Survey on Sensor Networks *

Praveen Rentala, Ravi Musunuri, Shashidhar Gandham, Udit Saxena

Department of Computer Science
University of Texas at Dallas, Richardson, TX 75080.
Email: {prave,musunuri,gshashi,udits}@utdallas.edu

Abstract

Networked sensors that coordinate among themselves to perform a large task are expected to revolutionize information gathering in any type of terrain and conditions in future. A wireless sensor network consists of groups of sensors or nodes using wireless links to perform distributed sensing tasks. Wireless sensors are employed for specialized tasks like surveillance and security, environmental monitoring, transport, smart spaces, precision agriculture, manufacturing and inventory tracking and health care. Their main advantage is their ability to be deployed in almost any kind of terrain with a hostile environment where it might not be possible to use traditional wired networks. In this paper we present an overview of wireless sensor networks and issues envolved in employing them. Here we make an attempt to describe solutions in recently published literature for issues like link layer establishment, routing the sensed data from sensor nodes to base station, power managemant, location management, clock synchronization and determination of theoritical bound on lifetime of sensor networks.

Keywords: Sensor Networks,Wireless Sensor Networks, Distributed Sensor Networks, WINS.

1 Introduction

Recent advances in commercial IC fabrication technology have made it possible to integrate signal processing and sensing in one integrated circuit. These devices are popularly known as wireless integrated network sensors (WINS) [10] and include microelectromechanical systems (MEMS) technology components such as sensors, actuators, RF components and CMOS building blocks. WINS combines microsensor technology and low power computing and wireless networking in a compact system. Sensor nodes are randomly dispersed over the area of interest and are capable of RF communication, and contain signal processing engines to manage the communication protocols and for data processing before transmission. The individual nodes have a limited capability, but are capable of achieving a big task through coordinated effort in a network that typically consists of hundreds to thousands of nodes. Examples of such activities are beamforming for enhanced target detection, distribution of timing and position information and multi-hop communication. Low power operation is critical and each device is limited in the range that it can communicate because of background noise and signal attenuation with distance.

Potential advantages of having a sensor network as compared to a stand alone satellite or ground based radar are [10]:

*Survey Paper submitted as per the requirements of Mobile Computing (CS 6392) Course.

- Ensuring greater signal to noise ratio (SNR) by combining information from sources with different spatial perspectives. Deploying a single powerful sensor might not be able to address issues of line of sight and high SNR.
- Allowing greater fault tolerance through a high level of redundancy.
- Providing coverage of a very large area through the union of individual nodes coverage area.
- Sculpting the coverage for a given area to overcome holes or shadows.
- Improving sensing performance by including multiple sensor types.
- Overcoming environmental effects by having sensors at close proximity to the object of interest.
- Localizing discrete phenomenon through each sensors limited range and combining information with other nodes.
- Deployment in regions where infrastructure for replenishing energy is not available

A general operational scenario: To take an example let us take the case of seismic detection [9]. The earth generates broadband seismic noise, which becomes attenuated and distorted with distance. The higher frequencies will be less reliable and hence the closer the sensors are to the source, the higher the probability that a reliable identification can be made. Thus a distributed network of sensors will collect significantly different information than a system of a small number of highly sensitive nodes. Power constraints dictate that the volume of data is reduced or the data is transmitted at low RF, because this consumes less energy even if some additional processing at baseband is required. Sensors close to the event of interest having reasonable SNR can then use the beamforming protocol with timestamps to collect the data for aggregation and synthesis at one of the nodes or a subset of nodes. Data reduction can be done with local decision making. Diversity techniques can be deployed to counteract attenuation of signals with distance. With low cost nodes, it is possible to have a dense deployment, which enables multihopped communications. This is desirable to route around obstacles and allows magnitudes of reduction in power levels in the presence of fading. The topology may have access to a few mobile nodes that serve as gateways to the outside world.

The remaining part of this paper is organised as follows. In section 2, we describe how sensor networks are different from other kinds of wireless networks. In section 3 we list out, technical and operational challenges in deploying wireless sensor networks. Section 4 and 5 talk about sensor network architecture and physical layer principles respectively. In section 6, link layer issues are described and few solutions proposed in literature are elaborated. In section 7 we move on to flat routing protocols and hierarchical protocols. We then consider other important issues like power management and location management.

2 How are Sensor Networks different from other kinds of networks?

Sensor networks is a new family of wireless networks [7] and is significantly different from traditional networks like cellular networks and MANETs. In these traditional networks the tasks of organization, routing and mobility management is done to optimize QoS and high bandwidth efficiency. These networks are designed to provide good throughput/delay characteristics under high mobility conditions. Energy consumption is of secondary importance as the battery packs can be replaced as needed. However sensor networks consist of hundreds to thousands of nodes that are designed for unattended operation. The nodes are generally stationary after deployment except for a few mobile nodes. The traffic is of a statistical nature as compared to the multimedia rich data in MANETs and cellular networks. The data rate is expected to be very low to the order of 1-100kb/s. Unlike conventional networks the main goals are prolonging the life of the network and prevent connectivity degradation through aggressive energy management as the batteries cannot usually be replaced because of operations in hostile or remote environments. In sensor networks the flow of data is predominantly unidirectional from the sensor nodes to a sink.

3 Technical/Operational challenges in Sensor Networks

Most sensor networks encounter the following operational challenges [2].

- Ad hoc deployment, requiring that the system should be able to cope with the resultant distribution and form connections between the nodes.
- Dynamic environmental conditions requiring the system to be adaptive in nature to changing connectivity and node failure.
- Unattended operation requiring configuration to be done automatically and repeatedly.
- Unthetered for energy and communication, requiring maximum focus on energy efficiency.

This scenario imposes several primary architectural decisions and result in several system trade-offs: small size, rugged construction, energy efficient operation and low cost. The small size limits battery capacity requiring every operation to be done efficiently. The limited available energy imposes a limit on radio transmission range and suggests small multihop transmissions schemes. This also introduces closer pacing of nodes to reduce energy consumption. This reduces signal processing requirements for multitarget discrimination and suggests local processing of information to reduce the amount of information to be transmitted. To take an example for ground to ground communication [7] it takes 3 J of energy to transmit 1 Kb of data a distance of 100 m. A general purpose processor having a processing capability of 100 million instructions per second would execute 300 million instructions for the same amount of energy.

To address these technical challenges several strategies are going to be key building blocks for sensor networks [1]:

- *Cooperative Signal processing* - Collaborative signal processing among nodes can greatly enhance the energy efficiency. Once a common stimuli has been detected both coherent signal processing on a small cluster by a centralized entity and non coherent processing with less stringent time coordination requirements can be exploited.

- *Exploiting redundancy* - This has application when the cost of deploying the initial set of sensors is much less as compared to the cost of replacing defective or failed nodes or renewing node resources. Thus redundancy can be exploited to extend system lifetime. Another application is when sensors cannot be positioned carefully, redundancy can be exploited to extend coverage by using a subset of nodes which are positioned favorably.

- *Adaptive signal processing* - This strategy can be exploited to give a balance between energy, accuracy and rapidity of results. The timeliness and fidelity of the signal processing can be adapted keeping in mind the energy resources and latency requirements.

- *Hierarchical architecture* - Higher energy or capacity nodes can be used to offload the information while small capacity nodes can be used to give the required proximity to the object of interest. In systems with homogeneous elements, creation of clusters and assigning special functions to cluster heads can greatly contribute to overall system scalability, lifetime and efficiency.

4 Sensor Node Framework/Architecture

In sensor networks, the architecture of a node is completely dependent on the purpose of the deployment. But, we can generalize the architecture as shown in the following Figure 1. Each node consists of a sensor, processor, radio, battery, OS and protocols, and memory.

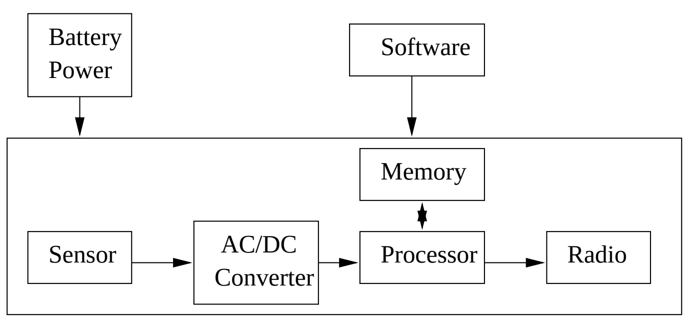

Figure 1: Sensor Node Architecture

4.1 General Layered Architecture

Figure 2 [20], shows the general layered architecture of a sensor node. The basic components in this architecture are,

- The sensor node processing subsystem running on sensor node main CPU
- The sensor subsystem and
- The communication subsystem.

This proposed architecture has a layered structure as shown. Here the network functionality is divided between main CPU and radio board. The main idea behind this architecture is to decrease the functionality on sensor CPU by transferring some of the functionalities to the radio board. The radio boards process the information in the form of Micro Controller Units (MCU), which are used for the physical and MAC layer implementation. Thus a part of the network layer functionality can be transferred to the radio board.

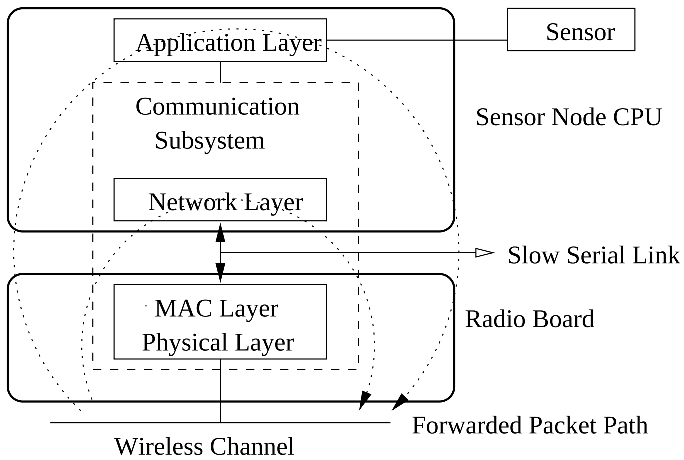

Figure 2: Layered Architecture 1

The radio board is equipped with a MCU as the main processing component along with some amount of configurable logic. This can be visualized in Figure 3 [20].

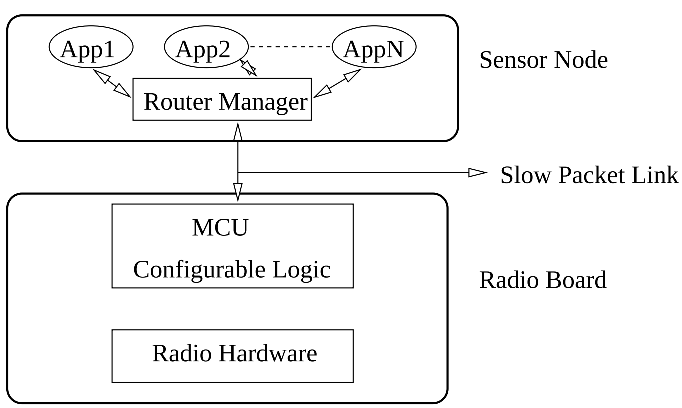

Figure 3: Layered Architecture 2

4.2 Sensor Information Networking Architecture (SINA)

This model assumes the sensor nodes as a massive collection of distributed objects in which SINA acts as a middleware. It facilitates adaptive organization of sensor information by allowing applications to issue queries into, collect replies from, and monitor changes within the sensor networks. With SINA organization and information provision is done at the lower layers, where as queries and monitoring tasks are done at higher layers. This is shown in Figure 4 [21].

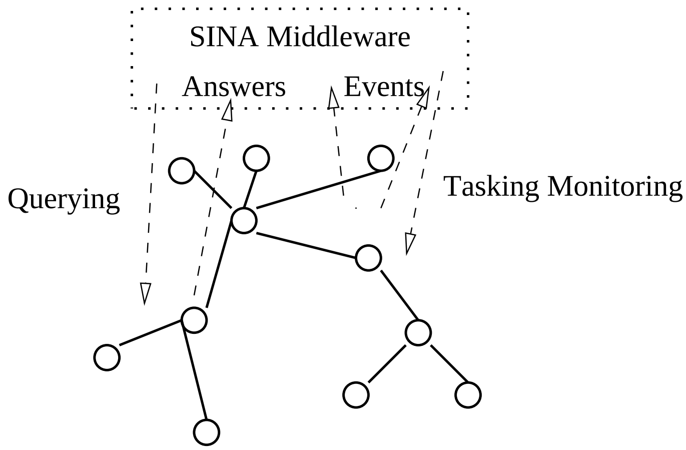

Figure 4: SINA Architecture

In this architecture sensor nodes autonomously form groups called as clusters. The clustering will be based on power level and proximity. This clustering process is applied recursively to form a hierarchy of clusters. Hence, it is known as *Hierarchical Clustering*. This is expected to increase the life of a sensor node by decreasing the power required for information exchange. This hierarchical structure is shown in Figure 5.

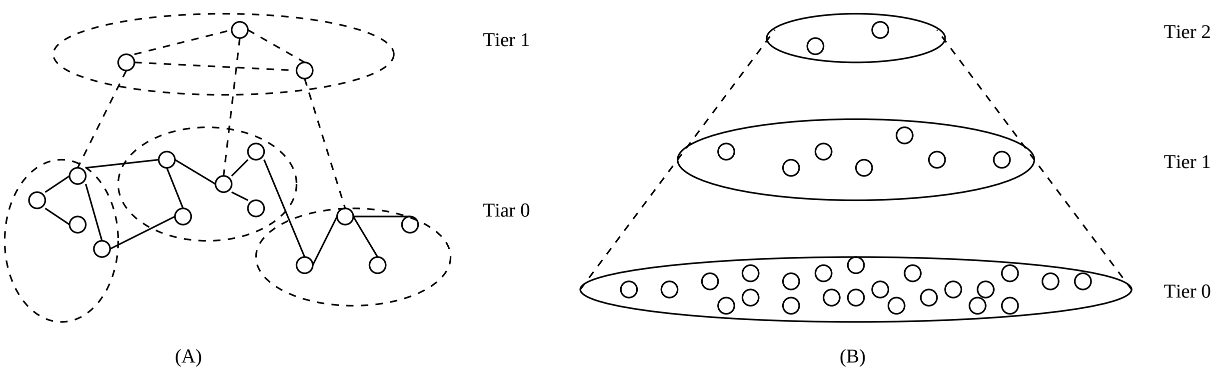

Figure 5: Hierarchical Clustering

Moreover, in this architecture information from the network is accessed via attribute-based naming instead of explicit addressing, which is very difficult to manage i.e., instead of asking for information at a particular node, the network is asked for the list of nodes which have the specified information. This is known as *Attribute-based Naming*. By using this type of queries, we can increase the overall network efficiency

By using hierarchical clustering and attribute-based naming, it is claimed that many complex queries can be answered with little overhead on the network.

5 Physical Layer Issues

Before going into the MAC layer and routing issues in sensor networks we will have to first understand the fundamental limits of sensing and detection theory [6].

- *Propagation laws of sensing* - It has been shown that if a system has to track objects reliably it has to be distributed. This makes sense because almost every scenario of interest in sensor networks involves heavily obstructed lines of sight.
- *Detection and estimation theory fundamentals* - A sensor is given a set of observables $\{X_i\}$ to determine which of several hypotheses $\{h_i\}$ are true. The observables may for example consist of the output of a seismic sensor, which may or may not include the background noise and the hypothesis may include the intersection of several discrete events. The decision concerning target presence, absence and type is based on the parameters of these hypotheses that are usually a small subset of the observables set to distinguish between different hypotheses. The set of parameters is known as the feature set $\{f_k\}$. To have a good estimate of any particular feature set we either need high SNR or a long set of independent observations. The way to choose among these hypothesis is to construct a decision space based on the conditional probability $p(h_i|\{f_k\}) > p(h_j|\{f_k\})$ for all j not equal to i.

  The complexity of the decision increases with the number of hypothesis that are to considered. To reliably distinguish among several hypothesis we need a large feature set. To build a minimum size feature set we have to see whether there is any marginal improvement in the error on including a certain feature. On these fundamentals rests the decision to deploy a dense sensor network. At short range the environment can be considered essentially homogeneous within the detection range and this reduces the size of the environmental features. Furthermore SNR is high at short distance and this gives the flexibility of using a variety of sensing nodes with small feature sets to distinguish targets.

6 Link Layer Protocols

The existing solutions to any channel access methods in ad hoc networks can be divided into two categories: contention based and organized methods. Contention based schemes are not suitable for sensor networks because of the task of continually sensing the access channel and wastage of resources whenever a collision occurs. The organized methods of channel access attempt to determine connectivity between nodes first and then handle the assignment of channel slots in a hierarchical manner by forming clusters and assigning cluster heads. This solution should be distributed because network wide synchronization for calculation of a schedule would be an energy intensive procedure in sensor networks. Another reason for distributed algorithms is that they scale well with an increase in network size and that they are robust to network partitions and node failures. Clustering also allows more energy efficient utilization for many distributed sensor coordination tasks like data aggregation and locating objects because it is better that only the cluster heads engage in such tasks rather than all the nodes. Mobile nodes are usually introduced in a sensor network to serve as gateways to the outside world. Thus solutions can further be subdivided into two kinds. One category caters to communication between the mobile nodes and the stationary nodes on the ground and the other category caters to communication between stationary nodes.

Some of the well known solutions proposed are:

6.1 Self-Organizing Medium access Control for Sensor Networks (SMACS)

SMACS is a distributed protocol [7], which enables nodes to discover their neighbors and build a network for communication without any master nodes. It builds a flat topology, i.e. there are no clusters or cluster heads. It is assumed that the bandwidth available is not a constraint and hence nodes can randomly choose any frequency to operate on. Nodes communicate intermittently and hence they can power themselves off when they have no data to send. The medium access channel consists of TDMA schedule that the authors refer to as a *super frame*. The structure of this super frame could change from time to time. The TDMA schedule consists of two separate regions. The first region is called the *bootup* period, when the nodes randomly search on a fixed frequency band for new nodes to include in the network or rebuild severed links. The other region is reserved for communication tasks with neighboring nodes.

After deployment each node wakes up according to a random distribution and performs neighbor discovery in the bootup period. As soon as a new link is discovered, the first time slot during which both nodes are free is assigned a channel and is added permanently to their schedule. To reduce collisions in channel assignment with other links, each link is required to operate at a different frequency or have a spreading code. As time goes on each node grows its neighbor list by attaching new nodes and eventually all the nodes are connected to each other. The ability to have nonsynchronous scheduled communication enables nodes to form links on the fly. The transmission/reception pattern repeats periodically every T_{frame}, which is fixed for all nodes and is a characteristic of super frame. After a link is formed a node knows when to turn on its transceiver ahead of time for communication. This leads to substantial savings in energy.

6.2 Eavesdrop and Register (EAR)

This solution [7] has been designed for communication between mobile nodes and stationary nodes on the ground. It is desirable that the connection be setup with as few message exchanges as possible. Hence to conserve energy the mobile node assumes full responsibility for the connection setup. The mobile node keeps a registry of all the sensors in its neighborhood and makes handoff decisions whenever the SNR drops below a threshold value. The EAR algorithm is designed to be transparent to the protocol followed by the stationary nodes. The first slot following the bootup period is reserved for the mobile nodes thus giving them higher priority. The EAR algorithm uses the invitation messages broadcast during the bootup period as a trigger. The mobile node simply eavesdrops on to these messages and forms a registry of all the stationary nodes within hearing range. The algorithm makes use of four primary messages:

- Broadcast Invite (BI) - The stationary node invites other nodes to join.
- Mobile Invite (MI) - The mobile node responds to the BI message.

- Mobile Response (MR) - The stationary node accepts the MI request.
- Mobile Disconnect (MD) - The mobile node informs the stationary node of a disconnect operation.

The mobile node continues to add every node sending a BI broadcast into its registry. On getting an MI, a stationary node determines whether it has an available TDMA slot for the communication. An acceptance (MR) is sent if the mobile nodes request can be accommodated. As the SNR degrades or improves, a MD may be sent or a new connection requested. To prevent communication overhead all acknowledgements are absent and timeouts are used to prevent indefinite waiting when no response to an MI is received. Since the mobile node communicates with a few stationary nodes at a time, its TDMA frame length is a fraction of the super frame used by the stationary nodes.

7 Routing

Routing protocols can in general be divided into flat routing and hierarchical routing protocols. These can be further subdivided into multihop and cooperative network routing. Multihop routing involves routing between a source and a sink which is usually a mobile node. Ad Hoc multihop routing protocols like Ad Hoc Demand Distance Vector (AODV) and Temporally Ordered Routing Algorithm (TORA) are not suitable for sensor networks because of the low mobility of the system. These protocols eliminate the high cost of table updates involved in high mobility scenarios by going with a demand driven system. So it is preferable to go with a table based system for sensor networks. For multihop routing, robustness is needed. Hence a lot of energy might have to be expended for route setup and maintenance. Cooperative network routing as the name suggests involves cooperation between sensors that detect a common target and sending it for aggregation to a central node for further processing. For cooperative routing, where data traffic is light, reducing route setup cost assumes importance.

7.1 Flat Routing Protocols

The first category of protocols are the Multihop routing protocols.

7.1.1 Sequential Assignment Routing (SAR)

SAR [7] takes into consideration the energy resource and QoS on each path, and the priority level on each packet for making routing decisions. A multi path approach is used to avoid overheads of route recomputation on failure, localized path restoration schemes are used. To create multiple paths from each node to the sink, multiple trees each rooted from a one-hop to the sink are built. Each tree is grown outward from the sink by successively branching and going for higher hop paths while avoiding nodes with lower QoS and energy reserves. At the end of this process each node will belong to multiple paths and each sensor can control which one-hop neighbor of the sink will be used for relaying the message. For each node two parameters are associated with each path. An additive QoS metric and a measure of the energy resources by estimating the number of packets that can be routed without energy being depleted if given exclusive use of that path. SAR then calculates a weighted QoS metric as the product of the additive QoS metric and a weight coefficient associated with the priority level of the packet.

The SAR algorithm attempts to minimize the average weighted QoS metric throughout the lifetime of the network. A periodic recomputation of paths is triggered by the sink to account for any changes in topology. Failure recovery is done through a handshaking procedure between neighbors and local path restoration is used.

7.1.2 Directed Diffusion

A different kind of solution was proposed by Estrin et.al. [3] which they call as directed diffusion. In this solution attribute based naming is used by the sensor nodes. Each sensor names data that it generates using one or more attributes. A sink may query for data by disseminating interests. Intermediate nodes propagate these interests. Interests establish gradients of data towards the sink that expressed that interest. For example a seismic sensor may generate a data: (type = seismic, id = 12, location = NE, timestamp = 01.01.01, footprint = vehicle/wheeled/over 40 tons). A sink may send an interest of the form: (type = seismic, location = NE). The intermediate nodes may send an interest for data for vehicle data in the NE quadrant towards the approximate direction. The strength of the gradient may be different towards different neighbors resulting in different amounts of information flow.

7.1.3 Minimum Cost Forwarding Algorithm for Large Sensor Networks

The minimum cost forwarding approach proposed by Ye et.al. [24] exploits the fact that data flow in sensor networks is in a single direction and is always towards the fixed base station. Their method neither requires sensor nodes to have unique identity number nor maintain routing tables to forward the messages. Each node maintains the least cost estimate from itself to the base station. Each message to be forwarded is broadcasted by the node. On receiving a message the node checks if it is on the least cost path between the source sensor node and the base station. If so it would forward the message by broadcasting.

In principle, the concept behind minimum cost forwarding is similar to the natural gravity field that drives waterfalls from top of mountain to the ground. At each point water flows from a high post to a low post along the shortest path. For this algorithm to work each node needs to have the least cost estimate from itself to the base station. The base station broad casts an advertisement message with the cost set to zero. Every node initially has the estimate set to infinity. On receiving the advertisement message if the estimate in the message plus the cost of link on which it is received, is less than the current estimate, current estimate and the estimate in the advertisement message is update,d as the estimate in the message plus the cost of the link on which it is received. If the received advertisement message is updated with a new cost estimate it is forwarded else its purged.

As a result of forwarding advertisement message immediately after updating, it is noticed in the simulations that some nodes will get multiple updates and do multiple forwards as lesser cost estimates flow in. Its was also reported by the authors that nodes far away from the base station get more updates than those close to the base station. To avoid this instability during the setup phase of the algorithm a back off algorithm is proposed. According to this back off algorithm on updating the current cost estimate, the advertisement message is not forwarded until $A * C_{node}$ units of time from the time of update. A is a constant determined through simulations and C_{node} is the cost of link on which the advertisement message was received.

7.1.4 Sensor Protocols for Information via Negotiation (SPIN)

These solutions proposed by Kaulik et.al. [5] are designed to disseminate individual sensor information to all the sensor nodes, assuming all of them are potential sinks. The solutions work to overcome information implosion and overlap by using negotiation and information descriptors (meta-data). Classic flooding suffers from the problem of implosion in that the information is sent to all nodes regardless of whether they have already seen that information or not. Another problem is that of overlap of information where two pieces of information might have some components in common, so it might be sufficient to just forward the information after removing the common part. SPIN uses three kinds of messages to communicate.

- ADV - When a node has data to send it advertises this using this message.
- REQ - A node sends this message when it wishes to receive some data.
- DATA - Data message containing the data with a meta-data header.

The details are as follows:

1. SPIN-PP: This protocol is designed for point to point communication, assuming that two nodes can communicate with each other without interfering with other nodes communication. This protocol also assumes that energy is not a constraint and packets are never lost. This protocol works on a hop-by-hop basis. A node which has information to send advertises this by sending an ADV to its neighboring nodes. The nodes who are interested in receiving this information express their interest by sending an REQ. The originator of the ADV then sends the data to the nodes that sent a REQ. These nodes then send ADV messages to their neighbors and the process repeats itself.
2. SPIN-EC: This protocol adds an energy heuristic to the previous protocol. A node participates in the process only if it can complete all the stages in the protocol without going below a low energy threshold.
3. SPIN-BC: This protocol was defined for broadcast channels. The advantage is that all nodes within hearing range can hear a broadcast while the disadvantage is that the nodes have to desist from transmitting if the channel is already in use. Another difference from the previous protocols is that nodes do not immediately send out REQ messages on hearing an ADV. Each node sets a random timer and on expiry of that timer sends out the REQ message. The other nodes whose timer have not yet expired cancel it on hearing the request thus preventing redundant copies of the request being sent again.
4. SPIN-RL: This protocol was designed for lossy broadcast channels by incorporating two adjustments. First each node keeps track of the advertisements it receives and re-requests data if a response from the requested node is not received within a specified time interval. Second nodes limit the frequency with which they will resend data. Every node waits for a predetermined time period before servicing requests for the same piece of data again.

Multihop flat routing can also be subdivided according to the signal processing techniques. There are two types of cooperative signal processing techniques: noncoherent and coherent. For noncoherent processing, raw data is preprocessed at the node itself before forwarding it to the central node (CN) for further processing. For coherent processing, the data is forwarded after minimum processing to the central node. The processing at the node involves operations like time stamping. Thus for energy efficiency algorithmic techniques assume importance for coherent processing since the data traffic is low, while path optimality is important for coherent processing.

7.1.5 Noncoherent Processing

There are three phases involved [7]. Phase1 involves target detection, data collection and preprocessing. Phase 2 involves membership declaration and Phase 3 involves central node election. After preprocessing if a node finds that the information might be of interest, it declares its intention to participate in a cooperative fashion (Phase 2). In Phase 3 of the process, a CN is selected for more sophisticated information processing by taking into account the energy reserves and computational capability of nodes. The CN election algorithm is made up of two components: Single Winner Election (SWE) algorithm and Spanning Tree (ST) algorithm. The first component involves the signaling involved in the election of a single candidate. The second component creates a spanning tree rooted at that winner. An *Elect* message is broadcasted by each node willing to be a CN along with the set of parameters that serve as the election criteria. The nodes that receive the first batch of messages then compare the criteria with themselves, respond with a second set of messages with the result of that comparison and store the winner in their registry. The routing information is piggybacked along with the Elect message thus allowing the calculation of a minimum spanning tree rooted at the winner simultaneously. Thus the winning candidates information diffuses through the network and the spanning tree gradually increases its coverage and finally covers the whole network.

7.1.6 Coherent Processing

In this algorithm an Multiple Winner Algorithm (MWE) [7], a simple extension of the SWE process is used. Since the energy cost of sending large amounts of data to the CN can be very high this process is used to limit the number of source nodes sending that data. Instead of keeping a record of the best candidate, each node will now keep up to n of them. An SWE procedure can then be used to find the node that yields minimum energy consumption as the CN from these source nodes (SN) elected in the previous step.

7.2 Hierarchical Routing Protocols

7.2.1 Low Energy Adaptive Clustering Hierarchy (LEACH)

Chandrakasan et.al. [17] proposed LEACH as an energy efficient communication protocol for wireless sensor networks. Authors of LEACH claim that this protocol will extend the life of wireless sensor networks by a factor of 8, when compared to protocols based on multi hop routing and static clustering. LEACH is a cluster based routing algorithm in which self-elected cluster heads collect data from all the sensor nodes in their cluster, aggregate the collected data by data fusion methods and transmit the data directly to the base station. These self-elected cluster heads continue to be cluster heads for a period referred to as a round. At the beginning of each round, every node determines if it can be a cluster head during the current round. If it decides to be a cluster head for the current round it announces its decision to its neighbors.

Other nodes which choose not to be cluster heads opt to join one of the cluster heads on listening to these announcements, based on predetermined parameters like signal to noise ratio.

LEACH is proposed for routing data in wireless sensor networks, which have a fixed base station to which the recorded data needs to be routed. All the sensor nodes are considered to be static, homogeneous and energy constrained. The sensor nodes are expected to sense the environment continuously and thus have data to be sent at a fixed rate. This assumption makes it unsuitable for sensor networks where a moving source needs to be monitored. Furthermore, radio channels are assumed to be symmetric. The term symmetric here means that energy required to transmit a particular message between two nodes is same in either direction. A first order radio model is assumed to describe the transmission characteristics of the sensor nodes. In this model energy required to transmit a signal has a fixed part and a variable part. The variable part is directly proportional to square of the distance. Some constant energy is required to receive a signal by any receiving antenna. Based on these assumptions its clear that having too many intermediate nodes to route the data might consume more energy, on a global perspective, when compared to direct transmission to the base station. This argument supports the decision to transmit the aggregated data directly from cluster head to base station.

The key features of LEACH are localized coordination for cluster set-up and operation, randomized rotation of cluster heads and local fusion of data to reduce global communication costs. LEACH is organized into rounds where each round starts with a set-up phase followed by a longer steady-state data transfer phase. Here we describe various sub-phases involved in both these phases.

1. *ADVERTISEMENT PHASE*: A predetermined fraction of nodes, say p, elect themselves as cluster heads. The optimum value of p can be found from plot between normalized energy dissipation and percent of nodes acting as cluster heads. For detail description of this procedure we refer the reader to the paper on LEACH. The decision to be a cluster head is made by choosing a random number between 0 and 1. If the generated number is less than a threshold $T(n)$ then the node will be a cluster head for current round. The threshold $T(n)$ is given by the expression $p/[1 - p(rmod(1/p))]$. This would ensure that every node would be a cluster head once in $1/p$ rounds. Once the decision is made, cluster heads advertise their id and this is done by employing CSMA MAC protocol.

2. *CLUSTER SET-UP PHASE*: On listening to advertisements in the previous phase, non cluster head nodes determine which cluster head to join by comparing signal to noise ratio from various cluster heads surrounding it. Each node informs the cluster head of the cluster which it decides to join by employing CSMA MAC protocol.

3. *SCHEDULE CREATION:* On receiving all the messages, cluster head creates a TDMA schedule and announces it to all the nodes in the cluster. In order to avoid interference between nodes in adjacent clusters, cluster head determines the CDMA code to be used by all the nodes in its cluster. This CDMA code to be used in the current round is transmitted along with the TDMA schedule.

4. *DATA TRANSMISSION*: Once the schedule is known each node will transmit the data during the time slot allocated to it. When the cluster head receives data from all the nodes in its cluster it will run some data fusion algorithms to aggregate the data. The resulting data is transmitted directly to the base station. After carefully going through the paper

on LEACH and analyzing we made some very important observation which could question the authors claim that their protocol increases the life time of the wireless sensor networks by a factor of 8. We list these observations below.

- In the simulations conducted the possibility of nodes being destroyed due to hostile environment in which they might be deployed is not considered. Its well known that sensor nodes are expected to be deployed in very hostile environment.
- Energy dissipation is assumed to be 50 nJ/bit, while for the best available technology its 115 nJ/bit. Though authors of LEACH mention this fact, they do not give any arguments to support such assumption.
- It is proposed in LEACH that certain predetermined fraction of nodes will be cluster heads in each round and a method to determine the optimum value of this fraction was proposed. But the issue of how to make sure that these cluster heads are uniformly distributed in the network with respect to their geographic location is not addressed. As a result there is a possibility that cluster heads could be concentrated in one part of the network and major part of sensor nodes will be left with out any cluster head in their vicinity.
- In the simulation study done by authors of LEACH they considered a 100-node network, while sensor nodes are to be deployed in thousands spreading a considerably large geographical area. This could have a significant impact on the final outcome. In a wireless sensor network with thousands of nodes, cluster heads far from base station would drain their energy quickly. This results in cascading effect in which nodes far away from base station would drain out quickly and reducing the area covered by the sensor networks.

7.2.2 Threshold sensitive Energy Efficient sensor Network protocol (TEEN)

TEEN is a cluster based routing protocol proposed by Manjeshwar et.al. [25] and is similar to LEACH described in the previous section except that sensor nodes do not have data to be sent at a fixed rate. Authors of this protocol have proposed to classify sensor networks into proactive sensor networks and reactive networks. Proactive networks continuously monitor the environment and thus have data to be sent at a constant rate. LEACH suites such sensor networks in transmitting data efficiently to the base station with minimum energy consumption. In case of reactive networks nodes, only when the variable being monitored increases beyond a threshold, the sensing node needs to transmit the data.

TEEN employ's the cluster formation strategy of LEACH but adopts a different strategy in data transmission phase. TEEN makes use of two user defined parameters Hard Threshold (Ht) and Soft Threshold (St) to determine if it needs to transmit the value it sensed currently. When the monitored value exceeds Ht for first time, it is stored in a variable and is transmitted during the time slot of the node. Subsequently if the monitored value exceeds the currently stored value by a magnitude of St node decides to transmit the data. This transmitted value is stored for comparing in future.

All the issues that were un-addressed by authors of LEACH were left by authors of TEEN also. In addition to those described in the previous section for LEACH, TEEN suffers from following drawbacks.

- As all nodes do not have data to be transmitted continuously, some cluster heads will be draining more energy as most of the nodes in its cluster might have data to be transmitted. If there is a hot spot in the network, where there is continuous increase in the value of variable being monitored, all the sensor nodes in this region will drain their battery very soon resulting in a situation where there could be no nodes to monitor that hot spot.

- If a node does not have any data to be transmitted, its time slot is left unused. Some other node could have potentially used this slot.

7.2.3 Power-Efficient Gathering in Sensor Information Systems (PEGAGIS)

PEGASIS [22] is another power efficient sensor networking protocol, which is claimed an improvement over LEACH. This is a near optimal chain-based protocol, in which each node communicates only with its closest neighbor and takes turns to transmit to the base station. This reduces the power required to transmit data per round and hence will increase lifetime of the network by two times when compared to LEACH.

The main motive behind this protocol was to increase the lifetime of each sensor node, to decrease the bandwidth consumed by using local collaboration among the nodes and tolerate node failures.

This protocol considers that the network holds the following assumptions and properties.

- The base station is fixed at a distance from the nodes and each node has the capability of transmitting data to base station either directly or through any other node.

- Each node has location information about all other nodes.

- All the nodes are homogeneous and with uniform energy.

- These nodes are immobile.

Initially these nodes will find its closest neighbor by sending a power signal and then gradually reducing the power, till it heard by only one node. The data is transmitted to base station with data fusion, which will combines two or more data packets and sends as one. This is similar to LEACH protocol. The key idea in PEGASIS is to form a chain among the nodes so that each node will receive from and transmit to a close neighbor. The fused data moves from node to node and is transmitted to base station by one of the designated nodes (which are the members of the chain). Nodes will take turns to transmit to base station, reducing the power required per round.

7.2.4 Two Level Clustering Algorithm

A solution for clustering was proposed by Estrin et.al. [3]. They gave a two level clustering algorithm that can be extended to build a cluster hierarchy. In this algorithm every sensor at a particular level is associated with a radius or the number of hops that its advertisements will reach. All sensors start with level 0. Sensors at a higher level are associated with higher radii. Each sensor then sends out periodic advertisements to neighbors that are its current level distance away. The advertisements carry its current level, its parent's identity (if any) and remaining energy. It then waits for a wait time that is proportional to its radius to receive other nodes advertisements. At the end of this timer all level 0 nodes start a promotion timer that is proportional to its remaining energy reserves and the number of level 0 nodes whose advertisements it received. When the promotion timer expires the node promotes itself to level 1. It then starts sending out periodic advertisements at level 1. In these new advertisements it lists its potential children which are the level 0 nodes that it previously heard. A level 0 node then picks up as its parent one of the level 1 nodes, whose advertisements included its identity. Once a level 0 node picks up its parent it cancels its promotion timer and drops out of the race. At the end, each the level 1 node starts a wait timer and waits for its potential children's acknowledgements. If no level 0 node selected it as its parent or its energy dropped below a certain level it demotes itself to a level 0 node. All level 0 and level 1 nodes periodically enter the wait stage to take into account any change in network conditions and reclustering takes place.

8 Other Important Issues

8.1 Power management in sensor nodes

Lifetime of distributed micro sensor nodes is a very important issue in the design of sensor networks. Sensor networks should operate with minimum possible energy to increase the life of sensor nodes. So we need power aware computation/communication component technology, low-energy signaling and networking, power aware software infrastructure. Nodes partitioning can be considered for low energy computation and communication.

Design challenges encountered in the building of wireless sensor networks can generally be classified into hardware, wireless networking, and applications. All three categories should minimize the power usage to increase the life of sensor node. Hardware includes the design activities related to all hardware platforms that make up sensor networks. MEMS, digital circuit design, system integration, RF are important categories in the design of hardware. This topic is outside the scope of this paper.

Second aspect includes design of power efficient algorithms and protocols. Once the system is designed additional power savings can be obtained by using dynamic power management (DPM) [13]. The basic idea behind DPM is to shutdown the devices when not needed and get them back when needed. So we need an embedded operating system [14] that is able to support DPM. We can also conserve energy by using dynamic voltage scheduling (DVS) which tries to minimize the idle processor cycles using feed back control system. We can also save energy by using efficient link layer [18], data aggregation [12] and system partitioning [14] as explained in the following sections.

8.1.1 Power-aware Sensor Node Model or Dynamic Power Management (DPM)

The switching of node states takes some finite time and resource [13]. So we have to carefully use DPM to get maximum life of sensor node. This shutdown yields good savings. But in many cases we may not know beforehand when we need a particular device. So we need stochastic analysis to predict the future events. The following model of sensor deals with switching of node state in power efficient manner.

This model describes the power consumption in different levels of node-sleep states. Every component in a node can be in different states. In general if we have N components in sensor node, each node-sleep state corresponds to particular a combination of component power modes.

Let us assume that all sensor nodes will have components such as processor, memory, sensing with A/D converter, Radio. So a sensor node will have the following sleep states.

States	Processor	Memory	Sensor with A/D	Radio
S0	Active	Active	On	T_x/R_x
S1	Idle	Sleep	On	T_x
S2	Sleep	Sleep	On	R_x
S3	Sleep	Sleep	On	Off
S4	Sleep	Sleep	Off	Off

Table 1: Sensor node sleep states

Each sleep state is characterized by latency and power consumption. The deeper the sleep state, the lesser the power consumption, and more the latency. Here we can see from the above table that not all combinations of states are useful. For example if a processor is in idle state then memory should be in sleep state. This removes some combinations from the node states.

Let us assume $P_k(T_{th,0})$ is the probability that no events occur under area covered by node k. We should switch the states if we can achieve savings in power usage. Figure 6, shows the latency and power transitions.

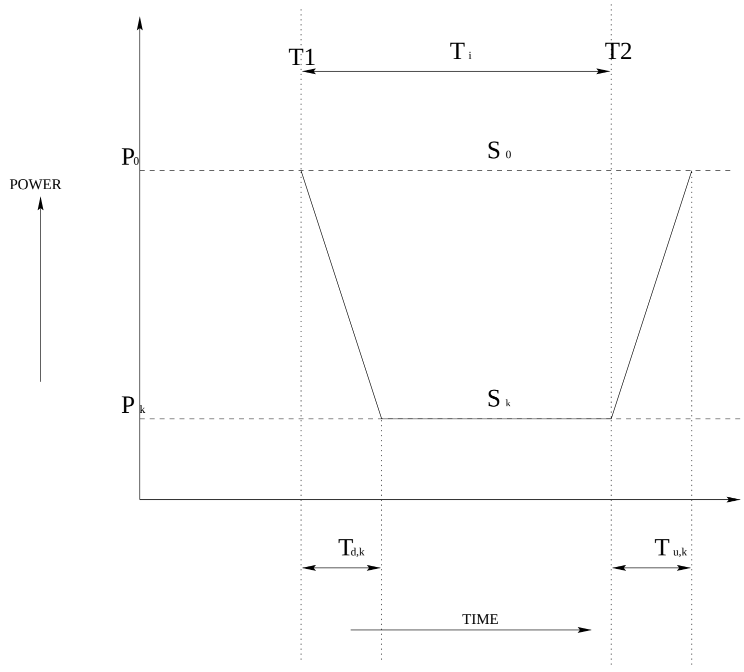

Figure 6: State Transition Latency and Power

Let us assume an event occurred at T_1 and next event occurred at $T_2 = T_1 + T_i$. At T_1, node k wants to go into sleep state k from active state S_0. Each state has a power consumption P_k, transition times $T_{d,k}$ and $T_{u,k}$ as shown in Figure 6.

Energy savings:

$$E_{save,k} = P_{0,T_i} - (P_0 + P_k/2) * (T_{d,k} + T_{u,k}) - P_k(T_i - T_{d,k})$$

Such a transition is only useful when $E_{save,k} > 0$. This leads to the threshold value:

$$T_{Th,k} = [T_{d,k} + [P_0 + P_k/P_0 - P_k] * T_{u,k}]$$

So the system puts the node into sleep state S_k by testing the probability of an event occurring in the corresponding sleep time threshold $T_{Th,k}$ against system defined $P_{th,0}$. Node k also updates $P_{th,0}$ after every event.

When system is in S4, there is a chance that some events get lost. If the task is critical then we can't allow S4. If event is not so critical then we can ignore some events. All states must be controlled by the operating system present in the node.

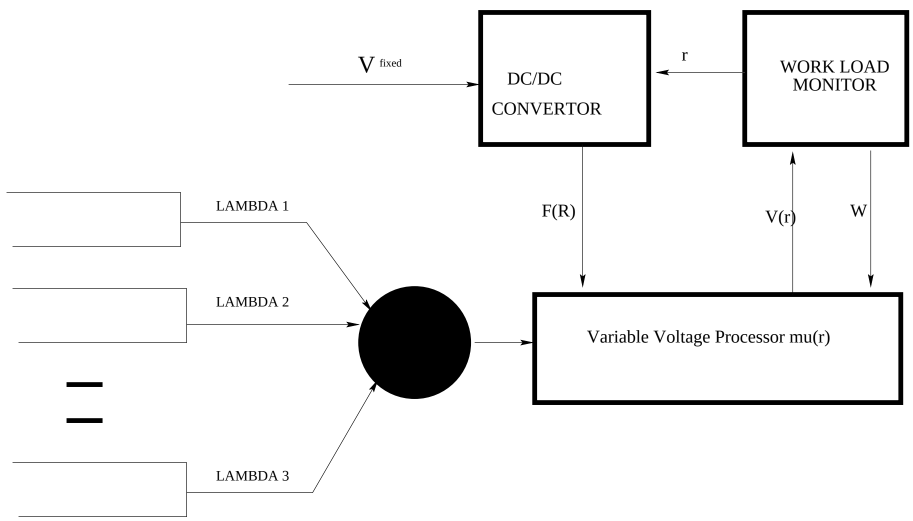

Figure 7: Block Diagram of a DVS Processor System

8.1.2 Energy Work-load Model or Dynamic Voltage Scheduling (DVS)

We can also save the energy by optimizing the sensor nodes performance in active state. Dynamic voltage scheduling (DVS) [13] [19] is an effective tool for this case. The main idea behind DVS is to change the power supply to match the workload. We have to tune the processor to deliver the required throughput to avoid idle cycles. The crux of the problem lies in the fact that future workloads are non-deterministic. So the efficiency depends on predicting the future workloads. Figure 7, shows the block diagram of variable voltage processing system. The *Task Queue* models the various events for a processor e.g. I/O, disk drives, network links etc. Each of the n sources produces events at an average rate $\lambda_k, (k = 1, 2, ..., n)$. An OS scheduler manages tasks and decides which event should run on the processor with average rate (λ) which is equal to the sum of n average arrival rates. The processor offers a time varying processing rate $\mu(r)$. The OS kernel measures the idle cycles and computes the normalized workload over the observed period. Work load monitor sets processing rate r based on workload w and previous workloads over some observation period. This rate r decides operating voltage *V(r)* and operating frequency *f(r)*. Thus this acts as a closed loop feedback control system.

8.1.3 Radio Energy Model or Energy Efficient Link Layer

Energy consumption can be achieved by compromising on quality of link layer established. This is possible by maintaining bit error rate (BER) [18] just below the user requirements. Different error controlling algorithms like Bose-Chaudhuri-Hocquen (BCH) coding, convolution coding and turbo-coding can be employed for error control. We need to choose the algorithm with lowest power consumption to support the predetermined BER and latency.

Total energy consumed for communication can expressed by the equation:

$$E = P_{tx} * (T_{on-tx} + T_{startup-tx}) + P_{out} * T_{on-tx} + E_{dsp}^{(e)} + d * P_{rx} * (T_{on-rx} + T_{startup-rx}) + E_{dsp}^{(d)}$$

P_{tx}, P_{rx} - power consumption for T_x, R_x

T_{on-tx}, T_{on-rx} - actual data transmission/reception time on $T_x and R_x$

$T_{startup-tx}, T_{startup-rx}$ - startup time for T_x, R_x

P_{out} - output transmit power

$E_{dsp}^{(e)}$ - encode power

$E_{dsp}^{(d)}$ - decode power

From the above equation we can calculate average energy to transmit, receive, encode, and decode each bit. From the calculated values we can make a choice between existing error control algorithms.

8.1.4 Power Efficient System Partitioning

Local computation [14] [17] of sensor data in wireless networks can be highly energy efficient. Partitioning the computation among multiple sensor nodes and performing the computation in parallel permits a greater control on latency and results in energy consumption through frequency scaling and voltage scaling.

8.1.5 Power Efficient Topologies

Biomedical wireless sensor networks (WSN) [16] are a class of sensor networks, with applications in medical field. These sensor networks include monitors and implantable devices intended for long-term placement in the human body. These sensor nodes should not dissipate any energy to avoid damage to surrounding tissues. Since topology is predetermined we need to evaluate a power efficient topology for these situations. Ayad et. al. proposed directional source aware routing protocol for this class of sensor networks and explored various topologies to determine what is the best one.

Directional Source-Aware routing Protocol (DSAP) incorporates power considerations into routing tables. In this protocol each node is given a unique identifying set. Each number in this set corresponds to how many nodes separate it from the edge of the network in a particular direction. We can hard code this information into each node, since topology is predetermined.

For example consider 2D topology with up to 4 neighbors as shown in Figure 8. Here Node 6 will have identifier (1,1,2,2) meaning 1(left), 1(up), 2(right), 2(down) nodes to reach the edges.

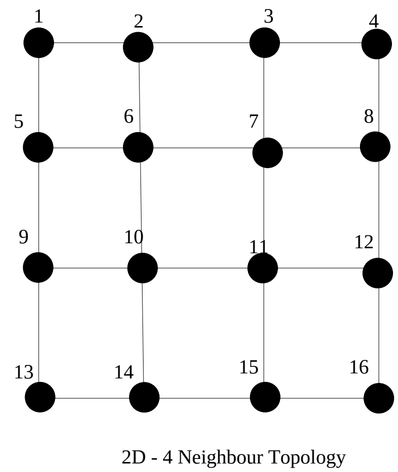

Figure 8: Example 2D Topology

When transmitting a message, the destination node identifying set is subtracted from source node identifying set. This gives at most two positive numbers (2D topology with 4 neighbors) that describes the direction in which the message can be forwarded. This could be either north or south, and east or west. Negative numbers are disregarded. The decision to move right/left or up/down is determined by the directional value (DV) of the neighboring nodes. Taking each of the neighbor's identity set and subtracting it from the destination identity set, we will get the DV. These four numbers in each resulting set are added together, and the node which has the set with smaller number is chosen as the next hop.

For example, node 1(0,0,3,3) wants to transmit message to node 16(3,3,0,0). So (0-3,0-3,3-0,3-0) = (-,-,3,3) (neglect negative values). Thus the next hop could be either node 2 or node 5 in this case. DV of node 2(1,0,2,3) = (3-1,3-0,0-2,0-3) = (2,3,-2,-3). Similarly DV for node 5(0,1,3,2) = (3-0,3-1,0-3,0-2) = (3,2,-3,-2). Here DVs are equal. So we can choose either node 2 or node 5 as the next hop and proceed. By using the above routing algorithm,the authors considered energy dissipation in various topologies like 2D mesh with 3 neighbors, 2D mesh with 4 neighbors, 2D mesh with 6 neighbors, 2D mesh with 8 neighbors and 3D mesh with 6 neighbors. Based on the simulation results, authors concluded that topology consisting of 3D mesh with 6 neighbors has least energy dissipation and is most suited for biomedical sensor networks.

8.2 Clock Synchronization

Some of the communication algorithms, for wireless sensor networks, that we presented in this paper make an inherent assumption that there exist some mechanism through which local clocks of all the sensor nodes are synchronized. Though this assumption is valid we need to have an explicit way of synchronizing local clocks of all sensor nodes. Apart from implementation of the communication algorithms, clock synchronization is required for accurate time stamps in cryptographic schemes, for recognizing duplicate detection of same event from different sensor nodes, for data aggregation algorithms like beam forming, for ordering of logged events and many other similar applications. In this section we present a post-facto clock synchronization algorithm proposed by Jeremy Elson and Deborah Estrin [26].

The post-facto clock synchronization algorithm discussed here is suitable to applications like beam forming, duplicate event detection and other similar localized systems. This algorithm is expected to be implemented on systems similar to WINS platform where a processor has various sleep modes and has the capability of powering down high-energy peripherals. Because of the capability of the sensor node processor to power down a device and power up when there is a requirement to sense data and transmit, existing clock synchronization methods for distributed systems are not applicable.

The basic idea behind post facto clock synchronization algorithm is, for certain applications like data fusion and beam-forming it is sufficient to order the events in a localized fashion. In this scheme nodes clocks are normally unsynchronized. When a stimulus arrives (time to sense and transmit data), each node records the stimulus with respect to its local clock. Immediately following this event a third party will broadcasts a synchronization pulse. Every node receiving this pulse normalizes their stimulus time stamp with respect to broadcasted synchronizing pulse. It is essential to note that the time elapsed between recording the stimulus and arrival of synchronized pulse needs to be measured accurately. For this reason the algorithm is inappropriate for systems, which need to communicate a time stamp over long distances.

8.3 Locationing

This consists of two different concepts.The first is finding the location of the sensor nodes with respect to each other. The second is finding the location of an external object e.g. an intruder entering into a building.

8.3.1 Locationing of Sensor Nodes

The distribution of sensor nodes in a location poses many challenges. The range of each sensor node is proportional to the power available in that sensor node. So, the prime factor in distributing these sensor networks is again limited power. In a sensor network, each node can communicate only with its neighbors. So, each node should be surrounded with sufficient number of neighbors. In other words the network should be well connected so that failure of a node or decrease in power level in a node or an incoming obstacle should not disconnect the network. In the following paragraphs some of the techniques used are briefly discoursed.

According to [23], the location of sensor nodes consists of two components, which are described below.

1. Distance or Range measurements: This is to know the maximum separation between two nodes in order to communicate with each other. This can be based on any physical variables of signals like received signal strength (RSSI), angle of arrival (AOA), time of arrival (TOA) or time-distance of arrival (TDOA).

2. Triangulation problem: This is a mathematical way of identifying the location of a node. If we are given the geometrical locations X_i, Y_i, Z_i and range R_i of each of its neighbors, then its geometric location X, Y, Z can be calculated using the set of equations as shown below. $(X_1 - X)^2 + (Y_1 - Y)^2 + (Z_1 - Z)^2 = R_{12}$

$(X_2 - X)^2 + (Y_2 - Y)^2 + (Z_2 - Z)^2 = R_{22}$
....
....
$(X_n - X)^2 + (Y_n - Y)^2 + (Z_n - Z)^2 = R_{n2}$
But the complication arises due to the difficulty of finding the accurate range measurement. Also the accuracy derived through triangulation method heavily depends on the geometry of position references and the configuration of network nodes. This problem is solved by taking the least-mean square value of all the range measurements of the neighbors of the node, whose position is unknown [23].

With the above method we can only know the relative position of a node with respect to its neighboring nodes. To know the accurate position of all sensor nodes this method follows two steps. The first one is called *Local Positioning* and the second one is called as *Global Positioning*.

The local positioning is done using *Assumption Based Coordinates* (ABC) algorithm. In ABC algorithm we assume that the first node is placed at the origin $(0, 0, 0)$. The next node is placed at $(r_{01}, 0, 0)$, where r_{01} is the range measurement between these two nodes. The coordinates of the third node are calculated by using the following equations.
$x_2 = r_{012} + r_{022} + r_{122}/2r_{01}$;
$y_2 = SQRT(r_{022} + x_{22})$;
$z_2 = 0$.

The fourth node is placed at the coordinates calculated using the following equations.
$x_3 = r_{012} + r_{032} + r_{132}/2r_{01}$;
$y_3 = r_{032} - r_{232} + x_{22} + y_{22} - 2x_2x_3/2y_2$;
$z_3 = SQRT(r_{032} - x_{32} - y_{32})$;

After finding the locations of the first four nodes, the locations of other nodes can be calculated using the equations of triangulation.

Global balancing is achieved through cooperating ranging approach. Cooperative ranging exploits the high connectivity of the network to translate the global positioning challenge into a number of distributed local optimization problems that iteratively converge to a global by interacting with each other. Though this approach takes some time to converge, it is not a big issue if the lifetime of the network is high. There are many methods to do it. Two of them are *Global Topology Discovery* and *Iterative Local Triangulation*.

In Global Topology Discovery, first ABC algorithm is applied to get the local positioning. The resulting information is forwarded to the neighboring nodes. Then each of the anchoring nodes removes a degree of freedom in the coordinate space and forces the neighboring nodes to linearly transform their own coordinate system. This is done in respect to both transposition and rotational perspective. This process continued through out the network till the converged solution is obtained. But the disadvantage with this approach is that the propagation of initial distance errors cannot be overcome with this approach. This will yield unacceptably large position errors.

In iterative local triangulation, each node uses the recently computed coordinates of each neighboring node along with range measurement to find its own coordinates. This process is iteratively applied till all nodes values converge.

8.3.2 Locationing of External Objects

Estrin et.al. [3] gave an application of their clustering algorithm for pinpointing the exact location of an object. For energy efficiency we need the fewest number of sensors participating in a triangulation process. The authors give a simple rule for determining whether a cluster head should participate in the triangulation process or not based on some assumptions. The assumptions are that each sensor can determine its position in 2D space and can specify the approximate direction in which the object lies relative to its own location. The rule is that if all the remaining cluster heads lay on the same side of a line drawn between itself and the object of interest then that clusterhead should participate in the computation. This ensures that the widest possible measurement baseline is found. To implement this rule a single message exchange between neighboring clusterheads suffices. The algorithm is robust to link or node failures and overhead is proportional to the local population density and a sublinear function of the total number of nodes.

8.4 Upper Bounds on the Lifetime of Sensor Networks

Every node in a sensor network is provided with a limited battery power supply and as there are thousands of nodes in a sensor network it is infeasible to replenish them with energy. As a result lifetime of the sensor network is an important parameter to determine. Here the term lifetime refers to the time period for which the sensor network is capable of transmitting the sensed data to the base station located away from the network. Determining the theoretical bound for life time will allow calibration of real world data gathering protocols and help in understanding the factors which prevent these protocols from reaching the theoretical limits. Here we describe a method for determining the upper bound on lifetime described in [27] by Manish Bhardwaj, Timothy Garnett and Anantha P. Chandrakasan. We describe the model of sensor networks considered by the authors [27] here. There is a specified region R covered by sensor nodes and these nodes need to track a moving source and route this data to the base station located away from the sensor network. Sensor nodes, which are within a distance 'ds from source, can sense it and will try to route the sensed data through multiple hops. As the source moves the path along which data is routed changes. The movement pattern of the source is non-deterministic and is described by a spatial probability distribution function (p.d.f). The parameters that model the node energy behavior are energy needed to sense a bit (E_{sense}), receive a bit (E_{rx}) and transmit a bit (E_{tx}). These are expressed by the equations

$E_{tx} = a_{11} + a_2 d^n$

$E_{rx} = a_{12}$

$E_{sense} = a_{13}$

It is established that optimum number of equidistant hops K_{opt} for routing data between two nodes separated by distance D is given by $\lceil \frac{D}{d_{char}} \rceil$ or $\lfloor \frac{D}{d_{char}} \rfloor$. Where d_{char} is given by the expression $(a_1/(a_2(n-1)))^{1/n}$.

For an exhaustive proof we encourage the reader to refer the paper. Now we will explain how to calculate upper bound of lifetime for various source behaviors.

- FIXED POINT ACTIVITY: For the case where the source location is fixed the theoretical limit for lifetime of sensor network with N nodes each with energy E is given by the following expression.
  $$\frac{N*E}{\frac{n}{(n-1)^n}\sqrt{a_1^{n-1}a_2(n-1)(d_B-d_S)r}}$$
  Here $d_B - d_S$ gives the distance of the multi hop path along which data is routed to the base station. The transmission rate is given by r.

- ACTIVITY DISTRIBUTED ALONG A LINE: For case where source is moving along a straight line, limit for lifetime of sensor network with N nodes each with energy E is given by the following expression
  $$\frac{N*E}{(\frac{n}{n-1})a_1(\frac{d_B+\frac{d_N}{2}+d_S}{d_{char}})}$$

Using the above equations and analysis given in the paper to calculate lifetime of sensor network with source moving in a rectangular region one can calculate the life time of sensor network for any movement pattern of the source. Dividing the coverage region into smaller regions, which fit any of the above cases, and then taking some sort of average can achieve this.

9 Conclusions and future work

In this paper we made an attempt to present an overview of wireless sensor networks and discuss various solutions proposed to prolong the lifetime of sensor networks by minimizing energy requirements for carrying out various tasks. Solutions to issues like link layer establishment, routing data from sensor node to sink, power management, location management and clock synchronization were discussed.

In various sections of this paper we highlighted some assumptions made by the authors of respective solutions which are questionable. For example few protocols like LEACH, and TEEN were simulated with sensor networks having 100 nodes. There is a need to explore if these protocols are scalable and if they can be effective in all conditions.

To start with we see a need for cooperation among research community to come to a consensus on some characteristics of wireless sensor networks and underlying assumptions that can be made while working on any solution. Furthermore we see a need for deploying a test bed where in any proposed solution can be deployed and a comparative study made.

References

[1] Estrin, D.; Girod, L.; Pottie, G.; Srivastava, M. Instrumenting the world with wireless sensor networks. *IEEE International Conference on Acoustics, Speech, and Signal Processing, Pages 2033 -2036*, 2001.

[2] Elson, J.; Estrin, D. Time synchronization for wireless sensor networks. *Proceedings 15th International Parallel and Distributed Processing Symposium, Pages 1965 -1970*, 2001.

[3] Estrin, D., Govindan, R., Heidemann, J., Kumar, S. Next century challenges: Scalable Coordination in Sensor Networks. *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking, Pages 263 - 270*, 1999.

[4] Heinzelman, W., Kulik, J., Balakrishnan, H. Adaptive protocols for information dissemination in wireless sensor networks. *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking, Pages 174 - 185*, 1999.

[5] Heinzelman, W., Kulik, J., Balakrishnan, H. Negotiation-based protocols for disseminating information in wireless sensor networks. *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, 1999.

[6] Pottie, G. J., Kaiser, W. J. Wireless integrated network sensors. *Communications of the ACM, Volume 43, Issue 5 Pages 51 - 58*, 2000

[7] Sohrabi, K.; Gao, J.; Ailawadhi, V.; Pottie, G.J. Protocols for self-organization of a wireless sensor network. *IEEE Personal Communications, Volume 7, Issue 5, Pages 16 -27* , 2000.

[8] Sohrabi, K.; Pottie, G.J. Performance of a novel self-organization protocol for wireless ad-hoc sensor networks *IEEE Vehicular Technology Conference, Volume 2, Pages 1222 -1226*, 1999.

[9] Pottie, G.J. Wireless sensor networks. *Information Theory Workshop Pages 139 -140*, 1998.

[10] Agre, J.; Clare, L. An integrated architecture for cooperative sensing networks. *Computer, Volume 33 Issue 5, Pages 106 - 108*, 2000

[11] Min, R.; Bhardwaj, M.; Seong-Hwan Cho; Shih, E.; Sinha, A.; Wang, A.; Chandrakasan, A. Low-power wireless sensor networks. *Fourteenth International Conference on VLSI Design, Pages 205 - 210*, 2001.

[12] Wang, A.; Heinzelman, W.R.; Chandrakasan, A.P. Energy-scalable protocols for battery-operated micro sensor networks. *IEEE Workshop on Signal Processing Systems, Pages 483 - 490*, 1999.

[13] Sinha, A.; Chandrakasan, A. Dynamic power management in wireless sensor networks. *IEEE Design and Test of Computers, Volume 18, Issue 2, Pages 62-74*, 2001.

[14] Wang, A.; Chandrakasan, A. Energy efficient system partitioning for distributed wireless sensor networks. *IEEE International Conference on Acoustics, Speech, and Signal Processing, Volume 2, Pages 905 - 908*, 2001.

[15] Slijepcevic, S.; Potkonjak, M. Power efficient organization of wireless sensor networks. *IEEE International Conference on Communications, Volume 2, Pages 472 - 476*, 2001.

[16] Salhieh, A.; Weinmann, J.; Kochha, M.; Schwiebert, L. Power efficient topologies for wireless sensor networks. *International Conference on Parallel Processing, Pages 156 - 163*, 2001

[17] Heinzelman, W.R.; Chandrakasan, A.; Balakrishnan, H. Energy-efficient communication protocol for wireless micro sensor networks. *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, Pages 3005 - 3014*, 2000.

[18] Shih, E.; Calhoun, B.H.; Seong Hwan Cho; Chandrakasan, A.P. Energy-efficient link layer for wireless micro sensor networks. *Proceedings of IEEE Computer Society Workshop on VLSI, Pages 16 - 21*, 2001.

[19] Im, C.; Huiseok Kim; Soonhoi Ha. Dynamic voltage scheduling technique for low-power multimedia applications using buffers. *International Symposium on Low Power Electronics and Design, Pages: 34 - 39*, 2001.

[20] Tsiatsis, V., Zimbeck, S., Srivastava, M. Architecture strategies for energy-efficient packet forwarding in wireless sensor networks. *Proceedings of the 2001 international symposium on Low power electronics and design, Pages 92 - 95*, 2001.

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

[22] Lindsey, S.; Raghavendra, C., S. PEGASIS: Power-Efficient Gathering in Sensor Information Systems. *International Conference on Communications*, 2001.

[23] Savarese, C., Rabaey J. Locationing in Distributed Ad-Hoc Wireless Sensor Networks. *IEEE proceedings on Acoustics, Speech, and Signal Processing, Pages 2037 -2040*, 2001.

[24] Ye F.; Chen, A.; Liu, S.; Zhang L. A scalable solution to minimum cost forwarding in large sensor networks. *Proceedings of Tenth International Conference on Computer Communications and Networks, Pages 304 -309*, 2001.

[25] Manjeshwar, A.; Agrawal, D.P. TEEN: a routing protocol for enhanced efficiency in wireless sensor networks. *International Proceedings of 15th Parallel and Distributed Processing Symposium, Pages 2009 -2015*, 2001

[26] Elson, J.; Estrin, D. Time synchronization for wireless sensor networks. *International Proceedings of 15th Parallel and Distributed Processing Symposium, Pages 1965 -1970*, 2001.

[27] Bhardwaj, M., Chandrakasan, A., Garnett, T. Upper Bounds on the Lifetime of Sensor Networks. *IEEE International Conference on Communications, Pages 785 - 790*, 2001.