ABSTRACT

Title: SELF ORGANIZING WIRELESS SENSOR NETWORKS

Kamiar Kordari, Ph.D., 2007

Directed By: Professor G.L. Blankenship, Department of Electrical and Computer Engineering

This dissertation is concerned with the properties of self-organizing network systems, where a large number of distributed sensor nodes with limited sensing, processing and communication capability organize themselves into a cooperative network without any centralized control or management. Due to the distributed nature of the management and lack of global information for in-node decision making, sensor management in such networks is a complicated task. The dynamics of such networks are characterized by constraints and uncertainty, and the presence of disturbances that significantly affect aggregate system behavior. In this dissertation we examine several important topics in the management of self-organizing wireless sensor networks.

The first topic is a statistical analysis to determine the minimum requirements for the deployment phase of a random sensor network to achieve a desired degree of coverage and connectivity.

The second topic focuses on the development of a viable online sensor management methodology in the absence of global information. We consider consensus based sensor data fusion as a motivating problem to demonstrate the capability of the sensor management algorithms. The approach that has been widely investigated in the literature for this problem is the fusion of information from all the sensors. It does not involve active control of the sensors as part of the algorithm. Our approach is to control the operations of the nodes involved in the consensus process by associating costs with each node to emphasize those with highest payoff. This approach provides a practical, low complexity algorithm that allows the nodes to optimize their operations despite the lack of global information.

In the third topic we have studied sensor networks that include “leaders,” “followers,” and “disrupters.” The diffusion of information in a network where there are conflicting strategies is investigated through simulations. These results can be used to develop algorithms to manage the roles in the network in order to optimize the diffusion of information as well as protect the network against disruption.

SELF-ORGANIZING WIRELESS SENSOR NETWORKS

By

Kamiar Kordari

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2007

Advisory Committee:
Professor Gilmer L. Blankenship, Chair
Professor Shuvra Bhattacharyya
Professor Rama Chellappa
Professor Neil Goldsman
Associate Professor Mark Austin

Dedication

To my mom, dad, and sister who made all of this possible, for their endless encouragement and patience.

Acknowledgements

I would like to take this opportunity to express my thanks to those who helped me with various aspects of conducting research and the writing of this dissertation. First and foremost, my advisor Dr. Blankenship for his guidance, patience and support throughout my graduate studies. His insights and words of encouragement have often inspired me and renewed my hopes for completing my graduate education. I would also like to thank my other committee members for their efforts and contributions to this work: Dr. Goldsman, Dr. Bhattacharyya, Dr. Chellappa, and Dr. Austin. I would additionally like to thank all the teachers and students I had the opportunity to learn from during my education.

My sincere gratitude goes to my family for all the love they gave me; my father who taught me to believe in myself and to follow my dreams, my mother whose love and encouragements gave me the power and determination to continue in pursuing those dreams, and my sister who was always supportive of my decisions.

Finally, I would like to thank all the people and friends who made my stay at the beautiful Washington DC metropolitan area a wonderful and pleasant experience.

Table of Contents

List of Figures

Chapter 1: General Introduction

"We may be very busy, we may be very efficient, but we will also be truly effective only when we begin with the end in mind"

- Stephen R. Covey

1.1 Introduction

Sensor networks have received significant attention in recent years due to the attractive solutions they provide for data acquisition and monitoring applications. These networks usually consist of a large number of nodes, sometimes called agents. Typically, each node is a small processing unit with the capability of sensing physical quantities of its surrounding environment, for example temperature, radiation, etc., and communicating the (processed) data to other nodes for a collective decision about features of the environment (detection, estimation, tracking, classification, etc.).

In traditional single sensor systems the data acquisition, analysis, and decision processeses were performed within the processing unit of the sensor. The decision algorithm was part of the sensor software or firmware, where it processed the low dimensional signals generated by the sensor. In a multi-sensor system, each sensor is a contributor to an aggregate decision process. Available data may vary in quality as a function of location and state. Proactive decisions to use additional sensors at some cost (increased process interaction, power, computational, communication requirements, etc.) require algorithms to select the best configuration to achieve the objective.

A class of wireless sensor networks that have received recent attention are self-organizing networks, where a large number of distributed sensor nodes with limited sensing, processing and communication capability organize themselves into a cooperative network without any centralized control or management. Such a network is usually deployed to perform a processing mission over a period of time. In order to survive this network should be robust to node and link failures and it should be self-healing.

In most applications, energy should be efficiently used in a (self-organizing) wireless sensor network; otherwise the network will not be able to perform its required mission over time. In this dissertation we assume that each node gets its energy for sensing, processing, and communication from a source, such as a battery with a fixed amount of energy. Therefore the total energy that has been distributed over the network is limited. If nodes overuse their energy they will die before the mission time is over. Therefore, an efficient sensor management protocol is critical for the operations of this type of energy-limited wireless sensor networks.

Self-organizing sensor management is complicated because of the distributed nature of the management and lack of global information for in-node decision making. Decentralized control of communicating-agent systems has emerged as a challenging research area in recent years. Applications in which effective distributed control algorithms are critical to the success of the application are increasingly common. Typically, the dynamics of such large scale systems are characterized by constraints and uncertainty, and by the presence of disturbances that significantly affect system behavior. Traditional off-line control design techniques may prove inadequate for

such applications. Hence there is a need to develop efficient distributed control schemes. This work is aimed at the development of a practically viable on-line sensor management methodology for distributed self-organizing networks.

To focus the development, we consider a surveillance application where the goal is to detect events in a region as they happen and then estimate a parameter related to each event. This is a general class of problems that can model many different applications. We assume that the network is going to operate for an extended period of time and that a minimum detection and estimation performance is required from the network. We shall show that a sensor management task for such an application can be decomposed into temporal and spatial control parts.

Temporal sensor management and control is the distribution of the energy in time. In order to detect events, nodes need to sense the environment. But a node will fail quickly if it is active and sensing all the time. Therefore nodes should take turns sensing the environment. A node that is not sensing powers down in some specific way to save energy. We will say that the node "goes into sleep." Event occurrences may have a pattern in space and/or time that can be "learned" and modeled by the network, and used to adjust the scheduling times. For example, in instances or locations where an event occurrence is improbable, it would be wise to turn off most of the nodes. A sensor temporal control algorithm involves modeling event occurrence and assignment of energy use in time.

Spatial sensor management and control involves the selection, based on quantitative performance measures (and costs), of a configuration of sensors to collect data for sensor data fusion. Sensor data fusion as widely investigated in the literature is the aggregation of information from all sensors. Generally, it does not involve the active "control'' of the sensor elements as part of a fusion algorithm. A node may be activated either by the detection of a surveillance event, or by a message from a neighbor. The technical objective for this part of the project is the development and evaluation of algorithms for the management of sensor networks containing many sensor nodes. This is critical technology in a sensor network application where there could be tens, hundreds, or thousands of individual sensor elements. The aggregate signal processing should "multiplex" the individual sensor signals, emphasizing those with the "highest payoff" (for example, those sensors processing the signals with the greatest uncertainty).

For this part of the work we use the consensus problem as a motivating problem to demonstrate the capability of our algorithms. One very common problem that arises in different computations in sensor networks is that of *agreement*, or *consensus* between the nodes regarded as active "agents." Typically, agents must achieve consensus with respect to a certain state variable of interest. Applications involving multi-agent networked systems solving consensus problems arise in different disciplines, including distributed computing, flocking and alignment problems, synchronization of coupled oscillators, and cooperative control.

Research on the consensus problem started in the field of computer science where distributed agreement, synchronization problems, and load balancing in parallel computers have been considered [Xu 1997]. Olfati and Murray [Olfati 2003] studied agreement problems in networks with dynamic agents and showed how simple nearest neighbor rules in networks of integrator agents will result in all nodes reaching a consensus over time. Other research has concentrated on developing fast converging algorithms, and proof of convergence of the consensus algorithms under certain conditions [Xiao 2004]. For the most part these studies concentrate on reaching consensus among all the nodes. They do not address the use of active control in a consensus protocol.

Our goal in developing practical distributed wireless sensor management schemes is to address three major challenges: Lack of global information, distributed computation and communications, and randomness.

(1) **Lack of global information:** A node should have enough global information to be able to make good decisions and tune its operation parameters. Our approach is to associate "weights" with each sensor that include sufficient global information for reasonable management decisions. Then the sensor systems operations management problem can be expressed in terms of feedback control laws that take local information and modify the network operation according to the information's global weight to optimize the computation. The weights are potentially capable of incorporating in them global information about the relative position of the node in the

communication graph, the link quality, the residual energy, the cost of operation, and quality of information. This solution is very attractive where it provides a practical low complexity algorithm that provides the nodes some sufficient global information to optimize their operations.

(2) **Distributed computation and communications:** The computational effort and communications volume should not become problematic for large-scale networks. The algorithms need to be *lightweight* and not require huge amounts of computation or communication.

(3) **Randomness:** Although wireless communication provides cost and flexibility advantages, it also presents reliability challenges; Communication links are variable and unpredictable. A wireless link that is strong today may be weak tomorrow due to environmental conditions, new obstacles, and/or unanticipated interferers. Therefore the topology of a network is in constant change. This change is generally due to one of these three major causes:

(i) RF interference: Sensors may communicate in bands of the electromagnetic spectrum devoted to general-purpose wireless communication devices. These bands may be crowded with traffic from Wi-Fi networks, cordless telephones, and other devices. Because there is no way to predict what interferers will be present in a given

location, and time, a reliable network must be able to continually deal with these interferers.

(ii) Blocked communication links: When a network is first deployed, wireless paths are established between sensor nodes based on the available neighbors. However, paths may later be blocked by new equipment, moving vehicles, or very small changes in node's position. Assuring reliability for the life of the network, not just the first few weeks after installation, requires continually working around these blockages in an automatic fashion.

(iii) Node Loss: A node failure may happen because of hardware malfunction, damage, or it may be removed from the network. More importantly a node with no residual energy will not be able to operate.

Any of these problems will compromise a wireless link. However, with a network protocol designed to protect against these issues, the network can isolate individual points of failure and reduce their impact, allowing the network as a whole to function effectively in spite of local failures.

The algorithms we propose in this work adapt to changes in the environment, allowing long-term operation with zero-touch maintenance. By dynamically adapting the associated "weights", each sensor has access to the updated global information, and therefore is capable of handling the randomness and uncertainties in the network.

This method is suited best for a network with relatively slow random changes, because the weight update process takes some time to converge.

The sensor fusion management protocols for self-organizing networks developed in this dissertation provide mechanisms to increase both efficiency and reliability of the network and to form a solid foundation for wireless sensor network applications.

1.2 Mission-Oriented Sensor Management

Beginning with the assumption that the nodes in a network perform a mission over a period of time under energy constraints, we develop an algorithm that manages the operation to guarantee the desired performance during the mission time.

The network with n nodes is supposed to perform a mission over a period of time $t \in [0,T]$. The mission time T is known in advance and it is the minimum acceptable lifetime for the network. This may not be the case for applications where there is no such predefined network lifetime and network operates until all the nodes fail. The mission of the network is to (1) detect when an event happens in the region being monitored, and (2) estimate a parameter of that event. The constraint is that each node has a total energy of E^i. Thus, the total energy distributed over the field is $E = \sum_{i=1}^{n} E^i$.

Therefore, in general it is not possible for all the nodes to participate in all the processing activities over the entire interval of interest. For each detection and estimation task a group of nodes should be selected to achieve a certain level of

performance. The global performance of the network over the mission time is a function of the individual node performances. The problem is how to distribute the total available energy to gain the maximum global performance. We formulate the problem as an optimization problem.

Assume that during the time period T (an integer) the maximum number of event occurrences is M. Partition the time axis to M equally distance time instance $k_1,...,k_M$, where each time slot has length $T/(M-1)$, and $k_M = T$. At each instance k an event may happen with probability $P(k)$. When an event is detected the network has a time equal to $T/(M-1)$ to finish the estimation task related to that event. Suppose the network assigns a total energy of x_k for each event detection task. Therefore, the total detection energy budget for each node is αE^i. α is the percentage of a node's total energy that is a set aside for detection. The share of node i from the energy x_k is x_k^i. Also, assume that there is a utility function $U(x_k)$ that defines the detection performance of the configuration of sensors. Assume that the utility function is differentiable and concave. The sensor management problem can be formulated as follows:

$$\begin{aligned} \max_{\{x_k^i\}} \quad & \sum_{k=1}^{M} E[U(x_k)]P(k) \\ \text{subject to } & \sum_{k=1}^{M} x_k^i \le \alpha E^i \quad \forall i \\ & x_k = \sum_{i=1}^{N} x_k^i \end{aligned} \tag{1-1}$$

This problem is solved sequentially and when an event happens the algorithm updates the energy that can be used by the network for processing the next event.

We solve this problem by decomposing it into two problems. The first problem is the *temporal control problem* that finds the total energy allocated for detecting each event.

$$\max_{\{x_k\}} \quad \sum_{k=1}^{M} E[U(x_k)]P(k) \qquad (1\text{-}2)$$
$$\text{subject to } \sum_{k=1}^{M} x_k \le \sum_{i=1}^{N} \alpha E^i \quad \forall i$$

The second problem is the *spatial control problem*. At each time instant k the energy allocated for the current event is distributed between nodes as follows,

$$\max_{\{x_k^i\}} \quad U(\{x_k^i\}) \qquad (1\text{-}3)$$
$$\text{subject to } \sum_{i=1}^{N} x_k^i \le x_k \quad \forall i$$
$$\sum_{s=1}^{k} \frac{x_s^i}{E^i} = \frac{\sum_{s=1}^{k} x_s}{\sum_{i=1}^{N} E^i}$$

The objective is to maximize the performance while keeping the total energy consumption over the participating nodes under a limit. Also we have a constraint to enforce nodes to consume energy at the same rate, so that all the nodes fail at the same time at the end of the mission and not before that. There are applications that all the nodes do not need to consume energy at the same rate and therefore this algorithm

does not address them. However the ideas presented here may be leveraged to apply for those types of applications.

The above problem deals with efficient distribution of the energy. However one other important issue in a sensor management scheme is *role management*. Role management is about assigning different roles and responsibilities to sensors in the network for more efficient processing as well as protecting against possible conflicts.

In this dissertation we investigate different issues related to the energy management problem as well as the role management. In Chapter Two we look into the achievable expected detection performance with a given total energy using statistical methods. In Chapter Three we study the spatial control and the distributed methods to solve that problem. And finally in Chapter Four, we look into the role management in a network that includes leaders, followers and disrupters.

1.3 Related Work

In general, wireless sensor network management is used for topology control and sensing mode selection [Perillo 2004]. Topology control is used when sensors are deployed with densities high enough so that not all sensors are needed to route data to the sink. The goal of a topology control is to ensure that enough nodes are active to provide a connected network so all the sensors that have data to send can get their data to the base station while turning off any unnecessary sensors to save energy. Sensor mode selection is needed when sensors are deployed with densities high enough so that activating every sensor in the network provides little more quality of

service for the application. The goal of the sensor selection is to have only certain sensors gather data so that there is no unnecessary redundancy, network congestion, and energy waste and the cumulative sensor data quality is sufficient to meet the application's goal.

Topology control algorithms include GAF (Geographic Adaptive Fidelity) [Xu 2001], SPAN [Chen 2000], ASCENT (Adaptive Self-Configuring sEnsor Networks Topologies) [Cerpa 2002], and STEM (Sparse Topology and Energy Management) [Schurgers 2002]. The idea in GAF is to form a virtual grid throughout the network and to allow only one node in a cell to be active at any given time. In SPAN a connected routing backbone is formed and other nodes that are not involved go to sleep for extended periods of time. The set of the nodes forming the backbone changes to ensure a balance in energy consumption among the nodes in the network. In ASCENT certain nodes are chosen to be active while others go into the sleep mode to conserve energy. The active nodes are chosen to provide connectivity and reliability based on the observed data loss rates among the neighbors. STEM is different from other protocols in that it activates nodes reactively rather than proactively. When data packets are generated, the sensor generating the traffic uses a paging channel to awaken its downstream neighbors.

Sensor mode selection protocols include PEAS (Probing Environment and Adaptive Sensing) [Ye 2003], NSSS (Node Self-Scheduling Scheme) [Tian 2003], and CCP (Coverage Configuration Protocol) [Wang 2003]. In PEAS nodes provide a consistent

coverage that is robust to node failures by periodically entering into probing state to check for the active nodes in the probing range. If there are no active nodes in the probing range, the node becomes active. In NSSS, a node measures its neighborhood redundancy as the union of the sectors covered by neighboring sensors in the node's sensing range and decides to turn off or not. In CCP a node finds all intersection points between the borders of its neighbors sensing radii and edges in the desired coverage area, and then deactivates itself if these intersection points are K-covered covered by at least K other sensor nodes).

The IDSQ [Zhao 2002] algorithm considers the information contribution of sensors against the cost of communicating with them. In this algorithm a central node calculates a probability distribution function that shows the target presence probability for each point of the field. Moreover this node improves the accuracy of this function by selecting one node (the highest payoff) at a time and incorporating the probability distribution function calculated by those nodes.

Algorithms such as these are simple in nature and focus mostly on sleep scheduling for coverage problems and the aggregation of information from sensors for estimation tasks. Specifically they do not usually include active control techniques. Most of the applications considered are centralized, where a central node receives the information from the rest of the network and decides on the best configuration of the nodes. Moreover, energy management over the mission period of the sensor network has not been addressed. This dissertation is an effort to address these issues and to study

different components of a general management methodology for wireless sensor networks.

Chapter 2: Minimum Requirements in Perimeter Surveillance

"Quality starts in the boardroom"

-W. Edwards Deming

2.1 Introduction

Surveillance missions defined an important class of applications for wireless sensor networks [Huang 2003]. Often surveillance networks are deployed in areas like battlefields where it is hard to position individual nodes precisely. So the nodes maybe relatively or highly randomly distributed in the region of interest. Achieving a desired target detection performance in the presence of randomness can be a considerable challenge. In designing an application the characteristics of the network should be known beforehand to be able to achieve a certain level of quality of service. Because of the uncertainty of the network topology, analysis for fixed networks can not be directly applied for random wireless sensor networks.

An important problem in these applications addressed in the literature is the *sensor coverage problem*. The extent of coverage is a measure of the *quality of service* of the sensing function. It is subject to a wide range of interpretations due to a large variety of sensors and applications [Meguerdichian 2001]. There are many different coverage measures including the area coverage and node coverage. For intrusion detection applications, the measure of coverage is the capability of the network to detect targets that move into or through a region of interest. In [Meguerdichian 2001], the paths in a network which are least likely to be detected by the sensors are found.

In wireless sensor networks energy efficiency is an important issue, because of the limited battery sources. Mechanisms that conserve energy resources are highly desirable, since they have a direct impact on network lifetime. Several protocols have been proposed to reduce the energy consumption by turning off the redundant sensors while maintaining the coverage at a desired level [Tian2002], [Ye 2003], [Clouqueur 2002]. Most of these protocols and algorithms are for a fixed network. In this work we will study the coverage problem as a design problem. Specifically, we shall investigate the fundamental coverage properties of a wireless sensor network which are governed by basic network parameters such as the number of the sensors deployed in the region. A well designed network will provide a protocol or algorithm the opportunity to perform at its best.

The surveillance scenario we are interested in this chapter is to detect intruders when they move into or out of a region of interest. In other words, we are monitoring the boundary of the region. The sensors are randomly scattered on the boundary and they report trespassing to a central node in the network. For the simplicity we will assume that the boundary is a circle. However all the results in here are valid for any shape of closed boundary. We will characterize the coverage properties on a circular boundary. Consequently, the requirements of the number of sensors or sensor density to achieve a target area coverage will be derived. We assume a Boolean sensing model where each sensor has a fixed sensing range. Moreover, we will introduce *detection percentage* as a measure of coverage for a boundary coverage problem. One other

issue that we will address is to predict the lifetime of the network by exploring the redundancy. Also we will explore the difference between the case where the number of sensors is fixed and the case where it is a random variable. In this regard, we consider two type of random networks, uniform and Poisson. In the uniform distribution a finite and fixed number of sensors are placed uniformly and independently around a circle. In the second case sensors are distributed according to a stationary Poison process with a fixed density.

In the first part of this chapter we are not concerned about the connectivity of the sensors and the communication and networking aspects of the network and we assume that all sensors are capable of communicating to a central point. In the second part we will investigate coverage and connectivity together.

2.2 Problem Formulation

Suppose *n* sensors are distributed independently and uniformly over circle with radius R. These sensors are located at points $\xi_1, \xi_2, \ldots, \xi_n$, with each sensor covering an arc of fixed length 2*a*. Let's denote these random arcs by $\chi_1, \chi_2, \ldots, \chi_n$ defined as follows:

$$\chi_i = \left\{ x \in R : |x - \xi_i| \le a \right\} \tag{2-1}$$

These arcs may overlap. We denote the Lebesgue measure on the circle by μ and the set-theoretical sum of the arcs by $\mathbf{X}$. A point on the circle is said to be *covered* if it is contained in at least one of the arcs $\chi_1, \chi_2, \ldots, \chi_n$.

We define the *coverage* as follows

$$C_{(n,a)} = \mu\left(\bigcup_{i=1}^{n} \chi_i\right) / \mu(R) \tag{2-2}$$

$C_{(n,a)}$ is the random proportion of the circumference that is contained in some arc. We define the *vacancy* to be

$$D_{(n,a)} = \mu\left(\bigcap_{i=1}^{n} \chi_i^c\right) / \mu(R) = 1 - \mathrm{C}_{(\mathrm{n,a})} \tag{2-3}$$

where χ_i^c denotes the complement of χ_i in R. $C_{(n,a)}$ and $D_{(n,a)}$ are random variables taking values in $[2a/\mu(R),1]$ and $[0,1-2a/\mu(R)]$. The moments of $C_{(n,a)}$ about zero are called *moments of coverage*. Those of $D_{(n,a)}$ are called *moments of vacancy*.

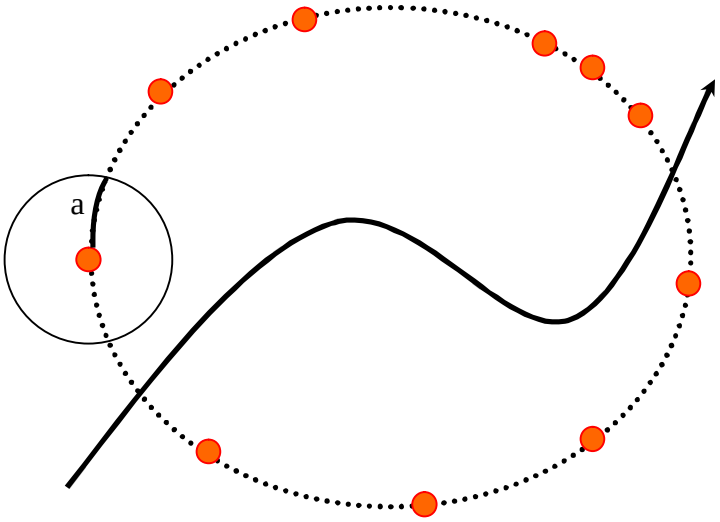

Figure 2-1. Perimeter Surveillance Problem. Each sensor node covers and arc of length *a* and the curve crossing the circle is the path of a moving target.

We are interested in characterizing the capability of this network to detect targets as they cross the circle at random locations as a function of the number of sensors and the sensing range. Also, we examine the lifetime of such a network and the effectiveness of sensor scheduling.

2.3 Minimum Number of Sensors

Sensors are uniformly distributed over the circle, so the probability density function for the center of the i^{th} arc will be one in the interval $[0, \mu(R)]$. The probability that a point on the circle is not covered by the sensors is equal to the probability that in an arc of length 2*a* centered around the point there are no sensors. So the probability that a point is covered is

$$\Pr(x \in \mathbf{X}) = 1 - (1 - \frac{2a}{\mu(R)})^n \tag{2-4}$$

It follows that the expected coverage is

$$E[C_{(n,a)}] = \int_0^1 P(x)dx = 1 - (1 - \frac{2a}{\mu(R)})^n \tag{2-5}$$

For a non-random regular network where the neighboring sensors have equal distance from each other the covered part of the circle increases linearly as the number of sensors increase, until the full coverage is achieved. For such a network

$$D_{(n,a)} = \mu\left(\bigcap_{i=1}^{n} \chi_i^c\right) / \mu(R) = 1 - \mathrm{C}_{(\mathrm{n,a})} \tag{2-6}$$

This suggests the following theorem.

Theorem 2-1- The factor increase in the minimum number of sensors to achieve a desired average coverage of $\overline{C}$ over a circle with a uniform random network compared to a non-random regular network is

$$K = \frac{2a}{\overline{C}\mu(R)} \log_{\left(1-\frac{2a}{\mu(R)}\right)} (1-\overline{C}) \qquad (2\text{-}7)$$

This is the price we have to pay for the randomness of the network. What we will earn in return is redundancy, which is examined in the next section.

2.4 Lifetime Extension

As we showed in the previous section, in a random network with sensors positioned according to a uniform distribution, we have to deploy more sensors to achieve the same coverage compared to a non-random regular network. However, these extra sensors introduce redundancy to the network. In such a network some of the sensors can be turned off without affecting the whole coverage. In this section we will compute this redundancy as another measure of the network performance that shows how much we can extend the lifetime of the network.

A sensor is a *redundant* if the arc it is covering is already covered by other sensors. Equivalently a sensor is redundant if the distance between its immediate left and right neighbors is smaller than $2a$; that is,

$$\Pr(\textit{redundancy}) = 1 - \left(1 - \frac{2a}{\mu(R)}\right)^{(n-1)} - \left(\frac{2a}{\mu(R)}\right)\left(1 - \frac{2a}{\mu(R)}\right)^{(n-2)} \qquad (2\text{-}8)$$

In wireless sensor network applications it is desirable to schedule the operation of sensor nodes by turning them ON and OFF in order to extend the lifetime of the network, while achieving the same level of performance. For this reason we are going to investigate the possibility of dividing the sensors into separate disjoint sets, with each set having a guarantied coverage level. Each set can be scheduled to be operational during a different time instance. This capability will give the network flexibility in increasing the lifetime, redundancy and fault tolerance.

The probability of a random uniform network to be divisable into k disjoint sets is equal to the probability that each point on the circle is covered by at least k sensors; this is often referred to as *k-node coverage*. The value k is sometimes called the *degree of coverage*. Higher node coverage helps to reduce the false alarms in the network. We will characterize the relationship between the number of sensors, the number of possible disjoint sets, and the degree of coverage. On the other hand the probability of a random uniform network to be divisable into k disjoint sets gives us the possibility of extending the lifetime of the network by a factor of k compared to a

network with all operating sensor nodes. We assume that each sensor node has a fixed initial energy and its energy consumption is only a function of its operating time.

Assume that the network has a mission time $T_{mission}$, which is the duration of time it is going to perform a surveillance task, and the time each sensor node can be operational is only T_{sensor} where $T_{sensor} < T_{mission}$. We would like to increase the lifetime of the network by a factor of $k_{life} = \left\lfloor \frac{T_{mission}}{T_{sensor}} \right\rfloor$, where $\lfloor . \rfloor$ is the integer part. Moreover, we want to have node coverage of k_{node}. We are looking for the optimal number of sensors to be scattered on the circle to achieve the above goals.

Theorem 2-2- Suppose n sensors are uniformly distributed on a circle R. The probability of extending the lifetime of the network by a factor of k_{life} by dividing the network into disjoint sets, with each set having a node coverage of k_{node} is

$$\begin{aligned} &\Pr(node\,\mathrm{cov}erage = k_{node}, life\,extension = k_{life}) \\ &= \Pr(k = k_{node}k_{life}\ disjo\,\mathrm{int}\ sets) \\ &= \Pr(each\ po\,\mathrm{int}\ on\,the\,circle\,is\,\mathrm{cov}ered\,by\,k\ sensors) \\ &= \binom{n}{k}\left(\frac{2a}{\mu(R)}\right)^{k}\left(1-\frac{2a}{\mu(R)}\right)^{(n-k)} \end{aligned} \tag{2-9}$$

We assume that each set of sensors is connected, and they can communicate to a central node. The above formula combined with (1-5) gives us the average coverage achieved with n sensors to have at least k disjoint sets. This is plotted in figure (1-2)

for $\frac{a}{\mu(R)} = 0.05$. For example, this figure suggests that 15 sensors can achieve coverage of 80 per cent if they all operate all the time, and the life time of this network can be extended by a factor of 3 if 28 sensors are used and by a factor of 5 if 55 sensors are used. These help a network designer to find the optimal number of sensor nodes for deployment.

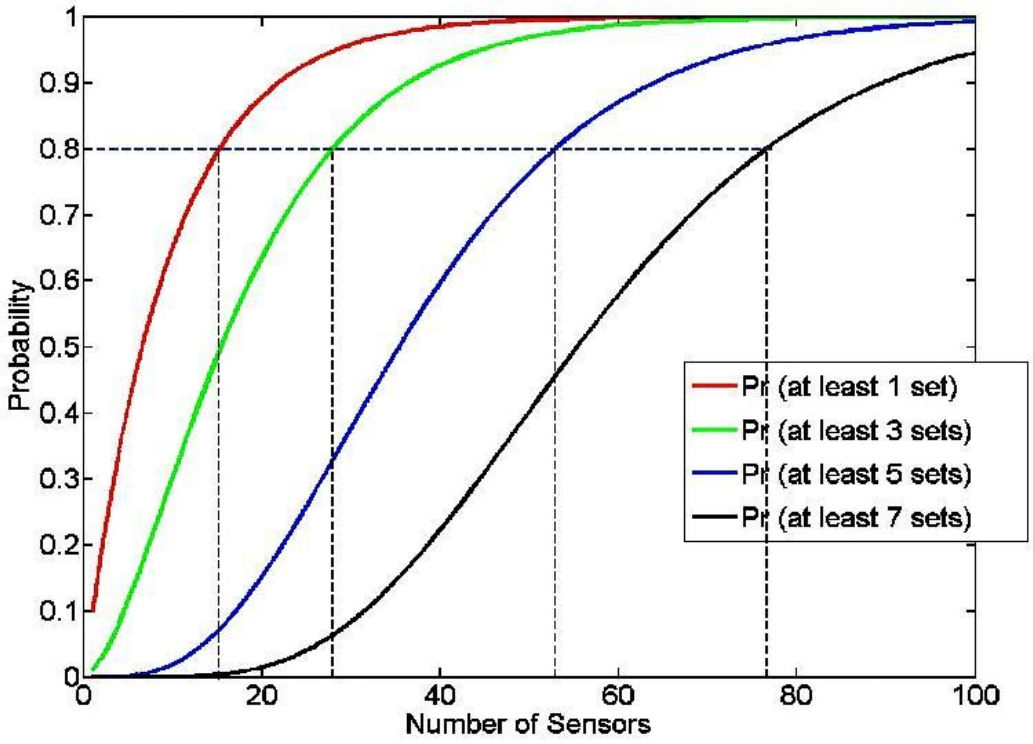

Figure 2-2. Probability of dividing the sensors into at least one (red), three (green), five (blue) and seven (black) disjoint sets versus the number of sensors.

There is an important question that how we should find the desired average coverage. In the following section we introduce *detection percentage* as a measure of coverage

that gives more insight into the quality of monitoring. The desired detection percentage can be translated into the average coverage, the measure we have used thus far.

2.5 Detection Percentage

In the previous section we calculated some coverage measures that are often used for showing the surveillance performance in detecting targets, e.g., coverage and node coverage. In this section we propose and calculate a different coverage measure: *detection percentage*. This measure is defined as the average percentage of targets detected crossing the circle. Each target crosses the circle in two points, and a target is detected if either of the crossing points is covered by the sensors.

We would like to calculate the probability that l targets cross the circle and exactly m of them are detected. For that reason we need the following theorem.

Theorem 2-3- If we randomly select m points on a circle with unit circumference monitored by a uniform network with n sensors with fixed sensing range of a, the probability that all these points fall into the covered part of the circle is the m th moment of the coverage $E[C_{(n,a)}^{m}]$.

Proof- Let $\mathbf{X}$ be the covered part of the circle for a given set of sensors. We randomly choose m points on the circle $x_1, x_2, \ldots, x_m$. Then for every point on the circle we define

$$g(x, \mathbf{X}) = \begin{cases} 1, & x \in \mathbf{X} \\ 0, & otherwise \end{cases} \tag{2-10}$$

It is clear that

$$\int_R g(x, \mathbf{X})\, dx = \mu(\mathbf{X}) \tag{2-11}$$

Thus,

$$\begin{aligned} E[C_{(n,a)}^m] &= E[\mu(\mathbf{X})^m] \\ &= \int_0^1 \mu(\mathbf{X})^m \Pr(\mu(\mathbf{X}) = u)\, du \\ &= \int_0^1 \left(\int_R g(x_1, \mathbf{X})\, dx_1 \ldots \int_R g(x_m, \mathbf{X})\, dx_m \Pr(\mu(\mathbf{X}) = u)) \right) du \\ &= \int_R \ldots \int_R \left(\int_0^1 g(x_1, \mathbf{X}) \ldots g(x_m, \mathbf{X}) \Pr(\mu(\mathbf{X}) = u))\, du \right) dx_1 \ldots dx_m \\ &= \int_R \ldots \int_R \Pr(x_1, x_2, \ldots, x_m \in X)\, dx_1 \ldots dx_m \end{aligned} \tag{2-12}$$

For the last step we have used the following property

$$g(x_1, \mathbf{X}) \ldots g(x_m, \mathbf{X}) = \begin{cases} 1, & x_1, \ldots, x_m \in X \\ 0, & otherwise \end{cases} \tag{2-13}$$

■

Theorem 2-4- If we randomly select m points on a circle monitored by n sensors uniformly distributed over the circle, the probability that all these points fall into the uncovered part of the circle is the m^{th} moment of vacancy $E[D_{(n,a)}^m]$.

The proof is similar to the proof of Theorem 3.

The m th moment of vacancy for n random arcs of length $2a$ on the circle R are given by [Siegel 1978]

$$E[D_{(n,a)}^m] = \binom{m+n-1}{n}^{-1} \sum_{l=1}^{m} \binom{m}{l} \binom{n-1}{l-1} \left(1 - \frac{2la}{\mu(R)}\right)_+^{m+n-1} \tag{2-14}$$

Also the moments of vacancy for n random arcs of length $2a$ on the circle R are given by [Siegel1978]

$$E[C_{(n,a)}^m] = 1 + \sum_{k=1}^{m} (-1)^k \binom{m}{k} E[D_{(n,a)}^k] \tag{2-15}$$

We combine the these theorems and results to prove the following theorem.

Theorem 2-5- Suppose we have a uniformly distributed network of sensors on a circle with n sensors. The probability that l targets cross the circle and exactly m of them are detected is

$$\Pr(l,m) = \binom{l}{m} \sum_{i=0}^{m} 2^{m-i} \binom{m}{i} E[C_{(n,a)}^{m+i}] E[D_{(n,a)}^{m-i}] E[D_{(n,a)}^{2(l-m)}] \tag{2-16}$$

Proof- The probability we are looking for can be stated as follows

$$\Pr(l,m) = \binom{l}{m} \Pr(m \text{ targets cross and all are detected}) \tag{2-17}$$
$$\Pr(l-m \text{ targets are not detected})$$

Each target represents two crossings, and a target is detected if one of the crossing points intersects with the covered part of the circle. We say a double detection happens if both crossing points of a target are detected.

$$\begin{aligned} &\Pr(m\ target\ cross\ and\ all\ are\ detected) \\ &= \sum_{i=0}^{m} \Pr(m\ detection | i\ double\ detection) \\ &= \sum_{i=0}^{m} 2^{m-i} \binom{m}{i} E[C_{(n,a)}^{m+i}] E[D_{(n,a)}^{m-i}] \end{aligned} \tag{2-18}$$

Combining together the above equations will give $\Pr(l,m)$.■

This probability function is concave. We call its expected value normalized to the number of targets the *detection percentage*. This measure is plotted in figure (1-3) as the number of sensors increases for $\frac{a}{\mu(R)} = 0.005$ and for $l = 20$ targets. On the same plot we can also see the coverage of the network. This figure suggests that to achieve a high percentage of crossing detection the network does not need to cover a large area of the circle. For example, 80 percent of the targets will be detected with

only 55 percent coverage of the circle. For a desired detection percentage, figure (1-3) helps us to find the corresponding coverage we need and thus the number of sensors we need to scatter on the circle.

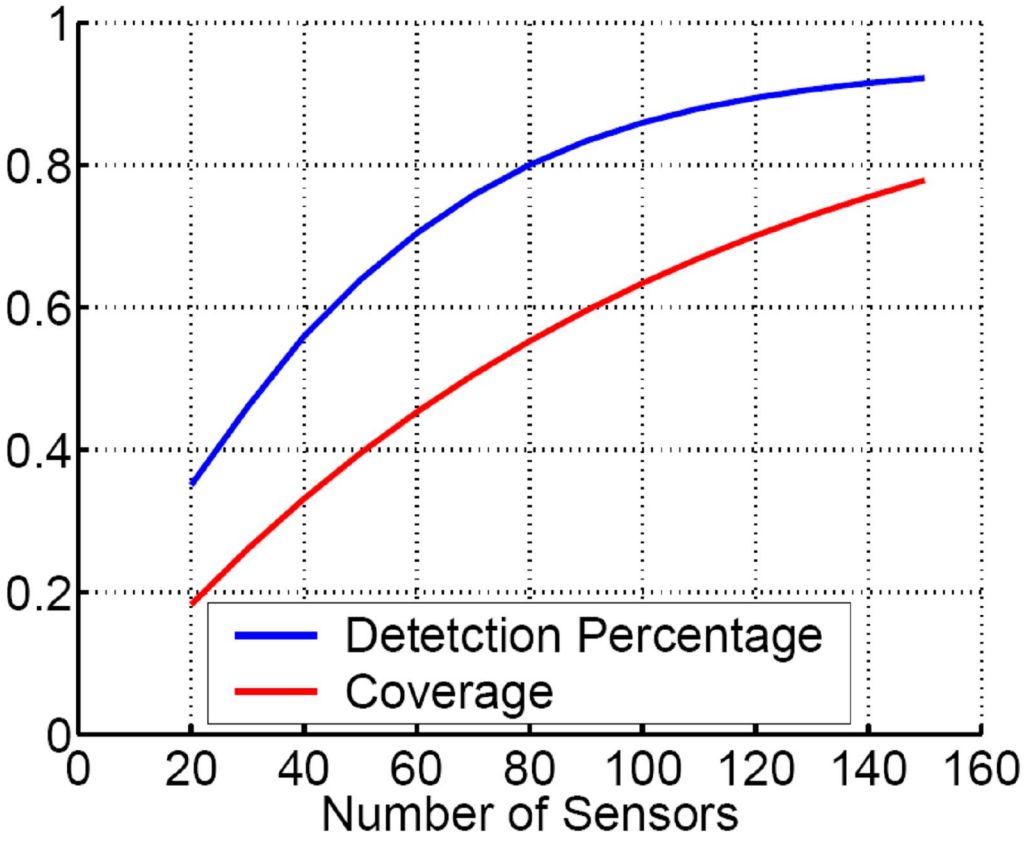

Figure 2-3. A high detection percentage can be achieved with a low coverage.

2.6 Poisson Network

Next we study the asymptotic behavior of the coverage of the network as the network gets larger. Suppose the circumference of the circle R and the number of sensors on

the circle, n, increases such that $\frac{n}{\mu(R)} \to \lambda$ where $\lambda > 0$. If S is an arbitrary arc in R, then the probability that there are k sensors in S equals

$$\begin{aligned} \Pr(k) &= \binom{n}{k}\left(\frac{\mu(S)}{\mu(R)}\right)^k\left(1-\frac{\mu(S)}{\mu(R)}\right)^{n-k} \\ &\to \frac{(\lambda\mu(S))^k}{k!}\exp(-\lambda\mu(S)) \end{aligned} \tag{2-19}$$

So, as R and n increase, the number of points in S converges to a Poisson distribution with mean $\lambda\mu(S)$, where λ is the density of sensors per unit length. Moreover, the numbers of sensors in disjoint arcs are independent. These are the two properties of a stationary Poisson process with mean λ. Such a sensor network is called a *Poisson network*. In this section we briefly review the behavior of a Poisson network, which are asymptotically derived from the results of the uniformly distributed network (UDN).

2.6.1 Minimum Number of Sensors

We should note that the total number of the sensors in a UDN is a fixed finite number; however it is a random number for a Poisson network. In a Poisson network the probability that a point on the circle is not covered by a sensor is equal to the probability that there are no sensors with a proximity of a around that point. Thus, the probability that a point is covered by the sensors is

$$P(x \in X) = 1 - e^{-2\lambda a} \tag{2-20}$$

And the average coverage is

$$E[C_{(a)}] = \int_0^1 P(x)\,dx = 1 - e^{-2\lambda a} \tag{2-21}$$

The minimum sensor density to achieve a desired average coverage of $\overline{C}$ is

$$\lambda^* = -\ln(1-\overline{C})/2a \tag{2-22}$$

2.6.2 Redundancy

A sensor is *redundant* if the arc it is covering is already covered by other sensors or equivalently if the distance between its immediate left and right neighbors is smaller than $2a$. This probability for a Poisson network is

$$\Pr(\textit{redundancy}) = 1 - e^{-2\lambda a} - 2a\lambda e^{-2\lambda a} \tag{2-23}$$

2.6.3 Lifetime Extension

The probability of extending the lifetime of a Poisson network by a factor of k_{life} by dividing the network into disjoint sets, with each set having a node coverage of k_{node} is

$$\begin{aligned} & pr(\textit{node coverage} = k_{node},\ \textit{life extension} = k_{life}) \\ &= \Pr(k = k_{node}k_{life}\ \textit{disjoint sets}) \\ &= \Pr(\textit{each point on the circle is covered by k sensors}) \\ &= \frac{(2\lambda a)^k}{k!} e^{-2\lambda a} \end{aligned} \tag{2-24}$$

2.7 Connectivity

The average coverage of a network is highly affected by the connectivity of the nodes. In this section we related this to the perimeter coverage problem. We use a probabilistic approach to find the relationship between the density of the sensors and the covered part of the boundary and the probability that a node is connected to the central node. Our analysis is both for homogenous and inhomogeneous networks and it helps in designing a network with a required performance. Different protocols have been designed to schedule the node operation in such a way that the communication graph is always connected. In [Wang 2005] the problem of coverage and connectivity is considered together.

In this section we offer a different perspective on the connectivity and coverage problem. We will formulate the quality of service in covering an area while maintaining the connectivity of the nodes in a probabilistic framework as a function of network parameters. The network parameters include the density of nodes and the communication and sensing ranges.

The surveillance scenario we are interested in here is to detect intruders when they move into or out of a region of interest. In other words, we are just monitoring the boundary of the region. The sensors are randomly scattered on that boundary and they report intrusion to a central node on the boundary. For the simplicity we will assume

that the boundary is a circle. However all the results are valid for any closed boundary.

Here, we study coverage and connectivity for both homogenous and inhomogeneous one-dimensional networks.

2.7.1 Connectivity problem Formulation

Suppose n nodes are distributed independently and uniformly over circle R. For simplicity we assume that the circumference of the circle is 1. The number of nodes n may be fixed or a random variable.

Nodes send their messages directly or by relaying through the nodes to a central node. Without loss of generality we assume that the distance on the circle is measured counterclockwise from the central node. The distance between two nodes is the length of the shortest arc on the circle connecting those nodes. We assume that nodes have a limited and fixed communication range r, where $r < 1$. Thus, two nodes can communicate if and only if the distance between them is less than r. The neighbor nodes that can communicate with each other form *clusters*. The number and the length of the clusters are related to the number of sensors and their communication range. Those clusters that do not contain the central node are isolated and cannot communicate their messages to the central node. There is only one cluster that is connected to the central node. We refer to the members of that cluster as connected nodes. We would like to know the number of sensors that we need to scatter over the circle to achieve a desired coverage by the connected nodes. We will investigate the

minimum coverage provided by the connected nodes, and will find the design parameters of the network to achieve the minimum required coverage. We consider design parameters such as the number of sensors or sensor density, communication range and sensing range.

We also consider an inhomogeneous network where there are two types of sensors: regular and master nodes. The regular nodes transmit their messages to the nearest master node. The master nodes are capable of transmitting a longer distance than regular nodes. Thus, a message is relayed through a master nodes to the central node. A master node is isolated if it cannot communicate to other master nodes. A regular node is isolated if it cannot communicate with a master node.

In both homogenous and inhomogeneous networks, we will find lower bounds on the probability that a regular node is connected to the central node. And from there we will find the minimum achievable coverage.

2.7.2 Homogenous Network

In this section we consider a homogenous network with nodes randomly distributed on the circle according to a Poisson distribution with density λ. The total number of nodes is not fixed in this setting. A node at location *x* can send its message to the central node, if $x \le r$ or if the interval between that node and the origin is completely occupied by other nodes that are not distant more than *r* from each other. A node can be connected to the central node from its left or its right or both. We denote by $P_c(x)$

the probability that an arbitrary node at location x is connected to the central node from one side. Then the probability that the node is connected to the central node $P_{co}(x)$ is:

$$P_{co}(x) = P_c(x) + P_c(1-x) - P_c(x)P_c(1-x) \tag{2-25}$$

The lower bound on $P_{co}(x)$ is [Dousse 2002]:

$$P_{co}(x) \geq 1 - (1+\lambda/2)^2 e^{-2\lambda r} \tag{2-26}$$

It is clear that the closer the node is to the central node the more likely it is connected. The nodes that can communication with each other make clusters. Evidently, clusters are isolated and only the cluster that contains the central node is capable of sending out its messages. We refer to this cluster as the *connected cluster*. Coverage of the circle (which includes communication to the central node) is only provided by the connected cluster. In the next theorem we find a lower bound on the coverage.

Theorem 2-6 -The lower bound on the expected coverage provided by the connected cluster is:

$$E[C_{(s)}^{\text{hom}\,ogeneous}] \geq (1-e^{-2\lambda s})\left(1-(1+\lambda/2)^2 e^{-2\lambda r}\right) \tag{2-27}$$

$C_{(s)}$ is the coverage provided by nodes that each cover an arc of length $2s$.

Proof- The average length of the connected cluster $\overline{X}$ is the average of the connection probability of a node to the central node $\int_0^1 P_{co}(x)\,dx$. The lower bound on $\overline{X}$ is:

$$\begin{aligned}\overline{X} &= \int_0^1 P_{co}(x)\,dx \\ &\geq \int_0^1 \left(1-(1+\lambda/2)^2 e^{-2\lambda r}\right)dx \\ &\geq 1-(1+\lambda/2)^2 e^{-2\lambda r}\end{aligned} \tag{2-28}$$

The probability that an arbitrary point x on the circle is covered by a node is:

$$P(x) = 1 - e^{-2\lambda s} \tag{2-29}$$

Therefore, the average coverage by the connected cluster is

$$E[C_{(s)}^{\text{hom}\,ogeneous}] = \int_0^{\overline{X}} P(x)\,dx \tag{2-30}$$

This is lower bounded by:

$$E[C_{(s)}^{\text{hom}\,ogeneous}] \geq (1-e^{-2\lambda s})\left(1-(1+\lambda/2)^2 e^{-2\lambda r}\right) \tag{2-31}$$

■

This inequality gives the minimum density to achieve a desired coverage level for a given communication range.

2.7.3 Inhomogeneous Network

Random distributions of nodes result in isolated clusters of nodes, and only the cluster that contains the central node is able to report its messages. We will be able to connect isolated clusters to the central node if we add nodes that are capable of communicating over a larger distance. We will refer to these nodes as *master nodes*. They have more initial energy or they expend energy faster and their role is to relay messages of a cluster through other master nodes to the central node. Obviously this will connect more clusters to the central node and will increase the coverage. In this section we want to know how the coverage changes with this enhancement, and we want to know the minimum number of master nodes needed to achieve a desired performance level.

2.7.4 Uniform Distribution

Assume that we randomly and uniformly distribute n master nodes on the circle. A master node can communication with other master nodes within a distance less than a. A master node can be connected to the central node from its left or its right or both. Denote by $P_m(x)$ the probability that a master node at location x is connected to the central node through q other master nodes. This probability is [Foh 2004]

$$P_m(x) = \begin{cases} 0, & q < (x-a)/a \\ (1-(1-\frac{a}{x})^q)^{q+1}, & q < (x-a)/a \end{cases} \tag{2-32}$$

The probability that a master node is connected to the central node $P_{cm}(x)$ is:

$$P_{cm}(x) = P_m(x) + P_m(1-x) - P_m(x)P_m(1-x) \tag{2-33}$$

The lower bound of P_{cm} happens when the master node is at $x = 0.5$ and there are $n^* = \lfloor (n-1)/2 \rfloor$ other nodes on each side,

$$\begin{aligned} P_{cm^-}^{Uniform} &= \inf_{0 \le x \le 1} P_{cm}(x) = P_{cm}(0.5) \\ &= 2P_m(0.5) - P_m^2(0.5) \\ &= 2\left(1-(1-2a)^{n^*}\right)^{(n^*+1)} - \left(1-(1-2a)^{n^*}\right)^{2(n^*+1)} \end{aligned} \tag{2-34}$$

Next we calculate the probability that a regular node is connected to the central node. Suppose an arbitrary regular node is located at x and the closest master nodes on the right and left side of his node are located at X and Y. The probability that this regular node is connected to the central node is:

$$\begin{aligned} P_{cg}(x) = {} & P_c(|X-x|)P_{cm}(X) + P_c(|Y-x|)P_{cm}(Y) \\ & - P_c(|X-x|)P_{cm}(X)P_c(|Y-x|)P_{cm}(Y) \end{aligned} \tag{2-35}$$

Theorem 2-7- If L is such that $L \ge 2r$ then $P_{cg}(x)$ the probability that a regular node located at x, with $x \ge r$, is connected to the central node is lower bounded by:

$$P_{cg}(x) \ge P_{cg^-}^{Uniform} = 1-(1+\frac{\lambda L}{2})^2 e^{-2\lambda r} + 2\left(P_{cm^-}^{Uniform} - 1\right)\left(1-e^{-\lambda r} + \lambda r e^{-\lambda r}\right) \tag{2-36}$$

Proof- The lower bound on this probability is achieved for the worst case when $x = 0.5$, $X = x + L/2$ and $Y = x - L/2$ for some L. Then,

$$P_{cg}(x) \geq 2P_c(L/2)P_{cm}(0.5-L/2) - P_c(L/2)^2 \qquad (2\text{-}37)$$
$$\geq \left\{2(1-e^{-\lambda r})e^{-\lambda(L-4r)e^{-\lambda r/2}} - 2\lambda r e^{-\lambda r}\right\} P_{cm^-}^{Uniform} - (1-e^{-\lambda r})^2 e^{\lambda(L-2r)e^{-\lambda r}}$$

By using Taylor series and some manipulations the above inequality yields. ■

The spacing between two master nodes L, is a random variable. The collection of n master nodes and the central node divides the circle into $n+1$ spacing. The spacing is identically distributed. Their common distribution is given by:

$$\Pr(L \leq l) = 1-(1-l)^n \qquad (2\text{-}38)$$

with mean value $E[L] = \frac{1}{n+1}$.

Theorem 2-7 gives the minimum spacing between master nodes required to achieve a minimum connectivity probability between regular nodes and the central node, and equation (2-38) gives a measure of uncertainty on the number of master nodes to achieve that minimum spacing. Note that n also affects $P_{cg^-}^{Uniform}$. From another point of view $P_{cg^-}^{Uniform}$ shows the number of connected nodes. This will give an estimate on the minimum achievable coverage by the following theorem:

Theorem 2-8-The lower bound on the average coverage provided by the connected nodes is:

$$E[C_{(s)}^{Uniform}] \geq (1-e^{-2\lambda s})P_{cg^-}^{Uniform} \qquad (2\text{-}39)$$

where $P_{cg^-}^{Uniform}$ is derived in theorem 2.

2.7.5 Poisson Distribution

In this section we consider the case that master nodes are randomly placed with a Poisson distribution with density ν. A master node can communication with other master nodes that are at a distance less than a. We will be able to derive the lower bounds on connectivity and coverage as we did in the previous section. Without going through the proofs we have the following theorems:

Theorem 2-9- If L is such that $L \geq 2r$ then $P_{cg}(x)$, the probability that a regular node located at x, with $x \geq r$, is connected to the central node is lower bounded by:

$$P_{cg}(x) \geq P_{cg^-}^{Poisson} = 1-(1+\frac{\lambda L}{2})^2 e^{-2\lambda r} - 2(1+\nu/2)^2 e^{-2\nu a}(1-e^{-\lambda r}+\lambda r e^{-\lambda r}) \qquad (2\text{-}40)$$

Proof- We used the lower bound at (1-26), for the worst case as we did in the uniform case. ■

For master nodes with Poisson distribution the spacing is exponential and is given by:

$$\Pr(L < l) = 1 - e^{-\nu l} \qquad (2\text{-}41)$$

with mean value $E[L] = \frac{1}{\nu}$.

And for the lower bound of coverage we have:

Theorem 2-10- The lower bound on the average coverage provided by the connected nodes is:

$$E[C_{(s)}^{Poisson}] \geq (1-e^{-2\lambda s})P_{cg^-}^{Poisson} \tag{2-42}$$

where $P_{cg^-}^{Poisson}$ is derived at theorem 4.

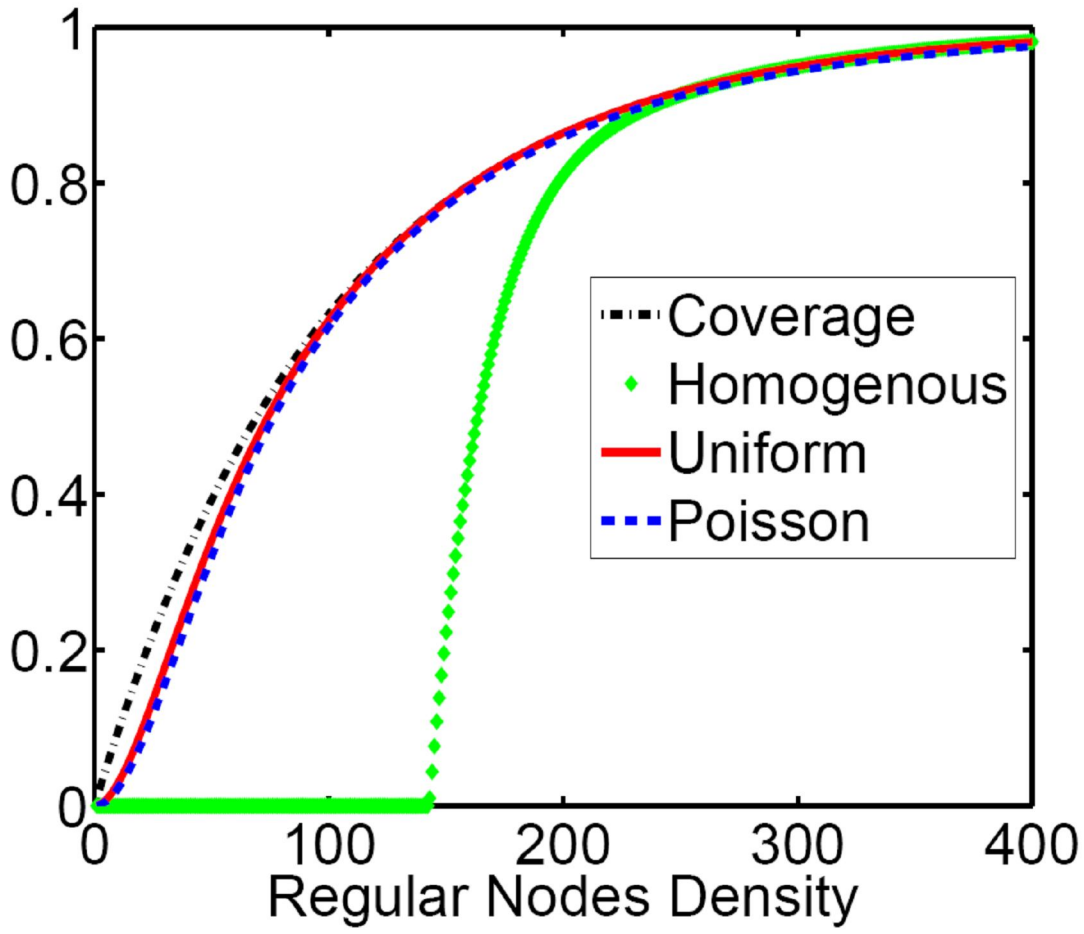

Figure 2-4. Average coverage achieved by a fully connected network, a homogeneous network and an inhomogeneous network with uniformly distributed and Poisson distributed master nodes versus the density of regular nodes

2.7.6 Analysis

The lower bounds on average coverage for the homogenous and homogenous network that we discussed are plotted in figure (1-5) as λ the density of the regular nodes increases. For comparison we have also plotted the average coverage in a homogenous network with the assumption that there is full connectivity between the nodes, which is,

$$E[C_{(s)}] = 1 - e^{-2\lambda s} \tag{2-43}$$

The node parameters are $r = 0.03$, $a = 0.2$ and $s = 0.005$. The number of master nodes n in the uniformly distributed network and the density of the master nodes ν in Poisson network is selected such that they produce the same performance. The lower bound for these two networks are for $n = 40$ and $\nu = 30$. In both cases the average spacing between master nodes is about 0.03, and the probability that the distance between two master nodes is less than that is 0.64. With this spacing the probability that a master node is connected to the central node is very close to 1. Thus, the above choice of parameters results in full connection of master nodes. In the lower density of regular nodes the coverage probability is small. As the density gets larger the coverage probability goes to 1. Adding master nodes in low densities increases the coverage significantly. At lower densities the network is sparse and there are many isolated clusters. Adding master nodes connects these isolated clusters to the central node. As the density goes beyond a certain value all the networks

behave the same. At that density the regular nodes are fully connected and the role of master nodes in decreasing isolation becomes negligible. However this density is much smaller for the inhomogeneous network. The reason that beyond that density the average coverage is still below 1 is because the sensing range is smaller than the communication range and the full connectivity does not guarantee full coverage. As the density keeps increasing the full coverage is achieved. After full connectivity we can approximate the average coverage by (2-43). This gives the minimum density to achieve a desired average coverage of $\overline{C}$:

$$\lambda^* = -\ln(1-\overline{C})/2s \tag{2-44}$$

So if the number or density of master nodes is selected in such a way that they are almost surely fully connected, then with a low density of regular nodes the full connectivity of the network is achieved. In this case the dominant factor in choosing the density of regular nodes will become the average coverage that we expect from the network. From the other hand if the number or density of master nodes is not enough to guarantee full connectivity of master nodes, and if the regular nodes are only allowed to communicate their messages through the master nodes as the case in our analysis, then increasing the density of regular nodes will not result in full connectivity and full coverage will never happen. The asymptotic average coverage of the Poisson network will be then,

$$E[C_{(s)}] = (1-e^{-2\lambda s})\left(1-2(1+\nu/2)^2 e^{-2\nu a}\right) \tag{2-45}$$

as is predicted by Theorem 2-4, and for the Uniform network it is,

$$E[C_{(s)}] = (1 - e^{-2\lambda s})\left(2\left(2\left(1 - (1-2a)^{n^*}\right)^{n^*+1} - \left(1 - (1-2a)^{n^*}\right)^{2(n^*+1)} - 1 \right) \right) \tag{2-46}$$

as is predicted by Theorem 2-3 where $n^* = \lfloor n - 1 \rfloor / 2$.

Chapter 3: Spatial Management

"It is not enough to do your best; you must know what to do, and then do your best"

-W. Edwards Deming

3.1 Introduction

In this chapter we study the spatial management of energy in a wireless sensor network. As a motivating application we use a distributed parameter estimation based on the consensus among the nodes in the network. First, we investigate the consensus problem in wireless sensor networks in general and look into its characteristics. In developing consensus protocols the convergence conditions and increasing the convergence speed is of particular interest. We will give a summary on this and then demonstrate an algorithm for distributed parameter estimation using the consensus concept. And we show how the dynamics of the process vary among the nodes in the network as a function of the relative position of the node in the communication graph. And we show this is closely related to the centrality concept in networks. We then leverage a well known eigenvector centrality concept to assign to each node a weight. A node will use the weights assigned to its neighbors to control the interactions with them. Then we show an algorithm for a constrained localized estimation where the total energy budget is limited and the goal is to minimize the uncertainty of the parameter estimation. The problem is formulated as a team formation problem, where an efficient group of nodes are to be selected to perform the task.

3.2 Consensus

Consider a network of n nodes that collaborate to compute a scalar function of their data $G(\mathbf{x})$ where

$$\mathbf{x} = [x_1, x_2, ..., x_n]^T \tag{3-1}$$

This vector is called the *state of the network* and x_i is the scalar noisy observation of the environment parameter θ at the ith sensor node. In this work we consider algorithms that generate at each node i a sequence $x_i[n]$ of approximations to $G(\mathbf{x})$. $G(\mathbf{x})$ is an estimate of a function of θ. We assume that each node can establish bidirectional communication with a subset of the nodes in its neighborhood.

Let's focus on local interactions modeled by a first order LTI fusion rule

$$\mathbf{x}[k+1] = W\,\mathbf{x}[k] \quad \text{with} \quad \mathbf{x}[0] = \xi \tag{3-2}$$

This is an iterative algorithm with the objective of asymptotically converging to the desired function $G(\mathbf{x})$,

$$\lim_{k\to\infty} \left\| \mathbf{x}[k] - \mathbf{1}G(\mathbf{x}) \right\| = 0 \tag{3-3}$$

where $\|.\|$ is the ordinary Euclidean norm, and $\mathbf{1}$ is $n \times 1$ vector of 1's. $\mathbf{x}[k]$ is the vector of the approximations and we call it the *state of the network* at k

Let's review some definitions and results in the consensus problem related literature that may be used in this chapter:

Adjacency Matrix of a Network- Matrix $\mathbf{A} = [a_{ij}]$ whose (i, j) entry is equal to the number of edges originating at the node i and terminating at node j. Elements on the diagonal are zero.

Asymptotically Converging – A rule is asymptotically converging to the desired scalar function $G(\mathbf{x})$, if the sequence $\mathbf{x}[n]$ satisfies,

$$\lim_{n\to\infty} \| \mathbf{x}(n) - \mathbf{1}G(\mathbf{x}) \| = 0 \tag{3-4}$$

Geometrically Convergence- $\| \mathbf{x}(n) - \mathbf{1}G(\mathbf{x}) \| \leq c\rho^n$

Where ρ is a positive constant smaller than 1 and the smaller the value of ρ the faster the convergence of the algorithm.

Approach: Direct vs. Iterative Methods- There is a variety of methods for solving distributed estimation problems. These methods can be classified as *direct* and *iterative* methods. Direct methods find the exact solution by considering all the available data at the same time. Iterative methods do not obtain an exact solution in finite time, but they converge to a solution asymptotically. Iterative methods are preferred to direct methods when the number of nodes is large or the communication graph is sparse (it will require less storage and communication requirements).

Linear Iterative Process-Sensor nodes start with an initial state $\mathbf{x}[0]$ and evolve iteratively by the rule,

$$\mathbf{x}[n] = \mathbf{W}\,\mathbf{x}[n-1] \tag{3-5}$$

Or,

$$\mathbf{x}[n] = \mathbf{W}^n\,\mathbf{x}[0] \tag{3-6}$$

We will refer $\mathbf{W}$ as the local interaction or local rule matrix.

Convergence Condition- A linear iterative process converges for every initial state $\mathbf{x}[0]$ if and only if the eigenvalues of $\mathbf{W}$ distinct from 1 have modulus $|\lambda| < 1$, and, if $\lambda = 1$ is an eigenvalue, its eigenspace is of full rank.

Perron's Theorem [Perron 1907, MacCluer 2000]- The eigenvalue of largest absolute value of a positive (square) matrix is both simple and positive and belongs to a positive eigenvector. All other eigenvalues are smaller in absolute value.

When $\mathbf{W} > 0$ (is positive-definite), the iterative process will converge when the dominant eigenvalue λ_1 is less than 1; the process converges to 0 if $\lambda_1 < 1$, and when $\lambda_1 = 1$ to the component of the initial condition in the eigenspace spanned by the associated eigenvector $\mathbf{v} > 0$.

When $\mathbf{W} \geq 0$, it has a nonnegative dominant eigenvalue λ_1 belonging to a nonnegative eigenvector $\mathbf{v} \geq 0$, but there may be other eigenvalues with the same modulus. Having multiple eigenvalues with the same modulus complicates convergence. If $\lambda_1 = 1$ it may produce periodic or even orbits that diverge to infinity.

Average Consensus: The goal is to find a local rule matrix that guarantees asymptotic convergence to the average of the sensor readings,

$$G(X) = \frac{1}{N}\sum_{i=1}^{N} x_i = \frac{1}{N}\mathbf{1}^T X \tag{3-7}$$

A set of sufficient conditions that guarantee asymptotic convergence is [Scherber 2005],

$$\begin{aligned} &W\mathbf{1} = W\mathbf{1} = \mathbf{1} \\ &\lambda_1(W) = 1 \\ &|\lambda_i| < 1, \quad \text{for } 2 \leq i \leq N \end{aligned} \tag{3-8}$$

The following rule satisfies the above conditions,

$$\mathbf{W} = \mathbf{I} + \rho L \tag{3-9}$$

L is the **Laplacian** of the communication graph, defined as follows,

$$L_{ij} = \begin{cases} 1, \text{ if i} \sim \text{j} \\ 0, \text{ otherwise} \end{cases} \tag{3-10}$$

$$L_{ii} = -\sum_{j \neq i} L_{ij} \tag{3-11}$$

The choice of ρ determines the rate of convergence of the sequence $\mathbf{x}[n]$ to the desired function. The convergence rate is governed by the second largest eigenvalue. Thus, for the fastest convergence ρ should be selected such that minimizes the following objective function [Scherber 2005],

$$\max_{2 \leq i \leq N} | \lambda_i | \tag{3-12}$$

This yields $\lambda_2 = -\lambda_N$ and is given by,

$$\rho_{\min \max} = -\frac{2}{\mu_2 + \mu_N} \tag{3-13}$$

This requires global knowledge of the network. Instead, a good selection of ρ that requires only local knowledge can be,

$$\rho_{\max - con} = \frac{1}{\max_i | d_{ii} |} \tag{3-14}$$

This calculation just requires the nodes to find the maximum of $| d_{ii} |$, which is the number of connections (degree) of the ith node, via voting. The justification for this choice is the following inequality,

$$\max_i | d_{ii} | + 1 \leq | \mu_N | \leq 2 \max_i | d_{ii} | \tag{3-15}$$

And assuming that W has the above mentioned properties, the unity vector is an eigenvector of $W[k]$ associated eigenvalue of 1. The objective is to select $h[k]$ so that the rest of the resulting eigenvalues are as small as possible in the magnitude.

Continuous Consensus Problem: The average consensus problem explained above can be studied from a different point of view. That algorithm can be looked at as the discrete version of this continuous problem: starting from an initial state $x(0)$, the dynamic of the nodes is defined by

$$\dot{x}(t) = Lx(t) \tag{3-16}$$

With the goal of reaching to average initial state

$$\lim_{t\to\infty} X(t) = \frac{1}{n}\sum_{i=1}^{n} x_i[0] \tag{3-17}$$

We define the following quadratic function, which is the Lyapunov function in the convergence (stability) analysis,

$$f(x) = \frac{1}{2}x(t)^T Lx(\mathrm{t}) \tag{3-18}$$

Where L is the Laplacian and is always positive semi-definite ($\forall i, \lambda_i \geq 0$). This quadratic function is shaped like an elliptic paraboloid bowl and its minimum is a line that runs through the bottom of the valley. The Laplacian matrix is usually symmetric. With symmetry assumption the gradient of the quadratic function is

$$f'(x) = Lx(t) \tag{3-19}$$

The consensus point belongs to the intersection of two spaces:

(1) The space where all the nodes have the same value which is in the minimum of the quadratic function

$$s_1 = \{x \in R^n \mid x_1 = x_2 = \ldots = x_n\} \tag{3-20}$$

The minimum of the quadratic function is where its gradient is zero

$$f'(x) = 0 \tag{3-21}$$

(2) The space where the average of the states is the same as the average of the initial state

$$s_2 = \{x \in R^n \mid x_1 + x_2 + \ldots + x_n = c, c = \sum_{i=1}^{n} x_i(0)\} \tag{3-22}$$

The consensus point is the point that belongs to the intersection of these two spaces.

3.3 Distributed Estimation

In this section we examine a distributed parameter estimation problem based on the consensus among the nodes [Xiao 2005]. We assume that all nodes make a noisy measurement of a parameter. We denote by x_i the measurement of the ith sensor node.

$$x_i = a_i \theta + v_i, \quad i = 1,...,n \tag{3-23}$$

Here $x_i \in IR$, a_i is a known coefficient that relates the unknown parameter to the i th sensor measurement, and v_i is the noise of measurement with Gaussian distribution with zero mean and variance σ_i. We assume that the measurement noises are independent.

To get insight into the process, suppose that all the sensor nodes have the same measurement noise variance σ. Each sensor can estimate the mean based on its own data.

$$\hat{\theta}_i = x_i \tag{3-24}$$

This estimate has an error variance $E[(\hat{\theta}_i - \theta)^2]$ of σ. A node can improve its estimate by taking more measurements. After taking m measurements the error variance will decay to σ / m. However, a better and faster estimate with error variance $\sigma /(nm)$ can be obtained if n sensor nodes exchanged their estimates and computed the average of the estimates. Suppose each sensor node has only one measurement and they exchange it to make an estimate with final error variance of σ / n.

Now consider the original problem. A central node with access to all the measurements will make parameter estimation based on the following aggregation

$$x = a\theta + v \tag{3-25}$$

Where

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad A = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}, \quad v = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

The covariance matrix of v is $\Sigma = diag(\sigma_1, \sigma_2, \cdots, \sigma_n)$.The maximum likelihood (ML) estimate of θ is a weighted least-squares approximate solution

$$\begin{aligned} \hat{\theta}_{ML} &= \left(A^T \Sigma^{-1} A\right)^{-1} A^T \Sigma^{-1} x \\ &= \left(\sum_{i=1}^{n} A_i^T \sigma_i^{-1} A_i\right)^{-1} \sum_{i=1}^{n} A_i^T \sigma_i^{-1} x_i \end{aligned} \tag{3-26}$$

This estimation is unbiased with an error covariance

$$Q = E[(\hat{\theta}_{ML} - \theta)^2] = \left(A^T \Sigma^{-1} A\right)^{-1} \tag{3-27}$$

If the noises are not Gaussian, but are independent with zero mean, this estimation is the linear minimum-variance unbiased estimation given the measurement. How can this estimation be performed in a distributed fashion where only local exchange of information is possible? Two possible options are:

(1) **Flooding:** Each sensor broadcasts its data into the network through multi-hop communications until all the nodes have access to all the data. This method requires a large amount of communication and may not be practical in a wireless sensor network with limited resources (energy and bandwidth)

(2) **Iterative method:** It is possible to develop iterative estimation algorithms that compute for each sensor an estimate of the unknown parameter that eventually converge to the estimate achieved by the central estimation. This is based on simple average consensus. In this scheme each node iteratively updates its data by a linear weighted average of its neighbors' data until all the data is diffused in the network and a global ML estimate of the parameter is achieved.

The following algorithm computes the ML estimate of the unknown parameter. Each node maintains local composite information $P_i(t)$ and a local composite information state $q_i(t)$, initialized at $t = 0$ as

$$P_i(0) = a_i^2 \sigma_i^{-1} \qquad (3\text{-}28)$$
$$q_i(0) = a_i \sigma_i^{-1} x_i$$

Then the node computes the average consensus for $P_i(t)$ and $q_i(t)$

$$P_i(t+1) = W_{ii} P_i(t) + \sum_{j \in N_i} W_{ij}(t) P_j(t) \qquad (3\text{-}29)$$
$$q_i(t+1) = W_{ii} q_i(t) + \sum_{j \in N_i} W_{ij}(t) q_j(t)$$

In the limit

$$\lim_{t\to\infty} P_i(t+1) = \frac{1}{n}\sum_{i=1}^{n} a_i^2 \sigma_i^{-1} \qquad (3\text{-}30)$$
$$\lim_{t\to\infty} q_i(t+1) = \frac{1}{n}\sum_{i=1}^{n} a_i \sigma_i^{-1} x_i$$

Each node makes the following estimate at each time

$$\hat{\theta}_i(t) = P_i(t)^{-1} q_i(t), \quad i = 1,2,...,n \qquad (3\text{-}31)$$

In the limit this estimate convergences to

$$\lim_{t\to\infty} \hat{\theta}_i(t) = \lim_{t\to\infty} P_i(t)^{-1} q_i(t) = \left(\sum_{i=1}^{n} a_i^2 \sigma_i^{-1}\right)^{-1} \sum_{i=1}^{n} a_i^T \sigma_i^{-1} x_i, \quad i = 1,2,...,n \qquad (3\text{-}32)$$

This limit is the ML estimate.

Now, let's study the properties of the intermediate estimates $\hat{\theta}_i(t)$. There exist scalar coefficients $\Phi_{ij}(t)$, that we call a diffusion function, such that

$$P_i(t) = \sum_{j=1}^{n} \Phi_{ij}(t) a_j^2 \sigma_j^{-1} \qquad (3\text{-}33)$$
$$q_i(t) = \sum_{i=1}^{n} \Phi_{ij}(t) a_i^T \sigma_i^{-1} x_i$$

The coefficients $\Phi_{ij}(t)$ depend on the underlying local rule algorithm and the topology of the network. $\Phi_{ij}(t)$ is the ij th entry of the matrix $\Phi(t) = W^t$. In the limit $\Phi(t)$ converges to $(1/n)\mathbf{1}\mathbf{1}^T$ and all the entries converge to $1/n$ (if the algorithm converges).

They have the following properties,

(1) $0 \le \Phi_{ij}(t) \le 1, \quad \forall i, j, t$

(2) If the local rules are convergent, then the coefficients $\Phi_{ij}(t)$ converge geometrically as t goes to infinity. We denote this limit by $\Phi_i(t)$.

(3) $\sum_{t=0}^{\infty} \frac{\Phi_{ij}(t)}{t!} \le \sum_{t=0}^{\infty} \frac{\lambda^t}{t!} = e^{\lambda}$, where λ is the second largest eigenvalue of W.

(4) $\sum_{j=1}^{n} \Phi_{ij}(t) = 1$

All the intermediate estimates are unbiased

$$E[\hat{\theta}_i] = \theta, \quad i = 1,...,n \tag{3-34}$$

And the error covariance matrix at each node converges to that of global ML solution

$$\lim_{t\to\infty} E[(\hat{\theta}_i - \theta)^2] = \left(A^T \Sigma^{-1} A\right)^{-1}, \quad i = 1,...,n \tag{3-35}$$

However at time *t* the intermediate covariance matrix is

$$\begin{aligned} Q_i(t) &= E[(\hat{\theta}_i - \theta)^2] \\ &= P_i(t)^{-2} \left(\sum_{j=1}^{n} \Phi_{ij}^2(t) a_j^2 \sigma_j^{-1} \right) \end{aligned} \tag{3-36}$$

As t increases the coefficients $\Phi_{ij}(t)$ all converge to $1/n$ and the error covariance $Q_i(t)$ converges to its limit.

As we see the quality of the intermediate estimates varies among the nodes, just because of the relative position of a node in the communication graph. Next we find

out which nodes are faster in converging to the final estimate (nodes with faster dynamics).

Let $G=(V,E)$ be the connected undirected communication graph of the network. Let W be the matrix of local interactions. Let $v_1, v_2, \cdots, v_n$ be the eigenvectors of W associated with eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_n$. Let v_j^i denote the i th component of v_j.

Lemma 3-1- The projection of the unit vector e_i on v_j is

$$\Pr_j(e_i) = \frac{\langle e_i, v_j \rangle}{\|v_j\|^2} v_j = v_j^i v_j \tag{3-37}$$

Lemma 3-2- The diffusion function $\Phi_{ij}(t)$ and $\Phi_i(t)$ can be expressed as follows

$$\Phi_{ij}(t) = \sum_{l=1}^{n} \lambda_l^t \, v_l^i \, v_l^j \tag{3-38}$$

$$\Phi_i(t) = \sum_{j=1}^{n} \sum_{l=1}^{n} \lambda_l^t \, v_l^i \, v_l^j$$

Proof- We use lemma (3-1) for calculating the element (*i,j*) of the matrix W^t

$$\begin{aligned}
\Phi_{ij}(t) = e_i^{'} W^t e_j &= \left\langle W^t \sum_{l=1}^{n} \Pr_l(e_i), \sum_{l=1}^{n} \Pr_l(e_j) \right\rangle \\
&= \left\langle W^t \sum_{l=1}^{n} v_l^i v_l, \sum_{l=1}^{n} v_l^i v_l \right\rangle \\
&= \sum_{l=1}^{n} \lambda_l^t \, v_l^i \, v_l^j
\end{aligned}$$

■

We define the centrality of node i as

$$c(i,t) = \sum_{l=1}^{n} \lambda_l^t (v_l^i)^2 \tag{3-39}$$

Centrality has the following property,

$$c(i,t) \le 1, \textbf{ since } \sum_{j=1}^{n} \left(v_j^i\right)^2 = 1 \tag{3-40}$$

In the limit as time goes to infinity the centrality of all nodes converges to the same value

$$\lim_{t\to\infty} c(i,t) = \frac{1}{n} \tag{3-41}$$

Lemma 3-3- The upper bound on the diffusion function is

$$\Phi_{ij}(t) \le c(i,t) \tag{3-42}$$

$$\Phi_{i}(t) \le n\, c(i,t)$$

Theorem 3-1 –The intermediate estimate at node i has an upper bound that can be expressed as the centrality of that node.

$$Q_i(t) \le P_i(t)^{-2} \left(\sum_{j=1}^{n} a_j^2 \sigma_j^{-1} \right) c(i,t)^2 \tag{3-43}$$

Proof- Just using Lemma (3-3) to change the equality in (?) will result in the above inequality. ■

Centrality is a measure that shows after t iterations how much the data of a node diffuses in the network. Centrality is a general concept in network theory that ranks nodes in a network according to different network metrics. Metrics may be local or may be global. The centrality measure introduced here shows the local effects of the diffusion at the beginning of the consensus and as consensus progress it incorporates the global effects as well.

These are some examples where the centrality concept comes into the picture:

(1) In an ecological study of food webs, a centrality measure might identify the most important organisms in an environment.

(2) In social networks a study of friendship networks might use a centrality measure to determine the most popular person.

(3) Centrality could be applied to traffic patterns to identify how well-used roads are, and perhaps identify where to spend maintenance funds.

The common centrality measures are as follows,

(1) Degree centrality: purely local measure, measure immediate influence. For example in a disease transmission network, if a node is infected those directly connected to that node will also be infected.

(2) Clustering coefficient: measures how close the neighborhood of each vertex comes on average to being a complete clique. This measure is local.

Clique: In graph theory, a clique in an undirected graph G is a set of vertices V such that for every two vertices in V, there exists an edge connecting the two. The size of a clique is the number of vertices it contains. Finding whether there is a clique of a given size in a graph (the clique problem) is NP-complete. There is significant correlation between centrality measures. However in specific cases they may differ significantly. Therefore, we distinguish between local effects from organizational effects.

(3) Average path length: this is a global measure and is the average number of steps along the shortest paths for all possible pairs of network nodes.

(4) Betweenness centrality: how influential a node is in communicating between node pairs. In other words, it measures the number of times that a shortest path between nodes i and j travels through a node k whose centrality is being measured.

(5) Closeness centrality: reciprocal of the sum of the lengths of geodesics to every other node. Closeness can be regarded as a measure of how long it will take information to spread from a given vertex to others in the network.

(6) eigenvector centrality: this centrality is proportional to the sum of the degree centralities of the node' neighbors. Therefore a node has high eigenvector centrality if

is connected to many other nodes or if it is connected to others that themselves have high degree.

(7) Subgraph centrality: The fourth measure we use within this study is the subgraph centrality, which is based on the idea that the importance of a node should depend on its participation in local closed walks where the contribution gets smaller the longer the closed walk is. The number of closed walks of length k starting and ending on node i in the network is given by the local spectral moments of the networks adjacency matrix A.

Assume all the sensor node measurements have the same noise variance σ and the measurement gain is unity ($a_i = 1$). The intermediate estimation variance is then

$$\begin{aligned} Q_i(t) &= E[(\hat{\theta}_i - \theta)^2] \\ &= P_i(t)^{-2}\left(\sum_{j=1}^{n}\Phi_{ij}^2(t)\sigma^{-1}\right) \\ &= \frac{\sum_{j=1}^{n}\Phi_{ij}^2(t)\sigma^{-1}}{\left(\sum_{j=1}^{n}\Phi_{ij}(t)\sigma^{-1}\right)^2} \\ &= \frac{1}{\sigma}\sum_{j=1}^{n}\Phi_{ij}^2(t) \end{aligned} \tag{3-44}$$

And from (3-42)

$$Q_i(t) \le \frac{1}{\sigma} n c^2(i,t) \tag{3-45}$$

And in the limit

$$\lim_{t\to\infty} Q_i(t) = \frac{1}{\sigma} n c^2(i,t) = \frac{1}{n\sigma} \tag{3-46}$$

Although all the nodes will asymptotically have the same estimation variance, they have different variance dynamics in the network. Some sensor nodes converge faster than others.

This representation of centrality in (3-39) is very similar to the Discrete Fourier Transform of a discrete signal with sequence $v_l^1, v_l^2, \cdots, v_l^N$, as both are sum of complex exponentials. The sequence of *N* numbers $x_0, x_1, \cdots, x_{N-1}$ is transformed into the sequence of *N* complex numbers $X_0, X_1, \cdots, X_{N-1}$ by the DFT according to the formula:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn} \quad k = 0,1,\cdots,N-1 \tag{3-61}$$

In the definition of centrality the eigenvalues are the exponentials and the set of complex numbers are $(v_l^i)^2$. However the Fourier representation is based on a fixed set of basis functions. But the exponential in the centrality representation are adjustable by tuning $\{\lambda_l\}$to the desired values with changing the local rules; this adjustment is an essential property in designing optimal local algorithms. In this sense designing algorithms with a desired response requires investigating the relationship between the local rules, the eigenvalues and the eigenvectors.

Therefore the centrality of a node $c(i,t)$ for $t = 0,1,\cdots,T$ can be seen as the filter response of that node, and the bandwidth of that determines the dynamics of the node. A local rule algorithm converges if the filter responses are low pass. The node with a filter response that damps the higher frequency components converges faster than the other nodes. The filter responses can be reshaped. One way to do this is to change the eigenvalues of the process. Assume that we want to reshape the filter introduced by the local interaction matrix W. If we change the local interactions matrix to $f(W)$ a polynomial function of W,

$$X(t+1) = f(W)X(t) \tag{3-62}$$

Then the eigenvectors $v_1, v_2, \cdots, v_n$ will be the same but the eigenvalues will be $f(\lambda_1), f(\lambda_2), ..., f(\lambda_n)$. We can choose a function f so that the eigenvalues have specific features, for example we may want to have a low pass filter so that,

$$f(\lambda_i) = \begin{cases} 1, \lambda_i < \lambda_c \\ 0, \lambda_c < \lambda_i \end{cases} \tag{3-63}$$

The following FIR filter does this:

$$f(\lambda_i) = (1 - a\lambda_i)(1 - b\lambda_i) = \begin{cases} 1, 0 < \lambda_i < \dfrac{1}{a} + \dfrac{1}{b} \\ 0, \qquad O.W. \end{cases} \tag{3-64}$$

3.4 Influence

Suppose we define the uncertainty of a node as the estimation error,

$$u(i,t) = \sum_{j=1}^{n} \Phi_{ij}^{2}(t) \tag{3-65}$$

This quantity has the following properties:

(1) Before starting the interaction the uncertainty is at its highest level $u(i,0) = 1$

(2) Asymptotically $u(i,t)$ converges to a limit that depends on the number of nodes involved in the process,

$$\lim_{t \to \infty} u(i,t) = \frac{1}{n} \tag{3-66}$$

(3) The uncertainty after the first interaction $u(i,1)$ is a function of the degree of that node,

$$u(i,1) = (1 - \rho d_i)^2 + d_i \rho^2 \tag{3-67}$$

Where d_i is the degree of the node.

Example- let's consider a network with 100 nodes plotted in figure (3-1). The uncertainty for 5 of the nodes is plotted in figure (3-2).

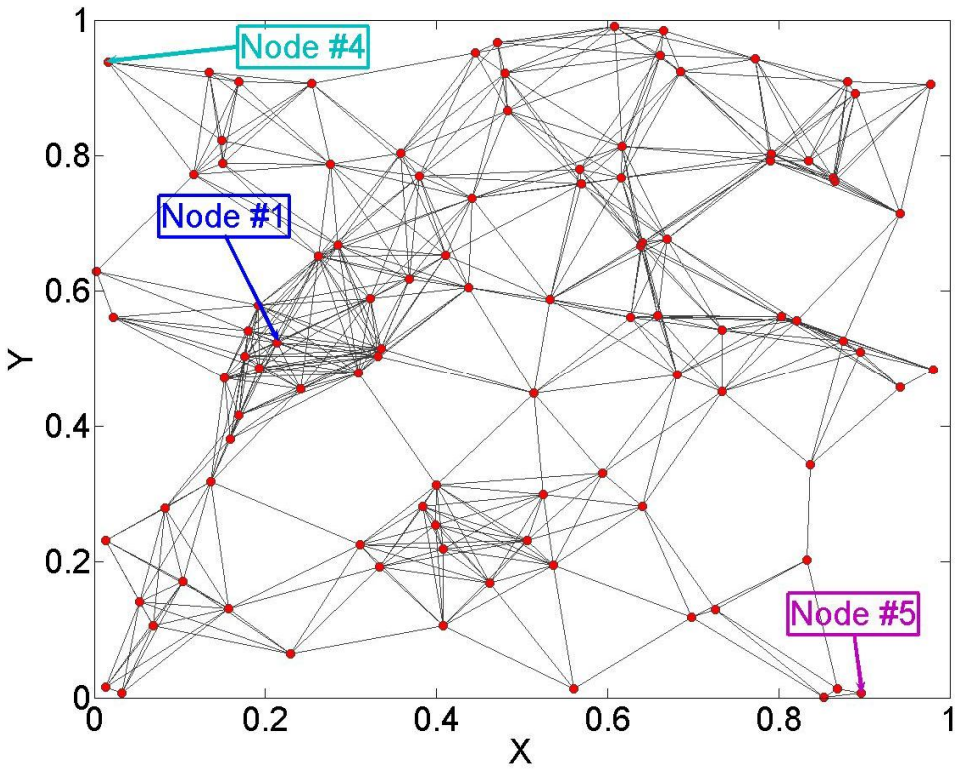

Figure 3-1. An example network with 100 nodes.

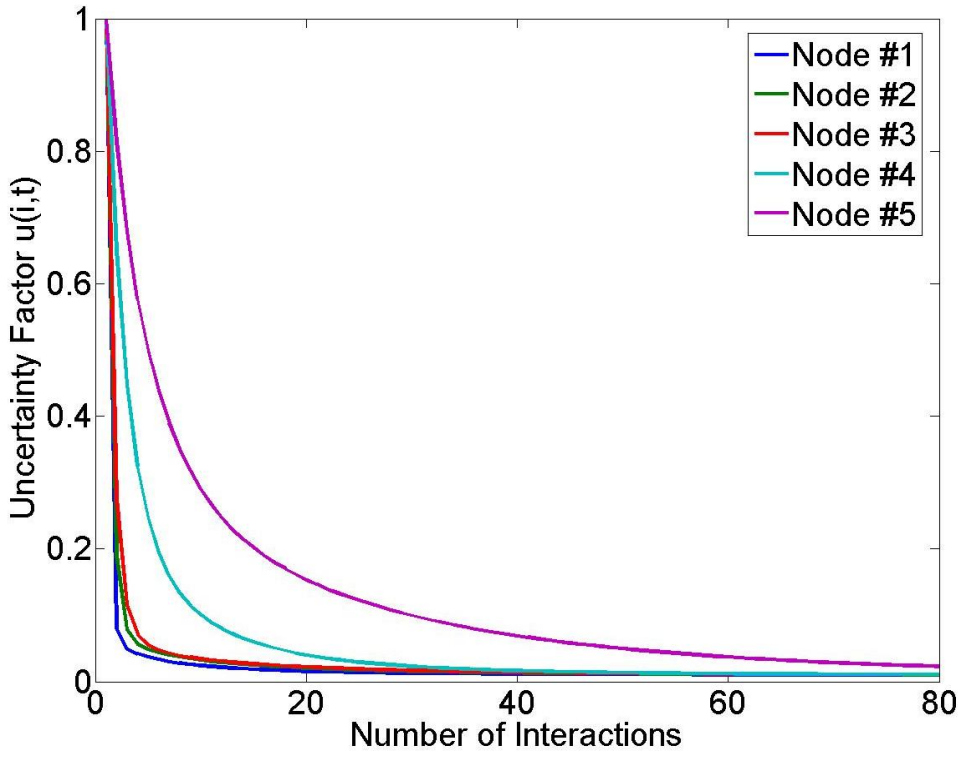

Figure 3-2. Utility function for the network in figure 3-1.

As this example shows, the uncertainty in some nodes decreases a lot slower than the uncertainty of other nodes. In the above example node 5 is the slowest node and node 1 is the fastest. The time it takes for node 5 to get close enough to the limit is almost 10 times larger than the time for node 1. The uncertainty in a node decreases as a node incorporates more information from other nodes. We define the *influence* of a node as follows,

$$\alpha(i,t) = \frac{1}{\sum_{j=1}^{n} \Phi_{ij}^{2}(t)} \tag{3-68}$$

Influence is a real number and varies between 1 and *n* the number of nodes.

(1) $\alpha(i,0) = 1$ means that the node only has its own information

(2) when the cooperative estimation process has converged a node has information from all the nodes in the network therefore,

$$\lim_{t\to\infty} \alpha(i,t) = n \tag{3-69}$$

This quantity is a measure of the estimation quality in a node as a function of time. The dynamics of influence depends on the relative position of a node in the communication graph of the network. A node with faster dynamics is capable of a faster injection (or absorption) of information in the network; therefore it has more influence in the network than others. Also it is more central in the operations of the network and is more important than the others in this sense.

Lemma 3-4- The influence of a node has the following relationship with the local rule matrix W

$$\alpha(i,t) = \frac{1}{(w_{ii}^t)^2} \tag{3-70}$$

Where w_{ii}^t is the (i,i) element of W^t.

Proof- Let's suppose the initial state vector is $X[0] = e_i = [0 \cdots 010 \cdots 0]'$ (impulse signal) with all elements zero except for its ith element. The state changes according to $X[k+1] = WX[k]$ and after t interactions we have:

$$x_j[t] = \Phi_{ij}(t) \tag{3-71}$$

And therefore,

$$\sum_{j=1}^{n} \Phi_{ij}^{2}(t) = \left\| X[t] \right\|^{2} \tag{3-72}$$

On the other hand

$$\left\| X[t] \right\|^{2} = \left\| W^{t} X[0] \right\|^{2} = X[0]' W^{2t} X[0] = e_i' W^{2t} e_i = (w_{ii}^{t})^{2} \tag{3-73}$$

■

It is possible for a node to adjust its operations based on the knowledge of its influence to save energy. A node can save energy by interacting less with its neighbors when this does not affect the entire performance of the network.

Let's restate the consensus protocol here. After a node interacts with its neighbors it updates its state as a weighted average of its own and its neighbors' states. Here by neighbors we mean the active neighbors or the neighbors that are interacting in that time instant.

$$x_i[k+1] = (1 - \rho\, d_i)\, x_i[k+1] + \sum_{j \in N_i(t)} \rho\, x_j[k] \tag{3-74}$$

First of all if a node stops interacting for some time steps the algorithm will still converge as long as the graphs produced as the result of this switching are jointly connected [Jadbabaie 2003].

Secondly, the following lemma guarantees that if a node stops interacting for one time step it does not change the asymptotic consensus value.

Lemma – In the consensus protocol $X[k+1] = WX[k]$ that is convergent if a node does not interact for one time step it does not change the consensus value and the state will remain in $s = \{x \in R^n \mid x_1 = x_2 = ... = x_n\}$ space.

Proof – Consider the immediate influence of the nodes on each other. Suppose that sensor node i and sensor node j communicate with each other. The change in the state of sensor node i as the result of this interaction is

$$\Delta x_k (i \leftrightarrow j) = \rho x_k^j - \rho x_k^i$$

And the change in the state of the sensor node j is

$$\Delta x_k (j \leftrightarrow i) = \rho x_k^i - \rho x_k^j$$

It is clear that these changes have the same value but different signs

$$\Delta x_k (j \leftrightarrow i) + \Delta x_k (i \leftrightarrow i) = 0$$

Therefore the change in the states of sensor nodes is such that the sum of all the changes in the network is zero. This implies that if two nodes do not interact for one time step the total sum of the states and as a result the average of the states does not change.

■

3.5 Computing the centrality

Centrality is a concept used often in the study of networks. It refers to a class of measures that are intended to capture the relative structural importance of a node or edge in a network. There are different ways to capture the concept of centrality. However, each of the measures attempts to quantify some sense of a node's or edge's overall importance in the network. Which one to use depends on the nature of "importance" underlying the relationships in the graph.

3.5.1 Modified Eigenvector Centrality

Eigenvector centrality ranks the node based on their importance in the network. We leverage the eigenvector centrality for our controlled consensus and use an iterative algorithm for distributed computation. After a couple of iterations the network converges to the optimal value and stays in the stable point unless a change happens in the network to which it will adjust itself. These changes include node failures and link and topology change.

The eigenvector centrality is based on the idea that an important node is connected to many other important nodes. Therefore if we represent the importance of node i by c_i then,

$$c_i \propto 1/|N_i| \sum_{j \in N(i)} c_j \tag{3-75}$$

$N(i)$ is the set of the neighbors of i. This equation can be written as

$$c_i \propto 1/|N_i| \sum_j a_{ij} c_j \tag{3-76}$$

Where a_{ij} is the component (i, j) of the adjacency matrix. In the matrix form

$$Hc = c \tag{3-77}$$

Matrix H has components h_{ij} defined as follows,

$$h_{ij} = \frac{a_{ij}}{\sum_{j=1}^{N(i)} a_{ij}} \tag{3-78}$$

The maximum eigenvalue of matrix H is 1. Also note that

$$c = [c_1, c_2, \cdots, c_N]^T \tag{3-79}$$

The largest eigenvalue results in an eigenvector with all positive elements; this is acceptable since the centrality of a node is a real number in $[0,1]$. The ith component of this eigenvector gives the centrality score of the ith sensor node. This automatically gives us a distributed method for the computation of the centrality measure. We start from an initial centrality vector with all components equal to $1/N$ and go through the following iteration

$$c^{k+1} = H c^k \tag{3-80}$$

c^k the centrality measure vector at iteration k will converge to the desired centrality measure. A network just needs to go through this convergence only one time. The

network and the above algorithm will reach to the new stable point if any change happens in the network and this is an attractive feature of using this centrality measure.

It is possible to leverage the above centrality measure by generalizing the *node importance concept.* The components of adjacency matrix A to belong to set $\{0,1\}$. If we define matrix B with the same structure of A but with components b_{ij} in the real number set $[0,1]$, then we have a matrix that can include other important factors in the centrality measure.

$$b_{ij} = \frac{E_i \times L_{ij} \times I_i}{C_i} a_{ij} \tag{3-81}$$

Here E_i is a metric representing the residual energy of node i. L_{ij} is a metric showing the quality of the communication link between node i and j. I_i is a metric showing the quality of the information in sensor node i. And finally C_i is a metric showing the cost of operations at node i.

Centrality measures computed from matrix B show the important nodes as far as energy, communication quality, information quality and cost of operations are the concerns. But how should we define these metrics? This remains as a future problem to be investigated.

3.6 Controlled Consensus

In a connected network all sensor nodes can, in principle, participate in maintaining a common operational picture of the field. Any sensor node may be able to provide valuable information. However, due to energy and bandwidth constraints it may be impossible for all nodes to participate fully. The responsibility of a distributed fusion management system is to select a subset of the large volume of sensor data that are potentially available, in a way where bandwidth and power constraints and operational fusion needs are simultaneously respected. In this kind of controlled consensus it is presumed that only a subset of nodes close to the event of interest have relevant measurements for fusion. This subset may be time varying. In this method we adaptively select this subset of nodes with the highest payoff. This is very attractive in the context of tracking and surveillance, as only a subset of the whole set of nodes in the network (those close to the source) have information-bearing measurements. Selecting the appropriate nodes can be a challenging task in an inhomogeneous network consisting of nodes with different sensing accuracy, link quality and energy consumption. Our objective is to develop methods to guarantee the desired global performance under the given constraints. We will refer to this type of reaching agreement as "*controlled consensus*".

A data fusion rule for consensus that has been widely investigated in the literature is the fusion of information from all the sensors. It does not involve the active control of the sensors as part of the algorithm. Our approach is to control the operations of the

nodes involved in the consensus process by associating weights with each node to emphasize those with highest payoff. These weights carry global information necessary for a node to control its operations and decisions to be part of the consensus.

A network with *N* nodes is monitoring an area. The first node that detects an event starts an algorithm to estimate a parameter related to that event. We will refer to this node as the *leader*. The leader wants to achieve the highest certainty possible about that parameter with the maximum allowable energy allocated for that estimation task. As described in the introduction, this maximum energy is determined by the temporal control layer.

We first introduce utility and cost models of sensors and then techniques that find optimal or nearly optimal assignments. A utility function assigns a scalar value to each sensor configuration that is

$$U : I \times K \to R \tag{3-82}$$

Where $I = \{1, 2, \cdots, N\}$ is the set of sensor indices and K is the discrete time interval. A sensor configuration is a subset V such that $V \subset I$. Each sensor configuration V also has a cost. The cost of sensing is $C_s(V)$, the communication cost is $C_c(V)$, and the computation cost is $C_p(V)$. To simplify the analysis we assume that the computation cost is negligible and there is no loss in the communication links, also the communication cost is due to transmission and not reception.

The sensor selection problem may be stated as follows:

Determine the set of configurations $V(k)$ that maximize the utility over a period of time

$$\max_{\{V(k)\}} \sum_{k=1}^{T} U(V(k)) \tag{3-83}$$

Subject to the following constraints,

1. Energy constraint

$$\sum_{k} C_s(V(k)) + \sum_{k} C_c(V(k)) \le C_{total} \tag{3-84}$$

2. The connectivity constraint: the sensor nodes in a configuration should be connected to each other as well as to the leader node.

The utility function of a configuration depends on the underlying communication structure in the configuration.

The main idea is to base sensor selection decisions on information content while respecting the constraints on energy and connectivity. Sensors can use the information utility they have received already to optimize the utility of future actions, and therefore efficiently manage the resources. We assume that for each event a fixed leader node is selected that is going to incorporate the measurements of other nearby sensors. The task here is to select an optimal subset of nodes in order to increase the

certainty in the leader node. The decision that a node belongs to an optimal configuration depends only on the characteristics of that node and prediction of its contribution.

In the surveillance application we consider here the certainty of information aggregated from two sensors is the aggregation of their certainty. For example if a node has certainty σ_1^{-1} and the other sensor has certainty σ_2^{-1} then the aggregation of the information from these two sensors has certainty $\sigma_1^{-1} + \sigma_2^{-1}$. This is based on the assumption that uncertainties are statistically independent.

In our algorithm given a current configuration, there is a procedure to determine which sensors to add to the current configuration among the remaining sensor nodes. This is based on a prediction and selecting the most likely best sensor. The information utility measure that we want to maximize is to have the maximum certainty in the leader node.

The algorithm is as follows:

1. Initialization: Sensors perform a distributed centrality computation, and a scalar number is assigned to each node as the centrality of that node. This number indicates the relative importance of a node in the communication graph. A leader is selected which is a node that first detects an event. If multiple nodes detect an event the node with a higher centrality starts the algorithm. The leader node knows how much energy

should be spent on this estimation task. As we explained in Chapter one the temporal energy control algorithm assigns the total amount of energy budget for processing each event. And this information is passed to all the nodes.

2. Request: The leader selects a group of its neighbors and requests information from them. A request includes the total amount of energy that node can spend in collecting information for the leader. The summation of these energy budgets and the energy used by the leader for sensing equals the total energy budget. The decision as to which neighbors to select and how much energy to allocate them is the leader's control mechanism. The goal of the leader is to maximize its information utility. This decision is based on the prediction of the leader about the performance of that neighbor, including the amount of increase in the certainty level and the total energy consumption for this certainty increase. The leader uses the centrality of the nodes as a measure for selecting the most likely best sensors. A neighbor with a higher centrality will be assigned more energy. As a metaphor this resembles hiring contractors to collect the necessary information.

3. Sensor addition: the neighbors that are already assigned to collect information for the leader may not be able to consume all the energy assigned to them to provide information. Therefore they request information from their neighbors by selecting a group of them and allocating energy to them. This stage is similar to a contractor hiring subcontractors to do the job. This node should estimate how much energy it is going to use by its own for sensing and communication. In the simulations we have

assumed that all the participating nodes make their own measures, which can be relaxed in the general case. Again this sensor selection and energy allocation is based on the centrality of the nodes. This process will continue and more nodes will be added to the current configuration. A node that has been allocated an energy which is only enough for its own operations does not request for more additions. As this proceeds a directional tree is formed with the leader node as its root.

4. Iteration: Nodes in the current configuration start a consensus algorithm and sending information iteratively in their neighborhood. Along their current estimate, they also send the certainty of that information as well the energy consumed to produce that information.

5. Adaptation: The leader node evaluates the performance of its selected neighbors and reconsiders its requests from neighbors. The performance of a node is defined as the certainty it has provided divided by the energy it has consumed to provide that level of certainty. The neighbor node with the highest performance will be assigned more energy

$$x^{ij}(k+1) = x^{ij}(k) + \alpha(x^{i}(k+1) - x^{ij}(k)) + (1-\alpha)(x^{i}(k+1) - x^{i}(k)) \qquad (3\text{-}85)$$

$x^{ij}(k)$ is the energy allocated by node i to node j at time k and $x^{i}(k)$ is the total energy node i is going to spend to get information from the neighbors. α is a coefficient that balances between two extremes: if $\alpha = 0$ there is no change in energy

allocations and $x^{ij}(k+1) = x^{ij}(k) + x^{i}(k+1) - x^{i}(k)$. On the other hand if $\alpha = 1$ then all the energy is requested only from node j and $x^{ij}(k+1) = x^{i}(k+1)$

The rest of the nodes will be assigned less energy, so that the total energy assignment remains constant and equal to $x^{i}(k+1)$,

$$x^{ij}(k+1) = (1-\alpha)x^{ij}(k) \tag{3-86}$$

The node with the least performance will be replaced with one of the neighbors that was not previously participating.

3.7 Simulation

We divide the time into periods of consensus and adaptation. The leader divides the total energy it is going to consume into M equal parts for M adaptation periods. Also in the selection of neighbors a node does not select all of them but only a percentage of them. The reason is to give some flexibility to the algorithm so that the operation is distributed over a larger area. The percentage parameter may be controlled by the leader node and be adapted in the adaptation process.

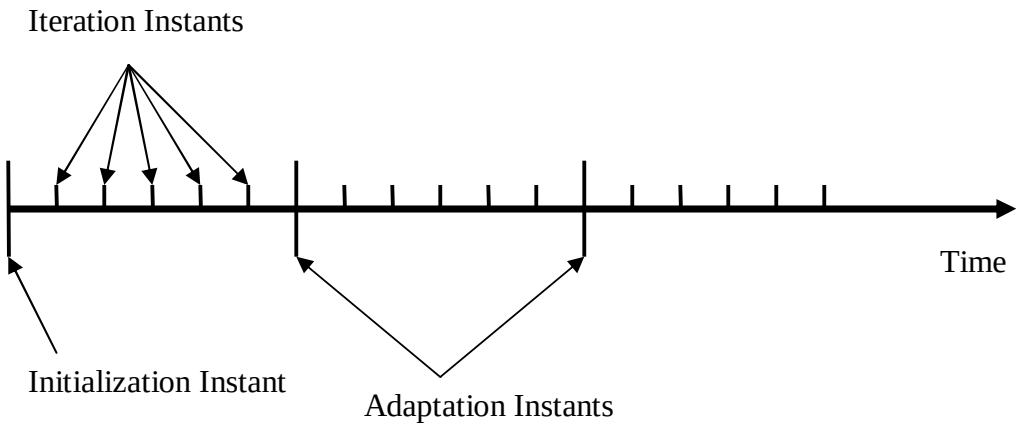

Figure 3-3. Two different time scales for adaptation and iteration.

As an example we show the result of a controlled consensus for the network shown in figure 3-4. There are 20 nodes in the network and the node marked with a star is the leader. The first simulation is an example to illustrate the usefulness of incorporating a controlled consensus.

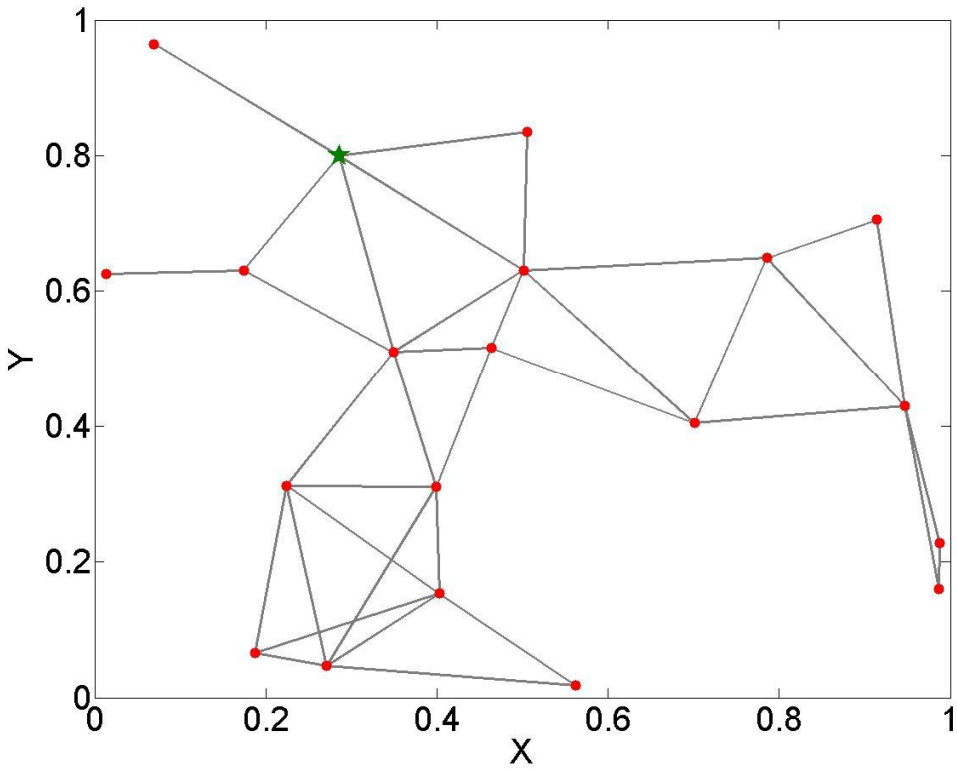

Figure 3-4. Example network with 20 nodes. The leader node is marked with a star.

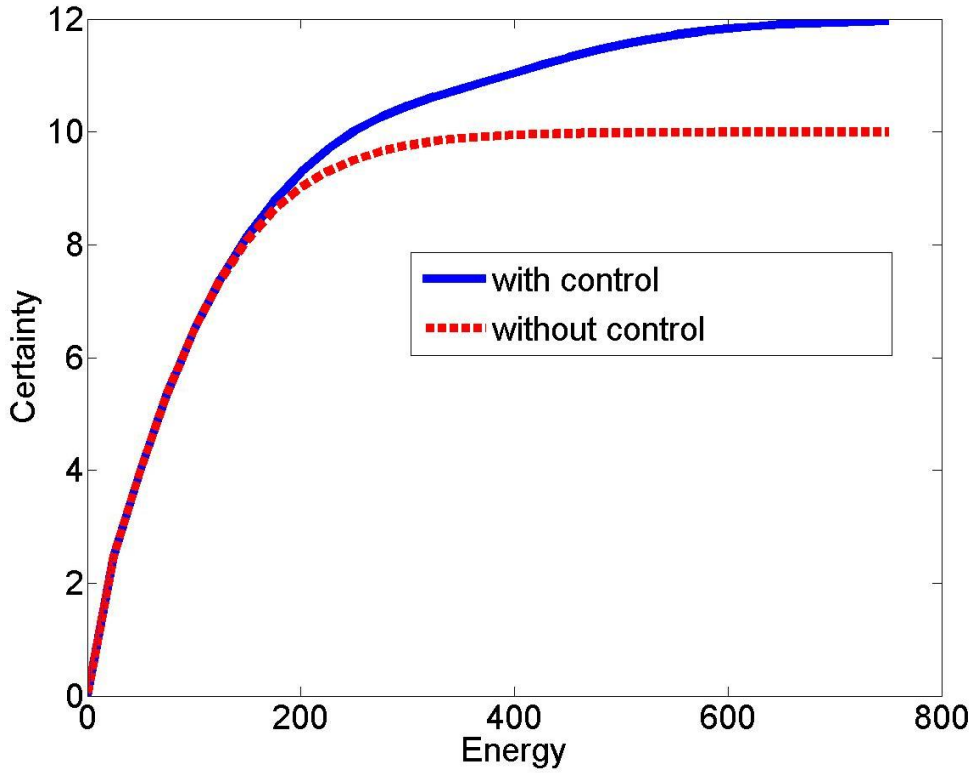

Figure 3-5. Energy consumption versus Information gain with and without control (or adaptation)

In the curve without control in figure 3-5 there is no adaptation process. And as we can see the certainty remains constant after a number of iterations while the algorithm with adaptation changes the nodes involved in the consensus to increase the certainty.

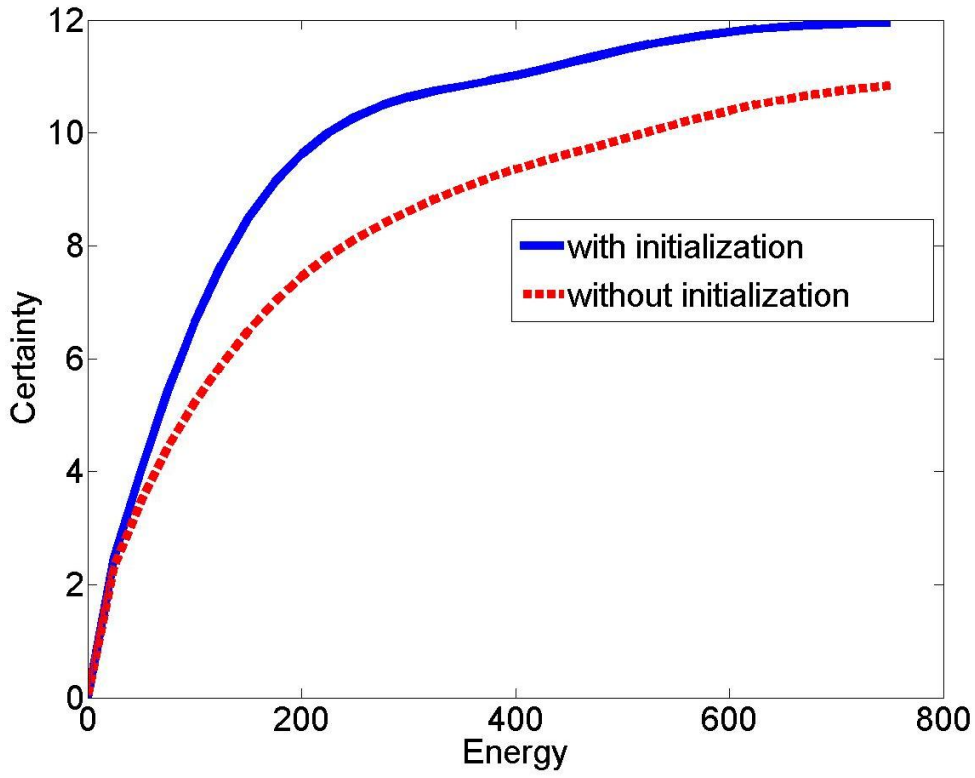

Figure 3-6. Initialization with and without centrality (Both with control).

In figure 3-6 the effect of the initial selection is illustrated. The initial selection based on the centrality measure results in a better information gain compared to an initial selection that is random.

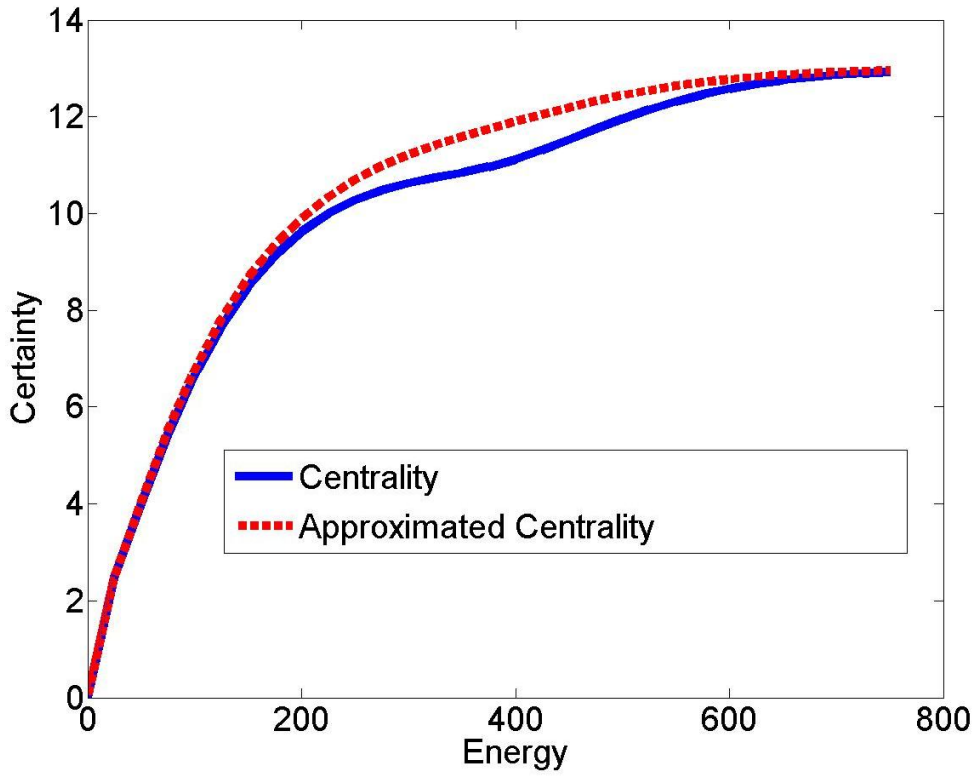

Figure 3-7. Eigenvector centrality (approximated centrality) versus the other centrality measure.

As we mentioned we have used the eigenvector centrality rather than the centrality measure we defined previously in this chapter. We chose eigenvector centrality because it is better suited for distributed computation. In figure 3-7 we can see that the performance of both methods is almost the same. However more investigation is necessary to proof this in general.

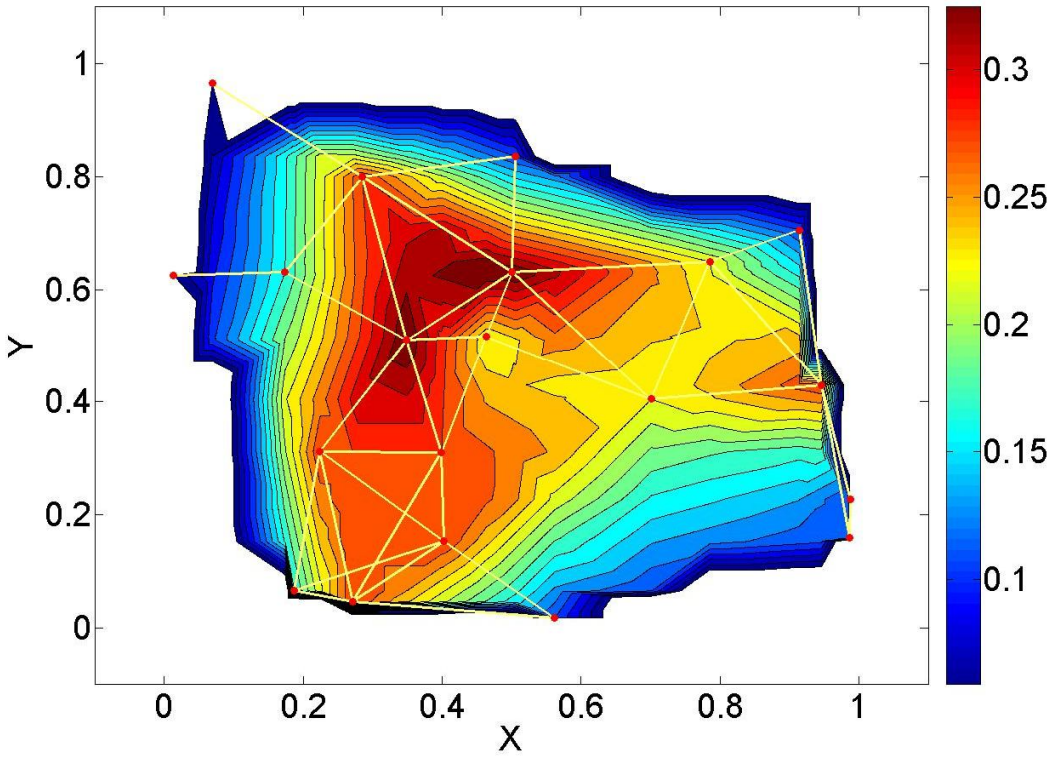

Figure 3-8. Centrality interpolated as a 2D function

Figure 3-8 shows the centrality of the network plotted as a landscape by giving each node a height equal to its centrality. The peaks of this landscape are the more central nodes. The other points of the landscape are the result of interpolation.

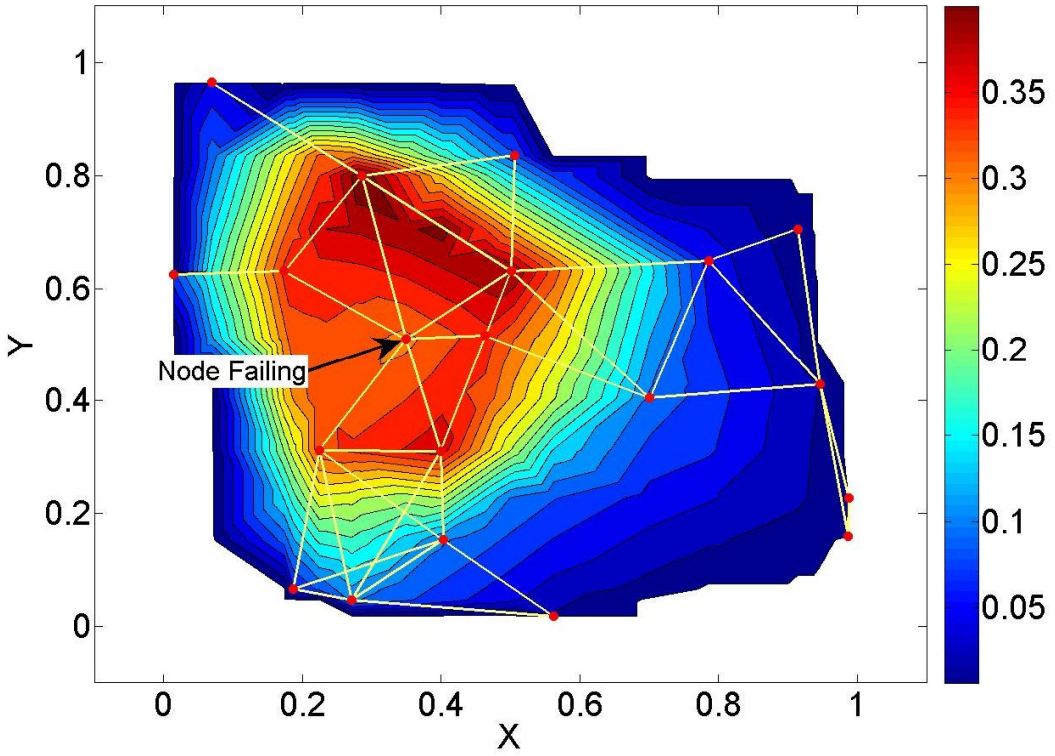

Figure 3-9. Change in centrality due to a node failure

Figure 3-9 shows how landscape changes as the topology of the network changes. In this example a node has started to fail and as we can see the centrality of the node has changes and the landscape around the failing node is becoming a valley rather than a peak.

3.8 Remark

In a complete management scheme each node controls its operations by considering three type of information. In general the control signal of a node will have the following form,

$$u_i(k) = f(x_i(k), z_i(k), r_i(k)) \tag{3-87}$$

$x_i(k)$ is the self knowledge of the node about the process.

$z_i(k)$ is the local knowledge gained through interaction with neighbors.

$r_i(k)$ is the global knowledge that include information about global performance and global resource consumption.

It is very important to notice that these three types of knowledge have different dynamics. The global knowledge changes very slowly. This is very desirable because its calculation is through iteration and involves information pass across the network. However it provides the advantages of an open loop control. On the other hand the self and local information have faster dynamics and they work as closed loop feedback controls that provide fast reaction to disturbances.

Chapter 4: Role Management

"Leadership appears to be the art of getting others to want to do something you are convinced should be done"

-Vance Packard

4.1 Introduction

Fusion from multiple sources can happen on three different levels: Data, Feature, and Decision. The dimension of the fusion space, and therefore the computation and communication requirements, is largest for data fusion and smallest for decision fusion. However the fusion becomes less dependent to the type of the sensing data in feature and decision fusion and it becomes easier to combine information from different type of sensors. Depending on the application the transformation from a high dimension data space to a lower dimension decision space may result in some information loss. So far we studied methods to increase the certainty in the fusion process in networks in which all the nodes have the same strategy when confronted with different decisions from other nodes. In this chapter we look at the diffusion of decisions in a different type of network where nodes have different strategies when confronted with different decisions and where there is conflict of interest among the nodes.

The network we consider in this chapter is based on a model that was considered in an article by Vicsek [Vicsek 1995]. In the Vicsek model a novel type of dynamics is

introduced in order to investigate the emergence of self-ordered motion in systems of particles with biologically motivated interaction. In that model particles are driven with a constant absolute velocity and at each time step assume the average direction of motion of the particles in their neighborhood. Vicsek presents numerical evidence that this model results in a continuous kinetic phase transition from no transport (zero average velocity) to finite net transport through spontaneous symmetry breaking of the rotational symmetry.

The concept of fusion of information is not limited to manmade systems. This phenomenon can be seen very often in social network, biology and psychology. Below we briefly state some examples.

In psychology there is a phenomenon called social proof or informational social influence or crowd psychology. Regardless of one's background, human behavior in groups is influenced by this principle. Simply, social proof means when human beings are not sure, and when they are uncertain about what should be done or what something means, they look to others to see how they should feel and what to do in certain situations [reference]. Individuals make the assumption that if enough people believe in something that must be right. This is of course not accurate in many situations. There are lots of examples of when a majority of people were used to believe that something was right but now we know that it was not. This is the behavior of people that are looking for simple solutions that are not necessarily the best. The commercial media use this fact to promote ideas or beliefs. This is how

beliefs, fashion, and viewpoints propagate in a social network. You are influenced by your neighbors in the social network. In fact we need to remember that the more important the decision is the more information we need to come up with an effective belief or decision. There can be a huge price for going with the flow. However, this behavior is part of how we learn by modeling. Therefore, a network performing based on this principle can be vulnerable to injection of false beliefs. Therefore, the members of the network should realize that there is a tradeoff between the simplicity and the accuracy on decisions based on social proof. However this provides simple learning methods.

The diffusion of innovations theory studies how new technology and ideas spread through cultures. This theory was formalized by Everett Rogers [Rogers 1962]. He says innovations or ideas spread through society in an S curve, as the early adopters select the technology, followed by the majority until the technology is common. This S curve is the model achieved when there is no competing technology or idea. He categorizes the adopters of any new innovation into five groups (table below)

Adopter Type	% of the population	Characteristics
Innovators	2.5	multiple information source, risk taker, educated
Early adopters	13.5	Social leader, popular, educated
Early majority	34.5	many informal social contacts
Late majority	34.5	skeptical, traditional, lower socio-economic status
laggards	16	neighbors and friends are main information source

A different decision making phenomenon in societies is known as the wisdom of crowds. Rather than the social proof this concerns collections of independently deciding individuals. Simply this principle is that a diverse collection of independently deciding individuals is likely to be more representative of the universe of possible outcomes, thereby producing a better prediction [Surowiecki 2004]. For example if a group has to decide between two alternatives, and each member has a probability of 0.75 of correctly identifying the better alternative, a decision made by one dominant member would be wrong with probability of 0.25. However if an equally shared consensus decision is made, with a simple majority voting, the probability of choosing the wrong alternative is 0.16 for a group of three members, 0.10 with five members, 0.07 with seven members, and so on. Not all crowds are wise. There are four elements required to form a wise crowd: Diversity of opinion, Independence, Decentralization, and Aggregation. Surowiecki argues that the reason that crowds make very bad judgments is that members of the crowd are too conscious of the opinion of others and began to emulate each other rather than think independently. The success of Google, wikis, and blogging is discussed in this context.

Consensus decision making is vital for the survival of certain species of social animals. Individual animals face decisions that are crucial to their fitness. In social species many of these decisions need to be made jointly with other members [Conradt 2005]. Scientists study the questions such as: who makes the decisions in an animal society? How many decision makers are there? Is this decision communicated locally

or globally to others? And what information is communicated? What are the underlying mechanisms and what happens if there is conflict if interest between animals? Consider, For example, a swarm of bees choosing a new nest site, a flock of birds deciding when to leave a foraging patch, a group of migrating insects navigating, and a group of cooperative hunters about prey targets. This decisions concern synchronization of the group movement direction, travel destination, and activity timing.

Although these examples look very different from the type of the networks that we study here, there are certain fundamental similarities. When nodes try to reach consensus at the decision level a common method to reach consensus is through local averaging. The reason for choosing an abstract method like decision fusion is its simplicity and speed, or because the nodes are learning through this. We should be careful that in the decision fusion process nodes are sacrificing accuracy and vulnerability to simplification. In this section we look into this tradeoff by studying a network of mobile nodes that consists of leaders, followers, and disrupters.

4.2 Model

In this chapter we study a network of mobile nodes in which each one is modeled as a particle moving in the plane at constant speed with their heading dependent on inter-particle interactions and possibly on prior information about the preferred directions. We consider N nodes capable of communicating with neighbors. They are divided into three subgroups. Let N_1 and N_2 be the number of nodes in two different

subgroups of informed nodes (with a preferred moving direction) and let N_3 be the number of naïve (uninformed) nodes such that $N_1 + N_2 + N_3 = N$ .The preferred heading direction for the nodes in group one is $\overline{\theta_1}$ and for those in group two is $\overline{\theta_2}$. These subgroups have the following dynamics:

Leaders dynamics: $\theta_i(t+1) = K_1\overline{\theta_1} + < \theta_i(t) >_r + n_i(t) \quad i \in N_1$

Disrupters dynamics: $\theta_i(t+1) = K_2\overline{\theta_2} + < \theta_i(t) >_r + n_i(t) \quad i \in N_2$

Followers dynamics: $\theta_i(t+1) = < \theta_i(t) >_r + n_i(t) \quad i \in N3$

where $< \theta(t) >_r$ denotes the average direction of motion of the nodes (including node i) within a distance r from the given node, $\overline{\theta_j}$ is a preferred direction for subgroup j , $n(t)$ is a Gaussian noise. Informed nodes balance their preferred direction and their social interactions with a weighting term K_i. Here we assume that there is no identification of the type of the nodes and no evaluation of the information of nodes. As a measure of consensus, we look at the heading of the centroid of the points in each subgroup.

4.3 Observations

Observation 1. For fixed network size (*N* fixed) and two subgroups consisting of leaders and followers the accuracy of group in following the preferred direction of the leaders increases asymptotically as the proportion of the leaders increase. Figure 4-2

shows this behavior as the percentage of the leaders increases in the network. Accuracy is a measure of how well the centroid of the points follows the preferred direction in average, where one means perfect following and zero means that the final heading of the centeroid of the points is completely random. This accuracy is a function of the weighting term.

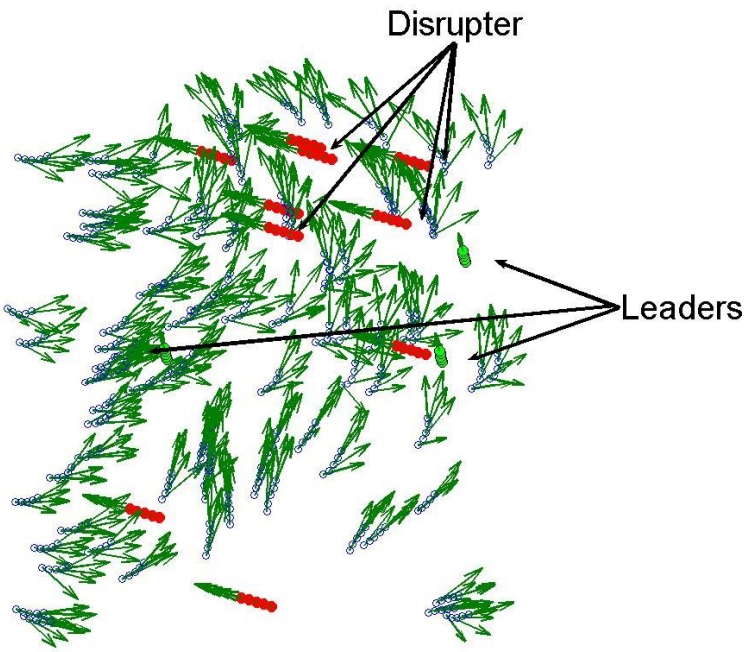

Figure 4-1. An Exmaple network with 100 nodes and 3 leaders and 10 disrupters. The change in the heading is plotted for 5 steps

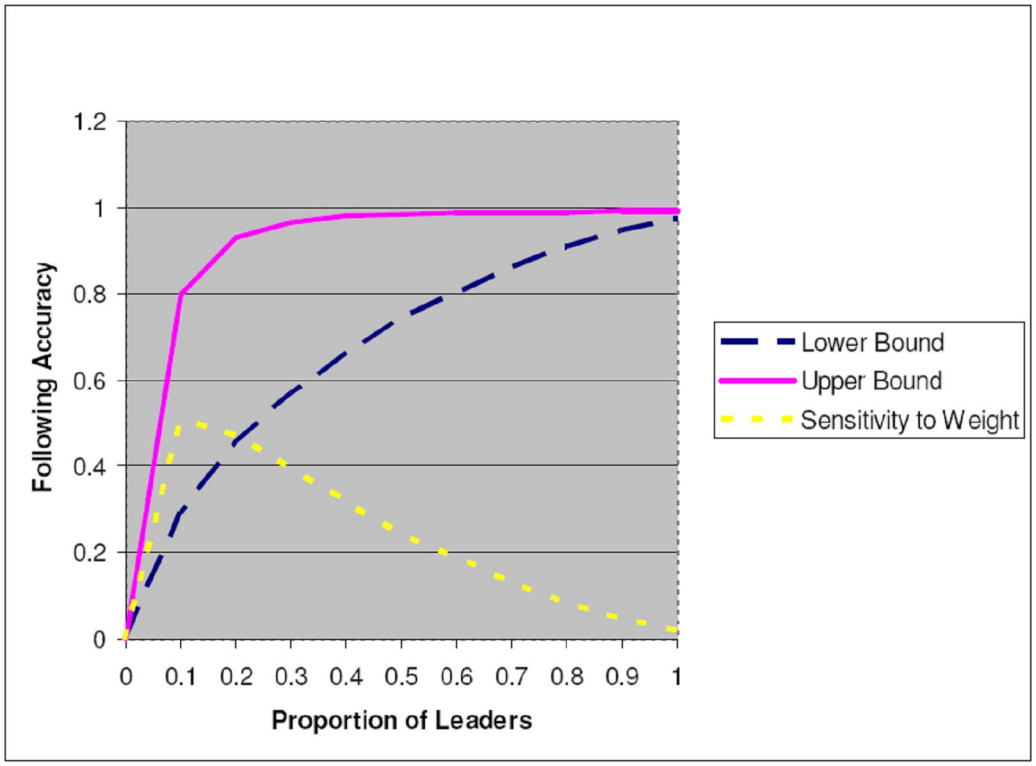

Figure 4-2. Following accuracy.

The accuracy is a function of the weighting term. Increasing the weighting term increases the accuracy of the following. However the sensitivity to this term changes with the number of leaders. For small and large numbers of leader this term is not important as much as in intermediate values.

Observation 2. For the case of followers and leaders as the size of the group increases a smaller proportion of leaders are needed to guide the group with given accuracy; and this relationship is non-linear.

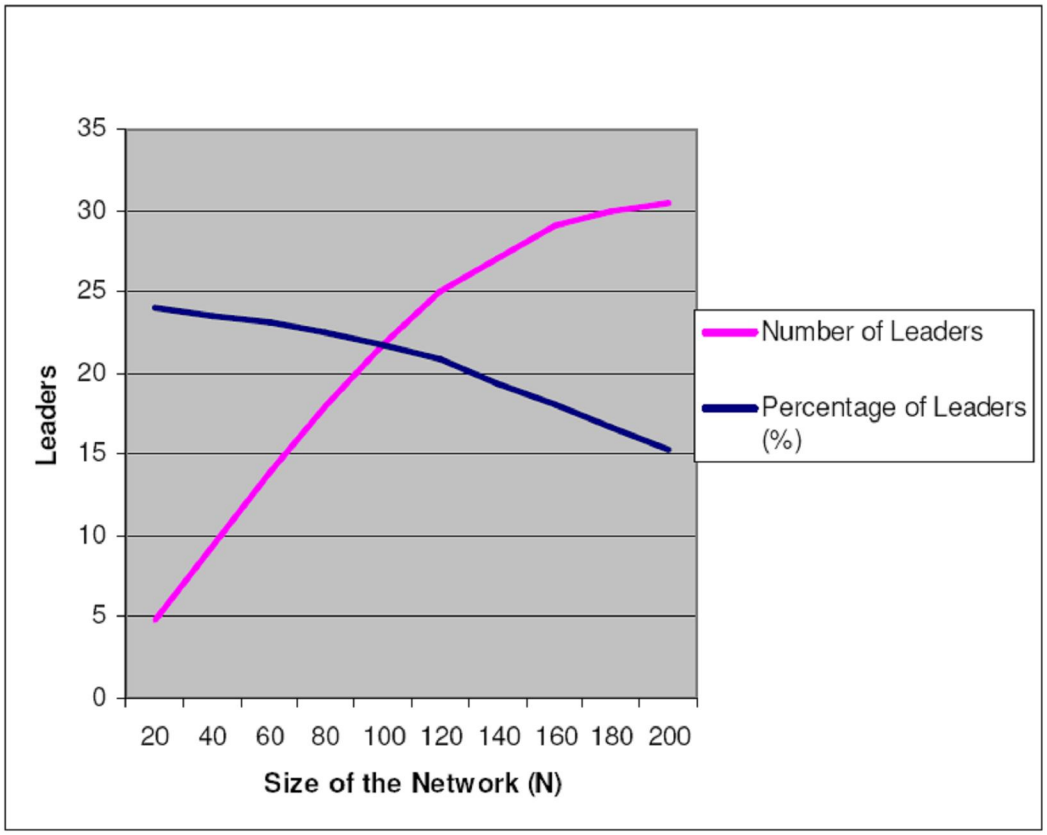

Figure 4-3. Number of leaders needed to achieve a certain level of accuracy.

Observation 3. Although it looks like we should increase the weighting term as much as we can, but there is a trade off between the accuracy of the group dynamics and the probability that the group split into disconnected parts. As we can see if the weighting term is high, then the leaders will go in their preferred direction before giving other nodes a chance to follow them.

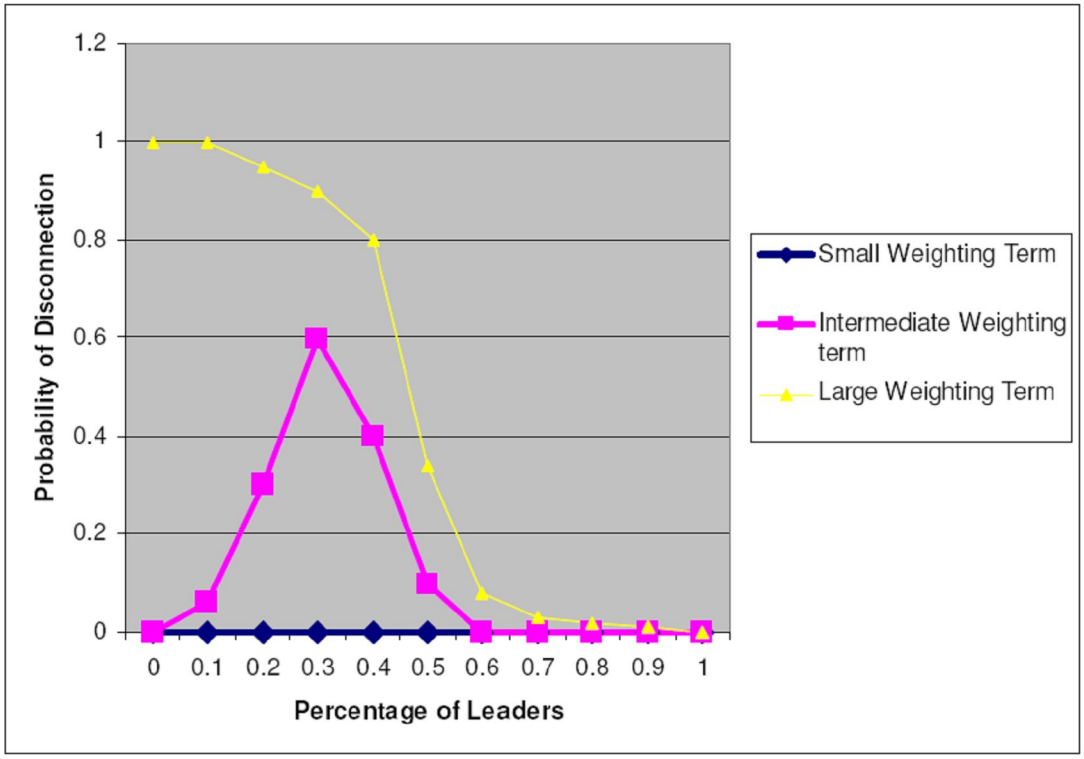

Figure 4-4. Weighting term affects the probability of the group split.

Observation 4. If there are leaders, followers, and disrupters in the network, and the leaders preferred direction is $\overline{\theta_1} = \frac{\pi}{2}$ and the disrupters preferred direction is $\overline{\theta_2} = \frac{3\pi}{2}$, then the network will most probably split into parts and the number of the nodes that are disrupted is a function of the relative number of leaders and disrupters. To plot figure 4-5, we have taken N1+N2=constant and we have changed N1/ (N1+N2). Disrupted nodes are those with heading $\left|\frac{3\pi}{2} - \theta_2\right| < \frac{\pi}{2}$.

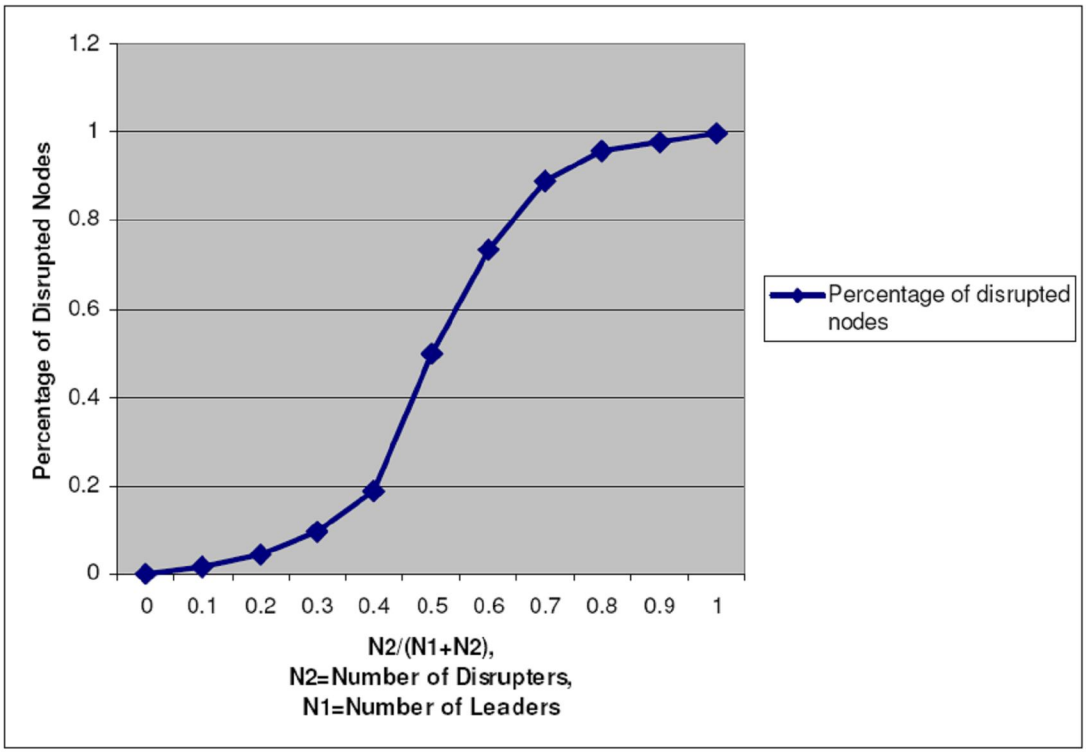

Figure 4-5. Percentage of disrupters to disrupt a network.

Chapter 5: Conclusion

5.1 Concluding Remarks

The contributions of this dissertation can be summarized as follows:

- Developing a distributed management algorithm for self-organizing wireless sensor networks under energy constraints by formulating the mission of such network as an optimal control problem and decomposing it into two temporal and spatial control problems.
- Addressing the temporal control of sensor networks by studying the coverage and connectivity in a perimeter surveillance application.
- Addressing the spatial control of sensor networks by proposing a distributed energy management algorithm, where the objective is to form a group of sensors with the maximum information gain under a limited energy budget.
- By utilizing the centrality concept in graphs our algorithm effectively finds the best configuration of sensor nodes for a parameter estimation problem, despite the randomness in the network and presence of disturbances and node failures.
- Investigating the characteristic of a network consisting of moving sensor nodes with different roles as leader, follower, and disrupter. Using a flocking analogy where mobile nodes try to align themselves with other nodes, we demonstrate the fusion of information in such a network.

5.2 Future Work

We identified the following directions for future research in this area:

- Studying the coverage problem and applying the same concepts in higher dimensions by charactering the relationship between energy consumption and the coverage in those spaces.
- Applying the management ideas presented in this work to clustered networks where there are master nodes as well as regular nodes.
- Improving the temporal control problem by:
    - Studying the statistical methods to model event occurrence for different applications.
    - Developing a distributed database for keeping track of the events..
- Improving the spatial control problem by:
    - Applying the ideas presented here to other methods of estimation
    - Applying the algorithm for a time changing parameter.
    - An algorithm that selects new leader as a target moves in the field and predicting the direction of the target to activate the nodes in that direction before the target gets there.
    - Defining the metrics for energy, link quality, cost and information for the leveraged centrality measure.
- Analysis and backing up the results we learned about roles in a network by Applying theories from social network.

- Use the centrality concept for the evaluating the best points of attacks in a network.

Bibliography

[Cerpa 2002] A. Cerpa, D. Estrin, “ASCENT: Adaptive Self-Configuring sEnsor Networks Topologies,” Proceedings of the twenty first International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), 2002.

[Chen 2000] B. Chen, K. Jamieson, H. Balakrishnan, R. Morris, “Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless networks,” Processdings of the sixth Annual International Conference on Mobile Computing and networking 2000.

[Conradt 2005] L. Conradt, T. J Roper, “Consensus decision making in animals,” Trends in Ecology and Evolution, Vol. 20, No. 8, August 2005.

[Clouqueur 2002] T. Clouqueur, V. Phipatanasuphorn, P. Ramanathan and K. K. Saluja, "Sensor deployment strategy for target detetction," in First ACM International Workshop on Wireless Sensor Networks and Applications, 2002.

[Dousse 2002] O. Dousse, P. Thiran, M. Hasler "Connectivity in ad-hoc and hybrid networks," IEEE Infocom, 2002,pp. 1079-1088.

[Foh 2004] C. H. Foh, B. S. Lee "A closed form network connectivity formula for one-dimeniosnal MANETs," IEEE ICC, 2004, vol. 6, pp. 3739-3742.

[Huang 2003] G. T. Huang, "Casting the wireless sensor network," Technology Review, 2003, pp. 50-56.

[Jadbabaie 2003] A. Jadbabaie, J. Lin, and A. S. Morse “Coordination of groups of mobile autonomous agents using nearest neighbor rules,” IEEE transactions on Automatic. Control, Vol. 48, No. 6, pp 988-1001, June 2003.

[MacCluer 2000] C. R. MacCluer, "The Many Proofs and Applications of Perron's Theorem." SIAM Rev. 42, 487-498, 2000.

[Meguerdichian 2001] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. Srivastava, "Coverage problems in wireless ad-hoc sensor networks," IEEE Infocom, 2001, pp. 1380-1387.

[Olfari 2003] Olfati-Saber, RM Murray, “Consensus protocols for networks of dynamic agents” Proceedings of the American Control Conference, 2003.

[Perolli 2004] M. Perolli, W. Heinzelman, “Wireless Sensor Networks”**,** Kluwer Academic Publishers, 2004.

[Perron 1907] O. Perron, "Grundlagen für eine Theorie des Jacobischen Kettenbruchalgorithmus." Math. Ann. 64, 11-76, 1907.

[Rogers 1962] E. M. Rogers, “Diffusion of Innovations,” 1962.

[Siegel 1978] A. F. Siegel, "Random arcs on the circle," Journal of Applied Probability, Vol. 15, No. 4, 1978, pp. 774-789.

[Scherber 2005] Scherber, Papadopoulos, “Distributed computation of averages over ad hoc networks” IEEE journal on Selected Areas in Communications April 2005.

[Schurgers 2002] C. Schurgers, V. Tsiatsis, S. Ganeriwal, M. Srivastava, “Optimizin Sensor Networks in Energy-Latency-Density Design Space,” IEEE transactions on Mobile Computing 1(1), pp 70-80, 2002.

[Surowiecki 2004] Surowiecki, James (2004). The Wisdom of Crowds: Why the Many Are Smarter Than the Few and How Collective Wisdom Shapes Business, Economies, Societies and Nations.

[Tian 2002] D. Tian, N. D. Georganas, "A coverage-preseving node scheduling scheme for large wireless sensor networks," in First ACM International Workshop on Wireless Sensor Networks and Applications, 2002, pp. 32-41.

[Tian 2003] D. Tian, N. D. Georganas, “A node scheduling scheme for energy conservation in large wireless sensor networks,” Wireless Communications and Mobile Computing Journal 3(2), pp 271-290, 2003.

[Vicsek 1995] T. Vicsek, A. Czir, E. Ben-Jacob, I. Cohen, O. Shochet, “Novel Type of Phase Transition in a System of Self-Driven Particles,” Physical Review Letter, 75(6), pp 1226 – 1229, 1995.

[Wang 2003] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, C. Gill, “Integrated Coverage and Connectivity Configuration in Wireless Sensor Networks,” In Proceedings of Sensys 2003.

[Wang 2005] X. Wang, G. Xing, Y. Zhang*, C. Lu, R. Pless, C. Gill, "Integrated Coverage and Connectivity Configuration in Wireless Sensor Networks", ACM Transactions on Sensor Networks, August 2005, 1(1), pp. 36-72.

[Xiao 2005] Xiao, Boyd, Lall, “A scheme for robust distributed sensor fusion,” 2005.

[Xiao 2004] L Xiao, S Boyd, “Fast linear iterations for distributed averaging,” Systems and Control Letters, 2004.

[Xu 1997] Cheng-Zhong Xu, Francis C.M. Lau, Load Balancing in Parallel Computers: Theory and Practice, The Springer International Series in Engineering and Computer Science, 1997.

[Xu 2001] Y. Xu, J. Heidemann, D. Estrin, “Geography-informed Energy Conservation for Ad Hoc Routing,” Proceedings of the ACM/IEEE International Conference on Mobile Computing and networking, 2001.

[Ye 2003] F. Ye, G. Zhong, J. Cheng, S. Lu, L. Zhang, “PEAS: A Robust Energy Conserving protocol for Long-Lived Sensor Networks,” Proceedings of the Twenty Third International Conference on Distributed Computing Systems, 2003.

[Zhao 2002] F. Zhao, J. Shin, J. Reich, “Information-driven dynamic sensor collaboration,” IEEE Signal Processing Magazine, 19(2), pp 61-72, 2002.

[Zhong 2003] F. Ye, G. Zhong, S. Lu and L. Zhang, "Peas: a robust energy conserving protocol for long-lived sensor networks," In Proceedings of ICDCS, Vol. 23, 2003,pp. 28-37.