Sensor Networks: An Overview

Archana Bharathidasan, Vijay Anand Sai Ponduru

Department of Computer Science
University of California, Davis, CA 95616
Email: {bharathi, ponduru}@cs.ucdavis.edu

Abstract

Sensor networks are dense wireless networks of small, low-cost sensors, which collect and disseminate environmental data. Wireless sensor networks facilitate monitoring and controlling of physical environments from remote locations with better accuracy. They have applications in a variety of fields such as environmental monitoring, military purposes and gathering sensing information in inhospitable locations. Sensor nodes have various energy and computational constraints because of their inexpensive nature and ad-hoc method of deployment. Considerable research has been focused at overcoming these deficiencies through more energy efficient routing, localization algorithms and system design. Our survey attempts to provide an overview of these issues as well as the solutions proposed in recent research literature

1. Introduction:

Recent technological improvements have made the deployment of small, inexpensive, low-power, distributed devices, which are capable of local processing and wireless communication, a reality. Such nodes are called as sensor nodes. Each sensor node is capable of only a limited amount of processing. But when coordinated with the information from a large number of other nodes, they have the ability to measure a given physical environment in great detail. Thus, a sensor network can be described as a collection of sensor nodes which co-ordinate to perform some specific action. Unlike traditional networks, sensor networks depend on dense deployment and co-ordination to carry out their tasks.

Previously, sensor networks consisted of small number of sensor nodes that were wired to a central processing station. However, nowadays, the focus is more on wireless, distributed, sensing nodes. But, why distributed, wireless sensing? [12] When the exact location of a particular phenomenon is unknown, distributed sensing allows for closer placement to the phenomenon than a single sensor would permit. Also, in many cases, multiple sensor nodes are required to overcome environmental obstacles like obstructions, line of sight constraints etc. In most cases, the environment to be monitored does not have an existing infrastructure for either energy or communication. It becomes imperative for sensor nodes to survive on small, finite sources of energy and communicate through a wireless communication channel.

Another requirement for sensor networks would be distributed processing capability. This is necessary since communication is a major consumer of energy. A centralized system would mean that some of the sensors would need to communicate over long distances that leads to even more energy depletion. Hence, it would be a good idea to process locally as much information as possible in order to minimize the total number of bits transmitted.

1.1 Applications of sensor networks:

Sensor networks have a variety of applications. Examples include environmental monitoring - which involves monitoring air soil and water, condition based maintenance, habitat monitoring (determining the plant and animal species population and behavior), seismic detection, military surveillance, inventory tracking, smart spaces etc. In fact, due to the pervasive nature of micro-sensors, sensor networks have the potential to revolutionize the very way we understand and construct complex physical system [11].

1.2 Challenges

In spite of the diverse applications, sensor networks pose a number of unique technical challenges due to the following factors:

- **Ad hoc deployment:** Most sensor nodes are deployed in regions which have no infrastructure at all. A typical way of deployment in a forest would be tossing the sensor nodes from an aeroplane. In such a situation, it is up to the nodes to identify its connectivity and distribution.

- **Unattended operation:** In most cases, once deployed, sensor networks have no human intervention. Hence the nodes themselves are responsible for reconfiguration in case of any changes.

- **Untethered:** The sensor nodes are not connected to any energy source. There is only a finite source of energy, which must be optimally used for processing and communication. An interesting fact is that communication dominates processing in energy consumption. Thus, in order to make optimal use of energy, communication should be minimized as much as possible.

- **Dynamic changes:** It is required that a sensor network system be adaptable to changing connectivity (for e.g., due to addition of more nodes, failure of nodes etc.) as well as changing environmental stimuli.

Thus, unlike traditional networks, where the focus is on maximizing channel throughput or minimizing node deployment, the major consideration in a sensor network is to extend the system lifetime as well as the system robustness [7].

1.3 Survey Focus:

A number of papers propose solutions to one or more of the above problems. Our survey focuses on the suggested solutions in the following areas:

Energy Efficiency: Energy efficiency is a dominant consideration no matter what the problem is. This is because sensor nodes only have a small and finite source of energy. Many solutions, both hardware and software related, have been proposed to optimize energy usage.

Localization: In most of the cases, sensor nodes are deployed in an ad hoc manner. It is up to the nodes to identify themselves in some spatial co-ordinate system. This problem is referred to as localization.

Routing: Communication costs play a great role in deciding the routing technique to be used. Traditional routing schemes are no longer useful since energy considerations demand that only essential minimal routing be done.

Besides the above topics, we will also look at some proposed sensor network systems. We also have a quick look at some of the simulators available today for simulating sensor networks.

1.4 Organization

Section 2 gives a brief overview of a sensor networking architecture and an operating system geared for sensor networks. In section 3, various solutions dealing with energy efficiency are looked at. Section 4 explains the localization problem and considers various techniques proposed to carry out localization. Section 5 considers various routing protocols for sensor networks. In section 6, we describe some systems which make use of sensor-networking concepts. Section 7 gives a very brief overview of some sensor network simulators available. Section 8 concludes.

2. Architecture:

To have a general idea of the kind of architectures and operating systems which are suitable for sensor networks, we give an example of each.

[23] proposes a middleware architecture called SINA (Sensor Information Networking Architecture). The architecture has the following components.

Hierarchical clustering: The sensor nodes are organized into a hierarchy, based on their power levels and proximity. A cluster head is elected to perform various functions, with ability for re-initiation should the cluster head fail.

Attribute-based naming: The sensor nodes are named based on their attributes. For example, consider a system which is used to measure temperature at a particular location. Then, the name [type=temperature, location=N-E, temperature=103] refers to all the sensors located at the northeast quadrant with a temperature reading of 103F. Thus, they can reply when a query like "which area has a temperature more than 100F" is posed. Such a scheme works because the nodes are by themselves *neither unique nor dependable*. So, applications access a particular data element by naming it directly. This approach has another advantage in that it eliminates the need for maintaining mapping/directory services, which is an extra overhead.

Most sensor data is associated with the physical context of the phenomena being sensed. Hence spatial coordinates are a natural way to name data. This makes localization - determination of the position of the node in some co-ordinate system - an important problem [rf based localization], which we is discussed in section 4.1.

The SINA architecture proposes Sensor Query and Tasking language (SQTL) as the programming interface between sensor applications and SINA middleware. The SQTL defines three events: receive, query and expire. An SQTL message consisting of a script should be interpreted and executed by any node in the network. The authors have described some sample applications like co-coordinated vehicle tracking which can be carried out using the nodes built using the SINA architecture.

TinyOS [25] is a component-based operating system that is specially designed for sensor networks. [9] describes an active message communication model using TinyOS which can be used as a building block for carrying out higher level networking capabilities.

3. Energy Efficiency :

Energy consumption is the most important factor to determine the life of a sensor network because usually sensor nodes are driven by battery and have very low energy resources. This makes energy optimization more complicated in sensor networks because it involved not only reduction of energy consumption but also prolonging the life of the network as much as possible. This can be done by having energy awareness in every aspect of design and operation. This ensures that energy awareness is also incorporated into groups of communicating sensor nodes and the entire network and not only in the individual nodes.

A sensor node usually consists of four sub-systems [21]:

- a **computing** subsystem : It consists of a microprocessor(microcontroller unit,MCU) which is responsible for the control of the sensors and execution of communication protocols. MCU's usually operate under various operating modes for power management purposes. But shuttling between these operating modes involves consumption of power, so the energy consumption levels of the various modes should be considered while looking at the battery lifetime of each node.
- a **communication** subsystem: It consists of a short range radio which is used to communicate with neighboring nodes and the outside world. Radios can operate under the Transmit, Receive, Idle and Sleep modes. It is important to completely shut down the radio rather than put it in the Idle mode when it is not transmitting or receiving because of the high power consumed in this mode
- a **sensing** subsystem : It consists of a group of sensors and actuators and link the node to the outside world. Energy consumption can be reduced by using low power components and saving power at the cost of performance which is not required.
- a **power supply** subsystem : It consists of a battery which supplies power to the node. It should be seen that the amount of power drawn from a battery is checked because if high current is drawn from a battery for a long time, the battery will die even though it could have gone on for a longer time. Usually the rated current capacity of a battery being used for a sensor node is lesser than the minimum energy consumption required leading to the lower battery lifetimes. The lifetime of a battery can be increased by reducing the current drastically or even turning it off often.

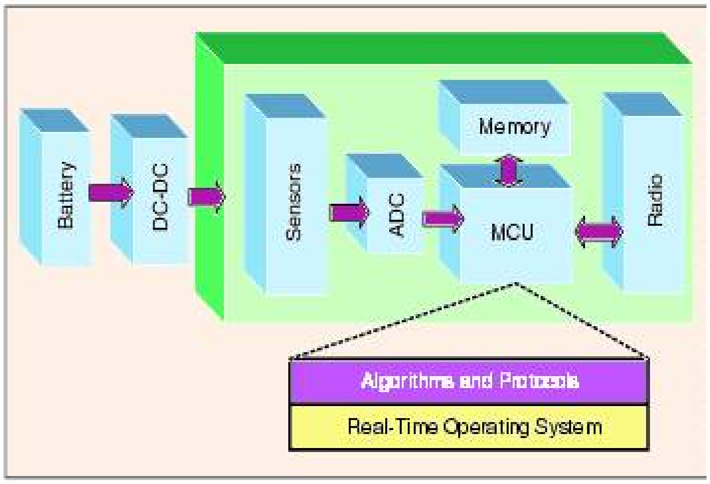

Figure 1. *System architecture of a typical wireless sensor node*

The power consumed by the sensor nodes can be reduced by developing design methodologies and architectures which help in energy aware design of sensor networks. The lifetime of a sensor network can be increased significantly if the operating system, the application layer and the network protocols are designed to be *energy aware*. Power management in radios is very important because radio communication consumes a lot of energy during operation of the system. Another aspect of sensor nodes is that a sensor node also acts a router and a majority of the packets which the sensor receives are meant to be forwarded. Intelligent radio hardware that help in identifying and redirecting packets which need to be forwarded and in the process reduce the computing overhead because the packets are no longer processed in the intermediate nodes.

Traffic can also be distributed in such a way as to maximize the life of the network. A path should not be used continuously to forward packets regardless of how much energy is saved because this depletes the energy of the nodes on this path and there is a breach in the connectivity of the network. It is better that the load of the traffic be distributed more uniformly throughout the network.

It is important that the users be updated on the health of a sensor network because this would serve as a warning of a failure and aid in the deployment of additional sensors. Younggang Zhao *et al*. [31] propose a mechanism which collects a *residual energy scan* (**eScan**) of the network which is an aggregated picture of the energy levels in the different regions of the sensor network. They also propose to use incremental updates to scans so that when the state of a node changes, it does not have to send its entire scan again thereby saving energy.

4. Localization:

In sensor networks, nodes are deployed into an unplanned infrastructure where there is no *a priori* knowledge of location. The problem of estimating spatial-coordinates of the node is referred to as localization. An immediate solution which comes to mind, is GPS [2] or the Global Positioning System. However, there are some strong factors against the usage of GPS. For one, GPS can work only outdoors. Secondly, GPS receivers are expensive and not suitable in the construction of small cheap sensor nodes. A third factor is that it cannot work in the presence of any obstruction like dense foliage etc. Thus, sensor nodes would need to have other means of establishing their positions and organizing themselves into a co-ordinate system without relying on an existing infrastructure.

Most of the proposed localization techniques today, depend on recursive trilateration/multilateration techniques [7]. One way of considering sensor networks is taking the network to be organized as a hierarchy with the nodes in the upper level being more complex and already knowing their location through some technique (say, through GPS). These nodes then act as beacons by transmitting their position periodically. The nodes which have not yet inferred their position, listen to broadcasts from these beacons and use the information from beacons with low message loss to calculate its own position. A simple technique would be to calculate its position as the centroid of all the locations it has obtained. This is called as *proximity based localization*. It is quite possible that all nodes do not have access to the beacons. In this case, the nodes which have obtained their position through proximity based localization themselves act as beacons to the other nodes. This process is called iterative multilateration. As can be guessed, iterative multilateration leads to accumulation of localization error.

Since most of the localization algorithms use some form of trilateration, a brief overview of trilateration based on [3], is given. Consider a person A, who wants to determine his position in 2-D space. Suppose A knows that he is 10kms from a point x. Then he can determine that he is anywhere on the circle of radius 10kms around the point x. Now, if A also knows that he is 20 kms from a point y, A can deduce that he is on either one of the two intersecting point of the circle of radius 10km around x and the circle of radius 20km around point y. Suppose A also has additional information that he is 15km from a point z. Now he knows at which of the two intersecting points he is one because only one of them will intersect with the third circle also. This is shown in figure 2 [3] below. Let x be Boise, y be Minneapolis and z be Tucson.

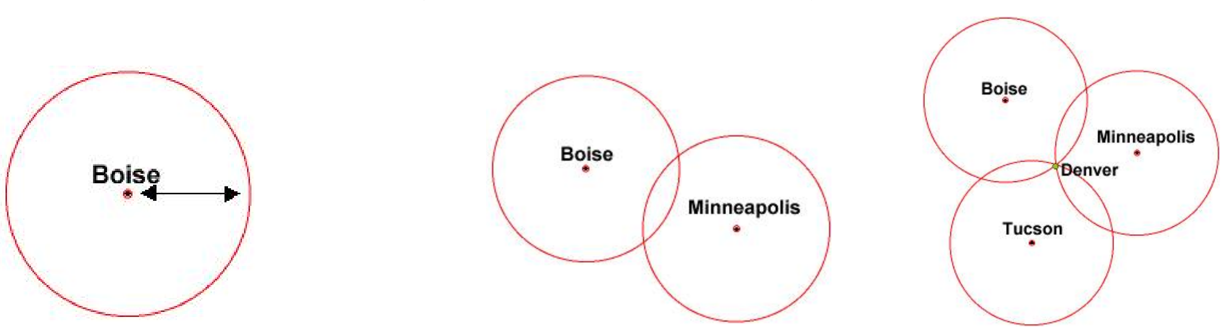

Figure 2. *Principle of trilateration in 2-D space*

Thus, trilateration is a geometric principle which allows us to find a location if its distance from other already-known locations are known. The same principle is extended to three-dimensional space. In this case, spheres instead of circles is used and four spheres would be needed. This is the principle used in GPS also. Figure 3 [3] demonstrates trilateration in 3-D space as used in GPS.

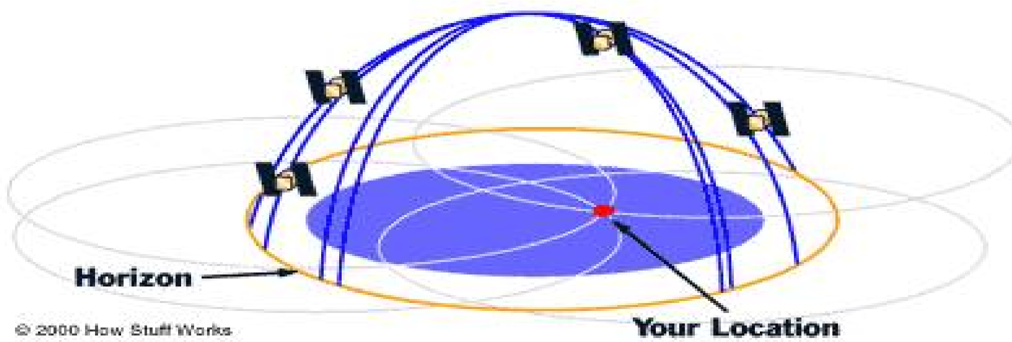

Figure 3. *Principle of trilateration in 3-D space as used in GPS.*

When a localization technique using beacons is used, an important question would be 'how many initial beacons to deploy'. Too many beacons would result in self-interference among the beacons while too less number of beacons would mean that many of the nodes would have to depend on iterative multilateration. Many papers research techniques to solve this problem. 4.2 discusses some of them. An associated problem would be to decide the total number of sensor nodes required in a given area. That is, determining the network density. [7] defines network density as :

$$\mu(R) = (N \,.\, \pi \,.\, R^2) / A$$

where, N is the number of nodes in a region of area A whose nominal range is given by R. Beyond a critical value λ, addition of extra nodes does not provide additional sensing nor coverage fidelity. Hence techniques would be required to decide optimum deployment.

4.1 Localization Techniques:

[4] gives an over-view of the various localization techniques. Localization can be classified as *fine-grained*, which refers to the methods based on timing/signal strength and *coarse-grained*, which refers to the techniques based on proximity to a reference point.

Examples of fine-grained localization are:

Timing: The distance between the receiver node and a reference point is determined by the time of flight of the communication signal.

Signal strength: As a signal propagates, attenuation takes place proportional to the distance traveled. This fact is made use of to calculate the distance.

Signal pattern matching: In this method, the coverage area is pre-scanned with transmitting signals. A central system assigns a unique signature for each square in the location grid. The system matches a transmitting signal from a mobile transmitter with the pre-constructed database and arrives at the correct location. But pre-generating the database goes against the idea of ad hoc deployment.

Directionality: Here, the angle of each reference point with respect to the mobile node in some reference frame is used to determine the location.

Examples of coarse-grained localization are:

Proximity based localization as described earlier. Some of the sensor systems [26,27,28] in use today, described in section 6 use coarse-grained techniques.

[4] proposes a localization system which is RF-based, receiver-based, ad hoc, responsive, low-energy consuming and adaptive. RF-based transceivers would be more inexpensive and smaller compared to GPS-receivers. Also in an infrastructure less environment, the deployment would be ad hoc and the nodes should be able to adapt themselves to available reference points.

A node, which has to calculate its position, receives signals from a collection of reference points. All these reference points have their connectivity metric above a pre-decided threshold. Connectivity metric is defined as the ratio of the total number of signals received by a node to the total number of signals sent by a node.

Once the node receives the signal, it calculates its position as the centroid of the positions of all the reference nodes as :

$$(X_{est}, Y_{est}) = ((X_{i1}+\ldots+X_{ik})/k, (Y_{i1}+\ldots+Y_{ik})/k))$$

where X_{i1}, Y_{i1} gives the position of the first reference point, X_{i2}, Y_{i2} gives the position of the second reference point and so on. The accuracy of the estimate can be determined by calculating the localization error .

$$LE= ((X_{est}-X_a)^2 + (Y_{est}-Y_a)^2)^{1/2}$$

By increasing the range overlap of reference points, the accuracy of the location estimate improves. [8] describes the prototype implementation of a model based on the above techniques. The prototype was found to be feasible for outdoor use in restricted domains.

A different approach towards localization has been adapted in [10]. Here, the task of determining the position of sensor node is treated as a mathematical problem. The nodes can be treated as nodes of a graph and the connections between them can be treated as edges. Also, the placement of the nodes is constrained by certain requirements like line of sight etc. Hence, the problem becomes similar to solving an LP (Linear Problem) with a set of equations and a set of constraints.

Another interesting approach is taken by Howard et al. [15]. This approach is geared towards robotics and considers the problem of robots identifying their own position as well as measuring the relative identity of nearby robots. The same authors view localization as a co-ordinate transform problem in [16]. Here, the authors state that if the position of a node were known in two different co-ordinate systems, in some arbitrary global co-ordinate system, these two locations would map onto the same point. If a co-ordinate transformation operator Γ, which maps points from the local to the global co-ordinate system, is known, we can write

$$\Gamma_a Z_a - \Gamma_b Z_b = 0$$

Where, Z_a and Z_b are the co-ordinates in the local systems a and b. Now, each local co-ordinate system is represented as a rigid body and each correspondence is represented as a spring joining two points on the rigid bodies. By allowing the spring to relax to its lowest energy configuration, the optimal set of co-ordinate transforms can be inferred.

4.2 Beacon/Node Placement Techniques:

Beacon placement techniques which suggest themselves at once are: *1. Uniform beacon placement 2. Very dense beacon placement.* However these methods are not sufficient. Consider airdropped beacons over a hill. Then the heavier beacons would roll down the hill while the lighter sensor nodes would remain atop. Similarly uniform placement does not necessarily ensure visibility. Cost/power might be a major consideration for dense placement. Even otherwise, as pointed out earlier, too many beacons cause self-interference.

Various techniques that have been proposed to ensure optimum placement of nodes. Mostly, problems arise due to the unpredictable nature of environmental conditions. Nodes thus will also need to be able to adapt to environmental changes.

[5] suggests incremental beacon placement based on empirical adaptation. By this, the authors mean that beacon placement is adjusted through adding more beacons rather than complete re-deployment and that additional deployment decisions are made by local measurements rather than complete off-line analysis of the whole model. Three algorithms have been suggested. All these algorithms just suggest the locations where the beacons could be placed - the method of actually deploying the beacons to the placing the beacons is left to the user.

1. **Random:** As the name suggests, any random location is chosen as a suitable candidate.

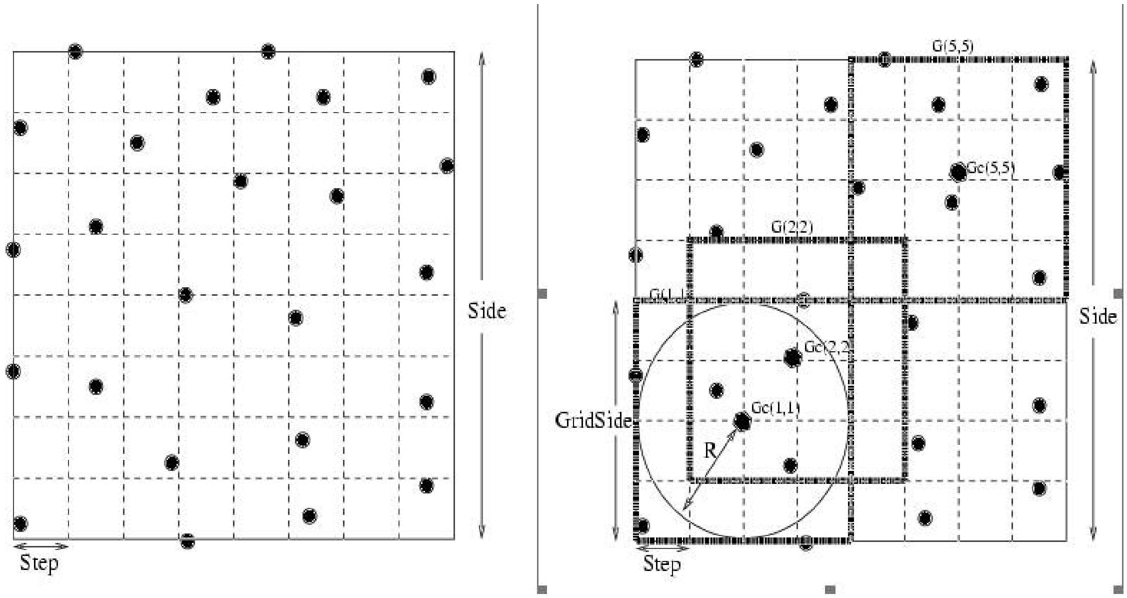

Figure 4. *The MAX and the GRID protocols*

2. Max: In this case, the terrain is divided into step*step squares. The localization error is calculated at each square corner. A beacon is added at the point which has the maximum localization error. Even though this approach is simple, it suffers from being overly influenced by propagation effects or random noises. Figure 4 [5] illustrates the Max algorithm.

3. Grid: The Grid approach computes the cumulative localization error over each grid for several overlapping grids as illustrated in figure 4 [5]. A new beacon is added at the center of the grid which has the maximum cumulative localization error.

The authors have shown that at low densities grid algorithm significantly improves the mean and median errors. Even though computationally expensive, the grid algorithm is superior to the other two.

[6] further builds upon the ideas above. The HEAP algorithm incorporates the concepts of the max and grid algorithms. It further details the actual implementation of the algorithm. It also proposes a technique called STROBE (Selectively Turning Off BEacons) for achieving adaptive operational density of beacons. The goal of the system is to extend the lifetime of the sensor network, achieve uniform granularity and minimize energy consumption. The crux of the idea is as follows. When a high density of beacons is deployed, only a certain percentage of the beacons are activated. This ensures that the duty cycles of individual beacons are reduced while still maintaining the same level of granularity thus meeting the above three goals.

In the STROBE technique, each node can be in one of the three self-explanatory states: BEACON_ONLY, LISTEN AND BEACON and SLEEP. Separate states are needed for BEACON and LISTEN AND BEACON since listening also consumes energy. All beacons start in the LB state. In the LB cycle, each node evaluates its connectivity, i.e, the total number of beacons it has heard from. When the connectivity exceeds a pre-determined threshold, the node goes either into SLEEP or BO state with a calculated probability. This method does not take into account factors like residual energy in other nodes etc.

[15] describes self-deployment algorithms for mobile sensor networks. This algorithm carries out actual self-deployment of nodes one by one into an unknown environment. The goal of the algorithm is to maximize the network coverage (blanket coverage which maximizes the total detection area) and at the same time ensure that the nodes retain line of sight communication with each other. The algorithm assumes that all the nodes are identical, the environment itself is static and that the position of each node is known in some arbitrary global co-ordinate system [16]. The algorithm has four phases: initialization, selection, assignment and execution.

Selection determines the next suitable deployment location. This is done by having the already deployed nodes form an occupancy grid. Each grid can be free, occupied or unknown. Bayesian techniques are used to determine the probability that a cell is occupied. Reachability grids are also used to determine if a grid is reachable (in some cases, even though a cell might be free it might be unreachable since these sensor nodes have finite dimensions). The next location for deployment is determined based upon the boundary - where the nodes should be deployed to the boundary between free and unknown space and coverage - nodes should be deployed to cover the maximum area of presently unknown space heuristics.

Once selection is done, **assignment** attempts to assign the selected goal to a waiting node. In case an already deployed node obstructs a path, the waiting node is deployed to that location and the obstructing node is moved to the new location. A more refined version of this procedure is carried out.

The next phase is **execution**, when active nodes are actually deployed to their goal locations. This deployment is carried out sequentially, where each node is allowed to reach its location before the next node is deployed. The authors are working on concurrent execution. Simulation experiments have been carried out which demonstrate the utility of the algorithm, since the algorithm achieves 75-85% of the coverage achieved by greedy techniques.

5. Routing:

Conventional routing protocols have several limitations when being used in sensor networks due to the energy constrained nature of these networks. These protocols essentially follow the flooding technique in which a node stores the data item it receives and then sends copies of the data item to all its neighbors. There are two main deficiencies to this approach [14].

Implosion : If a node is a common neighbor to nodes holding the same data item, then it will get multiple copies of the same data item. Therefore, the protocol wastes resources sending the data item and receiving it.

Resource management : In conventional flooding, nodes are not resource-aware. They continue with their activities regardless of the energy available to them at a given time.

The routing protocols designed for sensor networks should be able to overcome both these deficiencies or/and look at newer ways of conserving energy increasing the life of the network in the process. Ad-hoc routing protocols are also unsuitable for sensor networks because they try to eliminate the high cost of table updates when there is highly mobility of nodes in the network. But unlike ad-hoc networks, sensor networks are not highly mobile. Routing protocols can be divided into *proactive* and *reactive* protocols. Proactive protocols attempt at maintaining consistent updated routing information between all the nodes by maintaining one or more routing tables. In reactive protocols, the routes are only created when they are needed. The routing can be either source-initiated or destination-initiated. Some of the routing protocols which have been proposed for sensor networks aimed at eliminating the above-menntioned problems are the following.

5.1 Negotiation based protocols [18]:

These protocols, called the **SPIN***(Sensor Protocols for Information via Negotiation)* protocols aim at disseminating information among all the sensor nodes by using information descriptors for negotiation prior to transmission of the data .These information descriptors are called *meta-data* and are used to eliminate the transmission of redundant data in the network. In SPIN , each sensor node also has its own resource manager that keeps track of the amount of energy that the particular node has. Prior to transmission or processing data , the nodes poll their resource manager if they have enough energy or not. This allows the nodes to cut back on activities when their resources are low increasing the life of the node in the process. The SPIN family of protocols use three messages for communication.

ADV : When a SPIN node has some new data, it sends an ADV message to its neighbors containing meta-data(data descriptor)

REQ : When a SPIN node wished to receive some data, it sends an REQ message.

DATA : These are actual data messages with a meta-data header.

The following protocols make up the SPIN family of protocols.

1. SPIN-PP : This protocol has been designed to perform optimally for point-to-point communication. In this sort of communication, two nodes can have exclusive communication with each other without any interference from the other nodes. In such a network, the cost of communication for one node to communicate with n nodes is n times more expensive than communicating with one node. This protocol is a simple 3-way handshake protocol in which energy is not considered to be a constraint. When a node has some new data, it advertises this new data using the ADV messages to its neighbors. When a neighboring node receives this advertisement, it checks the meta-data to see whether it already has the data item or not. In case it does not, it sends an REQ message back requesting for the data item. Upon receiving the REQ message, the originating node sends DATA messages containing the missing data to the requesting node. One major advantage of using this protocol is its simplicity and that each node requires to know only about its single-hop neighbors and does not require any other topology information.

2. SPIN-EC : In this protocol, the sensor nodes communicate using the same 3-way handshake protocol as in SPIN-PP but there is a energy-conservation heuristic added to it. A node will participate actively in the protocol only if it is above a certain *energy threshold* and believes it can complete all the other stages of the protocol. If a node receives an advertisement , it will not send out an REQ message if it does not have enough energy to transmit an REQ message and receive the corresponding DATA message.

3. SPIN-BC : This protocol was designed for broadcast networks in which the nodes use a single shared channel to communicate. When a node sends out a message, it is received by all the other nodes within a certain range of the sender. In this protocol, a node which has received an ADV message does not immediately respond with an REQ message. It has to wait for a certain time before sending out the REQ message. When a node other than the advertising node receives the REQ message, it cancels its own request so that there are no redundant requests for the same message. When the advertising node receives an REQ message, it sends the data message only once because it is a broadcast network even though it might have got multiple requests for the same message.

4. SPIN-RL : This protocol makes two changes to the above SPIN-BC protocol. Each node keeps track of all the advertisements it hears and the nodes it hears them from. If it does not receive any requested data within a certain period of time, it sends out the request again. Next, the nodes have a limit on the frequency with which they resend the data messages. After sending out a data message, a node will wait for a certain period of time before it responds to other requests for the same data message.

5.2 Directed Diffusion [17]:

This is another data dissemination protocol in which the data generated by the nodes is named by *attribute-value* pairs. This is a destination-initiated reactive routing technique in which routes are established when requested. A sensing task or *interest* is propagated throughout the network for named data by a node and data which matches this interest is then sent towards this node. One important feature of the data diffusion paradigm is that the propagation of data and its aggregation at intermediate nodes on the way to the request originating node are determined by the messages which are exchanged between neighboring nodes within some distance (localized interactions). Tasks are described by a list of attribute-value pairs that describe the task. This description is called an interest. The data which is sent as a response to such an interest is also named in a similar manner. The querying node is the *sink* node and it broadcasts its interest message periodically to all of its neighbors. All nodes have an *interest cache* in which each item corresponds to a different interest. These entries do not contain any information about the sink node though.

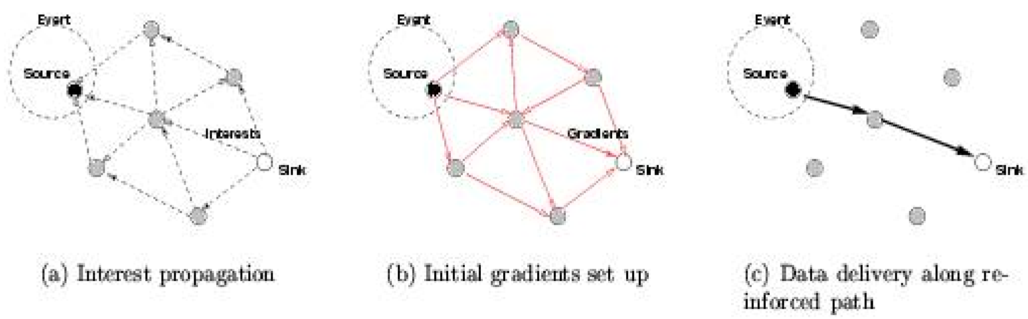

Figure 5. *Schematic diagram for directed diffusion*

An entry has several fields - *timestamp* field which contains the last received matching interest, the *gradient* fields contain the data rate specified by each neighbor, the *duration* field which contains the lifetime of the interest. When a node receives an interest, it checks its interest cache to check if it has entry. It creates one if there is no matching interest and a single gradient field is created towards the neighbor from which the interest is received. If the interest exists, the timestamp and the duration fields are updated in the entry. A gradient is removed from its interest entry when it expires. A gradient specifies both the data rate as well as the direction in which the events are to be sent. A node may send an interest it receives to some of its neighbors to whom it will appear as if this node itself is the originating node. Therefore, there is diffusion of interests throughout the network. A sensor node which detects an event searches its interest cache for a matching interest entry. If it finds one, it generates even samples at the highest data rate which it computes from the requested event rates of all its outgoing gradients. The event description is then sent to all its neighboring nodes for which it has gradients. Therefore the sink starts receiving low data rate events, when an event is observed, possibly along multiple paths. The sink then reinforces one particular neighbor to get the better quality events. This could result in more than one reinforced paths in which case, the better performing path is retained and the others are negatively reinforced by timing out all the high data rate gradients in the network while periodically reinforcing the chosen path.

5.3 Energy Aware Routing:

In this scheme, Rahul Shah et al.[22] proposed using sub-optimal paths occasionally to increase the lifetime of the network substantially. This protocol is also a destination initiated reactive protocol like Directed Diffusion with the difference being that instead of maintaining one optimal path, a set of good paths are maintained and chosen by means of a probability which depends on how low the energy consumption of each path is. Thus any single path does not get its energy depleted because different paths are chosen at different times. This ensures the graceful degradation of the network in low-energy networks because energy is burnt more equally in all nodes. This protocol has three phases:

Setup: The connection is initiated through localized flooding which is done to discover all the routes between the source and the destination and their costs thus building up the routing tables. The high-cost paths are discarded and the others are added to the forwarding table in which the neighboring nodes are chosen inversely proportional to their cost.

Data Communication: Data is sent from the source to the destination using one of the neighbors in the forwarding table with the probability of the node being chosen being the same as the probability of its being chosen in the forwarding table. In this way, the intermediate nodes forward the packet to a probabilistically chosen neighbor and this is continued till the packet reaches the destination node.

Route maintenance: Localized flooding is performed from the destination to the source now and then to keep the paths alive.

5.4 Rumor Routing [2]:

This routing protocol looks at routing queries to the nodes which have observed a particular event. It looks at creating paths leading to each event so that a query which is generated can be routed randomly till it finds the event path instead of flooding it across the network. The rumor routing algorithm uses a set of long-lived *agents* which create paths that are directed towards the events they encounter. Whenever an agent crosses path with a path leading to an event that it has not encountered, it adapts its behavior thus creating a path state which leads to both the events. When the agents come across shorter paths, they optimize the paths in the network by updating the routing tables to reflect the more efficient path. Each node maintains a list of its neighbors and an events table. When it encounters an event it adds it to its events table and might generate an agent in a probabilistic fashion. The agent also contains an events table like that of the nodes which it synchronizes with every node that it encounters. The agent has a lifetime of a certain number of hops after which it dies. Any node generating a query will transmit the query if it has a route to the event else it will transmit it in a random direction. If the node gets to know that the query did not reach the destination then it will flood the network. The lesser the number of queries which flood, the lesser the energy consumed.

5.5 Multipath Routing:

The resilience of a protocol is measured by the likelihood that an alternate path is available between a source and a sink when the optimal path fails. The solution proposed by Deepak Ganesan et al.[13] looks at increasing the resilience of a sensor network while ,keeping the maintenance overhead required for doing this, low. Maintenance overhead is the energy required to maintain the alternate paths by sending periodic messages. They propose a braided multipath scheme in which the paths are only partially disjointed. While disjointed multipaths are very resilient, they are highly energy inefficient because they are usually much longer and require much more energy to maintain the paths. *Braided* multipaths are not completely disjoint from the primary path. Each node tries to route around its immediate neighbor on the primary path towards the source to come up with an alternate path in case one of the nodes fails. Usually the energy costs of the alternate paths are comparable with the primary path because they tend to be very close. When there are isolated failures of nodes, braided multipaths perform display better resilience than the disjoint multipaths.

5.6 Media Access Control in Sensor Networks:

Media Access Control in sensor networks is very different than in the traditional networks because of its constraints on computational ability, storage and energy resources. Therefore media access control should be energy efficient and should also allocate bandwidth fairly to the infrastructure of all nodes in the network. In sensor networks, the primary objective is to sample the residing environment for information and send it to a higher processing infrastructure (base station) after processing it. The data traffic may be low for lengthy periods with intense traffic in between for short periods of time [29]. Most of the time, the traffic is multihop and heading towards some larger processing infrastructure. At each of the nodes, there is traffic originating out of the node and traffic being routed through the node because most nodes are both data sources and routers. There are several limitations on sensor nodes too. They have little or no dedicated carrier sensing or collision detection and they have no specific protocol stacks, which could specify the design of their media access protocol.

5.6.1 Fairness:

Alec Woo et.al [29] have suggested a transmission control scheme for media access in sensor network which looks at the following challenges in multihop sensor networks

- The originating traffic from a node has to compete with the traffic being routed through that node.
- An undetected node might exist in the network which might result in unexpected contention for bandwidth with route-thru traffic.
- The probability of corruption and contention at every hop is higher for the nodes which reside farther away from the higher processing infrastructure.

- Energy is invested in every packet when it is routed through every node. Therefore, the longer a packet has been routed, the more expensive it is to drop that packet.

Carrier sensing in sensor networks can be done by listening to the network which is expensive. Therefore the listening period should be short to conserve energy. The traffic also tends to be highly synchronized because nearby nodes tend to send messages to report the same event. Since there is no collision detection, the nodes will tend to corrupt each other's messages when they send them at the same time. This could happen every time they detect a common event. To reduce contentions, a *backoff* mechanism could be used. A node could restrain itself from transmitting for a certain period of time and hopefully the channel becomes clear after the backoff period. This will help in desychronising the traffic too. Contention protocols in traditional networks widely use the *Request to Send*(RTS), *Clear to Send*(CTS) and *acknowledgements*(ACK) to reduce contentions. A RTS-CTS-DATA-ACK handshake is extremely costly though when used in sensor networks because every message transmitted uses up the low energy resources of the nodes. Therefore, the number of control packets used should be kept as low as possible. Thus, only the RTS and CTS messages are used in the control scheme. If the CTS is not received by a node after sending the RTS for a long time, the node will backoff for a binary exponentially increasing time period and then transmit again. If it receives a CTS, which is not meant for it or receives a CTS before its own transmission, it will backoff to avoid collisions. Fairness in allocation between the originating traffic and route-thru traffic should be achieved. The media access controls the originating traffic when the route-thru traffic is high and when the originating traffic is high, it applies a backpressure to control the route-thru traffic deep down in the network from where it originated. A *linear increase* and *multiplicative decrease* approach is used for transmission control. The transmission rate control is probabilistic and it is linearly increased by a constant and it is decreased by multiplying it with, a, where $0<a<1$. Since dropping traffic which is being routed through is wastage of the network's energy resources, more preference is given to it by making its dropping penalty 50% lesser than for originating traffic.

The advantage of this scheme is that the amount of computation required for this is within the sensor nodes' computational capability and achieves good energy efficiency when the traffic is low while maintaining the fairness among the nodes.

5.6.2 S-MAC:

The solutions proposed by Wei Ye et al. [30] looks at the major sources of energy wastage while achieving good scalability and collision avoidance capability. The major sources of energy wastage are

- Collisions
- overhearing
- control packet overhead
- idle listening

Unlike Alec Woo et al. [29], they accept reduction in fairness because reduction in per-hop fairness may not necessarily mean reduction in end-to-end fairness. Unlike in traditional networks where all nodes require equal opportunity to transmit, sensor nodes all try to achieve a single common task. S-MAC introduces uses three techniques to reduce energy consumption. Firstly the nodes go to sleep periodically so that they do not waste energy by listening to an empty channel or when a neighboring node is transmitting to another node. This helps in avoiding the overhearing problem too. Secondly, nearby nodes form virtual clusters to synchronize their wake-up and sleep periods to keep the control packet overhead of the network low. Finally, message passing is used to reduce the contention latency and control overhead. S-MAC consists of three components

Periodic Listen and Sleep: Neighboring nodes are synchronized to go to sleep together so as to avoid a heavy control overhead. They listen together and sleep together. For this the nodes exchange schedules with their immediate neighbors. The nodes use RTS and CTS to talk to each other and contend for the medium if they want to communicate with the same node. Synchronized nodes form a virtual cluster but there is no real clustering and no inter-cluster communication problem. Synchronization is maintained by using SYNC packets which contain the sender's address and its next sleep time.

Collision and Overhearing Avoidance: S-MAC adopts a contention-based scheme to avoid collisions. A duration field is introduced in each transmitted packet which indicates how much longer the transmission will last. When a node receives a packet, it will not transmit any packets for at least the time that is specified in the duration field. This is recorded in a variable in the node called the *Network Allocation Vector* (**NAV**) which is reset every time the node received a packet whose duration field is larger than the current value. When the NAV is zero, the node can start transmitting packets. Overhearing is avoided by letting the nodes, which get RTS and CTS packets which are not meant for them, go to sleep. All immediate neighbors also go to sleep till the current transmission is completed after a sender or receiver receives the RTS or CTS packet.

Message Passing: Long messages are fragmented into smaller messages and transmitted in a burst. This is to avoid the high overhead and delay encountered for retransmitting when a long message is lost. ACK messages are used to indicate if a fragment is lost at any time so that the sender can resend the fragment again. The ACK messages also have the duration field to reduce overhearing and collisions. There is no contention to achieve fairness for each lost fragment. It is allowed to retransmit the current fragment but there is a limit on the number of retransmissions the node is allowed without any contention.

6. Sensor Networking Systems:

This section looks at some systems based on sensor-networking concepts, which have been developed by various groups. Most of the systems deal with location of persons/objects.

The **Active Badge Location System** [26, 27] is a system for locating personnel/objects inside a building. Each person/object, which has to be located, is tagged with an Active Badge which emits a unique infrared code every 10th of a second. These signals are picked up by networked sensors around the building. On the basis of the information provided by the sensor the location of the tag and hence the person/object can be determined. This system is actually in commercial use now.

Pin-point 3D-iD local positioning system described in [28] also deals with a similar problem of locating an item in 3-D space inside a location fixed by boundaries. The system consists of 3D-iD readers and tags. The readers emit codes that are received by the tags and transponded back to the reader after changing the signal's frequency. Based on the round trip time of flight, the distance of the tag from the antenna is calculated. It has an accuracy of 1-3metres.

[27] presents a sensor system which also allows the location of indoor people or equipment to be calculated but more accurately i.e. within 15cm of their actual location. A small wireless transmitter is attached to every object to be located. A matrix of receiver elements equipped with ultrasonic detectors are mounted on the room of the ceiling. The position of the transmitter is calculated using multilateration techniques.

Scalable Object-tracking through Unattended Techniques (SCOUT) [19] is a hierarchical, self-configuring approach for object location. It describes two schemes. SCOUT-AGG is a technique based

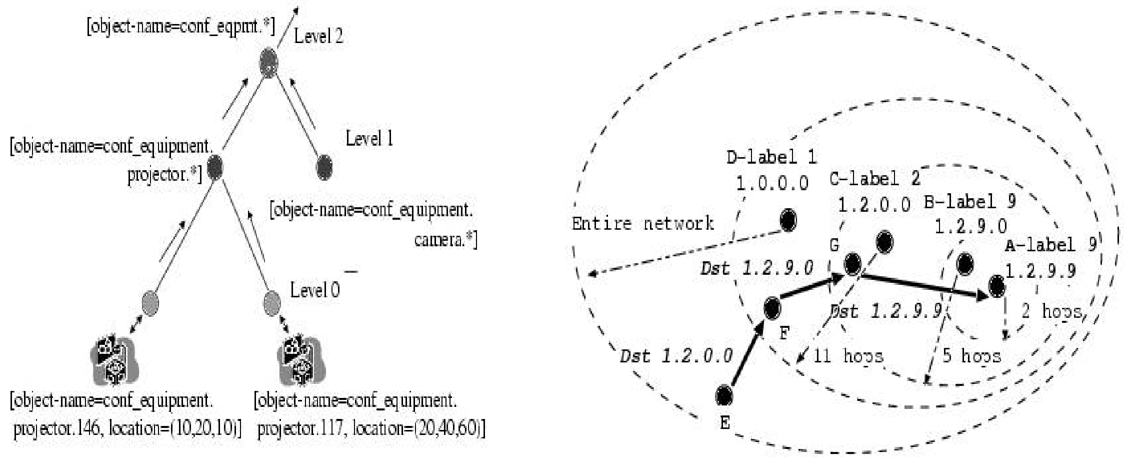

Figure 6. *The SCOUT_AGG and SCOUT_MAP Protocols*

on aggregation and naming. The objects are organized into a hierarchy. Each object has an object name based on the hierarchy. For example, conf_equipment.projector.146 defines a unique projector with id 146. On receiving a query, a sensor checks if the object is monitored locally, else it attaches its id to the query and forwards the query to its parent. The process is repeated till the sensor with the requested information is reached. Figure 6 [19] shows the aggregation in SCOUT-AGG.

SCOUT-MAP again organizes the sensors into a hierarchy. Also each sensor has a radius which determines the number of hops that an advertisement from that node will be allowed to propagate. Mapping of objects to locator sensor addresses is carried based on an algorithm. Figure 6 [19] shows query direction in scout-map.

Priyantha et al. discuss the Cricket Location-Support System [20] which helps devices learn where they are and lets them decide whom to advertise this information to. It does not rely on centralized management and there is no explicit co-ordination among the beacons. Here beacons are attached to some unobstrusive location. The objects which need to be located have listeners attached to them. When an object is deployed into the network, the listener infers its current location from the set of beacons it hears. The Cricket system uses a combination of RF and ultrasound hardware. Though the system has the advantage that it is decentralized and hence easy to manage and configure, it has the drawback that there is no explicit co-ordination. Hence RF signals from various beacons might collide. Thus it is the responsibility of the listener to analyze the various RF and ultrasound samples and deduce the correct RF,US pairs.

7. Simulators for Sensor Networks:

For the sake of completeness, this section very briefly looks at some of the more prominent simulators for sensor networks available today [http://www.cs.virginia.edu/~sb2jb/project/files/simulator_pres.ppt].

1. NS-2 [24]: The mother of all network simulators has facilities for carrying out both wireless and wired simulations. It is written in C++ and oTCL. Since it is object-oriented, it is easier to add new modules. It provides for support for energy models. Some example applications are included as a part of the package. It has the advantage of extensive documentation.

2. GloMoSim [1]: GLobal MObile Information systems SIMulator is a scalable simulation environment for wireless and wired network systems. It is written both in C and Parsec. It is capable of parallel discrete-event simulation. GloMoSim currently supports protocols for a purely wireless network. A basic level of Parsec knowledge and thorough C knowledge is sufficient to carry out simulations.

3. SensorSim [http://nesl.ee.ucla.edu/projects/sensorsim/]: is a simulation framework for sensor networks. It is an extension to the NS simulator. It provides the following: Sensing channel and sensor models, Battery models, Lightweight protocol stacks for wireless micro sensors, Scenario generation and Hybrid simulation. It is geared very specifically towards sensor networks and is still in the pre-release stage. It does not have proper documentation.

8. Conclusion:

Sensor Networks hold a lot of promise in applications where gathering sensing information in remote locations is required. It is an evolving field, which offers scope for a lot of research. Their energy-constrained nature necessitates us to look at more energy efficient design and operation. We have done a survey on the various issues in sensor networks like energy efficiency, routing and localization and the various schemes proposed for these issues and have given brief descriptions of these schemes. Further work is necessary in the areas of media access control, security and privacy.

.

9. References:

[1] L. Bajaj, M. Takai, R. Ahuja, K. Tang, R. Bagrodia, and M. Gerla, "GloMoSim: A Scalable Network Simulation Environment," UCLA Computer Science Department Technical Report 990027, May 1999.

[2] D. Braginsky and D. Estrin, "Rumor Routing Algorithm For Sensor Networks," Under submission to International Conference on Distributed Computing Systems (ICDCS-22), November 2001.

[3] M. Brain and T. Harris, "How GPS receivers work," http://www.howstuffworks.com/gps1.htm.

[4] N. Bulusu, J. Heidemann and D. Estrin, “GPS-less Low Cost Outdoor Localization For Very Small Devices,” *IEEE Personal Communications*, Special Issue on "Smart Spaces and Environments", Vol. 7, No. 5, pp. 28-34, October 2000.

[5] N. Bulusu, J. Heidemann and D. Estrin, “Adaptive Beacon Placement,” *Proceedings of the Twenty First International Conference on Distributed Computing Systems (ICDCS-21)*, Phoenix, Arizona, April, 2001.

[6] N. Bulusu, J. Heidemann, V. Bychkovskiy and D. Estrin, “Density-adaptive beacon placement algorithms for localization in ad hoc wireless networks,” *UCLA Computer Science Department Technical Report UCLA-CS-TR-010013,* July 2001.

[7] 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, Lake District,UK, July 2001

[8] N. Bulusu, V. Bychkovskiy, D. Estrin and J. Heidemann, “Scalable, Ad Hoc Deployable, RF-Based Localization”, *In Proceedings of the Grace Hopper Celebration of Women in Computing Conference 2002*, Vancouver, British Columbia, Canada, October 2002.

[9] D. E. Culler, J. Hill, P. Buonadonna, R. Szewczyk, and A. Woo, “A Network-Centric Approach to Embedded Software for Tiny Devices,” *EMSOFT 2001:First International Workshop on Embedded Software*, Oct. 2001.

[10] L. Doherty, “Algorithms for Position and Data Recovery in Wireless Sensor Networks,” *EECS Masters Report*,UC Berkeley, May 2000.

[11] 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 '99)*, August 1999, Seattle, Washington.

[12] D. Estrin, L. Girod, G. Pottie, M. Srivastava, “Instrumenting the world with wireless sensor networks,” *In Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP 2001),* Salt Lake City, Utah, May 2001.

[13] D. Ganesan, R. Govindan, S. Shenker and D. Estrin, “Highly-Resilient, Energy-Efficient Multipath Routing in Wireless Sensor Networks,” *ACM Mobile Computing and Communications Review*, Vol. 5, No. 4, October 2001.

[14] W. R. Heinzelman, J. Kulik, and H. Balakrishnan, “Adaptive protocols for information dissemination in wireless sensor networks,” *In Proceedings of the fifth annual ACM/IEEE international conference on Mobile computing and networking,* August 1999.

[15] A. Howard, M. J. Mataric and G. S. Sukhatme, “An Incremental Self-Deployment Algorithm for Mobile Sensor Networks,” *Autonomous Robots, Special Issue on Intelligent Embedded Systems,* 2002.

[16] A. Howard, M. J. Mataric and G. S. Sukhatme, “Relaxation on a mesh: A formalism for generalized localization,” *2001 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 29 - November 3, Maui, pp. 1055-1060.

[17] C. Intanagonwiwat, 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)*, August 2000, Boston, Massachusetts.

[18] J. Kulik , W. Heinzelman , and H. Balakrishnan, “Negotiation-based protocols for disseminating information in wireless sensor networks,” *Wireless Networks,* March 2002.

[19] S. Kumar, C. Alaettinoglu and D. Estrin, “SCalable Object-tracking through Unattended Techniques (SCOUT),” *In Proceedings of the 8th International Conference on Network Protocols(ICNP)*, Osaka, Japan, November 2000.

[20] N. Priyantha, A. Chakraborty.A, H. Balakrishnan, “The Cricket Location-Support System,” *6th ACM International Conference on Mobile Computing and Networking,* 2000.
“SensorSim : A Simulation Framework for Sensor Networks,” http://nesl.ee.ucla.edu/projects/sensorsim/.

[21] V. Raghunathan, C. Schurgers, Park.S, and M.B. Srivastava, “Energy-aware wireless microsensor networks,” *IEEE Signal Processing Magazine*, Volume: 19 Issue: 2 , March 2002 Page(s): 40 –50.

[22] R. C. Shah and J. Rabaey, "Energy Aware Routing for Low Energy Ad Hoc Sensor Networks," *IEEE Wireless Communications and Networking Conference (WCNC)*, March 17-21, 2002, Orlando, FL.

[23] Srisathapornphat, C.Jaikaeo and C.Chien-Chung Shen, “Sensor Information Networking Architecture,” *International Workshops on Parallel Processing*, Pages 23-30, 2000.

[24] “The Network Simulator – NS2,” http://www.isi.edu/nsnam/ns/.

[25] “ TinyOS”, http://webs.cs.berkeley.edu/tos/.

[26] R. Want, A. Hopper, V. Falcao and J. Gibbons, “The Active Badge Location System,” *ACM Transactions on Office Information Systems (TOIS)*, Vol. 10. N0. 1, Jan 1992 Pages 91-102.

[27] A. Ward, A. Jones, A. Hopper, “A New Location Technique for the Active Office,” *IEEE Personal Communications*, Vol. 4, No. 5, October 1997, pp 42-47.

[28] J. Werb and C. Lanzl, “A positioning system for finding things indoors,” *IEEE Spectrum*, 35(9):71-78, September 1998.

[29] A. Woo , D. E. Culler, “A transmission control scheme for media access in sensor networks,” *Proceedings of the seventh annual international conference on Mobile computing and networking,* July 2001.

[30] W. Ye, J. Heidemann and D. Estrin, “An Energy-Efficient MAC Protocol for Wireless Sensor Networks,” *In Proceedings of the 21st International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002)*, New York, NY, USA, June, 2002.

[31] Y.J.Zhao, R. Govindan, and D. Estrin, “Residual energy scan for monitoring sensor networks ,” *Wireless Communications and Networking Conference, 2002 (WCNC2002)*, 2002 IEEE , Volume: 1 , Mar. 17-21, 2002 Page(s): 356 –362.