ALGORITHMS FOR DATA-GATHERING IN WIRELESS SENSOR NETWORKS

SIAVASH RAHIMI

A THESIS

IN

THE DEPARTMENT

OF

ELECTRICAL AND COMPUTER ENGINEERING

PRESENTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF MASTER OF APPLIED SCIENCE

CONCORDIA UNIVERSITY

MONTRÉAL, QUÉBEC, CANADA

JULY 2008

Your file Votre référence
ISBN: 978-0-494-42496-4
Our file Notre référence
ISBN: 978-0-494-42496-4

Abstract

Algorithms for Data-Gathering in Wireless Sensor Networks

Siavash Rahimi

Wireless sensor networks consist of a large number of small battery powered sensor nodes with limited energy resources which are responsible for sensing, processing, and transmitting the monitored data . Once deployed, the sensor nodes are normally inaccessible to the user, and thus replacement of the battery is generally not feasible. A major concern in designing and operating dense Wireless Sensor Networks (WSNs) is the energy-efficiency. Hierarchical clustering and cross-layer optimization are widely accepted as effective techniques to ameliorate this concern. We propose two different novel energy efficient algorithms to gather data from sensor nodes. Energy-Efficient Media Access Control (EE-MAC) protocol is the first algorithm, which has excellent scalability and performs well for both small and large sensor networks. We will also provide a theoretical analysis of the protocol and give guidelines on how to find the optimal protocol parameters such as the number of clusters. In addition, we develop and analyze a novel and scalable Spiraled Algorithm for Data-gathering (SAD) that periodically selects cluster heads according to their geographic locations and residual energy by sorting nodes on virtual spirals. Theoretical analysis and simulation results show that SAD can achieve as much as a factor of three prolonging network lifetime compared with other conventional protocols like LEACH especially when the network is large. Moreover, SAD is also able to distribute energy dissipation evenly throughout the sensors such that 80% of the nodes run out of batteries in the last 20% of the network lifetime.

Acknowledgments

This thesis owes its existence to the help, support, and inspiration of many people. In the first place, I would like to express my sincere appreciation and gratitude to Prof. Dongyu Qiu for his support and encouragement. The discussions and cooperations with all of my colleagues have contributed substantially to this work: Zuhui Ma and Mojtaba Zokayee Ashtiany have been long-time companions during the graduate years, shaping my mind during many technical and non-technical discussions along the way.

Finally, I owe special gratitude to my family for continuous and unconditional support of all my undertakings, scholastic and otherwise.

Contents

List of Figures

List of Tables

Chapter 1

Introduction to Wireless Sensor Networks

1.1 Overview

With the popularity of laptops, cell phones, PDAs, GPS devices, RFID, and intelligent electronics in the post -PC era, computing devices have become cheaper, more mobile, more distributed, and more practical in daily life. It is now possible to construct, from commercial off-the-shelf (COTS) components, a wallet size embedded system with the equivalent capability of a 90's PC. Such embedded systems can be supported with scaled sown Windows or Linux operating systems. From this perspective, the emergence of wireless sensor networks (WSNs) is essentially the latest trend of Moore's Law toward the miniaturization and ubiquity of computing devices.

Typically, a wireless sensor node (or simply sensor node) consists of sensing, computing, communication, actuation, and power components. These components are integrated on a single or multiple boards, and packaged in a few cubic inches. With state-of-the-art, low-power circuit and networking technologies, a sensor node powered by 2 AA batteries can last for up to three years with 1%low duty cycle working mode. A WSN usually consists of tens to thousands of such nodes that communicate through wireless channels for information sharing and cooperative processing. WSNs can be deployed on a global scale for environmental monitoring and habitat study, over a battle field for military surveillance and reconnaissance, in emergent environments for search and rescue, in factories for condition based maintenance, in buildings for infrastructure health monitoring, in homes to realize smart homes, or even in bodies for patient monitoring[24, 34, 55, 63].

After the initial deployment (typically ad hoc), sensor nodes are responsible for self-organizing an appropriate network infrastructure, often with multi-hop connections between sensor nodes. The onboard sensors then start collecting acoustic, seismic, infrared or magnetic information about the environment, using either continuous or event driven working modes. Location and positioning information can also be obtained through the global positioning system (GPS) or local positioning algorithms. This information can be gathered from across the network and appropriately processed to construct a global view of the monitoring phenomena or objects. The basic philosophy behind WSNs is that, while the capability of each individual sensor node is limited, the aggregate power of the entire network is sufficient for the required mission.

In a typical scenario, users can retrieve information of interest from a WSN by injecting queries and gathering results from the so-called base stations (or sink nodes), which behave as an interface between users and the network. In this way, WSNs can be considered as a distributed database [64, 66]. It is also foreseen that sensor networks will ultimately be connected to the internet, through which global information sharing becomes feasible.

The era of WSNs is highly anticipated in the near future. In September 1999, WSNs were identified by *Business Week* one of the most important and impactive technologies for the 21st century[1]. Also in January 2003, The *MIT's Technology Review* stated that WSNs are one of the top ten emerging technologies[2]. It is also estimate that WSNs generated less than $150 million in sales in 2004, but would top $7 billion by 2010[80].

1.2 Enabling Technologies

1.2.1 Hardware

The hardware basis of WSNs is driven by advances in several technologies. First, System-on-Chip (SoC) technology is capable of integrating complete systems on a single chip. Commercial SoC based embedded processors from Atmel, Intel, and Texas Instruments have been used for sensor nodes such as UC Berkeley's motes [20, 65], UCLA's Medusa[60] and WINS [67], and MIT's μAMPS-1[56]. Several different research groups, such as the PicoRadio team from UC Berkeley [94], have been trying to integrate prototype sensor nodes (PicoNode I) onto a few chips (PicoNode II). Many interesting SoC designs related to wireless communication and sensor nodes can also be found at the SoC Design Challenge, 2004-2006 [16].

Second, commercial RF circuits enable short distance wireless communication with extremely low power consumption. Commercial products from RF Monolithics, Chipcon, Conexant Systems, and National Semiconductor have been used on various sensor nodes, including motes, Medusa, WINS, and μAMPS. A SoC based ZigBee radio is also available from Embre Cooperation[21]. These Commercial radios can usually achieve a data rate of tens to hundreds of Kbps, while consuming less than 20 mW of power for both packet transmission and receiving[61]. With wideband technology, enhanced modulation schemes and error detection mechanisms are employed to provide increased robustness.

Third, Micro-Electro-Mechanical Systems (MEMS) technology[88] is now available to integrate a rich set of sensors onto the same CMOS chip. Commercially available sensors now include thermal, acoustic/ultrasound, and seismic sensors, magnetic and electromagnetic sensors, optical transducers, chemical and biological transducer, accelerometers, solar radiation detectors, photo-synthetically active radiation detectors, and barometric pressure detectors[49]. These sensors can be used in broad range of applications, including acoustic ranging, motion tracking, vibration detection, and environmental sensing.

1.2.2 Wireless Networking

Besides hardware technologies, the development of WSNs also relies on wireless networking technologies. The 802.11 protocol, the first standard for wireless local area networks (WLANs), was introduced in 1997. It was upgraded to 802.11b with increased data rate and CSMA/CA mechanisms for medium access control (MAC). Although designed for wireless LANs that usually consist of laptops and PDAs, the 802.11 protocols are also assumed by many early efforts on WSNs. However, the high power consumption and excessively high data rate of 802.11 protocols are not suitable for WSNs. This fact has motivated several research efforts to design energy efficient MAC protocols[53, 99, 91]. Recently, the 802.15.4-based ZigBee protocol was released, which was specifically designed for short range and low data rate wireless personal area networks (WPAN). Its applicability to WSNs was soon supported by several commercial sensor node products, including MicaZ[20], Telos[61], and Ember products[21].

Above the physical and MAC layers, routing techniques in wireless networks are another important research direction for wireless ad hoc networks or wireless mobile networks. These protocols, including DSR[39] and AODV[73], are hardly applicable to WSNs due to their high power consumption. They are also designed to support general routing requests in wireless networks, without considering specific communication patterns in WSNs. Nevertheless, the customization of these protocols for WSNs and development of new routing techniques have become hot research topics[11, 22, 29, 33, 13, 44, 51, 79, 95]. The main idea behind these research efforts is to enable energy efficient and robust routing by exploiting link and path diversity.

1.2.3 Collaborative Signal Processing

Collaborative signal processing algorithms are another enabling technology for WSNs. While raw data from the environment are collected by sensor nodes, only useful information is of importance. Hence, raw data need to be properly processed locally at sensing nodes, and only processed data is sent back to the end users. since computation is much more energy efficient than wireless communication, this avoids wasting energy on sending large volumes of raw data. Such signal processing is often required to be performed by a set of sensor nodes in proximity, due to the weak sensing and processing capabilities of each individual node.

Information fusion is an important topic for collaborative signal processing. Since sensor readings are usually imprecise due to strong variations of monitoring entity or interference from the environment, information fusion can be used to process data from multiple sensors in order to filter noise measurements and provide more accurate interpretations of the information generated by a large number of sensor nodes. A rich set of techniques is applicable in this context, including Kalman filtering, Bayesian inference, neural networks, and fuzzy logic[5, 42, 82].

Other signal processing techniques that have been developed for WSNs include time synchronization[26, 45], localization[58, 76], target tracking[25, 52], edge and boundary detection[19, 48, 69], calibration[37, 92], adaptive sampling [59, 93], and distributed source coding[68, 75].

1.3 Application of Interest

1.3.1 Application Examples

The claim of wireless sensor network proponents is that this technological visions facilitate many existing application areas and bring into existence entirely new ones. This claim depends on many factors, but a couple of the envisioned application scenarios shall be highlighted.

Apart from the need to build cheap, simple to program and network, potentially long lasting sensor nodes, a crucial and primary ingredient for developing actual applications is the actual sensing and actuating faculties with which a sensor node can be endowed. For many physical parameters, appropriate sensor technology exists that can be integrated in a node of a WSN. Some of the few popular ones are mentioned in section 1.2.1. But even more sophisticated sensing capabilities are conceive, for example, toys in kindergarten might have tactile or motion sensors or be able to determine their own speed or location[84].

Actuators controlled by a sensor node are perhaps not quite as multifaceted. Typically, they control a mechanical device like a servo like, or they might switch some electrical appliance by means of an electrical relay, like a lamp, a bullhorn, or similar device.

On the basis of nodes that have such sensing and/or actuation faculties, in combination with computation and communication abilities, many different kinds of application can be constructed, with very different types of nodes, even of different kinds within one application. A brief list of scenarios should make vast design space and the very different requirements of various applications evident. overviews of these and other applications are included in references [4, 6, 9, 10, 12, 14, 17, 23, 24, 35, 36, 41, 54, 62, 70, 84, 85, 87, 98].

- **Disaster relief applications** One of the most often mentioned application types for WSN are disaster relief operations. A typical scenario is wildfire detection: Sensor nodes are equipped with thermometers and can determine their own location relative to each other or in absolute coordinates). These sensors are deployed over a wildfire, for example, a forest, or from an airplane. They collectively produce a "temperature map" of the area or determine the perimeter of areas with high temperature that can be accessed from the outside, for example, by firefighters equipped with Persona Digital Assistants (PDAs). Similar scenarios are possible for the control of accidents in chemical factories.

  Some of these disaster relief applications have commonalities with military applications, where sensors should detect, for example, enemy troops rather than wildfires. In such an application, sensors should be cheap enough to be considered disposable since a large number is necessary; lifetime requirements are not particularly high.

- **Environment control and biodiversity mapping** WSNs can be used to control the environment, for example, with respect to chemical pollutants, a possible application is garbage dump sites. Another example is the surveillance of the marine ground floor; an understanding of its erosion process is important for he construction of offshore wind farms. Closely related to environmental control is the use of WSNs to gain understanding of the number of plant and animal species that live in a given habitat (biodiversity mapping).

The main advantages of WSNs here are the long-term, unattended wirefree operation of sensors close to the objects that have to be observed; since sensors can be made small enough to be unobtrusive, they negligibly disturb the observed animals and plants. Often, a large number of sensors is required with rather high requirements regarding lifetime.

- **Intelligent building** Building waste vast amounts of energy by inefficient Humidity, Ventilation, Air Conditioning (HVAC) usage. A better, real-time, high resolution monitoring of temperature, airflow, humidity, and other physical parameters in a building by means of a WSN can considerably increase the comfort level of inhabitants and reduce the energy consumption (potential saving of two quadrillion British thermal units in the US alone have been speculated about[70]). Improved energy efficiency as well as improved convenience are some goals of "intelligent buildings"[47], for which currently wired systems like BACnet, LonWorks, Or KNX are under development or are already deployed[83]; These standards also include the development of wireless components or have already incorporated them in the standard.

  In addition, such sensor nodes can be used to monitor mechanical stress levels of building in seismically active zones. By measuring mechanical parameters like the bending load of grinders, it is possible to quickly ascertain via a WSN whether is it still safe to enter a given building after an earthquake or whether is on the brink of the collapse - a considerable advantage for the rescue personnel. Similar systems can be applied to bridges. Other types of sensors might be geared toward detecting people enclosed in a collapsed building and communicating such information to a rescue team.

  The main advantage here is the collaborative mapping of physical parameters. Depending on the particular application, sensors can be retrofitted into existing buildings (for HVAC-type applications) or have to be incorporated into the building already under construction. If power supply is not available, lifetime requirements can be very high - up to several dozens of years

- but the number of required nodes, and hence the cost, is relatively modest, given the costs of an entire building.

- **Facility management** In the management of facilities larger than a single building, WSN also have a wide range of possible applications. Simple examples include keyless entry applications where people badges that allow a WSN to check which person is allowed to enter which areas of a larger company site. This example can be extended to the detection of intruders, for example vehicles that pass street outside of normal business hours. A widearea WSN could track such a vehicle's position and alert security personnel - this application shares many commonalities with corresponding military application. Along another line, a WSN could be used in a chemical plant to scan for leaking chemicals.

  These application combine challenging requirements as the required number of sensors can be large, they have to collaborate (e.g. in the tracking example), and they should be able to operate a long time on batteries.

- **Machine surveillance and preventive maintenance** One idea is to fix sensor nodes to difficult-to-reach areas of machinery where they can detect vibration patterns that indicate the need for maintenance. Examples for such machinery could be robotics or the axles of trains. Other applications in manufacturing are easily conceivable.

  The main advantage of WSNs here is the cable free operation, avoiding a maintenance problem in itself and allowing a cheap, often retrofitted installation of such sensors. Wired power supply may or may not be available depending on the scenario; if it is not available, sensors should last a long time on a finite supply of energy since exchanging batteries is usually impractical and costly. On the other hand, the size of nodes is often not a crucial issue, nor is the price very heavily constrained.

- **Precision agriculture** Applying a WSN to agriculture allows precise irrigation and fertilizing by placing humidity/soil composition sensors into the fields. A relatively small number is claimed to be sufficient, about one sensor per 100 m $\times$ 100 m area. Similarly, pest control can profit from a high resolution surveillance of farm land. Also livestock breeding can benefit from attaching a sensor to each pig or cow, which controls the health status of the animal (by checking body temperature, step counting, or similar means) and raises alarms if given thresholds are exceeded.

- **Medicine and health care** Along somewhat similar lines, the use of WSN in health care applications is a potentially very beneficial, but also ethically controversial, application. Possibilities range from postoperative and intensive care, where sensors are directly attached to patients - the advantage of doing away with cables is considerable here - to the long term surveillance of (typically elderly)patients and to automatic drug administration (embedding sensors into drug packaging, raising alarms when applied to the wrong patient, is conceivable). Also, patient and doctor tracking systems within hospitals can be literally life saving.

- **Logistics** In several different logistics applications, it is conceivable to equip goods (individual parcels, for examples) with simple sensors that allow a simple tracking of these objects during transportation or facilitate inventory tracking in stores or warehouses.

  In these applications, there is often no need for sensor node to actively communicate; passive readout of data is often sufficient, for example, when a suitcase is moved around on conveyor belts in an airport and passes certain checkpoints. Such passive readout is much simpler and cheaper than active communication and information processing concept discussed in the other examples; it is realized by so-called Radio Frequency Identifier (RFID) tags.

  On the other hand, a simple RFID tag can not support more advanced applications. It is very difficult to imagine how a passive system can be used to locate an item in a warehouse; it can also not easily store information about the history of its attached object - questions like "where has this parcel been?" are interesting in many applications but require same active participation of the sensor node [24, 41].

- **Telematics** Partially related to logistics applications are applications for telematics context, where sensors embedded in the streets or roadsides can gather information about traffic conditions at a much finer grained resolution that what is possible today[31]. Such a so-called "intelligent roadside" could also interact with the cars to exchange danger warnings about road conditions or traffic jams ahead.

In addition to these, other application types for WSNs that have been mentioned in the literature include airplane wings and support for smart spaces[23], applications in waste water treatment plants[36], instrumentation of semiconductor processing chambers and wind tunnels [41], in "smart kindergartens" where toys interact with children [84], the detection of the floods [9], inter-active museums[70], monitoring a bird habitat on a remote island [54], and implanting sensors into the human body (for glucose monitoring or as retina prosthesis)[78].

While most of these applications are, in some form or another, possible even with today's technologies and without wireless sensor networks, all current solutions are "sensor starved" [70]. Most applications would work much better with informational higher spatial and temporal resolution about their object of concern than can be provided with traditional sensor technology. Wireless sensor networks are to large extent about providing the required informational the required accuracy in time with as little resource consumption as possible.

1.3.2 Types of Application

Many of these applications share some basic characteristics. In most of them, there is a clear different between **sources** of data - the actual node that sends data - and **sinks** -nodes where the data should be delivered to. These sinks sometimes are part of sensor network itself; sometimes they are clearly systems "outside" the network(e.g. the firefighter's PDA communicating with a WSN). Also, there are usually, but not always, more sources than sinks and the sink is obvious or not interested in the identity of the sources; the data itself is much more important.

The interaction pattern between sources and sinks show some typical patterns. the most relevant ones are:

- **Event detection** Sensor nodes should report to the sink(s) once they have detected the occurrence of a specified event. The simplest events can be detected locally by a single sensor node in isolation(e.g. a temperature threshold is exceeded); more complicated types of events require the collaboration of nearby or even remote sensors to decide whether a (composite) event has occurred(e.g. a temperature gradient becomes too steep). If several different events can occur, event classification might be an additional issue.

- **Periodic measurements** Sensors can be tasked with periodically reporting measured values. Often these reports can be triggered by a detected event; the reporting period is application dependent.

- **Function approximation and edge detection** The way a physical value like temperature changes from one place to another can be regarded as a function of location. A WSN can be used to approximate this unknown function (to extract its spatial characteristics), using a limited number of samples taken at each individual sensor node. This approximate mapping should be made available at the sink. How and when to update this mapping depends on the application's needs, as do the approximation accuracy and the inherent trade-off against energy consumption.

  Similarly, a relevant problem can be finding areas or points of the same given value. An example is to find the isothermal points in the forest fire application to detect the border of the actual fire. This can be generalized to finding "edges" in such functions or to sending messages along the boundaries of patterns in both space and/or time[28].

- **Tracking** The source of an event can be mobile (e.g. an intruder in surveillance scenarios). The WSN can be used to report updates on the event source's position to the sink(s), potentially with estimates about speed and direction as well. To do so, typically sensor nodes have to cooperate before updates can be reported to the sink.

These interactions can be scoped both in time and in space (reporting events only within a given time span, only from certain areas, and so on). These requirements can also change dynamically overtime; sinks have to have a means to inform the sensors of their requirements at runtime. Moreover, these interactions can take place only for one specific request of a sink (so-called "one-shot queries"), or they could be long-lasting relationship between many sensors and many sinks.

The example also has shown a wide diversity in deployment options. They range from well planned, fixed deployment of sensor nodes (e.g. machinery maintenance applications) to random deployment by dropping a large number of nodes from an aircraft over a forest fire. In addition, sensor nodes can be mobile themselves and compensate for shortcoming in the deployment process by moving, in a post deployment phase, to positions such that their sensing tasks can be better fulfilled [4]. They could also be mobile because they are attached to other objects (in the logistics applications, for example) and the network has to adapt itself to the location of nodes.

The applications also influence the available maintenance options. Is it feasible and practical to perform maintenance on such sensors - perhaps even required in the course of maintenance on associated machinery? Is maintenance irrelevant because theses networks are only deployed in a strictly ad hoc, short-term manner with a clear delimitation of maximum mission time (like in disaster recovery operations)? Or do these sensors have to function unattended, for a long time, with no possibility for maintenance?

Closely related maintenance options are the options for energy supply. In some applications, wired power supply is possible and the question is mute. For self-sustained sensor nodes, depending on the required mission time, energy supply can be trivial (application with a few days usage only) or a challenging research problem, especially when no maintenance is possible but nodes have to work for years. Obviously, acceptable price and size per node play a crucial role in designing energy supply.

1.4 Challenges for Wireless Sensor Networks

Handling such a wide range of application types will hardly be possible with any single realization of a WSN. Nonetheless, certain common traits appear, especially with respect to the characteristics and the required mechanisms of such systems. Realizing these characteristics with new mechanisms is the major challenge of the vision of wireless sensor network.

1.4.1 Characteristics Requirements

The following characteristics are shared among most of the application examples discussed above:

- **Type of service** The service type rendered by a conventional communication network is evident - it moves bits from on place to another. For a WSN, moving bits is only a means to an end, but not the actual purpose. Rather, a WSN is expected to provide a meaningful information and/or actions about a given task. Additionally, concepts like scoping of interactions to specific geographic regions or to time intervals will become important. Hence, new paradigms of using such a network are required, along with new interfaces and new ways of thinking about the service of network.

- **Quality of Service** Closely related to the type of a network's service is the quality of that service. Traditional quality of service requirements- usually coming from multimedia-type applications - like bounded delay or minimum bandwidth are irrelevant when applications are tolerant to latency[6] or the bandwidth of the transimtted data is very small in the first place, in some cases only occasional delivery of a packet can be more than enough; in other cases, very high reliability requirement exist. In yet other cases, delay is important when actuators are to be controlled in a real-time fashion by the sensor network. The packet delivery ratio is an insufficient metric; what is relevant is the amount and quality of information that extracted at given sinks about the observed objects or area.

  Therefore, adapted quality concepts like reliable detection of events or the approximation quality of a, say, temperature map is important.

- **Fault tolerance** Since nodes may run out of energy or may be damaged, or the wireless communication between two nodes may be permanently interrupted, it is important that the WSN as a whole is able to tolerate such faults. To tolerant node failure, redundant deployment is necessary, using more nodes than would be strictly necessary if all nodes functioned correctly

- **Lifetime** In many scenarios, nodes will have to rely on a limited supply of energy (using batteries). Replacing these energy sources in the field is usually not practical, and simultaneously, a WSN must operate at least for a given mission time or as long as possible. Hence, the **lifetime** of a WSN becomes a very important figure of a merit. Evidently, an energy-efficient way of operation of the WSN is necessary.

  As an alternative or supplement to energy supplies, a limited power source (via power sources like solar cells, for example recharging of batteries.) might also be available on a sensor node. Under such conditions, the lifetime of the network should ideally be infinite.

  The lifetime of a network also has direct trade-offs against quality of service: investigating more energy can increase quality but decrease lifetime. Concepts to harmonize these trade-offs are required.

  The precise definition of lifetime depends on the application at hand. A simple option is to use the time until the first node fails (or runs out of energy) as the network lifetime. Other options include the time until the network is disconnected in two or more partitions, the time until 50% (of some other fixed ratio) of nodes have failed, or the time when for the first time a point in the observed region is no longer covered by at least a single sensor node (when using redundant deployment, it is possible and beneficial to have each point in space covered by several nodes initially).

- **Scalability** Since a WSN might include a large number of nodes , the emplyed architectures and protocols must be scaled to these numbers.

- **Wide range of densities** In a WSN, the number of nodes per unit area - the density of the network can vary considerably. Different applications will have very different node densities. Even within a given application, density can vary over time and space because nodes fail or move. The density also does not have to homogeneous in the entire network (because of imperfect deployment, for example) and the network should adapt to such variations.

- **Programmability** Not only will it be necessary for the nodes to process information, but also they will have to react flexibly on changes in their tasks. These nodes should be programmable, and their programming must be changeable during operation when new tasks become important. A fixed way of information processing is insufficient.

- **Maintainability** As both the environment of a WSN and the WSN itself change (depleted batteries, failing nodes, new tasks), the system has to adapt. It has to monitor its own health and status to change operational parameters or to choose different trade-offs (e.g. to provide lower quality when energy resources become scarce). In these sense, the network has to maintain itself; it could also be able to interact with external maintenance mechanisms to ensure its extended operation at a required quality [54].

1.4.2 Required Mechanisms

To realize these requirements, innovative mechanisms for a communication network have to be found, as well as new architecture, and protocol concepts. A particular challenge here is the need to find mechanisms that are sufficiently specific to the given application to support the quality of service, lifetime, and maintainability requirement [24]. On the other hand, these mechanisms also have to generalize to a wider range of applications, otherwise, a complete from-scratch development and implementation of a WSN becomes necessary for every individual application - this would likely render WSNs as technological concept economically infeasible. Some of the mechanisms that will form typical parts of WSNs are:

- **Multihop wireless communication** While wireless communication will be a core technique, a direct communication between a sender and receiver is faced with limitations. In particular, communication over long distances is only possible using prohibitively high transmission power. The use of intermediate nodes as relays can reduce the total required power. Hence, for many forms of WSNs, so called multihop communication will be a necessary ingredient.

- **Energy-efficient operation** To support long lifetimes, energy-efficient operation is a key technique. Options to look into include energy-efficient data transport between two nodes (measured in J/bit) or, more importantly, the energy efficient determination of a requested information, Also, nonhomogeneous energy consumption- the formation of "hotspots" - is an issue.

- **Auto-configuration** A WSN will have to configure most of its operational parameters autonomously, independent of external configuration - the sheer number of nodes and simplified deployment will require that capability in most application, As an example, nodes should be able to determine their geographical positions only using other nodes of network - so-called "self-location". Also, the network should be able to tolerate failing nodes (because of a depleted battery, for example) or to integrate new nodes) because of incremental deployment after failure, for example).

- **Collaboration and in-network processing** In some application, a single sensor is not able to decide whether an event has happened but several sensors have to collaborate to detect an event and only the joint data of many sensors provide enough information. Information is processed in network itself in various forms to achieve this collaboration, as opposed to having every node transmit all data to an external network and process it "at the edge" of the network.

  An example is to determine the highest or the average temperature within an area and to report that value to a sink. To solve such tasks efficiently, readings from individual sensors can be aggregated as they propagate through the network, reducing the amount of data to be transmitted and hence improving the energy efficiency. How to perform such aggregation is an open question.

- **Data centric** Traditional communication networks are typically centered on the transfer of data between two specific devices, each equipped with at least one network address - the operation of such network is thus **address centric**. In a WSN, where nodes are typically deployed redundantly to protect against node failures or to compensate for the low quality of a single node's actual sensing equipment, the identity of the particular node supplying data becomes irrelevant. The answers and values themselves are important, not which node has provided them. Hence, switching from an address-centric paradigm to **data-centric** paradigm in designing architecture and communication protocols is promising.

  An example for such a data-centric interaction sould be to request the average temperature in a given location area, as opposed to requiring temperature readings from individual nodes. Such a data-centric paradigm can also be used to set conditions for alerts or events("raise an alarm if temperature exceeds a threshold"). In this sence , the data- centric approach is closely related to query concepts known from databases it also combines well with collaboration, in-network processing, and aggregation

- **Locality** Rather a design guideline than a proper mechanism, the principle of the locality will have to be embraced extensively to ensure, in particular, scalability. Nodes, which are very limited in resources like memory, should attempt to limit the state that they accumulate during protocol processing to only information about their direct neighbors. The hope is that this will allow the network to scale to large numbers of nodes without having to rely on powerful processing at each single node. How to combine the locality principle with efficient protocol designs is still and open research topic, however.

- **Exploit trade-offs** Similar to locality principles, WSNs will have to rely to large degrees on exploiting various inherent trade-offs between mutually contradictory goals, both during system/protocol design at runtime. Examples for such trade-offs have been mentioned already: higher energy expenditure allows higher result accuracy, or a longer lifetime of the entire network trades off against lifetime of individual nodes. Another important trade-off is node density: depending on the application, deployment, and node failures at runtime, the density of the network can change considerably - the protocols will have to handle very different situations, possibly present at different places of a single network. Again, not all the research questions are solved here.

1.5 Focus of Research

In many cases, sensor nodes must be left unattended e.g., in hostile environments, which makes it difficult or impossible to re-charge or replace their batteries. This requires devising novel energy-efficient solutions to some of the conventional wireless networking problems, such as medium access control, routing, self-organization, bandwidth sharing, and security. Exploiting the among energy, accuracy, and latency, and using hierarchical (tiered) architectures are important techniques for prolonging network lifetime [97, 43, 23].

Network lifetime can be defined in different ways such as the time elapsed until the first node or the last node in the network depletes its energy (dies). For example, in a military field where sensors are monitoring chemical activity, the lifetime of a sensor is critical for maximum field coverage. In this work, we define network lifetime as the number of rounds for which at least 20% of the nodes remain alive. Energy consumption in a sensor node can be either "useful" or "wasteful". Useful energy consumption can be due to (i) transmitting/receiving data, (ii) processing query requests, and (iii) forwarding queries/data to neighboring nodes. Wasteful energy consumption can be due to (i) idle listening to the media, (ii) retransmitting due to packet collisions, (iii) overhearing, and (iv) generating/handling control packets. Several MAC protocols attempt to reduce energy consumption due to wasteful sources, e.g.,[99, 40, 32, 91]. A number of protocols have also been proposed to reduce useful energy consumption. These protocols can be mainly classified into three classes. Protocols in the first class control the transmission power level at each node to increase network capacity while keeping the network connected [46, 74]. Protocols in the second class make routing decisions based on power optimization goals, e.g., [90, 13, 38, 18, 96]. Protocols in the third class control the network topology by determining which nodes should participate in the network operation (be awake) and which should not (remain asleep) [15, 95, 7]. Nodes in this case, however, require knowledge of their locations via GPS-capable antennae or via message exchanges.

Hierarchical (clustering) techniques can aid in reducing useful energy consumption [90]. Clustering is particularly useful for applications that require scalability to hundreds or thousands of nodes. Scalability in this context implies the need for load balancing and efficient resource utilization. Applications requiring efficient data aggregation (e.g., computing the maximum detected radiation around an object) are natural candidates for clustering. Routing protocols can also employ clustering [50, 8]. In [24], clustering was proposed as a useful tool for efficiently pinpointing object locations. Clustering can be extremely effective in one-to-many, many-to- one, one-to-any, or one-to-all (broadcast) communication. For example, in many-to-one communication, clustering can support data fusion and reduce communication interference.

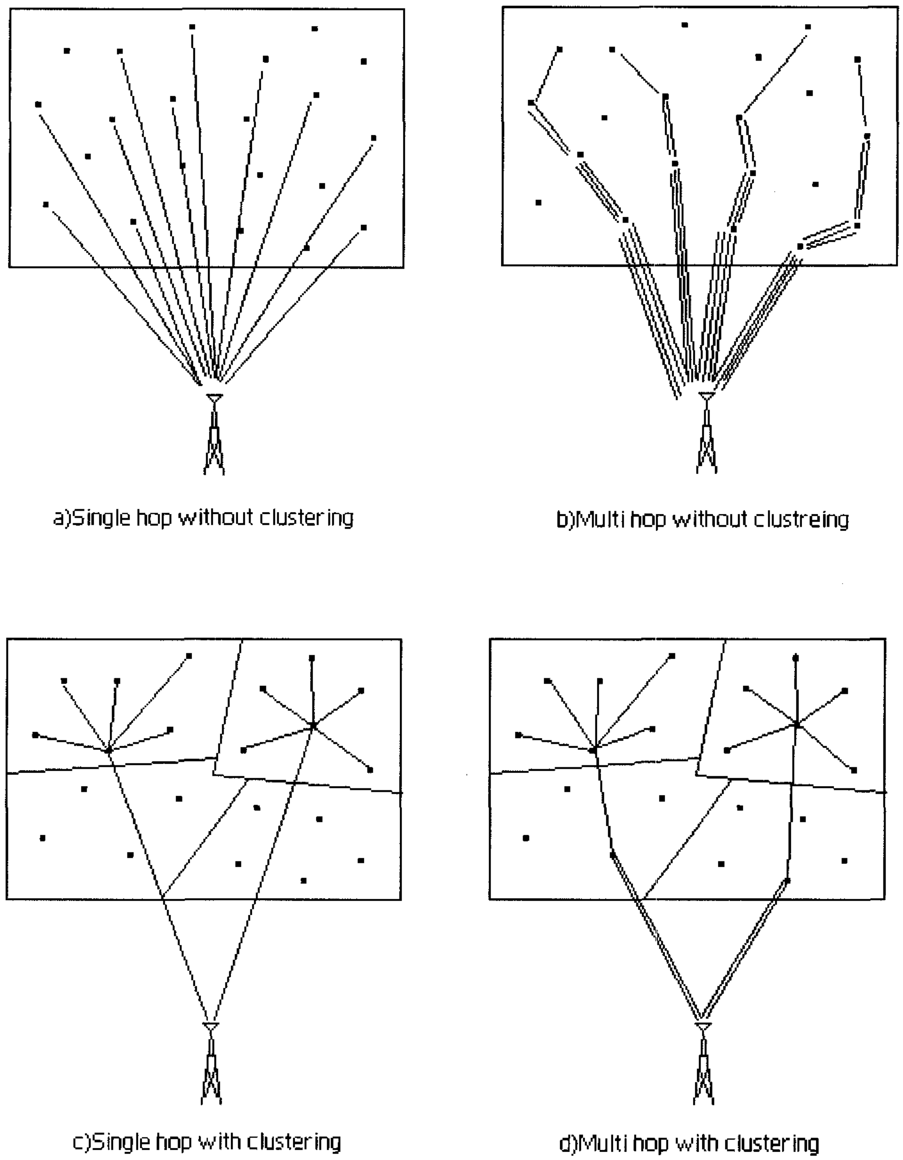

Figure 1: Clustering Effect

The essential operation in sensor node clustering is to select a set of cluster heads among the nodes in the network, and cluster the rest of the nodes with these heads. Cluster heads are responsible for coordination among the nodes within their clusters (intra-cluster coordination), and communication with each other and/or with external observers on behalf of their clusters (inter-cluster communication). Figure 1 depicts an application where sensors periodically transmit information to a remote observer (base station). The figure illustrates how clustering can reduce the communication overhead for both single-hop and multi-hop networks. With clustering, nodes transmit their information to their cluster heads. A cluster head aggregates the received information and forwards it over to the observer. Periodic re-clustering can select nodes with higher residual energy to act as cluster heads. Network lifetime is prolonged through (i) reducing the number of nodes contending for channel access, (ii) summarizing network state information and updates at the cluster heads through intra-cluster coordination, and (iii) routing through an overlay among cluster heads, which has a relatively small network diameter.

Many protocols proposed in the literature minimize energy consumption on routing paths. While these approaches increase energy efficiency, they do not necessarily prolong network lifetime if certain nodes are popular, i.e., present on most forwarding paths in the network. Even if dynamic routing (in which data is forwarded to nodes with the highest residual energy) is used, it may cause such problems as unbounded delay and routing loops. With clustering, a popular node is guaranteed to lose its popularity (no longer serve as cluster head) after a fixed interval of time. Of course, node popularity due to interest in the data it provides can only be reduced by deploying several redundant nodes, and rotating among them (e.g., [15]).

In wireless sensor networks, the main source of draining energy is radio transmission. Transmitting energy loss for long distances (e.g., greater than 88m) is at the order of of d^4. When the covered area is big, the distance between cluster heads and base station will increase. Therefore, none of the protocols mentioned above are suitable for a large network that covers a big area. Moreover, Finding evenly distributed optimum clusters in the network is a NP-hard problem[3], therefore, fast and good solution which leads to prolonging network lifetime is desirable. Motivated by this, we propose the EEMAC and SAD protocol that uses Cluster Head to Cluster Head transmission to the

Base Station.

1.6 Organization of the Thesis

The Thesis is organized as following. In Chapter 2 describes the network model and states the problem that we address in this work and briefly surveys the related work. In Chapter 3, we will give details on how EEMAC and SAD work. In chapter 4, we will theoretically analyze the protocols and give guidelines on how to find the optimal number of clusters and Spiral Factors. Chapter 5 shows their effectiveness via simulations and compares them to other clustering techniques. Finally, in Chapter 6, we will conclude this thesis and also discuss about future work.

Chapter 2

Problem Statement and Related Work

2.1 Problem Statement

Let the clustering process interval, T_C, be the time taken by the clustering protocol to cluster the network. Let the network operation interval, T_O, be the time between the end of a T_C interval and the start of the subsequent T_C interval. We must ensure that $T_O >> T_C$ to reduce overhead.

2.1.1 Network Model

Consider a set of sensors is dispersed on a circular field with radius of R. We assume the following properties about the network:

1. The nodes in the network are uniformly distributed and quasi-stationary.
2. The network serves one or more mobile/stationary base stations, which implies that energy consumption is not uniform for all nodes.
3. Nodes are location-aware, i.e. equipped with GPS-capable antennae or using message exchange

method [7].

4. All nodes have similar capabilities (processing/ communication), and equal significance.

5. Nodes are left unattended after deployment.

6. Each node can adjust its power level according to the distance to the receiver.

7. Sensors sense the environment at a fixed rate and always have data to send to the base station.

Our first assumption about mobility is typical for sensor networks. Clustering can still be performed, however, if only nodes that announce their willingness to be cluster heads are quasi-stationary during the T_C interval in which they are selected, and the ensuing T_O interval. Nodes that travel rapidly in the network may degrade the cluster quality, because they alter the node distribution in their cluster. The second network property motivates the requirement for re-clustering to select new cluster heads and re-distribute energy consumption. The third property is easily accessible by new cheap GPS technology and even in the cases that GPS is not an option we can use message exchange between nodes to determine the location of nodes [7] . The fourth and fifth properties of the network motivate the need for prolonging network lifetime and balancing cluster head loads. Regarding sixth assumption that each node adjusts its transmit power, there will be few overlapping transmissions and little spreading of the data is actually needed to ensure a low probability of collision. For the network, we use a model where nodes always have data to send to the end user and nodes located close to each other have correlated data. Although our work including EEMAC and SAD is optimized for this situation, they will continue to work if it were not true.

2.1.2 Radio Model

We assume a simple model for the radio hardware energy dissipation model[89, 90, 71] where the transmitter dissipates energy to run the radio electronics and the power amplifier, and the receiver dissipates energy to run the radio electronics, as shown in Figure 2. Both the free space (d^2 power

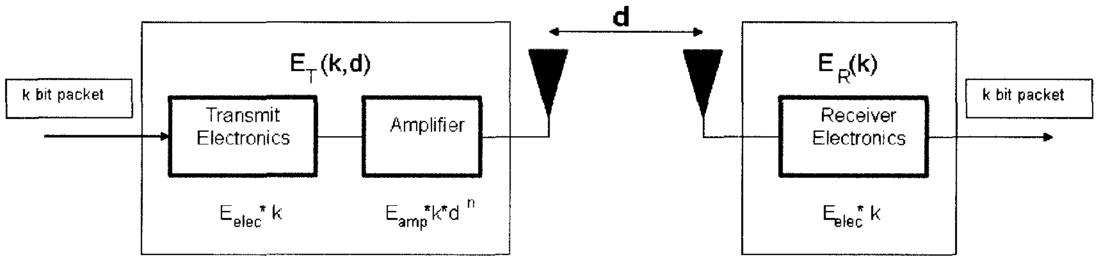

Figure 2: Radio Energy Dissipation Model

loss) and the multi-path fading (d^4 power loss) channel models can be used in our analysis, depending on the distance between the transmitter and receiver.

Thus, to transmit an l-bit message, we need energy

$$E_T(l,d) = E_{TX}l + E_{amp}(d)l \tag{1}$$

and to receive l-bit message, we have:

$$E_R(l,d) = E_{RX}l = lE_{elec} \tag{2}$$

where E_{TX} and E_{RX} are constants with equal values E_{elec}, i.e.,

$$E_{TX} = E_{RX} = E_{elec}$$

by defining d_0 as the threshold of free space and multi path fading channels, for 1 we will have:

$$E_T(l,d) = \begin{cases} lE_{elec} + lE_{fs}d^2 & \text{if } d < d_0 \\ lE_{elec} + lE_{mp}d^4 & \text{if } d \geq d_0 \end{cases} \tag{3}$$

where

$$d_0 = \sqrt{\frac{E_{fs}}{E_{mp}}}$$

2.1.3 The Clustering and Routing Problem

Assume that N nodes are dispersed in a field and the above assumptions hold. Our goal is to identify a set of cluster heads which cover the entire field. Each node n_i, where $1 \leq i \leq N$, is then mapped to exactly one cluster C_j, where $1 \leq j \leq K_{opt}$, and K_{opt} is the optimum number of clusters ($K_{opt} << N$). The node can directly communicate with its cluster head (via a single hop). The following desired properties must be met:

- Clustering terminates within a fixed number of iterations (regardless of network diameter).
- At the end of each T_C, each node is either a cluster head, or a non-cluster head node (which we refer to as regular node) that belongs to exactly one cluster.
- At the end of each T_C, each cluster head is aware of its next hop that finally leads to the base station.
- Clustering should be efficient in terms of processing complexity and message exchange.
- Cluster heads are well-distributed over the sensor field.
- Clustering can be distributed. Each node independently makes its decisions based on local information. This property is optional ,since distributed clustering is not the optimum solution.

2.1.4 Observation Ratio Factor

In this work we introduce a new factor, Observation Ratio(R_O), which is necessary to compare performance of different protocols in the area of wireless sensor network. Specially in monitoring applications, we need to have a minimum amount of data from monitored area per timeunit. With clustering approach, since data is aggregated in cluster heads, we receive limited data from specific cluster head and its covered area in each round, even if we had dense sensors. Assume two different number of clusters K_1 and K_2 when $K_1 > K_2$. By using these values for the same network, we will have a bigger covered area for each cluster in case of using K_2 comparing to the case of using K_1.

Consequently, since we receive same number of bits from each cluster head in each round, received data of K_2 cluster heads is less than K_1 cluster heads from the sensing field. By defining observation ratio $R_O = r_o$ bit/m^2/sec we can easily calculate the minimum number of cluster heads as a function of whole sensing area's size.r_o may differs in different application.

2.2 Related Work

LEACH protocol: Heinzelman, et al. [89] introduced a hierarchical clustering algorithm for sensor networks, called Low Energy Adaptive Clustering Hierarchy (LEACH). LEACH is a cluster-based protocol, which includes distributed cluster formation. LEACH randomly selects a few sensor nodes as cluster heads (CHs) and rotates this role to evenly distribute the energy load among the sensors in the network. In LEACH, the CH nodes compress data arriving from nodes that belong to the respective cluster, and send an aggregated packet to the BS in order to reduce the amount of information that must be transmitted to the BS. LEACH uses a TDMA/code-division multiple access (CDMA) MAC to reduce intercluster and intracluster collisions. However, data collection is centralized and performed periodically. Therefore, this protocol is most appropriate when there is a need for constant monitoring by the sensor network. A user may not need all the data immediately. Hence, periodic data transmissions are unnecessary, and may drain the limited energy of the sensor nodes. After a given interval of time, randomized rotation of the role of CH is conducted so that uniform energy dissipation in the sensor network is obtained. The authors found, based on their simulation model, that only 5 percent of the nodes need to act as CHs.

The operation of LEACH is separated into two phases, the setup phase and the steady state phase. In the setup phase, the clusters are organized and CHs are selected. In the steady state phase, the actual data transfer to the BS takes place. The duration of the steady state phase is longer than the duration of the setup phase in order to minimize overhead. During the setup phase, a predetermined fraction of nodes, p, elect themselves as CHs as follows. A sensor node chooses a random number, r, between 0 and 1. If this random number is less than a threshold value, $T(n)$, the node becomes a CH for the current round. The threshold value is calculated based on an equation that incorporates the desired percentage to become a CH, the current round, and the set of nodes that have not been selected as a CH in the last $\frac{1}{P}$) rounds, denoted G. It is given by

$$T(n) = \frac{p}{1 - p(r \bmod(\frac{1}{p}))}$$

where G is the set of nodes that are involved in the CH election. All elected CHs broadcast an advertisement message to the rest of the nodes in the network that they are the new CHs. All the non-CH nodes, after receiving this advertisement, decide on the cluster to which they want to belong. This decision is based on the signal strength of the advertisement. The non-CH nodes inform the appropriate CHs that they will be a member of the cluster. After receiving all the messages from the nodes that would like to be included in the cluster and based on the number of nodes in the cluster, the CH node creates a TDMA schedule and assigns each node a time slot when it can transmit. This schedule is broadcast to all the nodes in the cluster.

During the steady state phase, the sensor nodes can begin sensing and transmitting data to the CHs. The CH node, after receiving all the data, aggregates it before sending it to the BS. After a certain time, which is determined a priori, the network goes back into the setup phase again and enters another round of selecting new CHs. Each cluster communicates using different CDMA codes to reduce interference from nodes belonging to other clusters.

Although LEACH is able to increase the network lifetime, there are still a number of issues about the assumptions used in this protocol. LEACH assumes that all nodes can transmit with enough power to reach the BS if needed and that each node has computational power to support different MAC protocols. Therefore, it is not applicable to networks deployed in large regions. It also assumes that nodes always have data to send, and nodes located close to each other have correlated data. It is not obvious how the number of predetermined CHs (p) is going to be uniformly distributed through the network. Therefore, there is the possibility that the elected CHs will be concentrated in one part of the network; hence, some nodes will not have any CHs in their vicinity. Furthermore, the idea of dynamic clustering brings extra overhead (head changes, advertisements, etc.), which may diminish the gain in energy consumption. Finally, the protocol assumes that all nodes begin with the same amount of energy capacity in each election round, assuming that being a CH consumes approximately the same amount of energy for each node. The protocol should be extended to account for non-uniform energy nodes (i.e., use an energy-based threshold). An extension to LEACH, LEACH with negotiation or, was proposed in [89]. The main theme of the proposed extension is to precede data transfers with high level negotiation using meta-data descriptors as in the SPIN protocol discussed earlier. This ensures that only data that provides new information is transmitted to the CHs before being transmitted to the BS.

A centralized version of LEACH, LEACH-C, is proposed in [90]. Unlike LEACH, where nodes self-configure themselves into clusters, LEACH-C utilizes the base station for cluster formation. During the setup phase of LEACH-C, the base station receives information regarding the location and energy level of each node in the network. Using this information, the base station finds a pre-determined number of cluster heads and configures the network into clusters. The cluster groupings are chosen to minimize the energy required for non-cluster- head nodes to transmit their data to their respective cluster heads. Although the other operations of LEACH-C are identical to those of LEACH, results presented in [90] indicate a definite improvement over LEACH. The authors of [90] cite two key reasons for the improvement: (1) The base station utilizes its global knowledge of the network to produce better clusters that require less energy for data transmission. (2) The number of cluster heads in each round of LEACH-C equals a predetermined optimal value, whereas for LEACH the number of cluster heads varies from round to round due to the lack of global coordination among nodes.

- The base station utilizes its global knowledge of the network to produce better clusters that require less energy for data transmission.
- The number of cluster heads in each round of LEACH-C equals a predetermined optimal value, whereas for LEACH the number of cluster heads varies from round to round due to the lack of global coordination among nodes.

Power-Efficient Gathering in Sensor Information Systems: In [51], an enhancement over the LEACH protocol was proposed. The protocol, called Power-Efficient Gathering in Sensor Information Systems (PEGASIS), is a near optimal chain-based protocol. The basic idea of the protocol is that in order to extend network lifetime, nodes need only communicate with their closest neighbors, and they take turns in communicating with the BS. When the round of all nodes communicating with the BS ends, a new round starts, and so on. This reduces the power required to transmit data per round as the power draining is spread uniformly over all nodes. Hence, PEGASIS has two main objectives. First, increase the lifetime of each node by using collaborative techniques. Second, allow only local coordination between nodes that are close together so that the bandwidth consumed in communication is reduced. Unlike LEACH, PEGASIS avoids cluster formation and uses only one node in a chain to transmit to the BS instead of multiple nodes.

To locate the closest neighbor node in PEGASIS, each node uses the signal strength to measure the distance to all neighboring nodes and then adjusts the signal strength so that only one node can be heard. The chain in PEGASIS will consist of those nodes that are closest to each other and form a path to the BS. The aggregated form of the data will be sent to the BS by any node in the chain, and the nodes in the chain will take turns sending to the BS. The chain construction is performed in a greedy fashion. Simulation results showed that PEGASIS is able to increase the lifetime of the network to twice that under the LEACH protocol. Such performance gain is achieved through the elimination of the overhead caused by dynamic cluster formation in LEACH, and decreasing the number of transmissions and reception by using data aggregation. Although the clustering overhead is avoided, PEGASIS still requires dynamic topology adjustment since a sensor node needs to know about the energy status of its neighbors in order to know where to route its data. Such topology adjustment can introduce significant overhead, especially for highly utilized networks. Moreover, PEGASIS assumes that each sensor node is able to communicate with the BS directly. In practical cases, sensor nodes use multihop communication to reach the BS. Also, PEGASIS assumes that all nodes maintain a complete database of the location of all other nodes in the network. The method by which the node locations are obtained is not outlined. In addition, PEGASIS assumes that all sensor nodes have the same level of energy and are likely to die at the same time. Note also that PEGASIS introduces excessive delay for distant nodes on the chain. In addition, the single leader can become a bottleneck. Finally, although in most scenarios sensors will be fixed or immobile as assumed in PEGASIS, some sensors may be allowed to move and hence affect the protocol functionality.

An extension to PEGASIS, called Hierarchical PEGASIS, was introduced in [77] with the objective of decreasing the delay incurred for packets during transmission to the BS. For this purpose, simultaneous transmissions of data are studied in order to avoid collisions through approaches that incorporate signal coding and spatial transmissions. In the latter, only spatially separated nodes are allowed to transmit at the same time. The chain-based protocol with CDMA-capable nodes constructs a chain of nodes that forms a tree-like hierarchy, and each selected node at a particular level transmits data to a node in the upper level of the hierarchy. This method ensures data transmitting in parallel and reduces delay significantly. Such a hierarchical extension has been shown to perform better than the regular PEGASIS scheme by a factor of about 60.

Another clustering-based routing protocol called Base Station Controlled Dynamic Clustering Protocol (BCDCP)[57], which utilizes a high-energy base station to set up clusters and routing paths, perform randomized rotation of cluster heads, and carry out other energy-intensive tasks. The key ideas in BCDCP are the formation of balanced clusters where each cluster head serves an approximately equal number of member nodes to avoid cluster head overload, uniform placement of cluster heads throughout the whole sensor field, and utilization of cluster-head-to-cluster head (CH-to-CH) routing to transfer the data to the base station.

Chapter 3

Protocols

The application that typical sensor networks support is the monitoring of a remote environment. Since individual node's data are often correlated in a sensor network, the end user does not require all the (redundant) data; rather, the end user needs a high-level function of the data that describes the events occurring in the environment. Because the correlation is strongest between data signals from nodes located close to each other, we chose to use a clustering infrastructure as the basis for EEMAC[71] and SAD[72].

3.1 EEMAC

EE-MAC is a cluster-based protocol and the sensor nodes are geographically grouped into clusters and capable of operating in two basic modes: the cluster head mode and the sensing mode.

3.1.1 Cluster Formation and Setup Phase

The main activities in this phase are cluster setup, cluster head selection, CH-to-CH routing path formation, and schedule creation for each cluster. During each setup phase, the base station receives information of the current energy status from all the nodes in the network. Based on this feedback, the base station first computes the average energy level of all the nodes, and then chooses a set of nodes, denoted S, whose energy levels are above the average value. Cluster heads for the current round will be chosen from the set S, which ensures that only nodes with sufficient energy get selected as cluster heads, while those with low energy can prolong their lifetime by performing tasks that require low energy costs. The next tasks for the base station are then:

- To identify K cluster head nodes from the chosen set S.
- To group other nodes into clusters such that the overall energy consumption during the data communication phase is minimized.

In EE-MAC, we accomplish these tasks by means of an iterative cluster splitting algorithm [57]. This simple algorithm first splits the network into two subclusters, and proceeds further by splitting the subclusters into smaller clusters. The base station repeats the cluster splitting process until the desired number of clusters K is attained. The iterative cluster splitting algorithm ensures that the selected cluster heads are uniformly placed throughout the whole sensor field by maximizing the distance between cluster heads in each splitting step. Furthermore, in order to evenly distribute the load on all cluster heads, we utilize the balanced clustering technique [27] where each cluster is split so that the resulting subclusters have approximately the same number of sensor nodes. Accordingly, a single iteration of the cluster splitting algorithm consists of the following four steps:

1. From the nodes in the current cluster that are eligible to become cluster heads, choose two nodes, s_1 and s_2, that have the maximum separation distance.
2. Group each of the remaining nodes in the current cluster with either s_1 or s_2, whichever is closest.
3. Balance the two groups so that they have approximately the same number of nodes and forms two subclusters.
4. For each of the subcluster, perform the iterative splitting algorithm.

The second major activity in the setup phase is the formation of routing paths. As discussed earlier, the EE-MAC protocol uses a CH to CH multi-hop routing scheme to transfer the sensed

CH SCID	Time slot 1	Time slot 2	Time slot 3
00	01	10	11
01	00	10	11
10	00	01	11
11	00	01	10

Table 1: Scheduling for a cluster with four nodes

data to the base station. Once the clusters and the cluster head nodes have been identified, the base station chooses the lowest-energy routing path and forwards this info to the sensor nodes along with the details on cluster groupings and selected cluster heads. The routing paths are selected by connecting all the cluster head nodes using a minimum spanning tree [81] that minimizes the energy consumption for each cluster head.

Schedule creation is the last major activity in the setup phase. The EE-MAC protocol utilizes a time-division multiple access (TDMA) scheduling scheme for intra-cluster data transmission. Similar to [57], after cluster formation, the base station assigns an interim schedule creation ID (SCID) for all the nodes within a cluster. Table 1 demonstrates how this SCID is utilized for scheduling sensor node transmissions to the cluster head for a sample cluster with four nodes. As shown in Table 3.1.1, the sensor nodes use the 2-bit SCID to schedule themselves to the appropriate TDMA time slot. For instance, when SCID 00 is assigned to the cluster head, nodes with SCID 01, 10, and 11 transmit their data in time slots 1, 2, and 3, respectively. When the cluster head has an SCID of 01, the transmit order changes slightly, with node 00 transmitting at time slot 1, node 10 at time slot 2, and node 11 at time slot 3, and so on. In general, for a cluster with M nodes, an m-bit schedule creation scheme is used where m represents the smallest integer value greater than or equal to $\log_2 M$.

3.1.2 Steady State Phase

This phase consists of three major activities:

- Data gathering
- Data fusion

- Data routing

Using the TDMA schedule creation scheme described above, each sensor node transmits the sensed information to its cluster head. Since sensor nodes are geographically grouped into clusters, these transmissions consume minimal energy due to small spatial separations between the cluster head and the sensing nodes. Once data from all sensor nodes have been received, the cluster head performs data fusion on the collected data, and reduces the amount of raw data that needs to be sent to the base station. The aggregated data, along with the information required by the base station to properly identify and decode the cluster data, are then routed back to the base station via the CH-to-CH routing path created by the base station. We also assume that the fused data from a given cluster head may undergo further processing as it hops along the CH-to-CH routing path. Another key issue that needs to be addressed here is the radio interference caused by neighboring clusters that could hinder the operation of a given cluster [51]. EEMAC utilizes code-division multiple access (CDMA) to counteract this problem. Each cluster is assigned a spreading code that the nodes in the cluster use to distinguish their data transmissions from those of nodes in neighboring clusters. Once the data gathering process is completed, the cluster head uses the same spreading code assigned to the cluster to route data back to the base station.

3.1.3 Variations of EEMAC

- **Multi-hop EEMAC** This version utilizes all of EEMAC features mentioned above, including multi-hop data transmission between CHs and balance clusters.

- **One-hop EEMAC** When the network size is small , there is no need to transmit the data by multi-hop algorithm,since transmitting over long distances is already avoided .Therefore, in that case we introduce one-hop version of EEMAC, in which, CHs send their gathered data directly to the base station. Basically, Clusteering phase and steady state phase is same as multi-hop EEMAC, except that in routing path formation part which is handled by BS, all the CHs directly communicate with BS. Since the process is centralized (done by BS), dependeing on the network size BS can choose between different variations. Basically, the difference between one-hop EEMAC and LEACH or LEACH-C is in the way of clustering.

- **Unbalanced EEMAC** It is clear that implementing balanced clustering technique [27] imposes an extra energy dissipation to some of the regular nodes. When we transfer a node from its previously cluster to the new one in order to distribute the nodes evenly between clusters, we force that specific node to communicate with a CH of a new cluster which is further than its previous CH. This difference in distance between node and two CHs can have a big impact in the energy dissipation of that node. We investigate the results by removing balanced clustering feature from EEMAC under the name Unbalanced EEMAC.

3.2 The SAD Protocol

3.2.1 Generating Virtual Random Spirals

The idea of SAD is sorting cluster heads on virtual spirals. Archimedes'spiral [30]is the locus of points corresponding to the locations over time of a point moving away from a fixed point with a constant speed along a line which rotates with constant angular velocity, Figure 3 with following formula:

$$r = a\theta \tag{4}$$

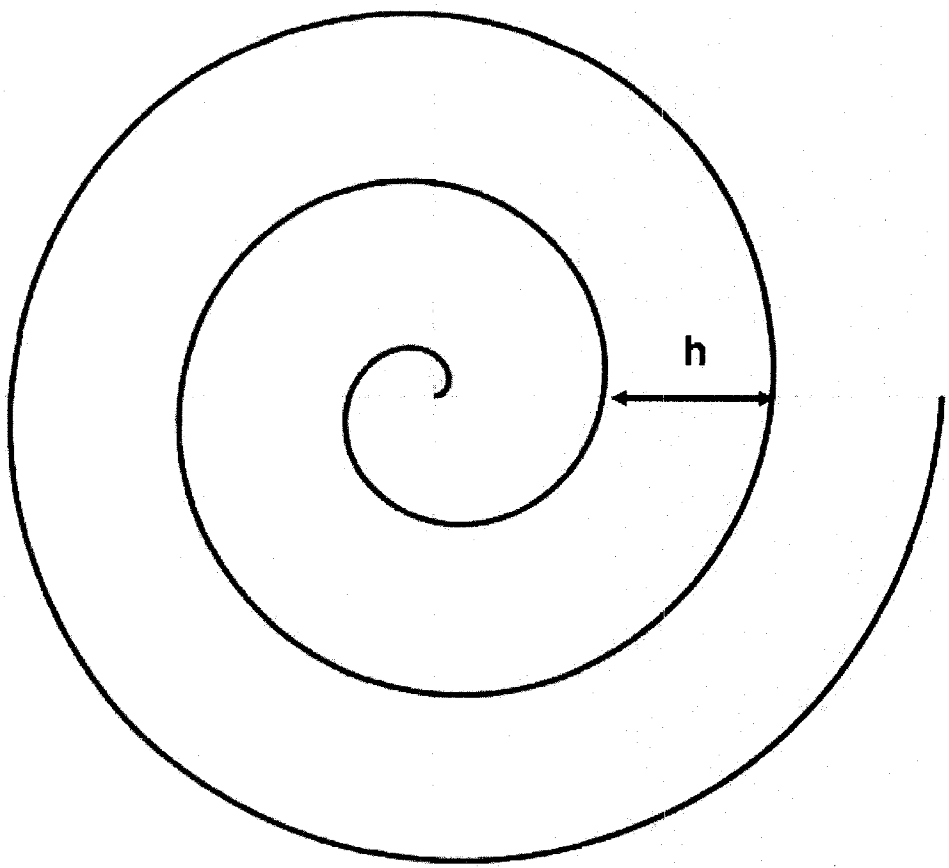

Figure 3: Archimedes' spiral

where the distance between two consecutive curves is $h = 2\pi a$ and it's length from center to the point (r, θ) can be deriven:

$$S(\theta) = \frac{a}{2}\left(\theta\sqrt{1+\theta^2} + \ln(\theta + \sqrt{1+\theta^2})\right) \tag{5}$$

By dividing length of Spiral to the number of clusters, we can find d_{CtoC},the distance between two consecutive clusters on the spiral. Setting clusters on the spiral with equal distances, leads to evenly distributed cluster heads that cover the whole sensing field.

$$d_{CtoC} = \frac{S(\theta_R)}{K_{opt}} \tag{6}$$

where $\theta_R = \frac{R}{a}$, therefore desired place of jth cluster head where $1 \leq j \leq K_{opt}$ can be found:

$$S(\theta_j) = j * d_{CtoC} \tag{7}$$

we can find θ_j and $r_j = a\theta_j$. During network life θ_js and r_js are fixed ,but in each round a random angle θ_{random} is added to θ_j where $1 \leq j \leq K_{opt}$, in other words the spiral will be rotated by θ_{random}.

By setting a good values for a and K_{opt}, we can distribute cluster heads over sensing field that they cover the whole area.

3.2.2 Clustering

During the set-up phase of SAD, each node sends information about its current location (possibly determined using a GPS receiver) and energy level to the BS. In addition to determining good clusters, the BS needs to ensure that the energy load is evenly distributed among all the nodes. To do this, the BS computes the average node energy, and whichever nodes have energy below this average cannot be cluster heads for the current round ,$E_i > E_{average}$(the general condition to be a cluster head candidate is $E_i > \alpha E_{average}$ where we use $\alpha = 1$, effect of changing α can be studied as a future work). Using the remaining nodes as possible cluster heads, base station selects node i as cluster head CH_j if i is the closest node to desired place of jth cluster head $(r_j, \theta_j + \theta_{random})$ between all of the candidates.

Base station at the beginning of each round sends a advertisement message ,which contains the ID of the selected cluster heads. Once cluster heads become notified of their roles, closest nodes to each cluster head will be chosen as members. This can be achieved either by base station's advertisement message or in distributed approach by advertisement message that each cluster head sends. Each regular node determines its cluster for this round by choosing the cluster head that requires minimum communication energy, based on the received signal strength of the advertisement from neighboring cluster heads. Assuming symmetric propagation channels for pure signal strength, the cluster head advertisement heard with the largest signal strength is the cluster head that requires the minimum amount of transmit energy to communicate with. Note that typically this will be the cluster head closest to the sensor, unless there is an obstacle impeding communication. In the case of ties, a random cluster head is chosen [1].

After each node has decided to which cluster it belongs, it must inform the cluster head node that it will be a member of the cluster. Each node transmits a join-request message (Join-REQ) back to the chosen cluster head using a non persistent CSMA MAC protocol. This message is a short message, consisting of the nodes ID and the cluster heads ID.Because nodes are uniformly distributed, all clusters will have almost equal members.

The cluster heads in SAD act as local control centers to coordinate the data transmissions in their cluster. The cluster head node sets up a TDMA schedule and transmits this schedule to the nodes in the cluster. Direct Sequence Spread Spectrum (DSSS) codes are used to minimize inter-cluster interference. This ensures that there are no collisions among data messages and also allows the radio components of each non-cluster head node to be turned off at all times except during their transmit time, thus reducing the energy consumed by the individual sensors. After the TDMA schedule is known by all nodes in the cluster, the set-up phase is complete and the operation phase(data transmission) can begin.

3.2.3 Operation Phase

The operation phase is broken into frames, where nodes send their data to the cluster head at most once per frame during their allocated time slot. The duration of each slot in which a node transmit data is constant, so the time to send a frame of data depends on the number of nodes in the cluster. We assume that nodes are all time synchronized and start the operation phase at the same time. this could be achieved, for example, by having the BS send out synchronization pulses to the nodes.

to reduce energy dissipation , each regular (non-cluster head) node uses power control to set the

[1] Communication energy often does not scale exactly with distance. However, gathering information about the communication channel between all nodes is impractical. Using distance calculated from the nodes GPS coordinates is, therefore, an approximation to the energy that will be required for communication.

amount of transmit power based on the location of the cluster head, or by strength of the received cluster head advertisement signal [2]. Furthermore, the radio of each regular node is turned off until its allocated transmission time. Since we optimize our design for the situation when all the nodes have data to send to the cluster head, using a TDMA schedule is an efficient use of bandwidth and represents a low-latency and energy-efficient approach.

The cluster head must be awake to receive all the data from the nodes in the cluster. Once the cluster head receives all the data, it performs data aggregation to enhance the common signal and reduce the uncorrelated noise among the signals. In our analysis, we assume perfect correlation such that all individual signals can be combined into a single representative signal. The resultant data are sent from the cluster head to the next hop which is discussed in the next section.

3.2.4 Routing on Spiral

Selected cluster heads on the spiral form a chain that is responsible for transmitting gathered data from nodes to the base station. Nodes closer to the base station will deplete their energy faster if cluster heads route data on the spiral in the direction of center. The approach that SAD puts into action is choosing a random cluster head as sink in each round. Afterward cluster head from both ends of spiral directs data into the direction of sink.

Having all the gathered data before transmitting them to the base station, sink(s) has the privilege of final decision or huge fusion of data by removing redundant data. In spite the fact that transmitting data is much more energy consuming than processing it, data fusion in sink as a last hop before BS is very energy efficient.

Finally sink sends all the gathered data to the base station. Since the BS may be far away and the data messages are large, this is a high-energy transmission. Our implementation is that we just have one sink on the spiral and aggregating data never happens in the middle hops on the spiral. In the advertisement message of base station, the cluster head that should acts as sink is defined,

[2] To ensure connectivity in a dynamic environment, the node can either set its transmit power slightly greater than the minimum needed to reach the cluster head, or the cluster head can send short feedback messages to each of the nodes telling them to increase or decrease their transmitted power, as is done in cellular systems.

therefore all the cluster heads know their next hop in the routing path of each round.

3.2.5 Distributed SAD and Other Variations

Although it is a centralized approach to divide the nodes to clusters, we can extend the protocol to distributed clustering approach by putting θ_js and r_js (location of desired cluster heads) and a sequence of θ_{random}s in all the nodes, therefore nodes can select themselves as jth cluster heads via transferring a message containing their distance with desired place of jth cluster head with their neighbors.

On the other hand, we can add some energy saving changes on SAD. As discussed in section 3.2.4 we can have more than one sink on the spiral, this feature is very promising since the delay of data transmission to the base station reduces approximately by the factor of number of sinks on the spiral. When the sensing field is very big this feature is a must and very time efficient. Moreover, we can extend this part to hierarchical layers.

Even though transmitting data is much more energy consuming than processing it, data might need to be sent unprocessed. Without final fusion SAD is when sink(s) can transmit all the gathered data to BS. Therefore, we receive more raw data that can be processed in BS which usually has more processing and energy power than sensor nodes. This is beneficial in the applications or events that need more detailed data from whole sensing field. This implementation of SAD without final fusion leads to the dead of sink(s) at the end of each round in most of the network lifetime. Moreover, This functionality can be activated in plain SAD upon request of network administrator in case of ambiguous data and again deactivated to reserve energy.

Another similar implementation of SAD is when all the system is looking for an upcoming event that can be anywhere in the field. In this version data is being aggregated in each hop during its transmission to the sink(s) and BS. Hence, large amount of energy can be saved by transmitting less data. Finally, SAD can be implemented as plain SAD or with features like distributed setup, multi sink, or without final fusion.

Chapter 4

Analysis

4.1 Analysis of the EEMAC Protocol

We assume that the whole area covered by N nodes is a circle with radios R. Base station divides this area to K clusters, and each cluster has a cluster head. Base Station tries to have balanced clusters in the terms of number of members and geographical distribution. The whole area A will then be

$$A = \pi R^2$$

We also assume that area of each cluster has a circular shape and the cluster head is in the center of that circle with radios r. This area is proportional to $\frac{A}{K}$ with factor $c \geq 1$ (because of overlapping areas). For the simplicity of analysis, we assume $c = 1$. So

$$A_{cluster} = \pi r^2 = c\frac{A}{K} \approx \frac{A}{K}$$

$$r^2 = cR^2/K \approx R^2/K \tag{8}$$

By balance clustering we have n_c nodes in each cluster which is:

$$n_c = \frac{N}{K} \tag{9}$$

4.1.1 Applicable Density

By preventing the transmission over long distances, we can reduce our energy consumption. In the meanwhile, for consuming less energy and consequently being able to only use free space model(d^2 power loss), the distance of any given hop shouldn't be more than d_0 which is the threshold between free space and multi path fading models (e.g., a typical value of d_0 is $88m$). If there exist obstacles in the sight line of the hops, this threshold could be less. We have two different kinds of node to node transmission. The first one, d_{toCH}, is from the sensing node (non-cluster-head) to the cluster head ,which is always less than the radius of the cluster $d_{toCH} \leq r$, and the second one is d_{CH-CH} is the distance between two consecutive cluster heads in the routing path to the base station. We have $d_{CH-CH} \approx 2r$. It is obvious that it implies:

$$r \leq d_0$$

$$2r \leq d_0$$

Hence from Eq. (8), we have

$$K \geq \left(\frac{2R}{d_0}\right)^2$$

It implies that we have a lower bound of number of clusters for a given R. Let $n_{c_{min}}$ be the minimum number of nodes in a cluster. Then for any give R, we have lower bounds for the number of clusters K and number of nodes N as

$$K_{min} = \left(\frac{2R}{d_0}\right)^2$$

$$N_{min} = n_{c_{min}} K_{min} = n_{c_{min}} \left(\frac{2R}{d_0}\right)^2 \tag{10}$$

4.1.2 Non-cluster-head Nodes

Each non-cluster head node only needs to transmit its data to the cluster head once in a give round. Thus, the energy used in each non-cluster head node is

$$E_{non-CH} = E_{elec} l + E_{fs} d_{toCH}^2 l$$

Since nodes are uniformly distributed in the cluster, it's easy to show that d_{toCH} has the following distribution.

$$f_D(d) = \frac{2d}{r^2} \qquad 0 \leq d \leq r \tag{11}$$

and the expected value of d^2_{toCH} is

$$E[d^2_{toCH}] = \frac{r^2}{2}$$

The expected value of the energy used in a non-cluster head node is then

$$\begin{aligned} E[E_{non-CH}] &= E[E_{elec}l + E_{fs}d^2_{toCH}l] \\ &= E_{elec}l + E_{fs}E[d^2_{toCH}]l \\ &= E_{elec}l + E_{fs}\frac{r^2}{2}l \\ &= E_{elec}l + E_{fs}\frac{R^2}{2K}l \end{aligned} \tag{12}$$

4.1.3 Cluster-head Nodes

There are $\frac{N}{K}$ nodes per cluster (one cluster head and $\left(\frac{N}{K} - 1\right)$ non-cluster head nodes). Each cluster head dissipates energy receiving signals from the nodes, aggregating the signals, transmitting the aggregate signal to the next cluster head, and forwarding other cluster heads packets. Therefore, the energy dissipated in the cluster head node in a given round is

$$\begin{aligned} E_{CH} &= E_{elec}l(n_c - 1) + E_{DA}l(n_c - 1) \\ &\quad + E_{elec}l + E_{fs}d^2_{tonextCH}l \\ &\quad + E_{forward} \end{aligned}$$

where E_{DA} is the energy required to process 1 bit information in data aggregation. Here, we also assume that the cluster head aggregate all messages received in a round to a l-bit message. Moreover, we have :

$$d_{tonextCH} \approx 2r \tag{13}$$

For $E_{forward}$, we have:

$$E_{forward} = E_{elec}ln_{fp} + (E_{elec}l + E_{fs}d^2_{tonextCH}l)n_{fp}$$

$$= \quad (2E_{elec}l + E_{fs}d_{tonextCH}^2 l)n_{fp} \tag{14}$$

where n_{fp} is the number of packets from other cluster heads which should be forwarded by the current cluster head. It's clear that the number of forwarded packets for a cluster head increases when it get close to the base station.

Forwarded Packets

Since nodes are distributed uniformly, the number of cluster heads in a specific area is proportional to its area. If we imagine our whole area which has a circular shapes is made of different rings, the cluster heads in a given ring are responsible for forwarding the packets of outer rings.

Let $A_{ring}(d)$ be the area between the circles with radius $d+r$ and $d-r$. Here d is the distance between a random cluster head and the base station. Then

$$A_{ring}(d) = \pi((d+r)^2 - (d-r)^2)$$

and the number of cluster heads inside this ring will be k_{ring}:

$$K_{ring}(d) = K\frac{A_{ring}(d)}{A} = K\frac{\pi((d+r)^2 - (d-r)^2)}{A}$$

Similarly, we can calculate the number of cluster heads outside this ring $K_{out}(d)$:

$$K_{out}(d) = K\frac{A - \pi(d+r)^2}{A}$$

The cluster heads inside the ring are responsible to transfer and forward the packets from outsider clusters[86]. So, on the average, a given cluster head in the ring is responsible for $n_{fp}(d)$ outsider cluster heads and

$$\begin{aligned} n_{fp}(d) &= \frac{K_{out}(d)}{K_{ring}(d)} = \frac{\pi(R^2 - (d+r)^2)}{\pi((d+r)^2 - (d-r)^2)} \\ &= \frac{R^2 - r^2}{4rd} - \frac{d^2}{4rd} - \frac{1}{2} \end{aligned}$$

Applying Eq. (8), we have

$$\begin{aligned} n_{fp}(d) &= \frac{R^2(1-\frac{1}{K})}{4\frac{R}{\sqrt{K}}d} - \frac{d}{4r} - \frac{1}{2} \\ &= \frac{R\sqrt{K}(1-\frac{1}{K})}{4}\frac{1}{d} - \frac{\sqrt{K}}{4R}d - \frac{1}{2} \end{aligned} \tag{15}$$

d and its expected values

Since cluster heads are uniformly distributed, d, the distance between a random cluster head and base station will follow the distribution

$$f_D(d) = \frac{2d}{R^2} \qquad 0 \leq d \leq R$$

Therefore we have:

$$E[d] = \frac{2R}{3} \tag{16}$$

$$E[\frac{1}{d}] = \frac{2}{R} \tag{17}$$

Expected Value of E_{CH}

From Eqs. (15)(16)(17), we have

$$\begin{aligned} E[n_{fp}] &= \frac{R\sqrt{K}(1-\frac{1}{K})}{4}E[\frac{1}{d}] - \frac{\sqrt{K}}{4R}E[d] - \frac{1}{2} \\ &= \frac{\sqrt{K}(1-\frac{1}{K})}{2} - \frac{\sqrt{K}}{6} - \frac{1}{2} \end{aligned} \tag{18}$$

Combined it with Eq. (14),

$$\begin{aligned} E[E_{forward}] &= (2E_{elec}l + E_{fs}E[d^2_{tonextCH}]l)E[n_{fp}] \\ &= (2E_{elec}l + 4E_{fs}r^2l)E[n_{fp}] \\ &= \left(2E_{elec}l + \frac{4E_{fs}R^2l}{K}\right) \times \\ &\quad \left(\frac{\sqrt{K}(1-\frac{1}{K})}{2} - \frac{\sqrt{K}}{6} - \frac{1}{2}\right) \end{aligned} \tag{19}$$

Now that we have all the items required to compute E_{CH} and it is straightforward

$$\begin{aligned} E[E_{CH}] &= E[E_{elec}l(n_c-1) + E_{DA}l(n_c-1) \\ &\quad + E_{elec}l + E_{fs}d^2_{tonextCH}l] + E[E_{forward}] \\ &= (E_{elec} + E_{DA})\, l\left(\frac{N}{K} - 1\right) + E_{elec}l \\ &\quad + 4E_{fs}\frac{R^2}{K}l + \left(2E_{elec}l + \frac{4E_{fs}R^2l}{K}\right) \times \\ &\quad \left(\frac{\sqrt{K}(1-\frac{1}{K})}{2} - \frac{\sqrt{K}}{6} - \frac{1}{2}\right) \end{aligned} \tag{20}$$

4.1.4 Total Energy Consumption

Total Energy for a given round is :

$$E_T = \sum E_{non-CH} + \sum E_{CH}$$

The expected total energy consumption is then

$$\begin{aligned}
E[E_T] &= (N-K)E[E_{non-CH}] + KE[E_{CH}] \\
&= (N-K)\left(E_{elec}l + E_{fs}\frac{R^2}{2K}l\right) \\
&\quad + K\left(E_{elec} + E_{DA}\right)l\left(\frac{N}{K} - 1\right) \\
&\quad + KE_{elec}l + 4E_{fs}R^2 l \\
&\quad + K\left(2E_{elec}l + \frac{4E_{fs}R^2 l}{K}\right) \times \\
&\quad \left(\frac{\sqrt{K}(1-\frac{1}{K})}{2} - \frac{\sqrt{K}}{6} - \frac{1}{2}\right) \qquad (21)
\end{aligned}$$

4.1.5 Optimum Number of Clusters

By taking the derivative of Eq. (21) with regarding to K and let it equal to zero, we get the equation for optimal K,

$$\begin{aligned}
\frac{\partial E[E_T]}{\partial K} &= E_{elec}K^{\frac{5}{2}} - \left(2E_{elec} + E_{DA}\right)K^2 \\
&\quad + \left(\frac{2}{3}E_{fs}R^2 - \frac{1}{2}E_{elec}\right)K^{\frac{3}{2}} \\
&\quad + E_{fs}R^2K^{\frac{1}{2}} - \frac{1}{2}NE_{fs}R^2 \\
&= 0 \qquad (22)
\end{aligned}$$

Solving Eq. (22) numerically, we will be able to find the optimal K for different R and N.

In Table 2, we show some of our numerical results. For the wireless sensor network model, the parameters are set as follows [89, 90, 57]: $E_{elec} = 50nJ/bit$, $E_{fs} = 10pJ/bit/m^2$, and $E_{DA} = 5nJ/bit/signal$. To have a reasonable clustering, we assumed the minimum required number of nodes in each cluster is $n_{c_{min}} = 10$. According to Eq. (10) there is a lower bound for number of

K_{opt}		N					
R	N_{min}	100	250	500	1000	2000	10000
100	51	8.9	12.2	15.5	19.8	25.3	45.7
200	206	n/a	17.3	23	30.4	39.9	74.9
300	464	n/a	n/a	28.6	38.6	51.7	99.7
400	826.4	n/a	n/a	n/a	45.2	61.5	121.5
500	1291	n/a	n/a	n/a	n/a	69.8	141
1000	5165	n/a	n/a	n/a	n/a	n/a	214.8

Table 2: Optimal K for different R and N

nodes in the network. In our numerical results, we exclude the not applicable combinations (very few nodes in big area which is not reasonable).

4.2 Analysis of the SAD Protocol

We consider that the whole area covered by N nodes is a circle with radios R. Base station divides this area to K clusters, and each cluster has a cluster head. Base Station tries to have balanced clusters in the terms of number of members and geographical distribution. The whole area A will then be

$$A = \pi R^2$$

We assume that area of each cluster has a circular shape and the cluster head is in the center of that circle with radios r. This area is equal to the product of distance between two consecutive cluster heads d_{CtoC} and the distance between two consecutive curved in spiral, $h = 2\pi a$. So:

$$A_{cluster} = \pi r^2 = 2\pi a d_{CtoC}$$

$$r^2 = 2ad_{CtoC} \tag{23}$$

As discussed in 3.2.2 we have fairly balance clustering or n_c nodes in each cluster:

$$n_c \approx \frac{N}{K} \tag{24}$$

By preventing the transmission over long distances, we can reduce our energy consumption. In the meanwhile, for consuming less energy and consequently being able to only use free space model(d^2

power loss), the distance of any given hop shouldn't be more than d_0 which is the threshold between free space and multi path fading models (e.g., a typical value of d_0 is $88m$). If there exist obstacles in the sight line of the hops, this threshold could be less. We have two different kinds of node to node transmission. The first one, d_{toCH}, is from the sensing node (non-cluster-head) to the cluster head ,which is always less than the radius of the cluster $d_{toCH} \leq r$, and the second one is d_{CtoC} is the distance between two consecutive cluster heads in the routing path to the base station. We have $d_{CtoC} = \frac{S(\theta_R)}{K}$. It is obvious that it implies:

$$r \leq d_0$$

$$d_{CtoC} \leq d_0$$

Hence from Eq. (23), we have

$$K \geq \frac{S(\theta_R)}{d_0}$$
$$a \leq \frac{d_0}{2}$$

It implies that we have a lower bound of number of clusters for a given R, confirming actual K is set according to the Observation Ratio. On the other hand upper bound of a help us in finding its optimal value.

Let $n_{c_{min}}$ be the minimum number of nodes in a cluster. For having a reasonable clustering, we should have enough nodes in each cluster (we can assume at least 10 nodes in each cluster, $n_{c_{min}}$) . Therefore, we can obtain a lower bound of number of nodes (N) for the whole area($A = \pi R^2$). Since this relation is linear, desired density of system should be bigger than a resulted lower bound density. Then for any give R, we have lower bounds for the number of nodes N as

$$N_{min} = n_{c_{min}} K_{min} = n_{c_{min}} \left(\frac{2R}{d_0}\right)^2 \qquad (25)$$

4.2.1 Non-cluster-head Nodes

Each non-cluster head node only needs to transmit its data to the cluster head once in a given frame. Thus, the energy used in each non-cluster head node for sending a l bit packet is:

$$E_{non-CH}(l) = E_{elec}l + E_{fs}d_{toCH}^2 l$$

Since nodes are uniformly distributed in the cluster, it's easy to show that d_{toCH} has the following distribution.

$$f_D(d) = \frac{2d}{r^2} \qquad 0 \leq d \leq r \tag{26}$$

and the expected value of d_{toCH}^2 is

$$E[d_{toCH}^2] = \frac{r^2}{2}$$

The expected value of the energy used in a non-cluster head node is then

$$\begin{aligned} E[E_{non-CH}] &= E[E_{elec}l + E_{fs}d_{toCH}^2 l] \\ &= E_{elec}l + E_{fs}E[d_{toCH}^2]l \\ &= E_{elec}l + E_{fs}\frac{r^2}{2}l \\ &= E_{elec}l + E_{fs}ad_{CtoC}l \end{aligned} \tag{27}$$

4.2.2 Cluster-head Nodes

There are $\frac{N}{K}$ nodes per cluster (one cluster head and $(\frac{N}{K} - 1)$ non-cluster head nodes). For ease of our calculations we assume $n_c = \frac{N}{K}$. Each cluster head dissipates energy receiving signals from the nodes, aggregating the signals, transmitting the aggregate signal to the next cluster head, and forwarding other cluster heads packets. If we define $E_{routing}$ as the energy that each cluster heads dissipates to receive packets from previous cluster head on the spiral and to send these data and its own data. Therefore, rest of the energy that dissipates in the cluster head node in a given round is

$$E_{CH} = E_{elec}l(n_c - 1) + E_{DA}l(n_c - 1) \tag{28}$$

The cluster head aggregate all messages received in a frame to a l-bit message.

Routing Energy

We want to calculate the number of received packets from previous cluster heads on the spiral which should be forwarded by the current cluster head. In addition, each cluster head should send its own gathered data.It's clear that the number of forwarded packets for a cluster head increases when it gets closer to the sink on the spiral. Number of received packets in each frame by CH_j where CH_s is sink and $1 \leq j, s \leq K$ is:

$$n_{rec} = \begin{cases} j-1 & \text{if } j < s \\ K-1 & \text{if } j = s \\ K-j & \text{if } j > s \end{cases} \tag{29}$$

It's clear that CH_j sends $n_{rec}+1$ packets in each frame , including its own data. Total number of received n_{rec} and sent n_{sen} packets by all the cluster heads excluding s will be:

$$\sum n_{rec} = \frac{(s-2)(s-1)}{2} + \frac{(K-s-1)(K-s)}{2}$$

$$\sum n_{sen} = \frac{(s-1)(s)}{2} + \frac{(K-s)(K-s+1)}{2}$$

by getting expected value of s which is $E[s] = \frac{K}{2}$we will have:

$$E\left[\sum n_{rec}\right] = \left(\frac{K}{2} - 1\right)^2$$

$$E\left[\sum n_{sen}\right] = \frac{K^2}{2}$$

For $E_{routing}$, we have:

$$E\left[\sum E_{routing}\right] = \frac{K^2}{2} E_{send} + \left(\frac{K}{2} - 1\right)^2 E_{receive}$$

$$E\left[\sum E_{routing}\right] = \frac{K}{2}^2 (E_{elec}l + E_{fs}d_{CtoC}^2 l) + \left(\frac{K}{2} - 1\right)^2 E_{elec}l \quad (30)$$

Since nodes are distributed uniformly, the number of cluster heads in a specific area is proportional to its area. If we imagine our whole area which has a circular shape is made of different rings, the cluster heads in a given ring are responsible for forwarding the packets of outer rings.

4.2.3 Sink

Since cluster heads are uniformly distributed, d_{BS}, the distance between a sink and base station will follow the distribution

$$f_D(d) = \frac{2d}{R^2} \qquad 0 \leq d \leq R$$

Therefore we have:

$$E[d_{BS}] = \frac{2R}{3} \quad (31)$$

$$E[d_{BS}^4] = \frac{R^4}{3} \quad (32)$$

Since d_{BS} is usually bigger than d_0 we have to use multi path fading radio model to calculate sink energy consumption in transmitting packets. The energy that sink dissipates in each frame is :

$$\begin{aligned} E_{sink} &= (K-1)E_{receive} + KE_{send}(d_{BS}) \\ &= (K-1)E_{elec}l + K(E_{elec}l + E_{mp}d_{BS}^4 l) \end{aligned}$$

and its expected value is:

$$E[E_{sink}] = (2K-1)E_{elec}l + (K-1)E_{mp}\frac{R^4}{3}l$$

4.2.4 Total Energy Consumption

Total Energy for a given frame is :

$$
\begin{aligned}
E_T &= \sum E_{non-CH} + \sum E_{CH} \\
&\quad + \sum E_{routing} + E_{sink}
\end{aligned}
$$

By combining 27,28,30 and 33the expected total energy consumption is :

$$
\begin{aligned}
E[E_T] &= (N-K)E[E_{non-CH}] + KE[E_{CH}] \\
&\quad + E[\sum E_{routing}] + E[E_{sink}]
\end{aligned}
$$

$$
\begin{aligned}
E[E_T] &= (N-K)\left(E_{elec}l + E_{fs} a d_{CtoC} l\right) \\
&\quad + N\left(E_{elec} + E_{DA}\right) l \\
&\quad + \frac{K}{2}^2 \left(E_{elec}l + E_{fs} d_{CtoC}^2 l\right) \\
&\quad + \left(\frac{K}{2} - 1\right)^2 E_{elec} l \\
&\quad + (2K-1)E_{elec}l + (K-1)E_{mp}\frac{R^4}{3} l
\end{aligned} \tag{33}
$$

4.2.5 Optimum Spiral Factors

As discussed in 6 d_{CtoC} is function of a where $\theta_R = \frac{R}{a}$. We will have:

$$
d_{CtoC} = \frac{a}{2K}\left(\frac{R}{a}\sqrt{1 + \frac{R}{a}^2} + \ln\left(\frac{R}{a} + \sqrt{1 + \frac{R}{a}^2}\right)\right) \tag{34}
$$

By taking the derivative of Eq. (33) with regarding to a and let it equal to zero, we will be able to find the optimal a for different R and K. As we mentioned , K is different for each application and according to the Observation Ratio ,R_O, of each application it can be set. We will further discuss the used values in section 5.1.

4.2.6 Observation Ratio R_O

Assume N nodes distributed uniformly in a circle with radius R .Having K cluster with $\frac{N}{K}$ members, we can calculate Observation Ratio during each frame, by dividing all the data gathered in each cluster during a frame to the frame time and whole area.

$$R_O = K \frac{\frac{N}{K} l}{\frac{\frac{N}{K} l}{B}} \frac{1}{\pi R^2}$$

where l is each packet length in bits and B is the channel bandwidth in bit/sec. therefore we can have:

$$R_O = \frac{KB}{\pi R^2} \tag{35}$$

Further we use this parameter to fairly compare protocols when they generate same amount of data.

Chapter 5

Performance Evaluation

5.1 Performance Evaluation

In this section, we evaluate the performance of the EEMAC and the SAD protocol via simulation. The bandwidth of the channel was set to 1Mb/s, each data message was 500 bytes long, including 25 bytes of the packet header. Therefore, the time slot required for each packet transmission is 0.004 sec. In these experiments, each node begins with 20J of energy and always has data to send to the BS while each round $T_C + T_O$ lasts for 20 s[1]. A node considered dead when it has lost 99.9% of its initial energy.

Since, most of the protocols in the literature work well for small networks, we set three different sizes for our simulations: small R = 100m, medium R = 400m and large R = 800m. In our simulation base station is located in the center of the sensing field (circle), though because all the data transferred to the base station through the sink in SAD; BS can be anywhere else that can be reached by sink.

Radio model that we use in our simulations is what we explained earlier in section 2.1.2. This

[1]The time for a round was chosen so that on average each node has enough energy to act as cluster head once and regular node several time throughout the simulation lifetime.

model assumes a continuous function for energy consumption. For the experiments , the communication energy parameters are set as : E_{elec} =50 nj/bit, E_{fs} =10 pj/bit/m^2, E_{mp} =0.0013 pj/bit/m^4, and d_0 =88m. The energy for data aggregation is also set as E_{DA} =5 nj/bit/signal [90].

As it explained earlier in section 3.2, SAD periodically selects cluster heads according to their geographical location and residual energy by sorting them on virtual spirals. We compare SAD to LEACH[89, 90] and EEMAC [71]. As we mentioned in section 2.2, LEACH, a distributed clustering protocol for micro-sensor networks, was introduced for prolonging the network lifetime. LEACH was proposed for an application in which sensor nodes are randomly distributed on a grid-like area and are continuously sensing the environment to send reports to a remote Base station. The application assumes that nodes are equally significant and data aggregation is possible. LEACH clustering proved to be 4× to 8× more effective in prolonging the network lifetime than direct communication or minimum energy transfer (shortest path multi-hop routing).

On the other hand EEMAC is a centralized protocol which uses a iterative clustering algorithm[71] to find well distributed cluster heads. Furthermore, in order to evenly distribute the load on all cluster heads, EEMAC utilize the balance clustering technique[27]. The other feature that EEMAC provided is having a choice of multi-hop or one-hop communication. Moreover, another implementation of EEMAC, called unbalanced EEMAC, is just limited to the iterative clustering, since balancing clusters impose extra energy dissipations to the regular nodes. It's necessary to mention that, since EEMAC utilizes iterative splitting algorithm, number of clusters in EEMAC should be of powers of 2. Therefore, we set our observation ratios in a way such that EEMAC can be fairly compared to SAD and LEACH.

We did our simulations with C++, and we kept the track of energy of each sensor during its lifetime. Each time that a node sends, receive, or aggregate data we reduce an specific amount from its energy.

5.1.1 Small Network

We define a small network as a circular field with radius $R = 100\text{m}$. Assuming $N = 100$ nodes dispersed uniformly and Observation Ratio $R_0 = 128\text{bits}/m^2/\text{sec}$ we will have 4 clusters which matches the optimum number of cluster head for LEACH and EEMAC. Moreover, our analysis for SAD optimal values in section 4.2.5 shows that $a = 4$ is optimum for $R = 100\text{m}$.

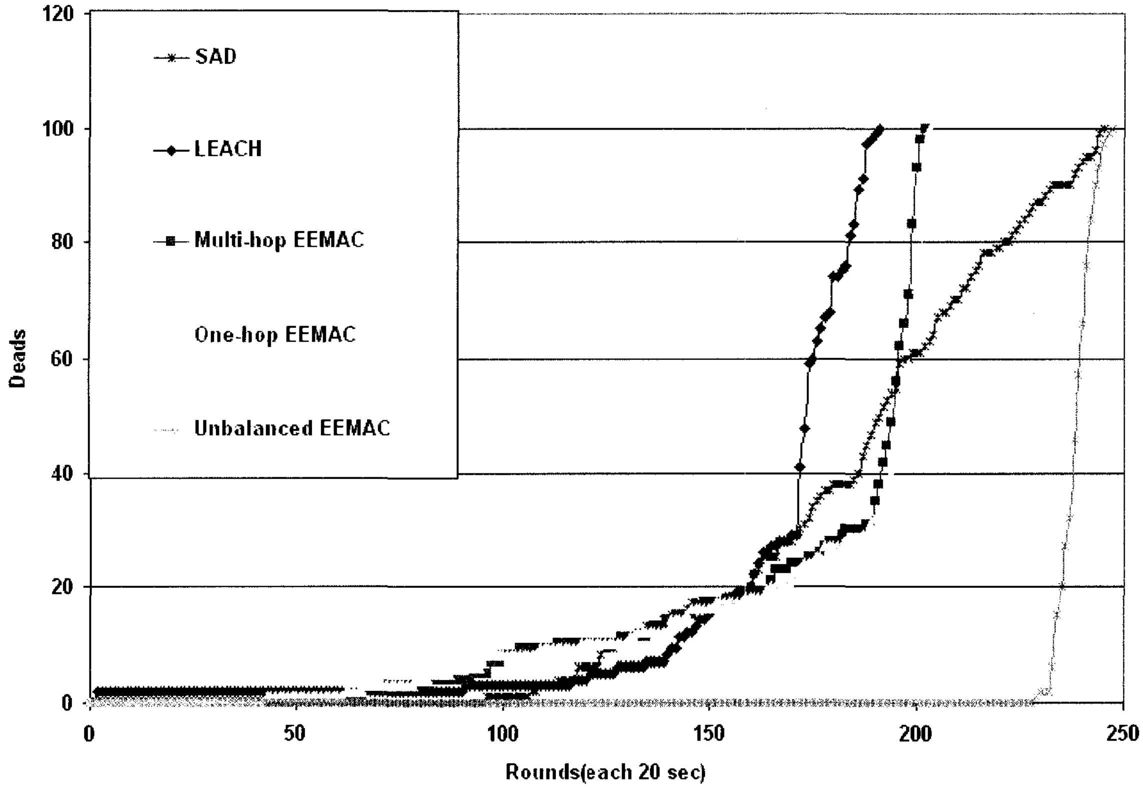

Figure 4: Number of dead nodes for N=100,R=100, K=4

In figure 4, we compare SAD with LEACH and EEMAC variations. This plot shows the number of dead nodes over the number of rounds. The unbalanced EEMAC and SAD that outperform others by factor of 20%, rest of the protocols including LEACH work almost the same. It was clear that SAD which is specifically designed for large networks doesn't make clear improvement in network lifetime.

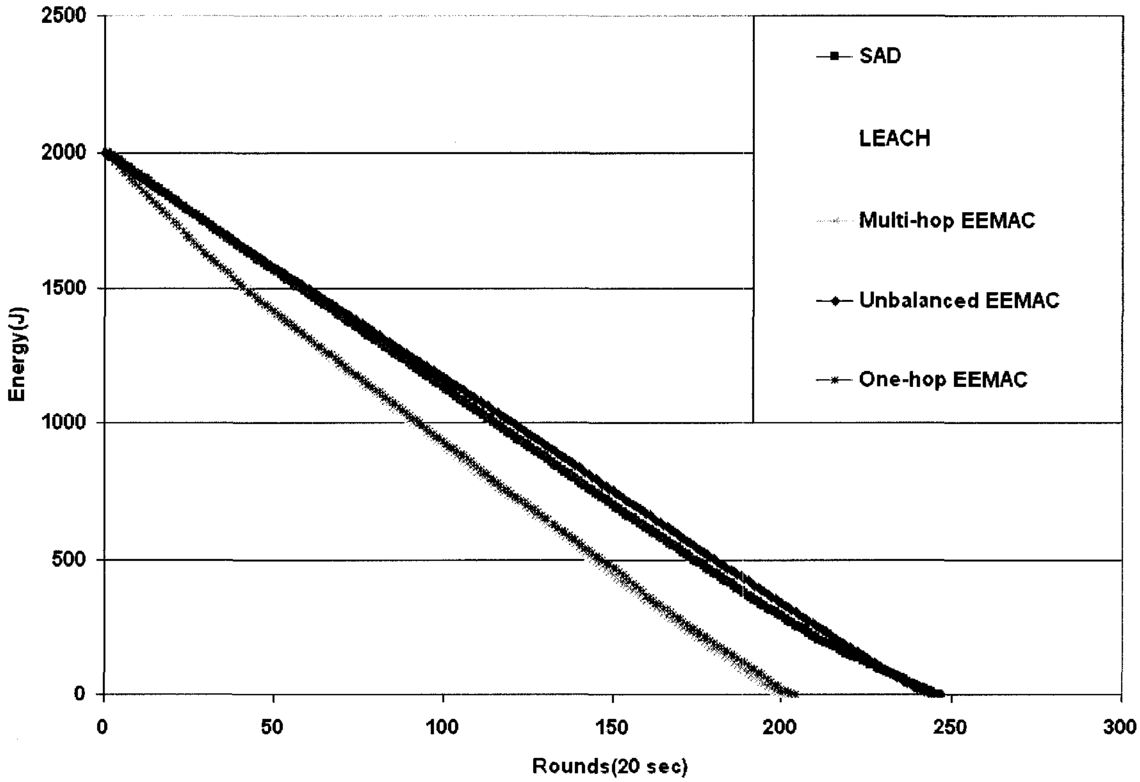

Figure 5: Energy dissipation for N=100,R=100, K=4

In addition, figure 5 states that from energy dissipation all the protocols have the same behavior. This plot shows the amount energy in Joule that dissipates in each round. Confirming the previous conclusion, unbalanced EEMAC and SAD dissipate less energy among their counterparts. This can be the result of good distribution of CHs in the network as long as not imposing extra energy for radio transition by not balancing the clusters. Multihop EEMAC and Onehop EEMAC almost have the same performance, since small network size doesn't give any privileges to multihop transmission.

5.1.2 Medium Network

By growing the size of network to R = 400m, we increased the number of nodes to N = 1600. Setting R_0 = 8bits/m^2/sec leads to having 64 clusters throughout network. Parameter a of SAD is set to a = 16 according to section 4.2.5 . Note that for medium network sizes and bigger optimal

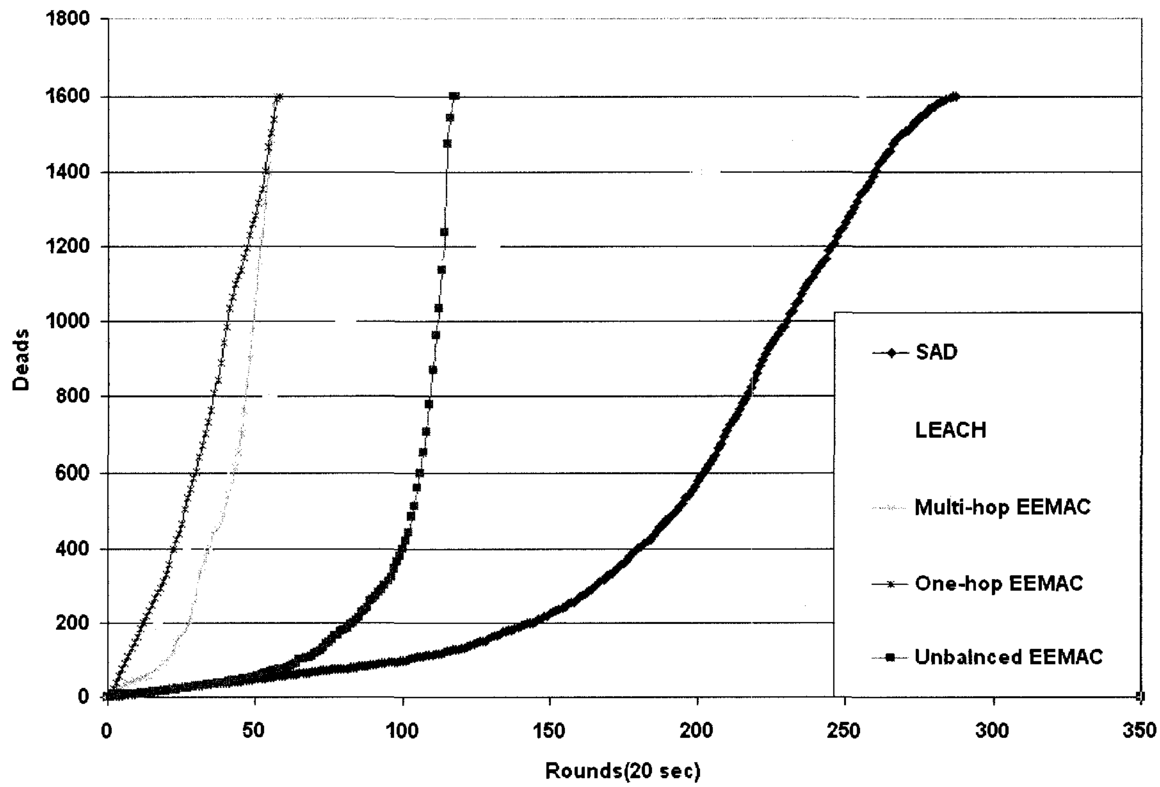

Figure 6: Number of dead nodes N=1600,R=400, K=64

value will be $14 < a < 18$, since the distance between two consecutive curves of spiral $2\pi a$ is around d_0.

Figures 6 and 7 show that SAD outperforms LEACH and EEMAC as expected. If Network lifetime is defined as the number of rounds for which at least 20% of the nodes remain alive, LEACH's and unbalanced EEMAC's Network lifetime is almost equal, while SAD lives 3× longer. It is interesting to mention that Multi-hop and One-hop EEMAC are not different from network life time point of view while Multi-hop EEMAC imposes more complexity to the network.

It is good to mention that, Unbalanced EEMAC has much more reliable behavior over LEACH, though last node of leach dies after EEMAC. The reason is that energy dissipation of unbalanced EEMAC is well distributed among the nodes, in such a way that 75% of the nodes deplete their energy in last 5% of its lifetime, or in other words unbalanced EEMAC works with full capabilities

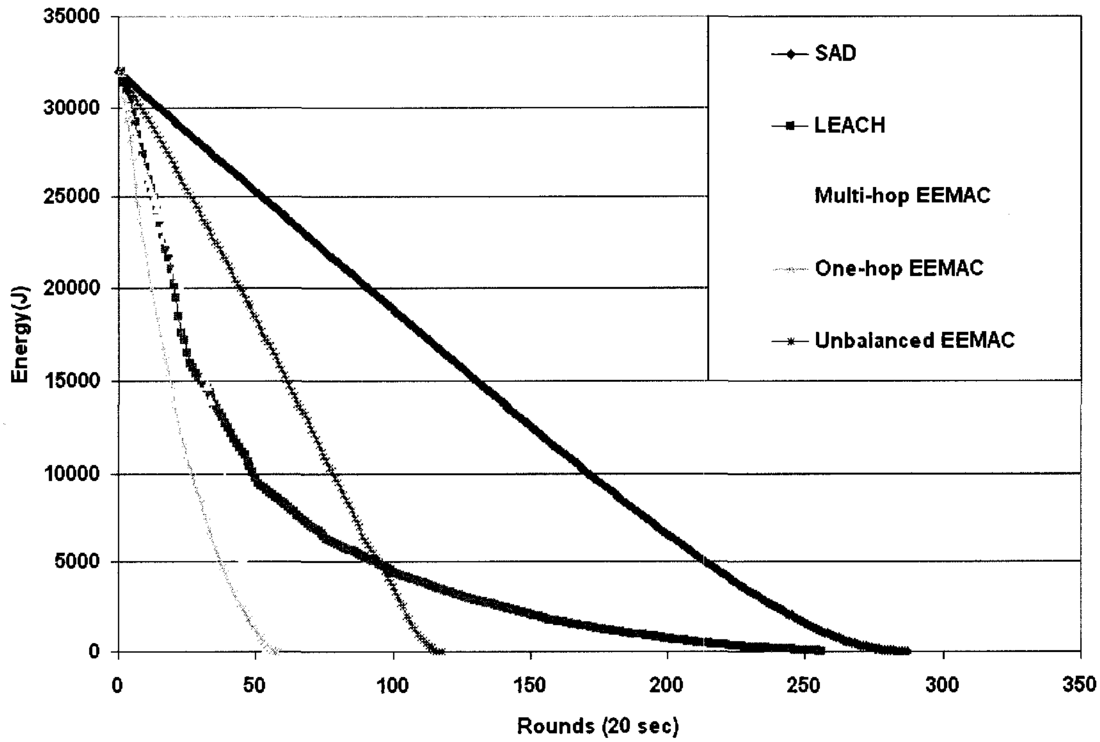

Figure 7: Energy dissipation for N=1600,R=400, K=64

most of its lifetime, while LEACH reaches 50% of its initial number of nodes after 50 rounds.

5.1.3 Large Network

Expansion of the application domain of sensor networks has resulted in a demand for large-scale networks of sensor nodes. Our simulation parameters for large networks are as follows: radius of field R = 800m, number of nodes N = 2500 and with observation ratio R_O = 6.5 we will have K = 128 cluster heads. Moreover, parameter a of SAD is again set to a = 16 according to section 4.2.5. SAD is specially designed for such large networks.

Figure 8 shows the number of dead nodes over network's operating rounds, while figure 9 reflects the amount of dissipated energy. SAD, as we expected spends less energy than LEACH and EEMAC. Furthermore according to our previous network lifetime definition, SAD significantly exceeds the

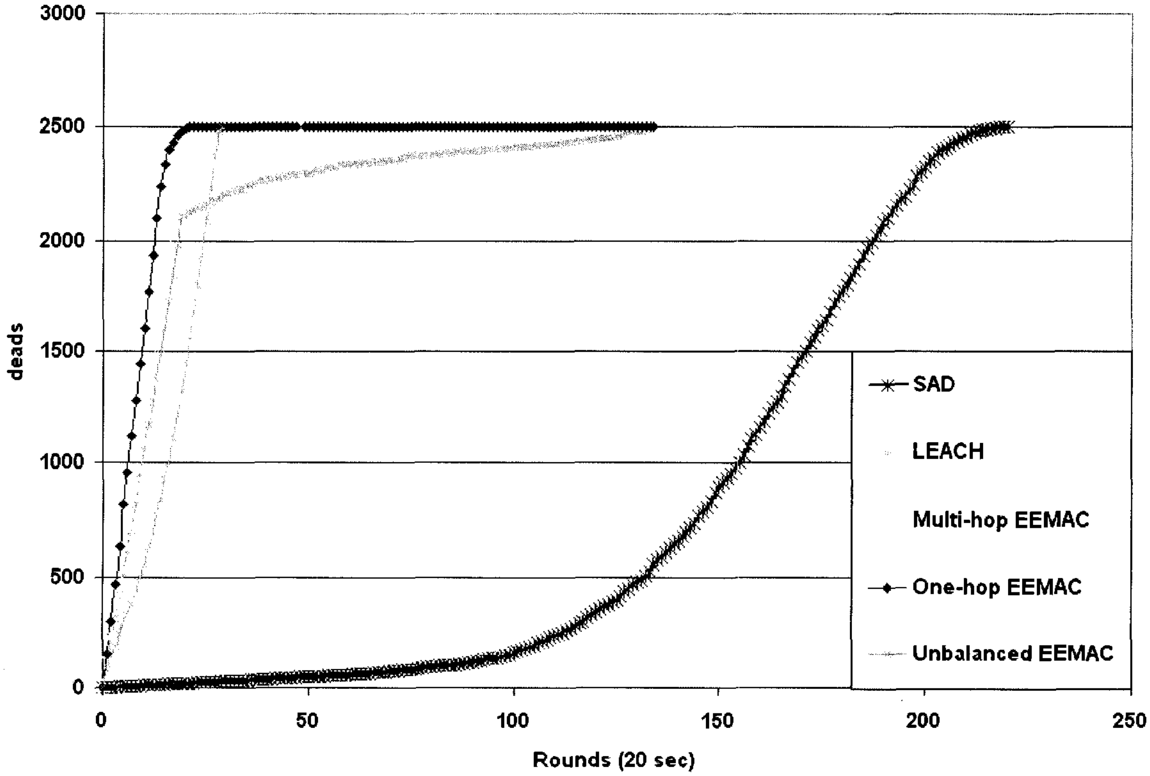

Figure 8: Number of dead nodes for N=2500,R=800, K=128

system lifetime compared to LEACH and EEMAC.

As it is shown in figure 9, Onehop EEMAC is dissipating more energy compared to others in big networks. Moreover, dissipation behaviors of all protocols except SAD are following the same form, and they are not suited for Large Networks. It is interesting to mention that effect of balancing clusters in EEMAC versions is very extravagant that they can not compete against the others.

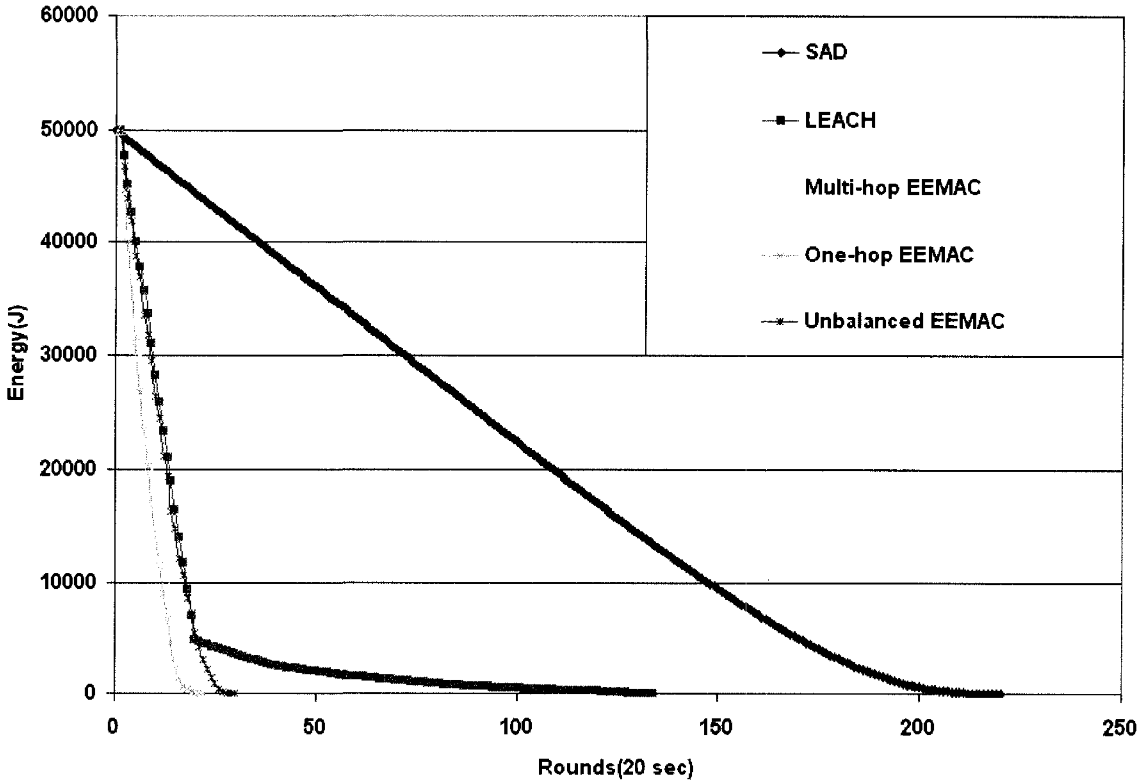

Figure 9: Energy dissipation for N=2500,R=800, K=128

Chapter 6

Conclusion

6.1 Conclusion and Future Work

In this work we proposed two different novel energy efficient algorithms to gather data from sensor nodes. Energy-Efficient Media Access Control (EE-MAC) protocol is the first algorithm, which has excellent scalability and performs well for both small and large sensor networks. In addition, we designed a new cluster-based, multi-hop CH-to-CH protocol, SAD, which utilizes the high-energy base station to perform most energy-intensive tasks and hence prolong the lifetime of the whole network. Considering that optimum clustering is a NP-hard problem, we provided a fast and close to optimum solution. By using the base station to coordinate nodes, the nodes are also relieved of performing cluster setup, cluster head selection, routing path information, and TDMA schedule creation. Our approach is to sorting cluster heads on random virtual spirals in each round of network operation and making a chain of cluster heads to route the data to the base station.

One big advantage of SAD compared with other protocols is its scalability. In SAD, radio transmission over long distance is avoided by using multi-hop CH-to-CH routing. Even if the area of the sensor field is very large, SAD will still perform well and hence provides an energy efficient scheme suitable for a vast range of sensing applications. It is also observed that because of using short transmissions the performance gain of SAD over its counterparts increases with the area of the sensor field. Our simulation results show that SAD prolongs network lifetime, and lowers energy dissipation.

Although we have only provided a protocol for building a single cluster layer with one sink, we can extend the protocol to multi-level hierarchies. This can be achieved by having more sinks on the spiral to limit the delay and moderate the energy consumption of the sink. At the same time, middle hops can aggregate the routed data with their own gathered data and sends it to the next station to lower the load on the chain. SAD can also be useful in the area of wireless sensor network with limited mobility.

We are currently investigating distributed version of SAD which is briefly explained in section 3.2.2 where clustering is performed by the nodes. Moreover, we are studying the effect of different αs in prolonging network lifetime.

Bibliography

[1] 21 ideas for the 21st century, Aug 1999.

[2] "". 10 emerging technologies that will change the world, Jan 2003.

[3] Pankaj K. Agarwal and Cecilia M. Procopiuc. Exact and approximation algorithms for clustering. In *SODA '98: Proceedings of the ninth annual ACM-SIAM symposium on Discrete algorithms*, pages 658–667, Philadelphia, PA, USA, 1998. Society for Industrial and Applied Mathematics.

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

[5] Saeed A. Aldosari and José M. F. Moura. Fusion in sensor networks with communication constraints. In *IPSN '04: Proceedings of the third international symposium on Information processing in sensor networks*, pages 108–115, New York, NY, USA, 2004. ACM.

[6] G. Asada, M. Dong, T. S. Lin, F. Newberg, G. Pottie, and W. J. Kaiser. Wireless integrated network sensors: Low power systems on a chip. In *Proc. 1998 European Solid State Circuits Conference*, 1998.

[7] H. Balakrishnan B. Chen, K. Jamieson and R. Morris. Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. *ACM Wireless Networks*, vol. 8(no. 5), September 2002.

[8] S. Banerjee and S. Khuller. A clustering scheme for hierarchical control in multi-hop wireless networks. In *Proceedings of IEEE INFOCOM, April 2001.*

[9] P. Bonnet, J. E. Gehrke, and P. Seshadri. Querying the physical world. *IEEE Personal Communications*, 7(5):10–15, 2000.

[10] G. Boriello and R. Want. Embedded computation meets the world wide web. *CACM*, 43(5):59–66, may 2000.

[11] D. Braginsky and D. Estrin. Rumor routing algorithm for sensor networks, 2002.

[12] J. Burrell, T. Brooke, and R. Beckwith. Vineyard computing: Sensor networks in agricultural production. *IEEE Pervasive Computing*, 3(1):38–45, January 2004.

[13] R. Govindan C. Intanagonwiwat and D. Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM), 2000.*

[14] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao. Habitat monitoring: Application driver for wireless communications technology. In *Proc. ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean*, 2001.

[15] A. Cerpa and D. Estrin. Ascent: Adaptive self-configuring sensor networks topologies. In *Proceedings of IEEE INFOCOM, New York, NY, June 2002.*

[16] SRC SoC Design Challenge.

[17] Anantha P. Chandrakasan, Rex Min, Manish Bhardwaj, Seong-Hwan Cho, and Alice Wang. Power aware wireless microsensor systems. In *Proc. ESSCIRC 2002*, September 2002. key note paper.

[18] J.-H. Chang and L. Tassiulas. Energy conserving routing in wireless ad-hoc networks. In *Proceedings of IEEE INFOCOM, Tel Aviv, Israel, Mar. 2000.*

[19] K. Chintalapudi, R. Govindan, G. Sukhatme, and A. Dhariwal. Ad-hoc localization using ranging and sectoring, 2004.

[20] The CrossBow corporation. http://www.xbow.com.

[21] The Ember corporation.

[22] D. De Couto, D. Aguayo, J. Bicket, and R. Morris. A high-throughput path metric for multi-hop wireless routing, 2003.

[23] G. Pottie D. Estrin, L. Girod and M. Srivastava. Instrumenting the world with wireless sensor networks. *International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2001)*, May 2001.

[24] J. Heidemann D. Estrin, R. Govindan and S. Kumar. Next century challenges: Scalable coordination in sensor networks. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM), August 1999.*

[25] Ashwin D'Costa and Akbar M. Sayeed. Collaborative signal processing for distributed classification in sensor networks. In *IPSN*, pages 193–208, 2003.

[26] J. Elson, L. Girod, and D. Estrin. Fine-grained network time synchronization using reference broadcasts, 2002.

[27] S. Ghiasi et al. Optimal energy aware clustering in sensor network. *MDPI Sensors*, vol.2:pp. 258 –269, 2002.

[28] D. Ganesan, D. Estrin, and J. Heidemann. Dimensions: Why do we need a new data handling architecture for sensor networks? *ACM SIGCOMM Computer Communication Review*, 33(1):143–148, 2003.

[29] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin. Highly-resilient, energy-efficient multipath routing in wireless sensor networks, 2001.

[30] M. Gardner. *The Unexpected Hanging and Other Mathematical Diversions.* Chicago University Press, pp. 106-107, 1991.

[31] S. Goel, T. Imielinski, K. Ozbay, and B. Nath. Sensor on wheels – towards a zero-infrastructure solution for intelligent transportation systems. In *Proc. 1st Intl. Conf. on Embedded Networked Sensor Systems (SenSys)*, pages 338–339, Los Angeles, CA, November 2003. ACM.

[32] Z. Haas and S. Tabrizi. On some challenges and design choices in ad-hoc communications. *IEEE MILCOM*, Oct 1998.

[33] W. Heinzelman, J. Kulik, and H. Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks, 1999.

[34] Jason Hill, Robert Szewczyk, Alec Woo, Seth Hollar, David Culler, and Kristofer Pister. System architecture directions for networked sensors. *SIGPLAN Not.*, 35(11):93–104, 2000.

[35] Jason L. Hill and David E. Culler. MICA: A wireless platform for deeply embedded computing. *IEEE Micro*, 22(6):12–24, December 2002.

[36] G. T. Huang. Casting the wireless sensor net. *Technology Review*, pages 51–56, July 2003.

[37] A. Ihler, J. Fisher, R. Moses, and A. Willsky. Nonparametric belief propagation for self-calibration in sensor networks, 2004.

[38] W. R. Heinzelman J. Kulik and H. Balakrishnan. Negotiation-based protocols for disseminating information in wireless sensor networks. *ACM Wireless Networks*, vol. 8(no. 2-3):pp. 169–185, 2002.

[39] David B Johnson and David A Maltz. Dynamic source routing in ad hoc wireless networks. In Imielinski and Korth, editors, *Mobile Computing*, volume 353. Kluwer Academic Publishers, 1996.

[40] V. Ailawadhi K. Sohrabi, J. Gao and G. J. Pottie. Protocols for self-organization of a wireless sensor network. *IEEE Personal Comm. Mag.*, vol. 7(no. 5), Oct 2000.

[41] J. M. Kahn, Randy H. Katz, and K. S. J. Pister. Next century challenges: Mobile networking for "smart dust". In *Proc. ACM Intl. Conf. on Mobile Computing and Networking (MobiCom)*, Seattle, WA, August 1999.

[42] R. E. Kalman. A new approach to linear filtering and prediction problems.

[43] Holger Karl and Andreas Willig. *Protocols and Architectures for Wireless Sensor Networks.* John Wiley & Sons, 2005.

[44] Brad Karp and H. T. Kung. GPSR: greedy perimeter stateless routing for wireless networks. In *Mobile Computing and Networking*, pages 243–254, 2000.

[45] R. Karp, J. Elson, D. Estrin, and S. Shenker. Optimal and global time synchronization in sensornets, 2003.

[46] V. Kawadia and P. R. Kumar. Power control and clustering in ad hoc networks. In *Proceedings of IEEE INFOCOM, April 2003.*, April 2003.

[47] C. D. Kidd, R. J. Orr, G. D. Abowrd, C. G. Atkeson, I. A. Essa, B. MacIntyre, E. Mynatt, T. E. Starner, and W. Newstetter. The aware home: A living laboratory for ubiquitous computing research. In *Proc. 2nd Intl. Workshop on Cooperative Buildings*, 1999.

[48] Bhaskar Krishnamachari and Sitharama Iyengar. Distributed bayesian algorithms for fault-tolerant event region detection in wireless sensor networks.

[49] F. L. Lewis. *Wireless Sensor Networks, Smart Environments: Technologies, Protocols, and Applications.* John Wiley, New York, 2004.

[50] C. R. Lin and M. Gerla. Adaptive clustering for mobile wireless networks. *IEEE J. Select. Areas Commun., September 1997*, 2002.

[51] Stephanie Lindsey, Cauligi Raghavendra, and Krishna Sivalingam. Data gathering in sensor networks using the energy*delay metric. *IEEE Transactions on Parallel and Distributed Systems*, vol.13(no.9):pp.924–935, September 2002.

[52] J. Liu, J. Liu, J. Reich, P. Cheung, and F. Zhao. Distributed group management for track initiation and maintenance in target localization applications, 2003.

[53] G. Lu, B. Krishnamachari, and C. Raghavendra. An adaptive energy-efficient and low-latency mac for data gathering in sensor networks, 2004.

[54] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson. Wireless sensor networks for habitat monitoring. In *Proc. 1st ACM Workshop on Wireless Sensor Networks and Applications*, Atlanta, GA, September 2002.

[55] R. Min, M. Bhardwaj, S. Cho, A. Sinha, E. Shih, A. Wang, and A. Chandrakasan. Low-power wireless sensor networks.

[56] MIT μAMPS Project.

[57] S.D. Muruganathan, D.C.F. Ma, R.I. Bhasin, and A.O. Fapojuwo. A centralized energy-efficient routing protocol for wireless sensor networks. *IEEE Communications Magazine*, vol.43(no.3):pp.S8–13, March 2005.

[58] D. Niculescu and B. Nath. Ad hoc positioning system (aps, 2001.

[59] Mark Perillo, Zeljko Ignjatovic, and Wendi Heinzelman. An energy conservation method for wireless sensor networks employing a blue noise spatial sampling technique. In *IPSN '04: Proceedings of the third international symposium on Information processing in sensor networks*, pages 116–123, New York, NY, USA, 2004. ACM.

[60] UCLA MEDUSA platform. http://www.cens.ucla.edu/project-description/sensor_node_platforms.

[61] Joseph Polastre, Robert Szewczyk, and David Culler. Telos: enabling ultra-low power wireless research. In *IPSN '05: Proceedings of the 4th international symposium on Information processing in sensor networks*, page 48, Piscataway, NJ, USA, 2005. IEEE Press.

[62] G. J. Pottie and W. J. Kaiser. Embedding the internet: Wireless integrated network sensors. *Communications of the ACM*, 43(5):51–58, May 2000.

[63] G. J. Pottie and W. J. Kaiser. Wireless integrated network sensors. *Commun. ACM*, 43(5):51–58, 2000.

[64] The Cougar Project. http://www.cs.cornell.edu/databse/cougar.

[65] The Smart Dust Project. http://robotics.eecs.berkeley.edu/pister/smartdust.

[66] The TinyDB Project. http://telegraoph.cs.berkeley.edu/tinydb.

[67] The WINS Project. Rockwell science center.

[68] Rohit Puri, Prakash Ishwar, S. Sandeep Pradhan, and Kannan Ramchandran. Rate-constrained robust estimation for unreliable sensor networks.

[69] U. Mitra R. Nowak. Boundary estimation in sensor networks:.

[70] Jan M. Rabaey, M. Josie Ammer, Julio L. da Silva, Danny Patel, and Shad Roundy. PicoRadio Supports Ad Hoc Ultra-Low Power Wireless Networking. *IEEE Computer*, 33(7), July 2000.

[71] S. Rahimi and D.Qiu. A novel energy-efficient media access control protocol for wireless sensor networks. In *Proceeding (565) Wireless and Optical Communications*, 2007.

[72] S. Rahimi and D.Qiu. Spiraled algorithm for data gathering in wireless sensor networks. *submitted to the SenSys08*, 2008.

[73] E. Royer and C. Perkins. Multicast ad hoc on- demand distance vector (maodv) routing, 2000.

[74] R. S. Sreenivas S. Narayanaswamy, V. Kawadia and P. R. Kumar. Power control in ad-hoc networks: Theory, architecture, algorithm and implementation of the compow protocol. In *Proceedings of European Wireless 2002. Next Generation Wireless Networks: Technologies, Protocols, Services and Applications, February 2002, pp. 156162.*

[75] Mina Sartipi and Faramarz Fekri. Distributed source coding in wireless sensor networks using ldpc codes: A non-uniform framework. In *DCC*, page 477. IEEE Computer Society, 2005.

[76] Andreas Savvides, Chih-Chieh Han, and Mani B. Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *Mobile Computing and Networking*, pages 166–179, 2001.

[77] Andreas Savvides, Chih-Chieh Han, and Mani B. Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 166–179, New York, NY, USA, 2001. ACM.

[78] L. Schwiebert, S. K. S. Gupta, and J. Weinmann. Research challenges in wireless networks of biomedical sensors. In *Proc. 7th Intl. Conf. on Mobile Computing and Networking (ACM Mobicom)*, pages 151–165, Rome, Italy, July 2001.

[79] Karim Seada, Marco Zuniga, Ahmed Helmy, and Bhaskar Krishnamachari. Energy-efficient forwarding strategies for geographic routing in lossy wireless sensor networks. In *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 108–121, New York, NY, USA, 2004. ACM.

[80] Wireless sensor networks spread to new territory. The new york times, 26 July 2004.

[81] Hong Shen. Finding the k most vital edges with respect to minimum spanning tree. *Acta Informatica*, vol.36(nr.5):pp.405–424, 1999.

[82] C. Siaterlis and B. Maglaris. Towards multisensor data fusion for dos detection, 2004.

[83] D. Snoonian. Smart buildings. *IEEE Spectrum*, pages 18–23, September 2003.

[84] M. Srivastava, R. Muntz, and M. Potkonjak. Smart kindergarten: Sensor-based wireless networks for smart developmental problem-solving environments (challenge paper). In *Proc. 7th Ann. Intl. Conf. on Mobile Computing and Networking*, pages 132–138, Rome, Italy, 2001.

[85] D. C. Steere, A. Baptista, D. McNamee, C. Pu, and J. Walpole. Research challenges in environmental observation and forecasting systems. In *Proc. 6th Intl. Conf. on Mobile Computing and Networking (ACM Mobicom)*, 2000.

[86] M. Sugano, Y. Kiri, and M. Murata. Performance analysis of large-scale wireless sensor network architecture with multi-cluster configuration. *Proceeding Networks and Communication Systems*, 2006.

[87] R. Szewczyk, E. Osterweil, J. Polastre, M. Hamilton, A. Mainwaring, and D. Estrin. Habitat monitoring with sensor networks. *Comm. ACM*, 47(6):34–40, 2004.

[88] MEMS technology application center.

[89] A. Chandrakasan W. R. Heinzelman and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the Hawaii International Conference on System Sciences*, 2000.

[90] A. Chandrakasan W. R. Heinzelman and H. Balakrishnan. An application-specific protocol architecture for wireless microsensor networks. *IEEE Transactions on Wireless Communications*, vol.1(no.4):pp. 660670, 2002.

[91] J. Heidenmann W. Ye and D. Estrin. An energy-efficient mac protocol for wireless sensor networks. In *Proceedings of IEEE INFOCOM,New York, NY, June 2002.*, 2002.

[92] Kamin Whitehouse and David Culler. Macro-calibration in sensor/actuator networks. *Mob. Netw. Appl.*, 8(4):463–472, 2003.

[93] Rebecca Willett, Aline Martin, and Robert Nowak. Backcasting: adaptive sampling for sensor networks. In *IPSN '04: Proceedings of the third international symposium on Information processing in sensor networks*, pages 124–133, New York, NY, USA, 2004. ACM.

[94] Picoradio: Berkeley wireless research center.

[95] J. Heidemann Y. Xu and D. Estrin. Geography-informed energy conservation for ad hoc routing. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM), Rome, Italy, July 2001, pp. 7084.*

[96] Osama Younis and Sonia Fahmy. Heed: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. *IEEE Transactions on Mobile Computing*, vol. 3(no. 4):pp. 366–379, Oct. 2004.

[97] Yang Yu, Victor K prasanna, and Bhaskar Krishnamachari. *Information Processing and Routing in Wireless Sensor Networks.* World Scientific Publishing Co. Pte. Ltd., 2006.

[98] F. Zhao, J. Shin, and J. Reich. Information-driven dynamic sensor collaboration for tracking applications. *IEEE Signal Processing Magazine*, March 2002.

[99] R. W. Zurek and L. J. Martin. Pamas: Power aware multi-access protocol with signalling for ad hoc networks. *ACM Computer Communication Review*, vol. 28(no. 3):pp. 526, May 1998.