Temperatures at different points of a certain area are highly correlated and the end users are only interested in a high-level description of the events occurring. The type of a high-level description of data or data aggregation that needs to be performed depends on the monitored events and user requirements. In this example, only the minimum, maximum, or the average of the temperatures might be needed.

One method of data aggregation, called beamforming [Oppe78, YaoH98], combines signals from multiple sensors by calculating the weighted sum of the signals as follows:

$$y[n] = \sum_{i=1}^{N} \sum_{l=1}^{L} w_i[l] s_i[n-l]$$

where $s_i[n]$ is the signal from the i^{th} sensor, $w_i[n]$ is the weighting filter for the signal from the i^{th} sensor, N is the number of sensors, and L is the number of taps in the filter. Although beamforming has a good property that the weighting filters can be chosen to satisfy an optimization criteria, such as minimizing mean squared error or maximizing signal to noise ratio, the weighted sum of signals may not be useful for many applications.

A functional decomposition can sometimes be used to perform local data processing on a subset of data [Hein00b]. The base station receives all data X and processes it to find $f(X)$. The function f can sometimes be broken up into several smaller functions $f_1, f_2, f_3, \ldots, f_n$ that operate on subsets of data $X_1, X_2, X_3, \ldots, X_n$ such that

$$f(X) \approx g(f_1(X_1), f_2(X_2), f_3(X_3), \ldots, f_n(X_n)).$$

Even though many data aggregation functions can not be decomposed in such a manner, we can find some applications where special data aggregation techniques can be applied for local data processing in order to reduce the communication.

Various aggregation techniques can be applied to the proposed architecture of the sensor network. The following methods are proposed.

4.2 Data Summarization

For some data types and applications, only the summarized information is needed to serve the purpose of monitoring environmental events. Different summarizations are suitable for different

applications. They include average, sum, minimum, maximum, median, mode, standard deviation, quartiles, and percentiles. In addition, one can use the number of nodes detected to have crossed a threshold and the total number of active nodes associated with the cluster head. For some applications, instead of using a single summarized value, a combination of the above values can be employed.

4.3 Finding Representative Data Items

The data can be summarized by a predefined number of representatives. Let n be the number of active basic sensors associated with a cluster head, and k be the desired number of representatives. The *k-means algorithm* [Seli94] is an iterative procedure that has received considerable attention in clustering (data clustering) analysis, since it produces a good minimization of the sum-of-the-squared-error, or the total variance, function. This algorithm first randomly selects k initial cluster centers. Next k clusters are formed by associating each data point with its closest cluster center. The centroids, or means, of these k clusters become the new cluster centers. The above procedure is repeated until there is no change in the cluster memberships.

Each cluster center is the representative of data items in its cluster. The cluster heads send these k representatives to the base station. (Note that cluster centers represent data, while cluster heads represent sensors.) If desired, each representative can be accompanied by the number of data items in its cluster or other parameter, which can be used to indicate the weight or importance of the representative.

The k-means algorithm converges very fast when the dimension of the data is small. For example, for a temperature sensor network the dimension is one, whereas for a sensor network measuring both temperature and humidity dimension is two. Usually the dimensionality of the sensor data is small. This fact justifies why the k-means algorithm is a suitable clustering method for sensor data.

4.4 Pattern Matching

In some applications, the basic sensors may find the pattern of data measured over a predefined time interval and send only this pattern information to the cluster head, instead of sending the raw data. Cluster heads collect patterns from their basic sensors and select the critical patterns that describe some critical events. These critical patterns can be sent to the base station.

For example, consider an application with sensors deployed to predict storms in a certain area. Each basic sensor measures temperature and pressure and collects data. Periodically, a basic sensor finds the pattern of changing temperature and pressure that is the best fit for collected data. Six example patterns for pressure changes are depicted in Fig. 2.

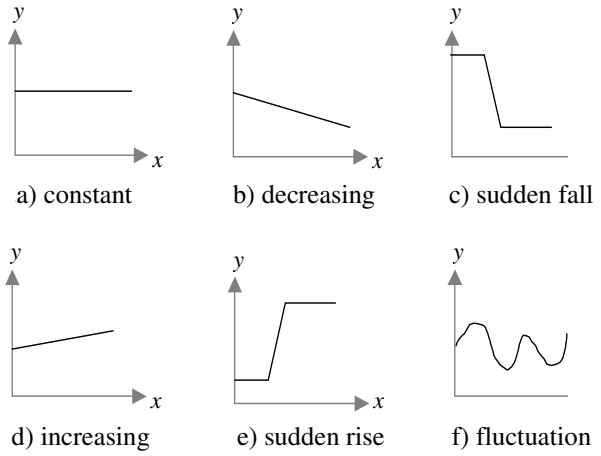

Fig. 2. Possible pressure patterns. Time is plotted on the x-axis and pressure on the y-axis.

Information sent periodically to the cluster head is concise. For example, we can send "c" denoting the sudden fall pattern of Fig. 2c. The sudden fall, sudden rise, and fluctuation in pattern pressure are categorized as critical patterns that forecast a storm. Similar patterns are defined for temperature. Each cluster head selects the patterns that predict a storm and sends them to the base station. The base station collects and analyzes the critical pressure and temperature patterns.

4.5 Tradeoffs between Data Aggregation and Communication

The main purpose of data aggregation is to reduce the required communication at various levels, and in turn to reduce the total energy consumption. Data aggregation saves energy if the energy required to perform aggregation is lower

than the energy required to send raw data to the upper level. Different data aggregation techniques require different amounts of energy to process raw data. The choice of data aggregation method depends not only on the application requirements but also on the relative energy savings obtained by using this method.

Another tradeoff between data aggregation and communication involves the time required to perform data aggregation at cluster heads or basic sensors, and time required to transmit raw data. There is a delay associated with processing data at sensor nodes due to their limited processing power. Depending on the application, partial processing of data can be done at the sensor nodes to maximize the data processing throughput, while still satisfying the real-time limitations of system operations.

5 Security Issues

5.1 Requirements and Solutions

The requirements for security in a sensor network, as stated in [Perr01], include: data confidentiality, data authentication, data integrity, and data freshness.

Data Confidentiality To assure data confidentiality within sensor levels, the standard encryption approaches can be employed. However, between sensor levels we propose adopting a strong security mechanism, with security protocols unique to each pair of levels.

Data Authentication Authentication is the mechanism by which the receiver of a message can ascertain its origin [Schn95]. Most of the existing authentication protocols require a trusted third party that generates secret keys for the communicating parties. Using a third party is not suitable for authentication of sensor nodes, deployed on a temporary basis and frequently reconfigured. Moreover, no IP address, required to communicate with a third party, is associated with a sensor node.

We propose the *Randomized Data Authentication* algorithm suitable for low-energy sensor networks. Every sensor node is given a unique *id* and a *security code*. All of the id–security pairs are stored at the base station. During data transmission from the basic sensors, the cluster heads periodically verify the sender of the data item as follows:

a. When a cluster head receives a data item from a basic sensor, it generates a random number x, where $0 \le x \le 1$.
b. If $x \le p$, the cluster head requests the sensor to send its security code. Here p is the predefined probability that a cluster head requests the sender for its security code.
c. After receiving the security code, the cluster head sends the id–security code pair to the base station for verification.

Energy of sensors is saved by not authenticating each data item. Also the risk of compromising security is reduced, since less frequent random authentication gives attackers fewer opportunities to capture a security code. On the other hand, more intrusions may remain undetected. However, in most cases a few intrusions can be tolerated. Since data is being gathered from a large number of sensors, a relatively few malicious data items do not affect the overall results significantly. In this algorithm, repeated intrusions are most likely to be detected. In each sequence of $1/p$ intrusions, we expect one intrusion to be detected. The probability p can be dynamically adjusted either to increase detection probability, e.g. when more intrusions are detected in the network, or to decrease detection chances, e.g. when no intrusions were experienced for a long period of time.

Data Integrity Data integrity in the networking environments includes processing data integrity, database (data storage) integrity, and data communication integrity. Standard approaches to data processing and database integrity can be used, since they are not more critical in sensor networks than in other networking environments. However, data communication integrity becomes even more critical, if only due to a sheer volume of transmitted sensor data. To protect data in transit from intentional attacks, we propose using the above Randomized Data Authentication algorithm. This solution is adequate, since authentication is a stronger property than data integrity for data communication [Perr01] (under the assumption that the sender is not

compromised). The standard protection against accidental failures includes using the error detection and correction facilities of the TCP/IP protocol suite [Perr01].

Data Freshness Data freshness means maintaining currency of data and ensuring that old messages do not masquerade as the current ones. There are two types of freshness: strong and weak [Perr01]. Strong freshness is used for time synchronization within the network, while weak freshness is required by sensor measurements.

5.2 Vulnerabilities in Sensor Networks

Vulnerabilities specific to sensor networks result from their capabilities to self-configure and from the wireless communication facilities embedded within the nodes. Potential solutions for prevention, fast detection of attacks, graceful degradation, and recovery should include both threat-avoiding and threat-tolerant approaches.

Some attacks may be prevented by using a tight authentication procedure during the registration of basic sensors and cluster heads with the base station. Otherwise, attackers may be able to deploy counterfeit sensors or take over sensors, even cluster heads. Even such a security breach should not enable an attacker to take over the entire network.

Data integrity and consistency mechanisms [Amma97] can be employed for detecting intrusions, especially in cases when an attacker wishes to corrupt data. For instance, in the field of precision agriculture, an adversary who wants to destroy crops could cause sensors to report acceptable levels of soil moisture, while actually the field needs watering. The integrity/consistency checker would detect bad data reports and alert the base station or the cluster head.

Under the worst scenario, an attacker gains control of the base station and compromises the entire network. This is dealt with by network-wide security means, including intrusion detection mechanisms of the base station and possibly the wider network. This well-researched issue is outside the scope of this paper, since we are concentrating on enforcing security at sensor node levels.

6 Conclusions

The proposed self-configuring clustering algorithm, called Low-energy Localized Clustering (LLC), is a generalization of the Localized and the LEACH algorithms. The ratio of candidates for cluster heads is the parameter used to control the behavior of LLC. The main advantage of LLC is that it can be energy-efficient while maintaining localization.

A number of data aggregation techniques, such as summarization, finding representative data items, and pattern matching have been proposed to provide an efficient way of processing data in a sensor environment.

A novel Randomized Data Authentication algorithm, which uses minimal energy as required in microsensor applications, has been developed.

References

[Amma97] P. Ammann, S. Jajodia, C. McCollum, and B. Blaustein, "Surviving information warfare attacks on databases," in *Proceedings of IEEE Computer Society Symposium on Security and Privacy,* Oakland, CA, May 1997.

[Bulu01] N. Bulusu, D. Estrin, L. Girod and J. Heidemann, "Scalable Coordination for Wireless Sensor Networks: Self-Configuring Localization Systems," in *Proceedings of the Sixth International Symposium on Communication Theory and Applications (ISCTA 2001),* Ambleside, United Kingdom, July 2001.

[Cerp02] A. Cerpa and D. Estrin, "ASCENT: Adaptive Self-Configuring Sensor Networks Topologies," in *Proceedings of the Twenty First International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002),* New York, NY, June 2002.

[Deli00] K. A. Delin and S. P. Jackson, "Sensor Web for In Situ Exploration of Gaseous Biosignatures," in *Proceedings of IEEE Aerospace Conference,* Big Sky, MT, March 2000.

[Estr99] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next Century Challenges: Scalable Coordination in sensor Networks," in *Proceedings of the Fifth*

Annual International Conference on Mobile Computing and Networks (MobiCom), Seattle, WA, August 1999.

[Heid01] J. Heidemann, F. Silva, C. Intanagonwiwat, R. Govindan, D. Estrin, D. Ganesan, "Building Efficient Wireless Sensor Networks with Low-Level Naming," in *Proceedings of the 18th ACM Symposium on Operating Systems Principles,* October 2001.

[Hein00a] W. R. Heinzelman, A. Chandrakasan, and H. Balakrisnan, "Energy-efficient Communication Protocol for Wireless Microsensor Networks," in *Proceedings of the 33rd International Conference on System Sciences (HICSS),* January 2000.

[Hein00b] W. Heinzelman, "Application-Specific Protocol Architectures for Wireless Networks," Ph.D. Thesis, Department of Electrical Engineering and Computer Science, MIT, Cambridge, MA, June 2000.

[Itan00] C. Itanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," in *Proceedings of the Sixth Annual International Conference on Mobile Computing and Networks (MobiCom),* 2000.

[Kris02] B. Krishanamachari, D. Estrin, and S. Wicker, "The Impact of Data Aggregation in Wireless Sensor Networks," in *Proceedings of the International Workshop on Distributed Event Based Systems (DEBS),* Vienna, Austria, July 2002.

[Oppe78] A. Oppenheim, *Applications of Digital Signal Processing,* Prentice-Hall, Inc., 1978.

[Perr01] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar, "SPINS: Security protocols for sensor networks," in *Proceedings of the Seventh Annual International Conference on Mobile Computing and Networks (MobiCom),* pp.189-199, 2001.

[Schn95] B. Schneier, *Applied Cryptography,* John Wiley and Sons, 1995.

[Seli94] S. Z. Selim and M. A. Ismail, "*k*-Means-Type Algorithms: A Generalized Convergence Theorem and Characterization of the Local Optimality," *IEEE Transactions on Pattern Analysis and Machine Intelligence*," 6(1), pp. 81-87, 1994.

[Tenn00] D. Tennenhouse, "Embedding the Internet: Proactive Computing", *Communications of the* ACM, 43(5), pp. 43-43, 2000.

[YaoH98] K. Yao, R. Hudson, C. Reed, D. Chen, and F. Lorenzelli, "Blind Beamforming on a Randomly Distributed Sensors Array System," in *Proceedings of the 1998 IEEE Workshop on Signal Processing Systems (SiPS '98),* October 1998.