2007-06

# Energy conservation in wireless sensor networks

Vincent, Patrick J.

Monterey, California. Naval Postgraduate School, 2007.

http://hdl.handle.net/10945/10228

# NAVAL POSTGRADUATE SCHOOL

## MONTEREY, CALIFORNIA

# DISSERTATION

**ENERGY CONSERVATION IN WIRELESS SENSOR NETWORKS**

by

Patrick J. Vincent

June 2007

Dissertation Supervisor: Murali Tummala

**REPORT DOCUMENTATION PAGE**

*Form Approved OMB No. 0704-0188*

|  |  |  |
|---|---|---|
| **1. AGENCY USE ONLY *(Leave blank)*** | **2. REPORT DATE**<br>June 2007 | **3. REPORT TYPE AND DATES COVERED**<br>Doctoral Dissertation |

**4. TITLE AND SUBTITLE**: Energy Conservation in Wireless Sensor Networks

**5. FUNDING NUMBERS**

**6. AUTHOR(S)** Vincent, Patrick J.

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**
Naval Postgraduate School
Monterey, CA 93943-5000

**8. PERFORMING ORGANIZATION REPORT NUMBER**

**9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)**
N/A

**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

**11. SUPPLEMENTARY NOTES** The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

**12a. DISTRIBUTION / AVAILABILITY STATEMENT**
Approved for public release; distribution is unlimited.

**12b. DISTRIBUTION CODE**

**13. ABSTRACT (maximum 200 words)**

This dissertation presents a system-level approach for minimizing the power expended in achieving communication between a ground-based sensor network and an overhead Unmanned Aerial Vehicle (UAV). A subset of sensor nodes, termed a *transmit cluster*, aggregates data gathered by the network and forms a distributed antenna array, concentrating the radiated transmission into a beam aimed towards the UAV. We present a method for more uniformly distributing the energy burden across the sensor network, specifying the time that should elapse between reassignments of the transmit cluster and the number of hops that should be placed between successive transmit clusters. We analyze the performance of two strategies for reconfiguring the communication burden between the sensor network and the UAV in order to bring the UAV and the sensor network's beam into alignment quickly, while minimizing the energy expenditure. We analyze the optimal number of nodes that should participate in a beamforming process in order to minimize the energy expended by the network, and we provide a framework to analyze the minimum energy expended in a simple beamforming algorithm. Finally, we analyze the probability that an arbitrarily selected sensor node is connected to a specified number of other nodes and we present an algorithm for the formation of near-linear arrays given random placement of nodes.

**14. SUBJECT TERMS** Sensor Networks, Distributed Antenna Array, Energy Conservation

**15. NUMBER OF PAGES**
202

**16. PRICE CODE**

|  |  |  |  |
|---|---|---|---|
| **17. SECURITY CLASSIFICATION OF REPORT**<br>Unclassified | **18. SECURITY CLASSIFICATION OF THIS PAGE**<br>Unclassified | **19. SECURITY CLASSIFICATION OF ABSTRACT**<br>Unclassified | **20. LIMITATION OF ABSTRACT**<br>UL |

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

# ENERGY CONSERVATION IN WIRELESS SENSOR NETWORKS

Patrick J. Vincent
Commander, United States Navy
B.S., Polytechnic Institute of New York, 1984
M.S., Naval Postgraduate School, 1992

Submitted in partial fulfillment of the
requirements for the degree of

**DOCTOR OF PHILOSOPHY IN ELECTRICAL ENGINEERING**

from the

**NAVAL POSTGRADUATE SCHOOL**
**June 2007**

Author: Patrick J. Vincent

Approved by:

| | |
|---|---|
| Murali Tummala<br>Professor of Electrical and<br>Computer Engineering<br>Dissertation Supervisor | Roberto Cristi<br>Professor of Electrical and<br>Computer Engineering |
| John McEachen<br>Associate Professor of Electrical<br>and Computer Engineering | Craig Rasmussen<br>Associate Professor<br>of Mathematics |
| Frank Kragh<br>Professor of Electrical and<br>and Computer Engineering | |

Approved by: Jeffrey B. Knorr, Chair, Department of Electrical and
Computer Engineering

Approved by: Julie Filizetti, Associate Provost for Academic Affairs

# ABSTRACT

This dissertation presents a system-level approach for minimizing the power expended in achieving communication between a ground-based sensor network and an overhead Unmanned Aerial Vehicle (UAV). A subset of sensor nodes, termed a *transmit cluster*, aggregates data gathered by the network and forms a distributed antenna array, concentrating the radiated transmission into a beam aimed towards the UAV. We present a method for more uniformly distributing the energy burden across the sensor network, specifying the time that should elapse between reassignments of the transmit cluster and the number of hops that should be placed between successive transmit clusters. We analyze the performance of two strategies for reconfiguring the communication burden between the sensor network and the UAV in order to bring the UAV and the sensor network's beam into alignment quickly, while minimizing the energy expenditure. We analyze the optimal number of nodes that should participate in a beamforming process in order to minimize the energy expended by the network, and we provide a framework to analyze the minimum energy expended in a simple beamforming algorithm. Finally, we analyze the probability that an arbitrarily selected sensor node is connected to a specified number of other nodes and we present an algorithm for the formation of near-linear arrays given random placement of nodes.

THIS PAGE INTENTIONALLY LEFT BLANK

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# ACKNOWLEDGEMENTS

I would like to thank my wonderful wife, Cynthia, who has always provided love, encouragement and support.

I would like to express my sincere appreciation to Dr. Murali Tummala for his mentoring, patience and guidance as my dissertation supervisor. My committee members—Dr. Murali Tummala, Dr. Roberto Cristi, Dr. John McEachen, Dr. Craig Rasmussen and Dr. Frank Kragh—provided many keen insights and suggestions, making this work far better than it would have otherwise been.

*Deo Gratias.*

THIS PAGE INTENTIONALLY LEFT BLANK

# EXECUTIVE SUMMARY

A wireless sensor network is an interconnected collection of *sensor nodes* that monitor and collect information about the physical environment, process the collected signals, and transmit the collected data to another location for further processing and interpretation. Sensor networks have attractive military applications since they can be deployed in dangerous, remote or inhospitable environments. For example, a large number of sensor nodes can be dropped from an aircraft, densely covering a large area. Left unattended, these nodes can then form a wireless ad hoc network, gather information about phenomena of interest, and transmit the collected data to the ultimate end-users via an unmanned aerial vehicle (UAV) that acts as an airborne relay.

For military applications, sensor nodes should be small (so as to be covert) and inexpensive (in order to be expendable). Furthermore, such applications presume that the hardware of a sensor node will not be serviced in the field; when a sensor's battery fails, the whole sensor node is rendered inoperable. Thus, careful power management is critical in order to extend the lifetime of the sensor network.

In this dissertation, we present a system-level approach for minimizing the power expended in achieving successful communication between a ground-based sensor network and an overhead UAV. Our approach is premised on the notion of using a subset of sensor nodes, termed a *transmit cluster*, to receive and aggregate data gathered by the entire network. The transmit cluster then forms a distributed antenna array, concentrating the radiated transmission into a narrow beam aimed towards the UAV.

We begin in Chapter II by first presenting a method for more uniformly distributing the energy burden across the sensor network. Because the duties of the transmit cluster are power-intensive, the role of transmit cluster must be shifted to different nodes as time progresses. We present and analyze an algorithm to reassign the transmit cluster, specifying the time that should elapse between reassignments and the number of hops that should be placed between successive transmit clusters in order to achieve three competing goals: First, we wish to better and more broadly spread the

energy load across the sensor network while, second, minimizing the energy expended in moving the transmit cluster, all the while, third, reducing to the extent practicable the time to bring the UAV and the sensor network's beam into alignment.

Successful communication requires that the sensor network's narrow transmission beam be directed such that the UAV falls spatially within it. In Chapter III, we describe, analyze and compare the performance of two strategies for reconfiguring the communication and computational burden between a ground-based sensor network and an overhead UAV. Both strategies bring the UAV and the sensor network's transmission beam into alignment, while minimizing the energy expended by the sensor network. The performance of the two strategies is compared in terms of probability of beam-UAV alignment as a function of time, and the expected time to alignment. We examine the performance tradeoff between the choice of strategy and parameters of the sensor network that affect power conservation and several scenarios are presented to illustrate the design tradeoffs that are made analytically accessible by our analysis.

Having developed an energy-efficient algorithm to move the transmit cluster to other locations in the sensor network to evenly spread the energy load and extend the lifetime of the network, and having also developed an energy-efficient approach to aligning the transmit cluster's beam with the UAV, we then examine how the transmit cluster should assemble its antenna array in an energy-efficient fashion. Specifically, in Chapter IV, we analyze the optimal number of sensor nodes that should participate in a beamforming process in order to minimize the energy expended by nodes in a sensor network. We provide a framework that a system designer can use to analyze the minimum energy expended in a simple beamforming algorithm, which, in turn, specifies the optimal number of nodes to use to minimize the total energy expenditure.

Finally, in Chapter V, we analyze our design approach in the face of randomness. We first propose a clustering methodology for sensor networks. We then analyze the probability that an arbitrarily selected sensor node is connected to a specified number of other nodes; i.e., the probability that a communication path exists between a selected

node and at least *m* other nodes for the purposes of forming a distributed antenna array. We present an algorithm for the formation of near-linear arrays given random placement of sensor nodes, and analyze the effectiveness of our approach through simulation.

THIS PAGE INTENTIONALLY LEFT BLANK

# I. INTRODUCTION

A wireless sensor network is an interconnected collection of *sensor nodes* that monitor and collect information about the physical environment and transmit the collected data to another location for processing and interpretation. Each individual sensor node in the network consists of one or more sensors, a radio transceiver, a microprocessor and a small battery housed in a common hardware unit, as shown in Figure 1. The individual sensors within a sensor node are designed to detect one or more aspects of the physical environment, such as motion, temperature, sound, the presence of chemical agents, the presence of nearby metallic objects, etc. Once a detection occurs, a sensor node transmits the sensed data using its on-board transceiver, and the data arrives at the ultimate end-user after traversing one or more wireless links.

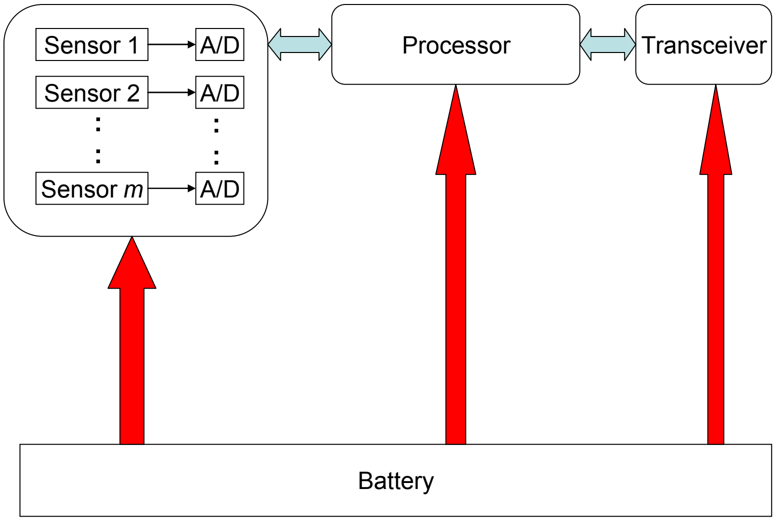


Figure 1. Sensor Node Composition

## A. WIRELESS SENSOR NETWORKS IN COMMERCIAL AND ENVIRONMENTAL APPLICATIONS

The variety of commercial and environmental applications involving wireless sensor networks can be illustrated by the following examples:

- BP, the oil and gas company—concerned over consistently losing track of 5 percent of its oil and gas inventory in the U.S. rail and shipyard networks—is outfitting railcars with sensors that continually monitor the car's location, weight and temperature. [1]
- A wireless sensor network was deployed in 2003 to monitor the nesting burrows of an endangered bird species on Great Duck Island, off the coast of Maine. The system monitored temperature, humidity, barometric pressure and bird nest occupancy, and consisted of a 50-node single hop network as well as a multihop 100-node network. [2]
- General Electric is developing a wireless sensor network as part of a $6 million project to boost the energy efficiency of electric motors. Motors throughout a plant are mounted with sensors that monitor motor efficiency, vibration and temperature, transmitting the results back to a remote computer for analysis. [3]
- A wireless sensor network was installed on a heavily trafficked bridge in Vermont to measure the dynamic and static strain placed on the structure. [4]
- Intel is using sensors in its semiconductor fabrication plant to monitor for vibrations that may indicate impending failure. [5]

In addition to the interests of individual industries, the National Science Foundation (NSF) has spent more than $100 million to foster research on sensor networks during the past several years. [6]

## B. IRELESS SENSOR NETWORKS IN MILITARY APPLICATIONS

Sensor networks have attractive military and homeland defense applications since they can be deployed in dangerous, remote or inhospitable environments. For example, a large number of sensor nodes can be dropped from an aircraft, densely covering a large area of interest. Left unattended, these nodes can then gather information about enemy movements, chemical agent deployment, the presence of radiological materials, or any other phenomena that might be of interest.

It is generally assumed that there is no existing infrastructure to support communication, except for the sensor nodes themselves. The initial exact positions of the sensor nodes cannot be predetermined since they might, as mentioned, be deployed from an aircraft or otherwise deployed in a rapid (not careful) manner. Ideally, the sensor nodes would collaborate among themselves to dynamically form a wireless ad hoc network, thereby transmitting the collected information back to the end-users located far from the scene of danger. Each sensor node, in addition to performing its own sensing and communication duties, should also be able to serve as a router to support multihop communications for other sensor nodes that cannot communicate directly.

The resulting sensor network is not static; individual sensor nodes may subsequently fail (e.g., due to battery depletion), or environmental factors (e.g., rain) may affect the ability of any two given sensor nodes to communicate. In each of these cases, the sensor network should adapt to these changes to ensure reliable communication of the sensed data to the ultimate end-users.

As attractive as many military applications might appear on paper, a host of technological hurdles must be overcome before the benefits of sensor networks might be fully exploited. For a detailed review of the history of sensor network research, from its humble beginnings at a DARPA-sponsored workshop in 1978, see [7]. For an excellent (and still quite timely) survey of wireless sensor networks, see [8].

### 1. The Problem of Power Management

The archetypical sensor node shown in Figure 1 contains one or more sensors that are used to detect aspects of the physical environment. A sensor node may have multiple sensors of the same type (e.g., three temperature sensors) for redundancy, or may have a heterogeneous mix of sensors to measure different phenomena (e.g., a temperature sensor, an audio sensor and a magnetic sensor). It is important for most applications that sensors of the same type (in a given sensor node and within all nodes in the network) be uniformly calibrated against the same standard scale. For example, if the sensor network is being used to measure the concentration of an airborne biological agent, the level detected by a sensor at a specified location should be the same as that provided by any other sensor in the network*, were it also at that same location*. We assume for our studies that all sensors are perfectly calibrated during the manufacturing process and do not drift following deployment in the field.

The analog signals resulting from these sensor observations are converted to a digital format, and sent to the processor. The processor is the "brains" of the unit, managing all aspects of the sensor node's operation. The transceiver transmits data to, and receives data from, external sources (e.g., other sensor nodes).

All of the aforementioned components of the sensor node draw their power from the sensor node's battery, as shown in Figure 1.

An examination of the obstacles preventing easy deployment of military sensor networks starts with the sensor nodes themselves. For military applications, sensor nodes should be small (so as to be covert) and inexpensive (in order to be expendable). To give some indication of *how small is small*, Figure 2 shows a picture of a commercially available sensor node next to a US quarter coin.

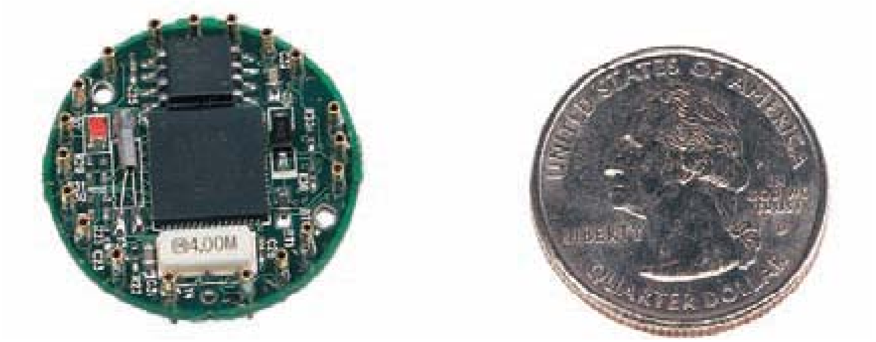


Figure 2. Crossbow © MICA2DOT mote Next to US 25¢ coin.

Furthermore, most military applications presume that the physical hardware of a sensor node will not be serviced in the field (although it may be possible to update or modify the software via wireless communication). So, for example, when a sensor's small battery fails, the whole sensor node (not just the battery) is rendered inoperable. It is important, therefore, that sensor nodes be inexpensive, so that employment of a sensor network would be cost-effective.

Therein lies a significant challenge: A small, inexpensive sensor node necessitates use of a small, inexpensive battery that can not be recharged. And use of such a battery, in turn, places a strong premium on power management in order to extend the lifetime of the network. Suppose an individual sensor node detects the presence of, say, a chemical agent and wishes to transmit this information. If the sensor node transmits at a high power, it will drain its battery quickly and soon die, never to be useful again. But if the sensor node decides to extend its life by transmitting at too low a power, the transmission may not be strong enough to reach any other node: the sensor node will have a long life but never communicate its data.

We will show in Chapter II that for military scenarios—where the sensor network is deployed in dangerous or remote environments—it is highly advantageous to transmit the sensor network's collected data to the ultimate end-users via an unmanned aerial vehicle (UAV) that acts as an airborne relay. But achieving successful communication from a sensor network to a UAV presents a formidable challenge since a UAV is considerably far from the nearest sensor node. Successful communication over such a distance would require that a sensor node transmit at a high power, but such action would quickly deplete the node's battery reserves. Thus the problem: the strong premium placed on power management (in order to extend the lifetime of the network) places severe restrictions on a node's transmit power, and consequently, on the maximum range at which a node's transmissions can be successfully received, making communication with a UAV very difficult.

### 2. The Use of a Distributed Antenna Array

To enable successful communication between a sensor network and a UAV, we propose having multiple sensor nodes coordinate their transmissions—each sending the same signal except for calculated phase and amplitude offsets. In such cases, the propagating electromagnetic waves from each of the sensor node antennas will interfere with each other, in some directions constructively (i.e., reinforcing each other) and in other directions destructively (i.e., tending to cancel each other), all depending on the relative amplitudes and relative phase differences between the individual node transmissions. The net effect is that the total radiated power from the multiple nodes can be focused in preferred directions. The participating sensor nodes are said to comprise a distributed antenna or an *antenna array*.

The key condition for this approach to work is that the sensor network's narrow transmission beam must be directed such that the UAV falls spatially within it. Bringing the sensor network's transmission beam into alignment with the UAV's position is, in turn, another difficult problem. Military applications must presume that the sensor network does not know *a priori* where the UAV is, nor does the UAV know the direction in which the sensor network has aimed its transmission beam. The techniques that an antenna can employ to estimate the direction of the UAV involve a large number of radio transmissions between sensor nodes, and a large number of computations performed by the nodes implementing the algorithm, all of which expend scarce battery resources, thereby limiting the lifetime of the involved nodes.

## C. THE SOLUTION APPROACH

In this dissertation, we present a system-level approach for minimizing the power expended in achieving successful communication between a ground-based sensor network and an overhead UAV. Our approach is premised on the notion of using a subset of sensor nodes, termed a *transmit cluster*, to receive and aggregate data gathered by the entire network. The transmit cluster then forms a distributed antenna array, concentrating the radiated transmission into a narrow beam aimed towards the UAV.

The problem of facilitating communication between a wireless sensor network and a UAV is complex. To solve this problem, we must attempt to coherently organize the various functions that must be carried out. We borrow the solution approach used in the study of computer networks. Computer networks are also complex, and the general approach to making the design more manageable is to organize networks as a series of layers, where each layer performs only a few specific well-defined functions [9]. The layers are built, one on top of the next, with each layer performing a service for the layer above it, and with a new layer created when a different level of abstraction is needed. Ideally, *how* a layer does its job is not known by the layers above and below, so that if we replace a layer with a different implementation that accomplishes the same task using a different mechanism, the other layers should not even be aware of it.

As a simple example from the field of computer networking, we might design an email application without worrying about such details as:

1) how the message should be encrypted
2) how end-to-end data errors should be minimized
3) how a route through the network should be chosen
4) how errors on individual links comprising the route should be minimized
5) how bits should be represented in the physical transmission media.

Once the email application is designed, we then might turn our attention to the encryption problem (the first task in list above, at the highest level of abstraction), without worrying about the other items on the list. Once the encryption problem is solved, we then turn our attention to the next level of abstraction, the next item on the list, minimizing end-to-end errors, without yet worrying about the lower-level details (i.e., choosing a route, etc.). We continue to incrementally solve the problem in this manner. Ideally, the layers for this simple problem would be built, one on top the next, as illustrated in Figure 3.

We note that the preceding example is very simplistic, and ignores a host of concerns that the network designer ordinarily must address (e.g., encoding, compression, synchronization, flow control, multiplexing, congestion control, addressing, internetworking, framing, controlling access to shared media, etc.). Additionally, the interfaces between layers are usually not so clean, and the algorithms employed at different layers will often have inter-dependencies. This simplified example is not intended to exhaustively summarize the principles for the design of a computer network for sending email. Nevertheless, while we do not intend to overemphasize a strict hierarchy and isolation of function across layers, we will adopt the generally accepted notion of layering for our problem as well.

| | |
|---|---|
| Layer 6 | Design of email application |
| Layer 5 | Design of encryption scheme |
| Layer 4 | Design of end-to-end error control scheme |
| Layer 3 | Design of routing algorithm |
| Layer 2 | Design of technique to control errors on an individual link |
| Layer 1 | Design of techniques to represent bits on the physical channel |

Figure 3. Simplified Email Communication Designed as a Series of Layers

We divide our problem into a series of layers as follows, starting at the highest level of abstraction. Specifically, we first decide how the distributed antenna should be moved about the sensor network in a manner that broadly spreads the energy load across the sensor network, while minimizing the energy expended in moving the antenna. We examine this problem without worrying about the details of how the antenna's beam is aligned with the UAV (in a manner that minimizes energy expenditure), and without worrying about how the sensor network is to successfully assemble nodes into a distributed antenna (also in a manner that minimizes energy expenditure). This first task, at the highest level of abstraction, is examined in Chapter II.

Once this problem has been solved, we move down to the next layer of abstraction: how should the distributed antenna's beam be aligned with the UAV in a manner that minimizes energy expenditure. We examine this problem without worrying about how the sensor network has successfully assembled nodes into a distributed antenna (in a manner that minimizes energy expenditure). This second problem is examined in Chapter III.

Having developed an energy-efficient algorithm to move the distributed antenna to other locations in the sensor network to evenly spread the energy load and extend the lifetime of the network, and having also developed an energy-efficient approach to aligning the distributed antenna's beam with the UAV, we then examine how the array should be assembled in an energy-efficient fashion. This is the task we undertake in Chapter IV.

Finally, in Chapter V, we analyze our design approach in the face of randomness. We propose a clustering methodology for sensor networks, analyze the probability that a cluster will be able to form an antenna array given random placement of sensor nodes, and present an algorithm for the formation of near-linear arrays.

Thus, in summary, we will use the layering approach shown in Figure 4. We will see that on occasion the interfaces between layers are not perfectly clean, and the algorithms employed at different layers will have inter-dependencies which will be discussed as they arise.

| | |
|---|---|
| Layer 4 | Design power-conserving techniques to shift the distributed antenna as time progresses |
| Layer 3 | Design power-conserving techniques to align the distributed antenna's beam with the UAV |
| Layer 2 | Design power-conserving techniques to form a distributed antenna for beamforming assuming deterministic node placement |
| Layer 1 | Analyze algorithms and protocols under stochastic conditions |

Figure 4. Sensor Network-UAV Communication Problem Designed as a Series of Layers

THIS PAGE INTENTIONALLY LEFT BLANK

# II. A NEW METHOD FOR DISTRIBUTING POWER USAGE ACROSS A SENSOR NETWORK[*]

As mentioned in Chapter I, a wireless sensor network is an interconnected collection of *sensor nodes* that monitor and collect information about the physical environment and transmit the collected data to another location for processing and interpretation. Each individual sensor node in the network consists of one or more sensors, a radio transceiver, a microprocessor and a small battery, as shown in Figure 1. Once a sensor detection occurs, a sensor node transmits the sensed data using its on-board transceiver, and the data arrive at the ultimate end-user after traversing one or more wireless links.

In Chapter I we noted that sensor nodes employed in military applications should be small (so as to be covert) and inexpensive (in order to be expendable). Furthermore, most military applications presume that the physical hardware of a sensor node will not be serviced in the field, so when a sensor node's small battery fails, the whole sensor node is rendered inoperable. It is important, therefore, that sensor nodes be inexpensive, so that employment of a sensor network would be cost-effective. These considerations, in turn, necessitate use of a small, inexpensive battery that can not be recharged. And use of such a battery, in turn, places a strong premium on power management in order to extend the lifetime of the network.

Before discussing our innovative approaches to power management, it is useful to frame the preceding discussion by examining the quantitative parameters of currently available technology. Several companies sell devices that have potential use as rudimentary sensor nodes in military applications. The hardware—sensors, batteries, transceiver and processor packaged into small palm-size devices—is commonly termed a *mote*.

---

[*] The results presented in this chapter were previously presented at the Third Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, SECON 2006 [10].

These commercially available motes can be deployed with an array of different user-selected sensors. Very small motes examined in [11] were found to exhibit communication ranges of approximately 20 m and lifetimes ranging from one week (continuous operation) to two years (1 percent duty cycle). The Berkeley Mote has a typical radio bandwidth of less than 100 kbps, whereas the Intel iMote is able to communicate at rates of 500 kbps [12]. The open-source operating system for motes, termed TinyOS, is designed to run on just 128 kB of memory. The operating system emphasizes power management of the various hardware components as well as low-level hardware control. Off-the-shelf sensor nodes range in price from \$100 to \$200 each [13], but such motes are not typically designed to meet military specifications (e.g., they are not intended to be dropped from aircraft). Even so, a sensor network of several hundred of these motes remains an expensive proposition.

Before delving further into the implications of the small inexpensive battery on the problems of power management in sensor networks, we briefly note a different problem that we will visit later in the dissertation. The utility of many sensor network applications often hinges on the ability to tie detected phenomena to a geographic coordinate system (e.g., latitude and longitude). For instance, if a sensor node detects the presence of anthrax, the first question an end-user will likely ask is: Where is it? Simply put: sensors need to know where they are. In many wireless networking applications, network nodes that have a clear view of the sky can determine their position by using Global Positioning System (GPS) receivers. Unfortunately, GPS receivers consume significant power and are not amenable for use by power-constrained devices. As such, we assume our sensor nodes do not have GPS capability, and, as such, no such location-awareness system is shown in Figure 1.

It is thus fair to say that rudimentary versions of our archetypical sensor node are available today, at a high price. For the foreseeable future, though, power conservation and management will continue to guide all aspects of sensor network design and employment.

## A. THE NEED FOR UNMANNED AERIAL VEHICLES

In military scenarios, it is highly advantageous to transmit the sensor network's collected data to the ultimate end-users via an unmanned aerial vehicle (UAV) that acts as an airborne relay. As shown in Figure 5, a communication signal might proceed from the sensor network to the UAV (via a line-of-sight link), then from the UAV to the intended receiver (via a second line-of-sight link).

Two questions arise: First, why an *airborne* relay? And, second, presuming we do decide to employ an airborne relay, why an *unmanned* airborne relay?

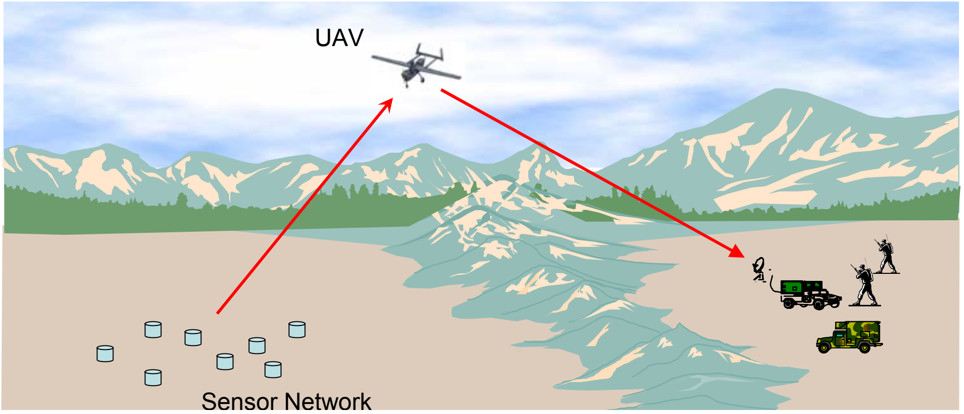


Figure 5. Use of a UAV

To answer the second question first: If the sensor network is located in a hazardous environment, then employing a UAV places only equipment—not human life—at risk. If the sensor network is located in "enemy territory," then low-altitude flights are very vulnerable to a host of cheap and readily available anti-air weapons, while U.S. policy (rightly) places a heavy premium on the life of each airman. Similarly, if the sensor network is located near a chemical cloud or a radiological event, then a nearby airborne platform would also necessarily be near the hazard. In acting merely as an airborne relay, a UAV can perform just as well as a manned aircraft, so we posit that the real question is: Why *not* use a UAV?

But why use an airborne relay at all? Why not communicate to the end-users via the sensor network's ad hoc network infrastructure? An airborne relay is advantageous for three reasons:

- First, for frequencies above 30 MHz, electromagnetic waves propagate in a straight line. Beyond the distance to the horizon, labeled *d* in Figure 6, the Earth will block the line-of-sight (LOS) path, and any receivers beyond this range will fall within the transmitter's "shadow zone," unable to receive transmissions without suffering diffraction loss. A formula for the distance to the horizon is given in [14] as

  $$d = \sqrt{2h},$$

  where $h$ is the height of the transmitting antenna (in feet) and $d$ is the distance to the radio horizon in statute miles (1 statute mile = 5280 feet). (Note that the formula accounts for the slight downward refraction of radio waves caused by a standard atmosphere.)

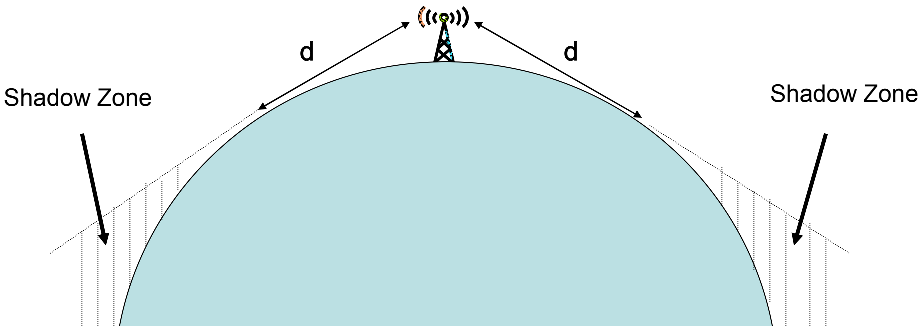


Figure 6. Line-of-sight (LOS) Radio Horizon

Now, consider the implications of this for sensor networks. If a sensor node's transmitting antenna is three inches high (a reasonable value, as we shall shortly see), the distance to the horizon is about 3700 feet. When one considers that, for a three-inch antenna, the local features of the ground (rocks, grass, etc) will very likely obscure the view to the horizon, the estimated transmission range of 3700 feet is surely optimistic. In any event, if sensed data are to be communicated to an end user using only the ground-based sensor network, then the end-user will need to be very close to the final sensor node in the communication path. On the other hand, if a UAV is employed, then the transmission range is no longer limited by the distance to the horizon.

- The second reason that an airborne relay is advantageous again relates to the fact that the transmitter and receiver must be within sight of each other. Consider again Figure 5, which shows a hilly terrain separating a sensor network from the users of the data. Even if the sensor network is nearby in range, the hills prevent line-of-sight communications. As a second example, consider Figure 7, which shows a sensor network deployed within an urban environment. (The extent of the sensor network is indicated by the white oval; the individual sensor nodes might be very small, and unnoticed by passers-by.) In such a case, buildings and other structures effectively block the line-of-sight communications in every direction except "skyward," meaning that the only way for the sensor network to communicate within the urban canyons of a built-up area is via an aircraft overhead.

Figure 7. Sensor Network in an Urban Setting

- The third reason that an airborne relay is advantageous relates to propagation loss. If a sensor node (say, Node *A*) needs to transmit a block of data to Node *B*, which is separated by a distance *d*, Node A's required transmit power as a function of the distance *d*, $P_S(d)$, may be estimated using the log-distance path loss model as

  $$P_S(d) = k\, d^n ,$$

  where the constant *k*, the *path loss coefficient*, accounts for the transmitting and receiving antenna gains, the wavelength of the radiated signal, and the power required at the receiver to attain a specified bit error rate (BER) as well as losses not related to propagation (e.g., filter losses)

> [15]. Concerning propagation, the important effects in indoor and urban wireless environments are reflection, diffraction and scattering. The *path loss exponent*, $n$, is equal to two if propagation is in free space, but for low-lying antennas the signal propagation is affected by partial signal cancellation by a ground-reflected ray as well as the surface roughness and the presence of reflecting and obstructing objects, and in such instances the power required to transmit a signal over a distance $d$ is proportional to $d^4$ [16]. Because the propagation loss is much greater (i.e., the transmit power decays more rapidly) for antennas near the ground, it is much more power-efficient to move communications "off the ground" as quickly as possible, and an effective way to accomplish this is by handing off the data stream to a UAV.

Use of a UAV has important implications for the tactical use of military sensor networks. Specifically, the decision to employ a UAV as described above has important implications on the *when* and *where* of sensor data delivery to the ultimate end-users.

In terms of *when* sensed data are delivered to a sink, sensor networks can, in general, be classified as either continuous, event-driven, sink-initiated or hybrid [17]. To illustrate this distinction, consider a hypothetical sensor network deployed over a terrain, where, say, each node measures the ambient temperature. In the *continuous* model, each node would provide a steady stream of temperature readings to one or more sinks. In the *event-driven* model, sensors will only report a temperature reading if a specified event occurs (e.g., a sensor detects a temperature over 80°F). In the *sink-initiated* model, sensors will only report temperatures upon receiving an explicit request from a sink. In the *hybrid* model, some combination of these three approaches is used. Now, in our applications, the sink is a UAV, and, since sensor data will be supplied only when a UAV comes within range, we clearly follow the *sink-initiated* paradigm.

In terms of *where* the data should be delivered, the obvious answer is: To the UAV for follow-on relaying to the ultimate end-users. A complication arises when the sensor network is operating with no UAV (i.e., no explicit sink) present. Suppose, for example, that a military sensor network is deployed in an inhospitable region to detect the presence of anthrax. In this case a UAV might be sent over the region every so often (as opportunities arise) to query the sensor network. In instances when no UAV is present, what does the network do with the data that arises in meantime? Should the network discard all sensed data—potentially important information—if there is no UAV present at the time of detection, in the interests of conserving power? This would mean, in effect, that we only use the sensor network at those specific times when the network has a direct communication link with the UAV, and we are not interested in any history outside of such time periods. Or, on the other hand, is sensed data so important that it should be stored pending the arrival of the UAV? If the answer to the latter question is yes, then *where* should the data be stored? Only at the node that originally sensed the event, in the interests of conserving power? Or, should we expend power to replicate important data at other nodes, just in case the node that first detected the data should expire (and take with it the data)? The answers to these questions will critically depend on the application that necessitates employment of the sensor network. We will assume for this dissertation that sensed data are so important that they must be stored pending the arrival of the UAV (i.e., we *are* interested in the history between UAV visits), and that the sensed data are routed to a node that will consolidate the sensor network's data for transmission to the UAV.

But, having decided to use a UAV as an intermediate relay, communication from a sensor network to a UAV still presents a formidable challenge. As mentioned, military applications will use small (covert) low-cost (expendable) nodes, and most applications presume that the physical hardware of a sensor node will not be serviced in the field, so when a node's small battery fails, the whole sensor node is rendered inoperable. The strong premium placed on power management (in order to extend the lifetime of the network) places severe restrictions on a node's transmit power, and consequently, on the maximum range at which a node's transmissions can be successfully received.

## B. THE USE OF A DISTRIBUTED ANTENNA ARRAY

For wireless communication, the electrical signal energy from the transmitter is converted to propagating electromagnetic waves by an associated antenna. Successful communication requires that some of the energy that originates at the transmit antenna arrives at the receiver's antenna, where the incident electromagnetic waves are converted back to electrical signals that, hopefully, are smaller-amplitude replicas of those that existed in the transmitter.

An antenna's *radiation pattern* displays the relative distribution of the radiated power as a function of the direction in space. With the antenna at the origin of a coordinate system, the radiation pattern can be found by measuring the power density at all points on the surface of a sphere at a distance $r$ from the origin (i.e., the antenna). Since we are interested in the environment at a specified distance from the antenna, a spherical coordinate system $(r, \theta, \phi)$ is usually used to describe the radiation pattern (see Figure 8). Note that the variable $r$ ranges from zero to infinity. The angle $\theta$, often termed the *elevation angle* (particularly when the earth is taken to be the *x*-*y* plane), ranges from 0° to 180° and the angle $\phi$, termed the *azimuth angle*, ranges from 0° to 360°. (Note that once $r$ is fixed, the power density is a function of only $\theta$ and $\phi$.)

Small, commercially available sensor nodes are advertised as having omnidirectional antennas. An omnidirectional radiator in the three-dimensional sense, i.e., an isotropic radiator, can not be physically realized. Nevertheless, we will assume that a sensor node's antenna radiates isotropically. This assumption greatly simplifies the analysis. As we will point out later, there are opportunities for antenna theorists to incorporate the actual antenna characteristics into our framework.

Any isotropic antenna radiates power uniformly in all directions over the entire solid angle of $4\pi$ steradians.[§] For an isotropic antenna, the power density will have the same value on any sphere centered at the antenna; that is, since the power radiates uniformly in all directions, the power density is the same at any point a distance $r$ from the antenna. Most of the transmitted power is wasted; only the small amount of power that happens to propagate in the direction of the UAV is useful. Note that for sensor nodes on the ground, an immediate power loss of 3 dB is encountered.

---

[§] If the radius of a sphere is $r$, then 1 steradian is defined as the solid angle subtended at the center of a sphere by an area on the surface of the sphere equal to $r^2$. Since the entire surface of an area of a sphere is $4\pi r^2$, the total solid angle of a sphere is $4\pi$ steradians.

Suppose, though, that the power could be preferentially radiated in certain directions. In this case, the power density in the preferred directions would be greater than that of the isotropic antenna, while the power density in the other directions will be less than that of an isotropic antenna (with both antennas radiating the same total power).

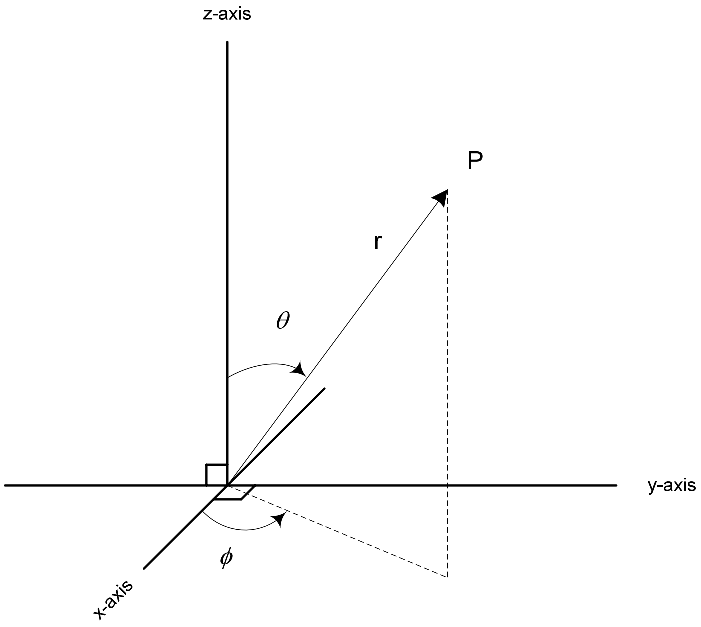


Figure 8. Spherical Coordinate System

We cannot focus power in preferred directions using the single antenna on an individual sensor node, since we are unwilling to equip our nodes with expensive directional antennas. (Recall that sensor nodes must be inexpensive in order to be expendable.) We have proposed [18] having multiple sensor nodes coordinate their transmissions—each sending the same signal except for calculated phase and amplitude offsets. In such cases, the propagating electromagnetic waves from each antenna will interfere with each other, in some directions constructively (i.e., reinforcing each other) and in other directions destructively (i.e., tending to cancel each other), all depending on the relative amplitudes and relative phase differences between the individual node transmissions. (This interference is a consequence of the fact that the total electric field at any point in space is the vector sum of the individual electric field vectors associated with the interfering waves.) The net effect is that the total radiated power from the multiple nodes can be focused in preferred directions. The ability to focus radiated power in certain directions results in antenna *directive gain*, *D*, while the solid angle through which the radiated power is focused is referred to as the *beam*, and the participating transmitting elements are collectively termed an *antenna array*. More formally, the directive gain, *D*, of an antenna array, at a certain point in space, is defined as the ratio of the power density present at that point to that which would be present (at the same point) if a single isotropic antenna were radiating the same total power as the elements in the array.

The *power gain* of an antenna, *G*, is equal to the directive gain reduced by the losses on the antenna. Thus, the power gain, *G*, is equal to the directive gain, *D*, if the antenna is 100% efficient. If the antenna is not 100% efficient, then

$$G = eD,$$

where *e*, the radiation efficiency, is the ratio of the radiated power to the input power. In this dissertation, we shall assume that dissipative losses and impedance mismatch losses can be neglected (i.e., that *e* in the preceding equation is very nearly equal to one, as is frequently the case for frequencies in the VHF band and above [19]), and we will henceforth not distinguish between power gain and directive gain. Accordingly, we will use the simple term *gain*, and the symbol *G*.[**]

Thus, an antenna with a gain of 1 and an input power of 10 watts will be as effective (at the receiver) as an antenna with a gain of 10 and an input power of 1 watt *provided the receiver (the UAV) falls within the beam*. Employment of an antenna array with a gain of 10 permits a reduction in the required total input power by a multiple of 10, while still placing the same amount of power at the receiver. Therefore, if we are interested in minimizing the energy expended by sensor nodes, then we should minimize

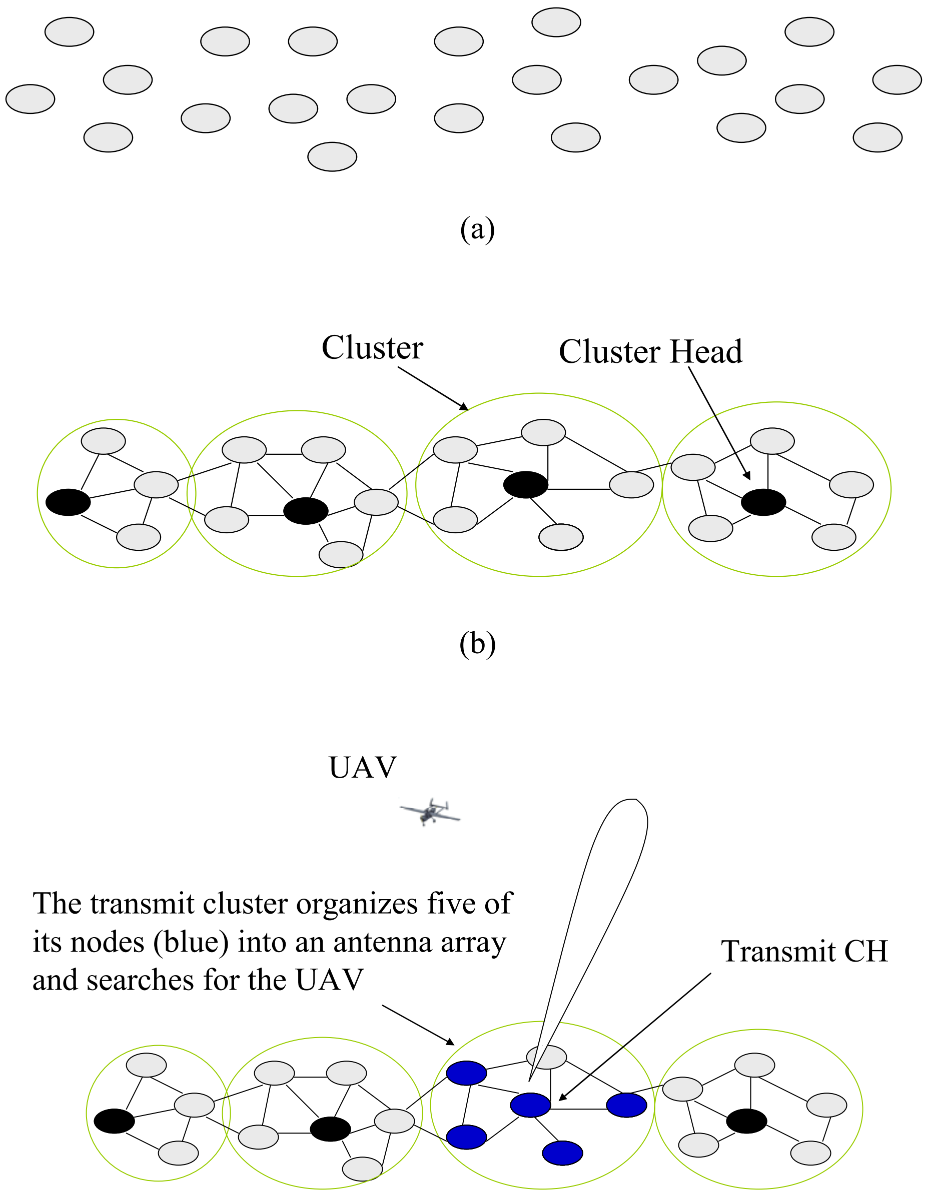
---

[**] The term “gain” is often used to refer to the value of the gain in the maximum-radiation direction. For example, an antenna with a gain of 2 in some directions, a gain of 1.5 in other directions (and a gain of zero in still other directions) will often be said to have a gain of two. The intended meaning should be clear from the context.

the radiated power required for communication. It is thus highly desirable to use an antenna array that manages to concentrate the radiated power within a solid angle less than 4π steradians, so that the gain of the antenna can be increased.

## C. THE SCENARIO FOR TACTICAL MILITARY SENSOR NETWORKS

The military sensor network thus should operate as follows. The sensor nodes are deployed over a dangerous or remote terrain—perhaps dropped from an aircraft—and land on the ground and awaken (Figure 9a). Upon awakening, the sensor nodes partition themselves into non-overlapping clusters, which collectively contain all the nodes in the network. Each cluster has a fixed single node designated as the clusterhead (CH) which oversees and manages the nodes in its cluster. Figure 9b, for example, shows the sensor network partitioned into four clusters, with the clusterheads shown as darkened nodes. (Actual sensor networks composed of hundreds or thousands of nodes will, of course, have very many clusters.) A specified cluster, termed the *transmit cluster*, is selected from among the many clusters to act as the transmission point for the sensor network. That is, any sensed data originating anywhere in the network are routed to the transmit cluster's CH (hereafter termed the *transmit CH*), where they are consolidated with any data already collected, and prepared for transmission to the UAV. The transmit CH organizes a subset of its cluster nodes into a distributed antenna array, coordinating their transmissions to direct the beam towards the UAV, as shown in Figure 9c. Once the beam is aligned with the UAV (Figure 9d) and the communication link is established, the sensor network's data are transmitted to the UAV, for follow-on routing to the ultimate end-users. Because these duties are power-intensive, the role of transmit cluster must be shifted to different clusters as time progresses in order to balance the energy load as evenly as possible across the network to extend the lifetime of the network (Figure 9e).

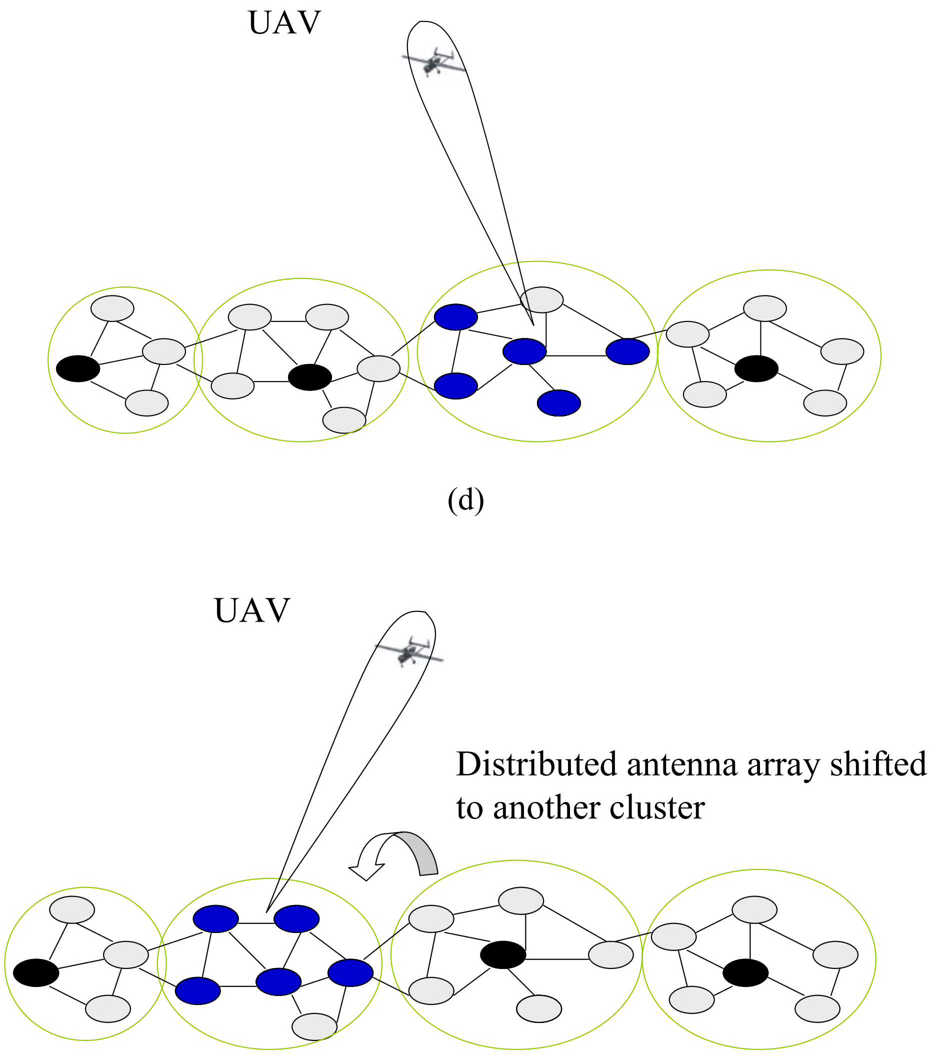


The transmit cluster organizes five of its nodes (blue) into an antenna array and searches for the UAV

(a)

(b)

(c)



(d)

(e)

Figure 9. Sensor Network Operation (continues on next page)

The transmit CH calculates the magnitude and phase offsets to be applied to the otherwise identical transmissions of the participating nodes in order to form a radiation pattern that concentrates the radiated power into a narrow beam. The key condition for this approach to work is that the sensor network's narrow transmission beam must be directed such that the UAV falls spatially within it, as shown in Figure 9d.

If the direction of approach of the UAV (i.e., its elevation and azimuth angles) is known *a priori*, then a number of techniques are available to determine the magnitudes and phase offsets necessary to synthesize the desired radiation pattern to point the beam towards the UAV [20]. Military applications, however, must presume that the sensor network does not know *a priori* where the UAV is, nor does the UAV know the direction in which the sensor network has aimed its transmission beam. The sensor network, lacking GPS capability, is, in fact, not even aware of an absolute geographic coordinate system.

There are a number of different techniques that an antenna array can employ to estimate the direction of arrival of the UAV, some of which are summarized in [21]. All of these methods share a feature in common: They are very complex. In one of the simplest methods, for example, a UAV flies over the sensor network transmitting a known reference signal. The antenna array steers a very narrow beam continually in space, analyzes the spatial spectrum as a function of position, and then selects the direction that yields the highest power as the direction of the UAV, assuming that this power arises from the UAV's transmission. For this method to be effective, the transmit CH will have to continually calculate different sets of amplitudes and phases for the participating sensor nodes comprising the antenna array (so as to steer the beam), transmit these values to the participating nodes, and these sensor nodes, in turn, will have to continually send their transceiver receptions to the transmit clusterhead for analysis of the spatial spectrum. When energy constraints require that the role of transmit cluster be shifted to a different cluster, all relevant data collected up to this point must be passed through the network to the new transmit CH, or the entire process of finding the UAV must begin anew. As all these operations involve a large number of radio transmissions between sensor nodes, and a large number of computations performed by the transmit CH(s), they expend scarce battery resources, thereby limiting the lifetime of the involved nodes.

## D. THE SOLUTION APPROACH

As mentioned in Chapter I, we will divide our problem into a series of layers as follows, starting at the highest level of abstraction. Specifically, we first decide how the transmit cluster functionality should be moved about the sensor network in a manner that broadly spreads the energy load across the sensor network while minimizing the energy expended in moving the transmit cluster. That is, we examine how the actions of Figure 9(e) should be carried out, without worrying about the details of how the transmit cluster's beam is aligned with the UAV (in a manner that minimizes energy expenditure) as in Figure 9(d), and without worrying about how the sensor network is to successfully assemble nodes into a distributed antenna (also in a manner that minimizes energy expenditure) as in Figure 9(c). This first task, at the highest level of abstraction, is examined in this chapter.

Note, for instance, that Figure 9 does not depict any assumptions about the separation of nodes. The actual separation of real sensor nodes is very important, since the separation between nodes determines whether the nodes are able to communicate. Additionally, as we shall see, the separation of the nodes is very important for beamforming; the desired spacing of the elements (nodes) is a function of the operating frequency, and the precise operating frequency is, in turn, application-dependant.

The present discussion, though, is not intended to be specific to any one application. At this point in our discussion we are explicitly deciding not to worry about, for instance, how the sensor network is to assemble nodes into a distributed antenna. The maximum inter-node separation between nodes that can directly communicate is, at this point in our discussion, just a variable (e.g., $d$), and we are developing a framework that can be used for any value of $d$.

Returning to the discussion of our solution approach, once this preliminary problem (i.e., deciding how the transmit cluster functionality should be moved about the sensor network in a manner that broadly spreads the energy load across the sensor network while minimizing the energy expended in moving the transmit cluster) has been solved, we move down to the next layer of abstraction: How should the transmit cluster's beam be aligned with the UAV in a manner that minimizes energy expenditure? That is, we examine how the actions of Figure 9(d) should be carried out, without worrying about how the sensor network has successfully assembled nodes into a distributed antenna (in a manner that minimizes energy expenditure) as in Figure 9(c). This second problem is examined in Chapter III.

Having developed an energy-efficient algorithm to move the transmit cluster to other locations in the sensor network to evenly spread the energy load and extend the lifetime of the network, and having also developed an energy-efficient approach to aligning the transmit cluster's beam with the UAV, we then examine how the transmit cluster should assemble its antenna array in an energy-efficient fashion. This is the task we undertake in Chapter IV.

Finally, in Chapter V, we analyze our design approach in the face of randomness. We propose a clustering methodology for sensor networks, analyze the probability that a cluster will be able to form an antenna array given random placement of sensor nodes, and present an algorithm for the formation of near-linear arrays.

## E. TRANSMIT CLUSTER MOVEMENT TO EXTEND NETWORK LIFETIME

Before communication can occur, the sensor network's transmit CH must form a distributed antenna array (Figure 9c) and then align the transmit beam with the UAV (Figure 9d). We assume, for the present discussion, that these two tasks have been carried out, and our aim in this section, at the highest level of abstraction, is to examine how the transmit cluster functionality should be shifted to other portions of the network as time progresses in order to more evenly distribute the energy load across the network.

### 1. The Need to Move the Transmit Cluster

The transmit cluster is tasked with gathering all sensor data originating anywhere throughout the network. Consider the sensor network shown in Figure 10, which displays a section of a larger sensor network distributed over a terrain, where the transmit cluster consists of the selected transmit CH and nodes 1, 2, 3 and 4. The transmit CH constitutes the data sink for the network; all sensor data gathered by any node in the network are routed to the transmit CH via one of its adjacent nodes (nodes 1 - 4). While the presence of a single network sink simplifies the routing problem (since sensor nodes need only be concerned with routing to a single destination), a heavy burden is placed on nodes 1 - 4, since all the data from the network must be transmitted by one of them to the transmit CH. These transmissions expend scarce battery resources. As a result, extending the network's lifetime also requires that the transmit cluster be shifted among the clusters of the network as time progresses to balance the load.

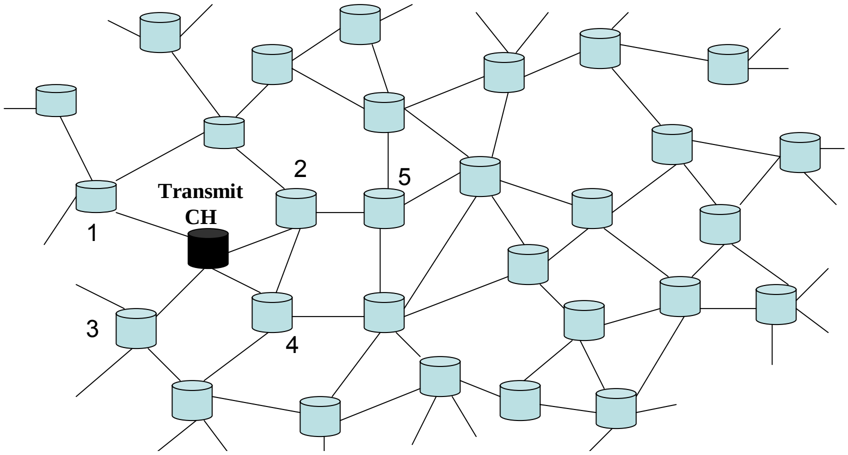


Figure 10. A Deployed Sensor Network

Since the transmit cluster is shifted about only to extend the lifetime of the sensor network, it is important that the reassignment itself be mindful of this objective.

In this chapter, we present and analyze an algorithm to reassign the transmit cluster, specifying the time that should elapse between reassignments and the number of hops[††] that should be placed between successive transmit clusters in order to achieve three competing goals: First, we wish to better and more broadly spread the energy load across the sensor network while, second, minimizing the energy expended in moving the transmit cluster, all the while, third, reducing to the extent practicable the time to bring the UAV and the sensor network's beam into alignment.

Three related questions must be considered. First, how many hops away from the previous transmit CH should the new one be? Second, how much time should elapse between transmit CH reassignments? Finally, how should this reassignment be carried out and, specifically, what protocol should be used?

---

[††] The number of hops separating a source node and a destination node is equal to the number of links (i.e., edges in the network's graph representation) between the two nodes. For example, two nodes that are within direct communication range are said to be one hop apart.

Let $h_0$ denote the number of hops separating a newly assigned transmit CH from the previous transmit CH. Our first question is thus rephrased: What is a good value for $h_0$? Referring to Figure 10, we see that $h_0 = 2$ would not be a wise selection; if, for example, node 5 was designated as the new transmit CH (assuming it is indeed a CH in a separate cluster), then node 2 would continue to be heavily taxed since it would remain adjacent to the network sink. Thus, $h_0 \geq 3$. But we also cannot make $h_0$ as large as practicable. There is a tradeoff: On the one hand, the objective of balancing the energy load across the network militates towards the selection of a large $h_0$; on the other hand, as we shall see, the energy expended in shifting the transmit CH increases as $h_0$ increases, which militates towards the selection of a smaller $h_0$.

Let $T_{hold}$ denote the time between shifts in the location of the transmit cluster. That is, once a cluster (with its associated CH) assumes the role as the sensor network's transmit cluster, it will retain these duties for a time $T_{hold}$ before the transmit cluster role is passed on to a different cluster. If $T_{hold}$ is arbitrarily large, the nodes in the transmit cluster will deplete their batteries. Thus, in the interests of extending the network's lifetime, it is necessary to place an upper limit on the value of $T_{hold}$. But, as with $h_0$, there is a tradeoff in the selection of $T_{hold}$ : A small value of $T_{hold}$ ensures the nodes in a transmit CH are not excessively depleted and works to distribute the energy load more evenly across the network, but a too-small value of $T_{hold}$ keeps the transmit cluster's upwards-pointing beam always shifting, thereby lengthening the time to beam-UAV alignment. Thus, a transmit cluster should retain its duties for both a maximum time as well as for a minimum time.

### 2. Algorithm to Transfer the Transmit CH

To formalize these ideas, consider the following simple algorithm for shifting the role of transmit cluster from one cluster to another. The algorithm assumes that each node in the sensor network has a unique ID-number, and that the network has already been partitioned into non-overlapping clusters, with each cluster managed by a single fixed clusterhead.

## ALGORITHM TO REASSIGN TRANSMIT CLUSTER

Step 1. Upon assuming the duties of transmit CH, the node starts a timer.

Step 2. When the timer reaches $T_{hold}$ , the transmit CH generates a Transmit Cluster Nominate (TCN) message, which invites another node to have its cluster assume the duties of the transmit cluster. The TCN message contains a sequence number which, together with the ID-number of the transmit CH, uniquely identifies the message. (If the node should subsequently have to transmit a new TCN, it will increment the sequence number.) The TCN message also contains a hop-counter, $h$, initiated to an integer start value, $h_0$.

Step 3. The TCN message propagates by flooding [22]; that is, the transmit CH sends the TCN message to all of its neighbors, which in turn forward the data to all of their neighbors, and so on, subject to the provision that a node will only forward a given TCN message the first time it is received. (In a wired network, a node would not forward the TCN message back to the neighbor from whom it received the data; in wireless networks, the broadcast nature of the radio channel does not permit selective transmission to a subset of neighbors.) When a node receives a TCN message, it decrements the hop-counter by one and appends its address to the message (for the purpose of enabling reverse tracing of the route followed by the message) prior to transmitting it onward. TCN messages are flooded only until $h = 1$.

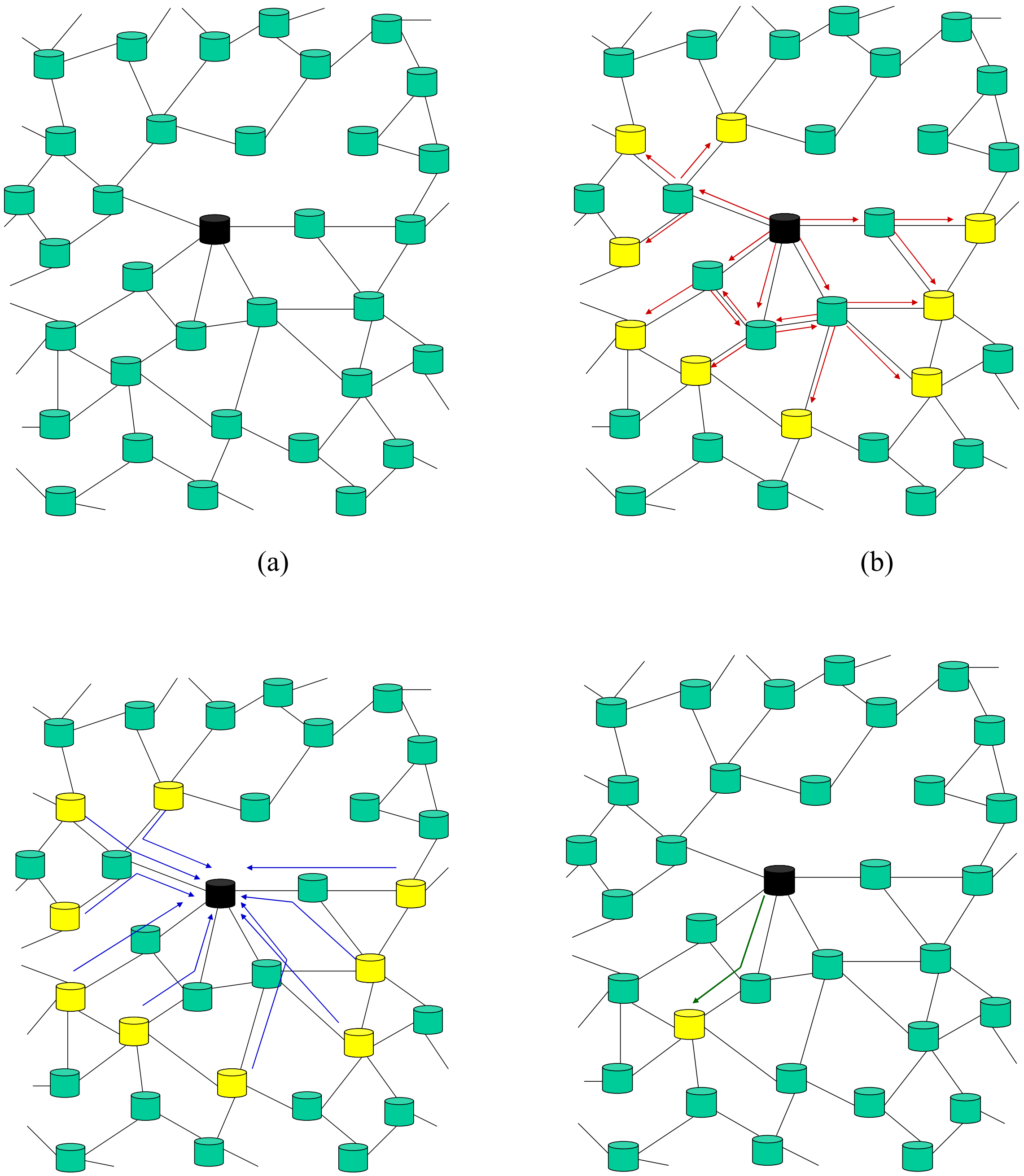
Step 4. When a node receives a given TCN message with a hop-count $h = 1$ (for the first time), it transmits a Transmit Cluster Response (TCR) message, nominating its cluster as the new transmit cluster (and its clusterhead as the new transmit CH). The TCR message propagates back to the current transmit CH along the reversely traced route contained in the received message.

Step 5. Upon receipt of the first arriving TCR message, the current transmit CH transmits a Transmit Cluster Assignment (TCA) message back to the node that initiated this TCR message. Any additional TCR messages that might be received (from other nodes that also received a copy of the TCN message with $h = 1$) are ignored. In addition to serving to transfer the transmit cluster duties to the new cluster, the TCA message contains all collected sensor data that have been aggregated up to that point. That is, the TCA message contains the historical sensor network data that the retiring transmit CH *would have* transmitted to the UAV, had the UAV been within its beam. If the destination node for the TCA message is not itself a clusterhead, then it forwards the TCA message to its clusterhead.

Step 6. Upon receiving the TCA message, the new transmit CH transmits a message announcing itself as the new transmit CH. This message, a Transmit Cluster Declaration (TCD) message, propagates by flooding to all nodes in the sensor network.

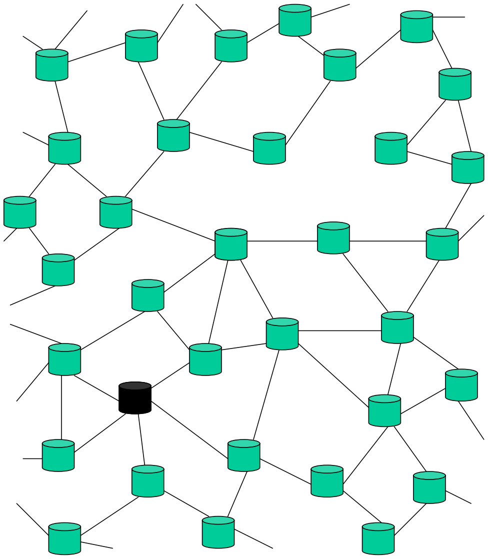
Step 7. Return to Step 1.

This algorithm is illustrated pictorially in Figure 11. In Figure 11a, the darkened node has just assumed the duties of transmit CH, and starts a timer. When the timer reaches $T_{hold}$ , the transmit cluster transmits the Transmit Cluster Nominate (TCN) message, which propagates by flooding. For this example, we assume that the hop-counter is initiated to $h = h_0 = 2$. (This assumption is made for ease of illustration; this choice of $h_0$ would not be wise.) The immediate neighbors of the transmit CH receive the TCN message, append their address, decrement $h$ (thus $h = 1$), and flood the message in turn. The TCN messages are shown in red in Figure 11b. The nodes that then receive the TCN message with $h = 1$ (shown in yellow in Figure 11b) do not further flood the TCN message. Each of these yellow nodes transmit a Transmit Cluster Response (TCR) message back to the transmit CH, as shown (in blue) in Figure 11c. The transmit CH then transmits the Transmit Cluster Assignment message to the node whose TCN message arrived first, as shown (in green) in Figure 11d. This node then assumes the duties of the Transmit CH, floods a Transmit Cluster declaration (TCD) message, and the process begins anew (Figure 11e).



(a) (b)

(c) (d)



(e)

Figure 11. Algorithm to Move the Transmit CH

### 3. Algorithm Motivation

Before analyzing the protocol, we discuss its motivation and features.

First, note that flooding is used in steps three and six of the algorithm (for TCN and TCD messages). Flooding results in a large number of unnecessary transmissions, a phenomena termed the implosion problem [23] and illustrated in Figure 12: Node A, the current transmit CH, transmits a TCN message that is received by nodes B, C and D, each of whom transmit the message to node E. Node E acts on the first reception of the TCN message, but disregards the two subsequent (duplicate) receptions. Put another way, since any location will likely be within the communication range of a number of other nodes, many broadcasts will be unnecessary (redundant). Since power is expended for each transmission, and many transmissions under flooding are unnecessary, this routing algorithm would seem to be a poor choice for a power-constrained network.

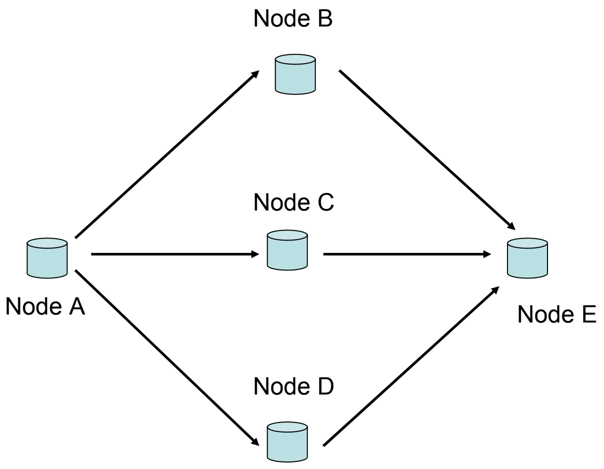


Figure 12. The Implosion Problem

But flooding results in another deficiency, termed the "broadcast storm."[24] As hosts in nearby proximity attempt to transmit the message, these transmitters contend for the channel, and collisions can be expected. Because these hosts are nearby each other, it is likely that they will sense the same phenomena, and, consequently, the timing of their onward transmissions will be highly correlated. It would seem that collisions are almost guaranteed!

Notwithstanding these inefficiencies, flooding has a number of virtues in its favor and is, in fact, employed as the routing algorithm in many sensor networks. First, flooding is an extremely simple algorithm. More efficient routing algorithms are also more complex, and each sensor node's processor must expend energy both to implement the more complex protocol machinery and to store in memory a potentially large set of computational results (e.g., tables of optimal routes). Second, more efficient point-to-point routing algorithms would require that sensor nodes exchange control messages with each other in order to learn something about the underlying network topology. In some applications, these control messages will be small compared to the actual data to be transmitted later, and so the investment sunk into transmitting and processing these control messages—which represents an inefficiency since these messages do not convey actual information that end-users are interested in—is more than made up for by the efficient transmission of large blocks of data later. But TCN and TCD messages may be quite small, consisting of tens of bytes. Certainly it would *not* be energy-efficient to transmit a large amount of control information just so that these very short blocks of data can be efficiently transferred later. Third, flooding is fault-tolerant and robust in the face of changes in the network topology since it does not depend on the network topology at all. Thus, should some nodes fail, the surviving nodes do nothing different. More efficient algorithms must employ some knowledge of the network topology, and as the topology changes, the remaining sensor nodes must consequently do *something* (e.g., exchange more control messages) to learn about the updated configuration of the network. Finally, although end-to-end delay is not a great concern for many sensor networks, it is worth noting that flooding will ensure that data are transferred as quickly as possible (since the data will arrive at the destination multiple times via all possible routes, including the fastest route).

The flooding protocol can be "tweaked" in an attempt to benefit from its core virtues while making it more efficient. For example, a hop-counter initiated to some integer start value can be added to each data packet and decremented on each hop. When a node receives a data packet with a hop-count equal to zero, the packet is discarded and not routed further. This technique will prevent packets from following excessively circuitous paths between source and destination and is also applicable to networks that do not employ flooding (e.g., a hop-counter is required by the IPv6 standard and already used de facto by IPv4 [25] ).

A variant of classic flooding that conserves power in sensor networks has recently been proposed by [23]. In this scheme, before transmitting any data, nodes negotiate with each other. Instead of exchanging sensor data as they arrive, nodes exchange *metadata*: data about the sensor data. Specifically, when a sensor node acquires new information, it transmits an advertisement that provides a brief description of the new data. A receiver can say "No thanks" by not responding (if it already has the data), or "Yes, please send it," or even "Send me only such-and-such portion of it." To be sure, storing, retrieving, comparing and managing metadata places demands on a sensor node's processor, but in return energy is never wasted on undesired transmissions. Such a scheme greatly alleviates the amount of traffic that is routed in networks that use classic flooding to propagate data.

Another method to improve performance under flooding is to selectively prevent some nodes from forwarding the message, which reduces the number of redundant transmissions while minimizing channel contention and message collision. For example, a node may decide to retransmit a received message with a probability $P < 1$. The researchers in [24] propose a counter-based scheme wherein a node may receive the same flooded message numerous times before having an opportunity to flood it. It can be shown by a simple geometric argument that when a node receives a packet, the amount of new area covered in an additional broadcast will be, on average, 61%. If a node receives the same flooded message from two nodes, the new area covered in its broadcast will be on average 19%. If a node receives the same flooded message from four nodes, the new area covered in its broadcast will be on average 5%. In general, if a node hears the same message from $k$ nodes, the additional area that will be covered by a new transmission will decrease as $k$ increases. Under the counter-based scheme, if $k$ exceeds a threshold, a node assumes that its transmission will cover only a small amount of new area, and it does not forward the message further.

Note that TCR messages (Step 4) and TCA messages (Step 5), on the other hand, use *point-to-point routing*. Each flooded TCN message records and updates the route traveled as the message progresses through the network. Thus, when a node receives a TCN message with a hop-counter equal to one, the arriving TCN message contains the route back to the transmit CH, which can then be employed as the point-to-point route used for the TCR message. This same route can then be used by the transmit CH to route the TCA message to the applicable node. Efficient routing of TCA messages is particularly important since these messages are presumably very large, containing all the network's historical sensor data aggregated up to that time.

The algorithm is designed such that the initial value of the hop-counter, $h_0$, determines how far (in hops) the newly assigned transmit CH will be from the prior transmit CH. Recall that the node seeking to relinquish its duties as transmit CH—say, node $u$—will transmit a Transmit Cluster Nominate (TCN) message with $h_0$ initiated to an integer value that is subsequently decremented on each hop as the TCN message floods through the network. A node $v$ that receives the TCN message with $h = 1$ will transmit a Transmit Cluster Response (TCR) message, nominating its cluster head as the new transmit CH. If $v$ is itself a clusterhead, and its TCR message is the first to arrive at $u$, then the new clusterhead ($v$) will be $h_0$ hops from the prior clusterhead. If $v$ is not a clusterhead, and is $j$ hops away from its clusterhead, then $v$ will be $h_0 \pm j$ hops from $u$ (i.e., it may be the case that node $v$'s CH is closer to node $u$ than is node $v$ itself). Many clustering algorithms in the literature (see, e.g., [26], [27], [28]) ensure all nodes are within one hop of their clusterheads, and our algorithm implemented on such a clustered network would ensure $h_0 \pm 1$ hops between successive transmit CHs.

Note that the algorithm to move the transmit CH does not treat three problems. First, the algorithm does not account for packet collisions in the wireless channel (the medium access problem). Second, we do not account for node/link failures that might occur as the algorithm progresses. Finally, we do not consider that sensor data might be gathered by the network during the handover. These points are left for future research.

### 4. Analysis of the Algorithm

We are interested in minimizing the amount of energy expended in moving the transmit CH. How much energy is expended by the algorithm in transferring transmit cluster duties from one cluster to another? We answer this question by first noting that communication consumes much more power than does data processing. A concrete example described in [29] illustrates this point: For a specific ground-to-ground transmission scenario, a sensor node transmitting 1,000 bits of data to a node 100 m away expended 3 joules of energy, the same amount of energy required for a modest processor to execute 300 million CPU instructions. Another example is provided in [30]: Sending 100 bits a distance of 100 m expends 10 μJ, but executing a single 32-bit instruction takes 0.06 nJ, over 100,000 times less energy. Thus, in calculating the energy cost in transferring the duties of the transmit cluster, we only consider the power expended in communicating, neglecting power expended on data processing that takes place within any node.

The algorithm has, in essence, two types of messages: small control messages (Transmit Cluster Nominate (TCN) messages in Steps 2 and 3, Transmit Cluster Response (TCR) messages in Step 4 and Transmit Cluster Declaration (TCD) messages in Step 6) and long data messages (Transmit Cluster Assignment (TCA) messages in Step 5). The length of the short control messages is denoted $m_1$ bits while the length of the long data message is denoted $m_2$ bits $\left(m_2 \gg m_1\right)$. We assume all transmissions occur at a rate of $R$ bits/sec.

We model the sensor network as an undirected graph $G = (V,E)$, where $V$ denotes the set of vertices (sensor nodes) and $E$ denotes the set of edges; an edge exists between two vertices if the corresponding sensor nodes are within direct communication range of each other. The total number of sensor nodes is denoted as $N$; that is $|V| = N$. Noting that the degree of a node is defined to be equal to the number of edges incident to it, let the maximum vertex degree in the graph be denoted as $\Delta$, and let the average degree be denoted as $\overline{\Delta}$. Sensor nodes communicate with each other at a transmit power $P$, and the maximum transmission range from one sensor node to another at this power $P$ is denoted $d$.

Since a given edge will never see the same message transmitted more than twice (once in each direction), flooding requires at most $2E$ message transfers [31]. Note that in a wireless network (such as our sensor network), a node's transmission will be broadcast on all of its links, but no node will transmit a flooded message more than once. Thus, a message flooded throughout the entire sensor network will require at most $N$ transmissions.

We calculate the energy expended in moving the transmit CH by first counting the number of occurrences of each type of message (TCN, TCR, TCA, and TCD) generated during the reassignment algorithm.

*TCN messages*

In Steps 2-3 of the algorithm, TCN messages are not flooded throughout the entire network; the presence of the counter that decrements on each hop ensures that only nodes within $h_0 - 1$ hops of the transmit CH will transmit a TCN message. Thus, for Steps 2-3, the number of message transmissions will be equal to the total number of nodes within $h_0 - 1$ hops of the transmit CH. Let the number of TCN messages generated by the algorithm be denoted as $N_{TCN}$.

It is straightforward to first calculate an upper-bound on $N_{TCN}$ by simple induction. First, if $h_0 = 1$, there is only a single transmission (from the transmit CH), and $N_{TCN} = 1$. Next, consider the case $h_0 = 2$. Now, the initial transmission from the transmit CH is received by at most Δ nodes, each of which decrements the hop-counter to one and transmits the message onward to each of its neighbors. Thus, in this case, $N_{TCN} \le 1 + \Delta$. Next, consider $h_0 = 3$. As before, we have the initial transmission, followed by at most Δ transmissions from the transmit CH's adjacent nodes. Now each of these Δ new transmissions is, in turn, received for the first time by at most $\Delta - 1$ nodes. Thus, $N_{TCN} \le 1 + \Delta + \Delta(\Delta - 1)$. Similarly, if $h_0$ is 4, we find $N_{TCN} \le 1 + \Delta + \Delta(\Delta - 1) + \Delta(\Delta - 1)^2$, and continuing in like manner, we find that for any value of $h_0$, $N_{TCN} \le 1 + \Delta \sum_{j=0}^{h_0 - 2} (\Delta - 1)^j$. Simplifying by employing the formula for the geometric sum (i.e., $\sum_{i=0}^{n} r^i = (r^{n+1} - 1)/(r - 1)$ ) and noting that the total number of transmissions cannot exceed the number of nodes, $N$, we find

$$N_{TCN} \le \min\left(1 + \left(\frac{\Delta}{\Delta - 2}\right)\left([\Delta - 1]^{h_0 - 1} - 1\right), N\right), \quad h_0 \ge 1. \qquad (1)$$

*TCR messages*

Only nodes that receive a Transmit Cluster Nominate (TCN) message with a hop-count equal to one will transmit a Transmit Cluster Response (TCR) message. Noting the development above, the maximum number of nodes that transmit a TCN message with a hop count of one is $\Delta\left(\Delta-1\right)^{h_0-2}$. Now, each of these messages is received by at most $\Delta-1$ nodes for the first time, and these are the nodes that transmit TCR messages. Each of these TCR messages propagates via a direct $h_0$-hop path back to the transmit CH. Thus, the number of transmissions required by the algorithm, $N_{TCR}$, is bounded above by

$$N_{TCR} \le \min\left(h_0\Delta\left(\Delta-1\right)^{h_0-1}, Nh_0\right) . \tag{2}$$

*TCA messages*

Since the TCA message generated by the transmit CH propagates directly to the first node whose TCR message was received, and that node, if not a clusterhead, forwards the TCA message to its clusterhead located at most one hop away, the number of TCA transmissions, $N_{TCA}$, is bounded above by

$$N_{TCA} \le h_0 + 1 . \tag{3}$$

*TCD messages*

The Transmit Cluster Declaration message propagates by flooding. Thus, since no node will transmit a flooded message more than once, the number of TCD messages, $N_{TCD}$, is bounded above by

$$N_{TCD} \le N . \tag{4}$$

In light of the foregoing, we now provide an upper-bound on the amount of energy expended in transferring a transmit CH.

**Theorem.** The amount of energy expended in executing the algorithm to transfer the transmit CH from one node to another is bounded above by:

$$\begin{aligned} E_{transfer} \le\ & P\left[\min\left(1+\left(\frac{\Delta}{\Delta-2}\right)\left(\left[\Delta-1\right]^{h_0-1}-1\right), N\right)\right]\left(\frac{m_1}{R}\right) \\ & + P\left[\min\left(h_0\Delta\left(\Delta-1\right)^{h_0-1}, Nh_0\right)\right]\left(\frac{m_1}{R}\right) + \frac{PNm_1}{R} + \frac{P\left(h_0+1\right)m_2}{R}. \end{aligned} \tag{5}$$

**Proof.** An upper-bound on the total number of TCN, TCR and TCD messages generated by the algorithm is given by Equations (1), (2) and (4), respectively. Each of these messages, being control messages, takes time $m_1/R$ seconds to transmit. Additionally, by (3), at most $h_0 + 1$ TCA messages are transmitted during the course of the algorithm, each taking time $m_2/R$ seconds. Each transmission occurs at power $P$ watts, and the energy expended by a transmission equals the transmit power multiplied by the duration of the transmission.

Note that the upper-bound (5) depends critically on the value of $h_0$, the number of hops to transfer the CH, as well as the maximum node degree $\Delta$. The computed upper-bound is $O\left(\Delta^{h_0}\right)$. As an example, Figure 13 displays the upper-bound on the total energy expended in implementing the algorithm to shift the transmit CH as a function of $h_0$, assuming that the sensor network consists of 10,000 motes, communicating with each other at a minimal RF transceiver power of -24 dBm (4μW). At low values of $h_0$, the last two terms in (5) dominate, i.e., (5) is

$$E_{transfer} \approx \left(PNm_1 + Ph_0m_2 + Pm_2\right)/R .$$

As $h_0$ increases, the first two terms of (5), growing exponentially, come to dominate.

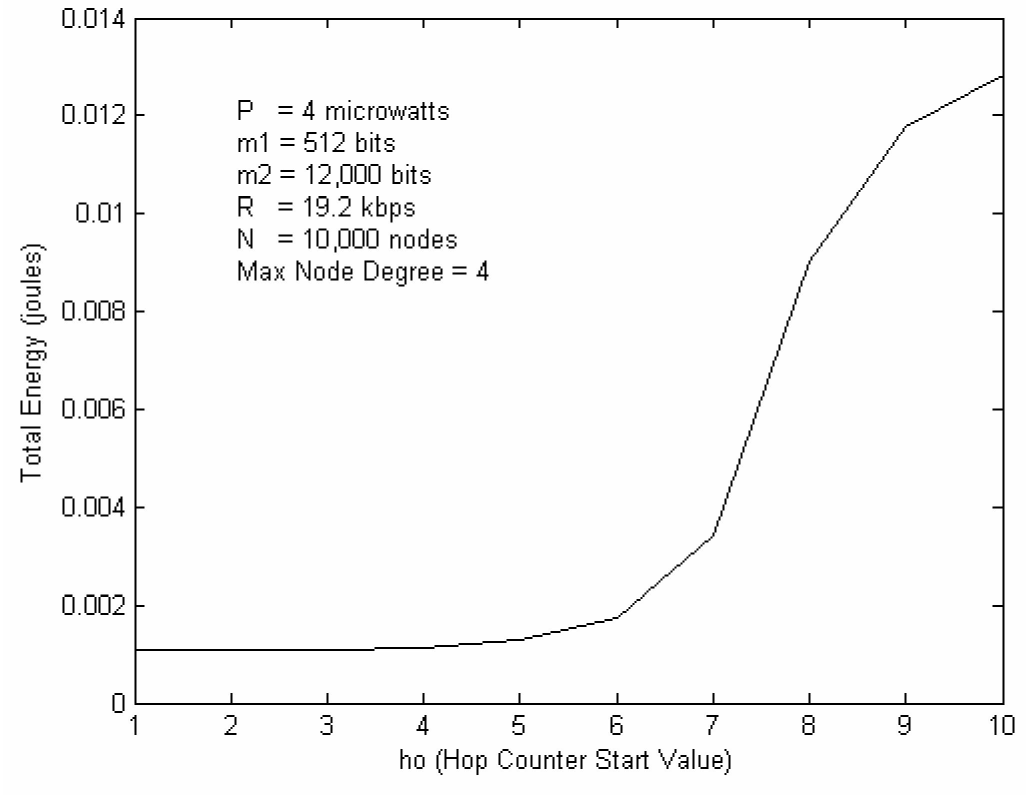


Figure 13. Upper Bound of Energy Expended to Shift the Transmit CH

With the exception of the maximum node degree Δ, all of the parameters in (5) are presumed known *a priori*. For example, the transmission rate ($R$) and transmit power ($P$) are determined by the transceiver design, the number of deployed nodes ($N$) is known, and the average packet sizes ($m_1$ and $m_2$) as well as the hop-counter start value ($h_0$) are preprogrammed. Since the maximum node degree is not known and can not be readily determined and an energy calculation based on the maximum node degree may not be very illuminating (e.g., a network may have a single node of maximum degree, say $\Delta = 10$, with the next highest-degree node being of degree 5), we replace Δ in (5) with $\overline{\Delta}$, the average node degree. Of course, this modification means that (5) will no longer be an upper-bound (and will neither provide an average of $E_{transfer}$ since the expression in (5) above is not linear in $\Delta$), but will instead serve as a more useful estimate of the energy expended in transferring the transmit CH from one location to another.

Denoting the degree of node $v \in V$ as deg($v$), the average node degree is $\overline{\Delta} = \frac{\sum_{v \in V} \deg(v)}{N} = \frac{2|E|}{N}$ where the latter equality follows from the Handshaking Lemma [32]:

$$\sum_{v \in V} \deg(v) = 2|E| \ .$$

However $\overline{\Delta}$ cannot be calculated from this formula since we do not know the degrees of all the nodes in the network (nor the number of edges). We instead determine a simple estimate of $\overline{\Delta}$ using the geometry of the problem.

The transmission power of a node, $P$, will correspond to a maximum reception range $d$. That is, a node transmitting at power $P$ will be within communication range of all nodes within a distance $d$ away. We assume that the sensor network is known to be confined to a specific finite planar region—termed the *sensor region*—of size $A$. Consider $N$ nodes distributed uniformly over this region. Focusing on a single node within the interior of the region, all nodes within a circle of radius $d$ will be within communication range of the node (see Figure (14)). The average number of nodes within this region is $N\left(\frac{\pi d^2}{A}\right)$. Since this quantity includes the node under consideration, we estimate the average node degree as

$$\overline{\Delta} = N\left(\frac{\pi d^2}{A}\right) - 1 \ . \tag{6}$$

We note that this estimate of $\overline{\Delta}$ is still an over-estimate (i.e., an upper-bound on the actual average node degree) since nodes near the boundaries of the region will have fewer neighbors, as a portion of their reception zone will lie outside the sensor region. As a simple numerical example, consider again the sensor network of 10,000 motes uniformly distributed over a 10 km × 10 km region. Using a transmission range of $d$ = 100 m, we find from (6) that $\overline{\Delta} = 2.1416$. Figure 15 displays an estimate of the total energy expended in implementing the algorithm to shift the transmit CH as a function of the start value of the hop counter, $h_0$, assuming the same sensor network parameters from

Figure 13 but using $\overline{\Delta}$ instead of Δ. We see in Figure 13 that the energy expended in transferring the transmit CH first exceeds 2 mJ when $h_0 \approx 6$. Comparing this to Figure 15, we note that the energy expended in transferring the transmit CH first exceeds 2mJ when $h_0 \approx 35$.

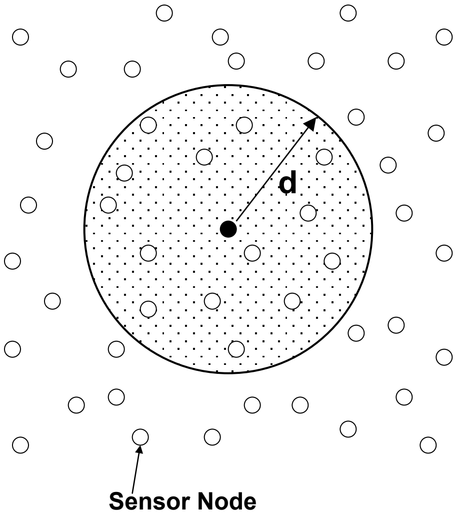


Figure 14. Estimate of $\overline{\Delta}$

Based on (5) (and its behavior as depicted in Figures 13 and 15), the value of $h_0$ chosen for use in the algorithm to reassign the transmit cluster should not exceed the point where the exponentially growing terms (the first two terms in (5)) exceed the sum of the constant term (the third term in (5)) and the linearly increasing term (the last term in (5)). Thus, in order to more evenly distribute the energy load across the network while minimizing the energy expended in moving the transmit cluster, we should select $h_0$ to be the maximum value of $h$ such that the following inequality is satisfied:

$$(\Delta-1)^{h-1}\left\{\frac{\Delta}{\Delta-2}+h\Delta\right\}-\frac{m_2}{m_1}h \;\leq\; N+\frac{m_2}{m_1}+\frac{\Delta}{\Delta-2}-1 \;. \qquad (7)$$

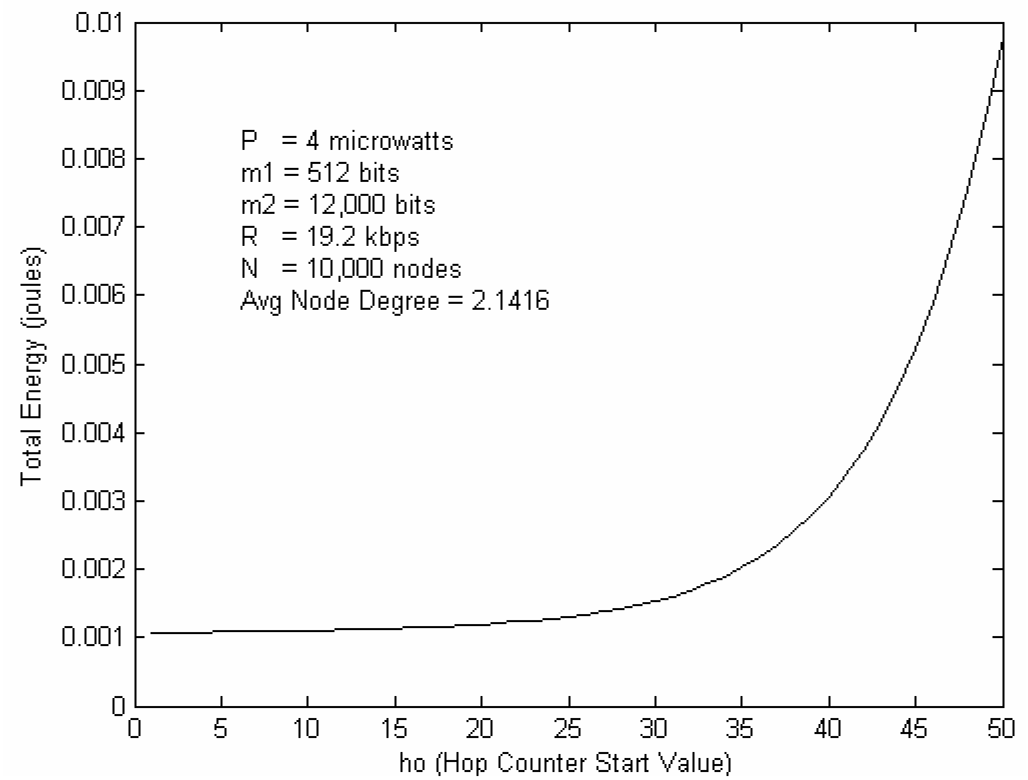


Figure 15. Heuristic Estimate of Energy Expended to Shift the Transmit CH

For the scenario depicted in Figure 13, (7) yields $h_0 = 6$, while for the scenario depicted in Figure 15, (7) yields $h_0 = 36$. Thus, we see that by considering the average node degree, we can move each transmit cluster head a considerable distance (i.e., 36 hops in this example) from the previous transmit cluster head, thereby more broadly and uniformly spreading the energy burden across the sensor network.

In this chapter, we presented a method for more uniformly distributing the energy burden across a wireless ground-based sensor network communicating with an overhead unmanned aerial vehicle (UAV). A subset of sensor nodes, termed a *transmit cluster*, receives and aggregates data gathered by the entire network and forms a distributed antenna array, concentrating the radiated transmission into a narrow beam aimed towards the UAV. Because these duties are power-intensive, the role of transmit cluster must be shifted to different nodes as time progresses. We presented an algorithm to reassign the transmit cluster, specifying the time that should elapse between reassignments and the number of hops that should be placed between successive transmit clusters in order to achieve three competing goals: First, we wish to better and more broadly spread the energy load across the sensor network while, second, minimizing the energy expended in moving the transmit cluster, all the while, third, reducing to the extent practicable the time to bring the UAV and the sensor network's beam into alignment. The algorithm thus extends the lifetime of the sensor network while meeting system-level performance objectives. In the next chapter, we describe in detail an energy-conserving technique to align the transmit CH's upward-pointing beam and present a series of examples illustrating the utility of the cluster shifting algorithm presented in this chapter.

THIS PAGE INTENTIONALLY LEFT BLANK

# III. AN ENERGY EFFICIENT APPROACH FOR INFORMATION TRANSFER FROM TACTICAL WIRELESS SENSOR NETWORKS[‡]

Recall that following deployment, the sensor nodes partition themselves into clusters, each managed by a single node designated as the clusterhead (CH). A specified cluster, termed the *transmit cluster*, is selected to act as the transmission point for the sensor network. In the previous chapter, we presented an algorithm to reassign the transmit cluster, specifying the time that should elapse between reassignments and the number of hops that should be placed between successive transmit clusters in order to more uniformly distribute the energy burden across a ground-based sensor network communicating with an overhead UAV. In this chapter, we describe in detail an energy-conserving technique to align the transmit CH's upward-pointing beam with the UAV.

## A. SOLUTION APPROACH FOR ESTABLISHING A COMMUNICATION LINK WITH A UAV

Any sensor data originating anywhere in the network is routed to the transmit cluster's CH (the *transmit CH*), where it is consolidated with any data already collected and prepared for transmission to the UAV. The transmit CH organizes a subset of its cluster nodes into a distributed antenna array, coordinating their transmissions to direct the beam towards the UAV. Specifically, the transmit CH calculates the magnitude and phase offsets to be applied to the otherwise identical transmissions of the participating nodes in order to form a radiation pattern that concentrates the radiated power into a narrow beam. The key condition for this approach to work is that the sensor network's narrow transmission beam must be directed such that the UAV falls spatially within it, as shown in Figure 9d.

---

[‡] The results in this chapter were previously presented at the IEEE International Conference on System of Systems Engineering (SoSE 2006) [33].

Now, as mentioned in Chapter II, if the direction of approach of the UAV (i.e., its elevation and azimuth angles) is known *a priori*, then a number of techniques are available to determine the magnitudes and phases necessary to synthesize the desired radiation pattern to point the beam towards the UAV [20]. Military applications, however, must presume that the sensor network does not know *a priori* where the UAV is, nor does the UAV know the direction in which the sensor network has aimed its transmission beam.

So, how can the sensor network "find" the UAV? As also mentioned in Chapter II, an antenna array such as that formed by the transmit CH can, in theory, employ a variety of techniques to estimate the direction of arrival of the UAV [21]. All of these methods, though, are very complex. In one of the simplest methods, for example, a UAV would fly over the sensor network transmitting a known reference signal. The sensor network's antenna array would steer a very narrow beam continually in space, analyzing the spatial spectrum as a function of position and then selecting the direction that yields the highest power as the direction of the UAV (i.e., assuming that this power arises from the UAV's transmission). For this method to be effective, the transmit CH will have to continually calculate different sets of amplitudes and phases for the participating sensor nodes comprising the antenna array (in order to steer the beam) and transmit these values to the participating nodes; these sensor nodes, in turn, will have to continually send their transceiver receptions to the transmit clusterhead for analysis of the spatial spectrum. When energy constraints require that the role of transmit cluster be shifted to a different cluster as described in Chapter II, all relevant data collected up to this point must be passed through the network to the new transmit CH, or the entire process of finding the UAV must begin anew. As all these operations involve a large number of radio transmissions among sensor nodes and a large number of computations performed by the transmit CH(s), they expend scarce battery resources, thereby limiting the lifetime of the involved nodes.

Still, regardless of the difficulties, it remains a fact that before communication can occur, the sensor network's transmit CH must align the transmit beam with the UAV.

Our approach to the problem begins by recognizing that the sensor nodes are very power-constrained, having limited transmission and computational capacity. The UAV, on the other hand, is highly capable. Not as limited in size and weight, it carries a much larger battery, and, unlike sensor nodes, it will fly back to its base and have its battery recharged or replaced once its mission is complete. Nevertheless, despite the strength of the UAV and the limitations of the sensor nodes, existing algorithms that determine the proper steering of the beam place all of the communication and computational burdens on the sensor network. The very capable UAV does nothing but fly around, waiting to be "found."

Our approach is to reverse these efforts: The sensor network will invoke a simple algorithm that requires minimal communication and only a single amplitude/phase offset calculation. Specifically, the transmit CH will direct a fixed beam, pointing straight up, at 0° elevation. The burden will then be placed on the more capable UAV to fly over the sensor network and find this beam.

The basic approach is illustrated in Figure 16. The transmit CH has organized a subset of its nodes into a distributed antenna. The transmit CH calculates the proper amplitude and phase offsets needed for the resultant beam of the desired gain to point straight up, and these values are passed to each participating node. No transmission to the UAV is attempted at this time since there is no reason to believe that the UAV will be overhead within the beam; the amplitude and phase values, calculated only once, are retained by the array elements in case a transmission should be required later. In fact, the distributed antenna "stands down," and the transmit CH now functions as an omnidirectional antenna again, waiting for the UAV to find it. (In the meantime, the sensor network continues to pass sensed data to the transmit cluster, for eventual transmission to the UAV.)

The UAV, meanwhile, travels over the sensor region as shown in Figure 16a, continuously transmitting a known reference signal, pointing its transmission beam straight down. If the transmit CH should fall within the UAV's beam, the clusterhead will send the data stream (containing the consolidated data gathered by the sensor network) to the elements in the antenna array, and these elements will, in turn, transmit the signal using the precomputed amplitudes and phase offsets, as shown in Figure 16b. If the distributed antenna array's gain is close to the UAV's antenna gain (and employs a similar conically shaped beam), we can be assured of the scheme's success since whenever the transmit CH falls within the UAV's transmission beam, the UAV will fall within the sensor network's transmission beam.

In short, instead of having the sensor network's beam search for the UAV, we simply propose having the UAV search for the sensor network's beam. Thus, the power-constrained sensor nodes are not burdened with finding the highly capable UAV; instead, the burden of aligning the transmission beam is shifted to the UAV.

Since the role of transmit cluster must be shifted to different locations as time progresses, it can be expected that the transmit CH will move during the course of the UAV's flight (see Figure 17). This movement of the transmit CH, discussed in the previous chapter, will be revisited in the present context after defining the system model and two rival search plans.

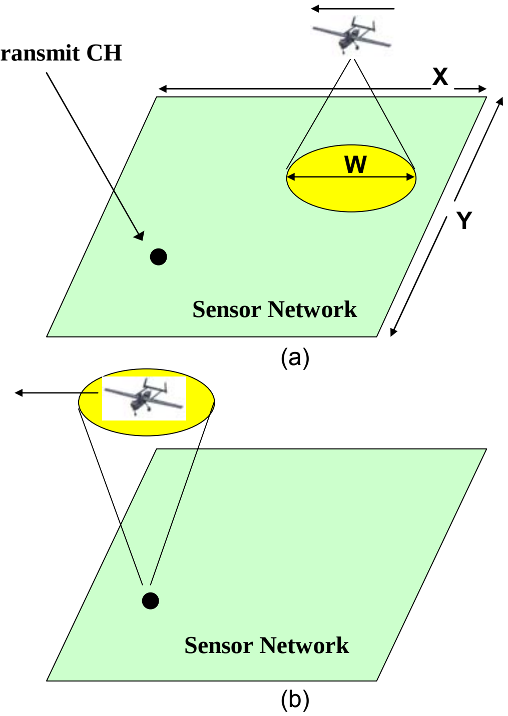


Figure 16. Solution Approach to Beam-UAV Alignment

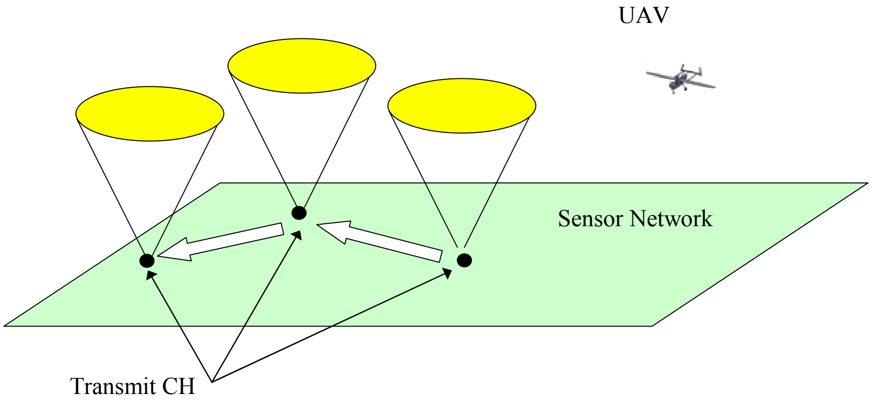


Figure 17. Transmit Cluster Movement

## B. SYSTEM MODEL AND TWO RIVAL SEARCH PLANS

We address the question: How should a UAV be employed to scan an area, seeking to place the transmit CH within its downward pointing beam? We may consider the UAV to be searching for, and trying to detect, the transmit CH. Three criteria guide our decision:

1. The cumulative probability of detection (as a function of time), $P_D(t)$, should be maximized.
2. The expected time to detect the transmit CH should be minimized.
3. The plan should be as simple as possible.

A considerable body of research (see, for example, [34],[35]) exists for determining how search effort should be allocated over a region to maximize the probability of detecting a target. Unfortunately, the known solutions offer results that are theoretical as opposed to practical since they do not yield flight trajectories that an aircraft can physically follow. For a survey of major results in search theory, see [36].

The theory of search plan design also shares elements in common with the field of robot motion planning and, in particular, with the subfields of terrain acquisition and coverage path planning. Here, the aim is to design algorithms that enable a sensor-equipped robot to generate an optimal path through an unexplored terrain populated with unknown (but fixed) obstacles such that the sensor "passes over" every point in the terrain. Unfortunately, the exhaustive search algorithms useful for robot motion planning offer limited utility for our search operations for two reasons. First, these algorithms all assume the target of the search (in our case, the transmit CH) remains stationary during the execution of the algorithm. The aim of these algorithms is to find an efficient path that ensures all regions are scanned at least once; a region that has been scanned is never intentionally revisited unless necessary to reach an as-yet-unexamined area. If the target of the search is mobile—a key assumption in our model where the beam must be allowed to be shifted to another transmitting cluster to extend the lifetime of the network—these algorithms fail since the target may move undetected into an already-scanned region, and never be detected. Second, many of these algorithms assume that the boundaries of the target area are unknown, and much of the emphasis is on employing robot sensors to explore and discover the borders of the region and the contours of the obstacles within. Clearly, this is not of concern in our searches where the boundaries are known because we deployed the sensors. Examples of terrain-covering algorithms that assume stationary targets and assume the region boundaries are unknown at the outset are presented in [37] and [38]. Exhaustive search algorithms that assume the search region is known a priori, but still do not adapt to moving targets are presented in [39] and [40]. For a survey of recent results in robot motion planning, see [41].

To conduct the search, we have a UAV that flies at a minimum velocity of $V$. The UAV transmits a known reference signal, using a conical beam pointed downward. Referring to Figure 16a, we assume that any sensor node within a radial distance of $W/2$ from the point on the ground directly below the UAV will detect the UAV's transmission, but will never detect the UAV's transmission beyond this range. The quantity $W$, termed the sweep width, is fixed for a given UAV altitude and antenna gain. The UAV is assumed to have the ability to ascertain the coordinates of its location at any time and, additionally, is aware of the geographic boundaries of the search region.

The sensor network is known to be confined to a region on the ground, termed the sensor region. Specifically, let us assume the sensor network's location is confined to a rectangular planar region of width $X$, length $Y$ and area $A = XY$. Without loss of generality, we assign the label $X$ to the shorter side (i.e., $X \leq Y$) and note that $X$ is almost always much larger than the sweep width, $W$. While we are certain that the transmit cluster (and its upward-pointing beam) resides within a region of known boundaries, we possess no *a priori* knowledge about the transmit CH's precise location within the region; as such we assume that the transmit CH's location is uniformly distributed over the region.

In summary, we are interested in examining search strategies that will ensure that the UAV finds the transmit CH as quickly as possible (i.e., brings the UAV and the sensor network's beam into alignment as quickly as possible), maximizing the probability of detection (as a function of time), using as simple a means as possible.

In this chapter, we describe and analyze two alternative strategies that bring the UAV and the sensor network's beam into alignment. Both schemes seek to minimize the energy expended by the sensor network in aligning the beam and the UAV. One strategy employs a very capable UAV and ensures, with probability one, that the beam is aligned with the UAV, also guaranteeing that it will be done as quickly as possible, given that alignment is to occur. The second scheme uses a less capable UAV, and, while not absolutely guaranteeing alignment, will find the beam more quickly in a variety of scenarios. The performance of the two strategies is compared in terms of probability of beam-UAV alignment as a function of time, and the expected time to alignment. We examine the performance tradeoff between the choice of strategy and parameters of the sensor network that affect power conservation.

Our first scheme uses a very capable UAV. In this scheme, the UAV flies a predefined trajectory, or pattern. Use of predefined patterns can ensure that the search is both effective (no interesting areas are missed) and efficient (a UAV does not unnecessarily duplicate its efforts). Consider, for example, that if the transmit CH is stationary, a UAV can perform parallel sweeps of the sensor region (where the sweeps are separated by $W$, the sweep width) and ensure detection in finite time. On the other hand, flying a pattern may prove inflexible in highly dynamic environments, and the pattern—being predictable—might make the UAV more susceptible to shoot-down. We term our second search plan the *progressive search* for reasons that will become clear shortly.

Our second search plan is a simple scheme in which the UAV searches randomly. This scheme employs an inexpensive UAV that has limited processing capability and has bare-bones navigational equipment, serving only to keep it within the boundaries of the sensor region. The UAV in this scheme follows a meandering track, searching randomly within the sensor region. This search is hereafter referred to as *random search*.

Our aim is to analyze these two rival UAV search schemes, quantitatively comparing them using several performance metrics. The progressive search, if properly designed, will not scan already-cleared areas and will not miss any areas. A random search, though, is cheaper, by virtue of its less capable UAV that does not have to implement the necessary search plan algorithm, with the attendant costs in calculation, navigation, etc. In the remainder of this section, we describe in detail the progressive search, and in the next sections, we compare the two rival schemes considering the probability of detection as a function of time and the expected time to find the transmit CH.

In the progressive search, the UAV follows equally spaced, straight back-and-forth parallel tracks. The movement of the UAV from one end of the sensor region to the other along the *x*-axis constitutes a sweep. Figure 18 shows two consecutive sweeps (with the second sweep still in progress). Note that the UAV must move all the way to the edge of the sensor region before turning 90°, so as to avoid any gaps in coverage. It is possible that the sensor network will move the transmit CH during the course of the UAV's search (so as to evenly distribute the energy load). For example, consider the case where the transmit CH is initially at Point "A" in Figure 18, just outside the UAV's beam as it flies past. Suppose that once the UAV has passed, the transmit CH is moved to Point "B." If consecutive sweeps did not overlap, the UAV would complete the search and never find the transmit CH. To avoid this scenario, a given sweep must have some overlap with the prior sweep, while making progress toward the eventual goal. The overlap between consecutive sweeps is designated $S$ (see Figure 18).

Recalling from Chapter II that $T_{hold}$ denotes the time between shifts in the location of the transmit CH, the new transmit CH will be $h_0 \pm 1$ hops from the prior transmit CH and the maximum reception range between two sensor nodes is denoted as *d*, then, from the UAV's perspective, the transmit CH moves around the network as time progresses with a maximum velocity *v*, given by

$$v = \frac{\left(h_0 + 1\right) d}{T_{hold}} \ . \tag{8}$$

Although the transmit CH's maximum velocity is known, the transmit CH's *actual* speed and direction of movement at any given time are unknown.

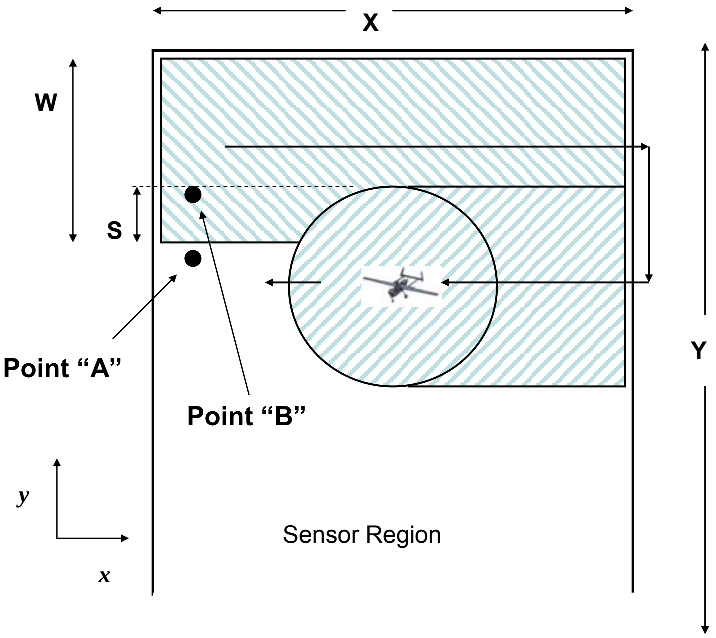


Figure 18. Progressive Search

To guarantee detection of the transmit CH, we presume the worst-case scenario shown in Figure 18: The transmit CH is at the leftmost boundary just beyond the UAV's beam (Point "A") and moves straight up at speed *v* as soon as the UAV has passed. The time for the UAV to go from the left edge back to the left edge is $\frac{2X + W - S}{V}$, and during this time the transmit CH can move a maximum distance of $\left(\frac{2X + W - S}{V}\right)v$ into the previously swept region. To assure detection, we require that this maximum distance be set equal to the overlap. Thus $S = \left( \frac{2X + W - S}{V} \right) v$. Solving for *S*, we determine the minimum overlap between sweeps necessary to assure detection of the transmit CH:

$$S = \frac{(2X + W)\, v}{V + v} \quad . \tag{9}$$

Note that when $v = 0$ (the transmit CH does not move during the entire search), then $S = 0$ (no overlap needed), as expected. If the UAV's downward movement between sweeps is less than $W - S$, the search will take longer than necessary, whereas if the downward movement between sweeps exceeds $W - S$, it is possible that the transmit CH might not be detected. For the search to be completed in finite time, successive sweeps must advance; the overlap between successive sweeps, *S*, must be less than the sweep width, *W*. Hence, we term such a search a *progressive search*.

A progressive search requires that $S < W$. From (9), this requires $v < (WV/2X)$. Of course, if $Y \le W$, then only one sweep is required. Thus, the smallest transmit CH velocity that prevents a progressive search is given by

$$v = \begin{cases} WV/2X & Y > W \\ \infty & Y \le W \end{cases} \quad . \tag{10}$$

Stated another way, alignment is always (eventually) assured with the progressive search provided $v < WV/2X$ (if $Y > W$).

Now, every progressive search consists of *M* sweeps, where *M* is given by

$$M = \max\left( \left\lceil \frac{Y - S}{W - S} \right\rceil , 1 \right) \quad . \tag{11}$$

To show this, first assume that $Y > W$ so that the area cannot be examined in a single sweep. Noting the *x* and *y* axes shown in Figure 18, we see that during each sweep the UAV travels a distance *X* along the *x*-axis (the full width of the sensor region). With the exception of the final sweep, each sweep progresses a distance of $W - S$ along the *y*-axis. During the last sweep, part of the UAV's beam may fall beyond the sensor region's boundary. If the last sweep (which completes the search) covers a distance of $Y_{last}$ along the *y*-axis, then, since the total breadth in the *y*-axis direction is *Y*, we have

$M = 1 + \frac{Y - Y_{last}}{W - S}$. Now, since $S < Y_{last} \le W$, $Y_{last}$ must differ from $W$ by an amount less than $(W - S)$. Thus, $M = 1 + \left\lceil \frac{Y - W}{W - S} \right\rceil = \left\lceil \frac{Y - S}{W - S} \right\rceil$. Finally, we remove our initial assumption that $Y > W$ by incorporating the max function (if $Y \le W$, then $M = 1$).

The time between when the search begins (i.e., when the outer periphery of the UAV's beam first enters the sensor region) to the time when the UAV itself is first over the sensor region is $\frac{W/2}{V}$. Each sweep except the final sweep takes time $\frac{X}{V} + \frac{W - S}{V}$. The final sweep takes time $\frac{X}{V}$. Thus, provided the transmit CH velocity does not exceed the conditions of (10), the total time to complete the progressive search, $t$, is given as: $t = \frac{W}{2V} + (M - 1)\left(\frac{X}{V} + \frac{W - S}{V}\right) + \frac{X}{V}$. Simplifying this expression, the time to complete the progressive search is

$$t = M\left(\frac{X}{V} + \frac{W - S}{V}\right) + \frac{2S - W}{2V} \; . \qquad (12)$$

Figure 19 shows the time to complete the progressive search for a sensor network located in a 10 km × 10 km region, as a function of the maximum transmit CH speed, $v$. The UAV moves at 120 km per hour and has a sweep width of 0.5 km.

## C. PROBABILITY OF BEAM-UAV ALIGNMENT AS A FUNCTION OF TIME

The progressive search can assure transmit CH detection ($P_D = 1$) through the careful selection of the amount of overlap between successive sweeps. The performance tradeoff between the search time $t$ and the transmit CH velocity has been previously discussed.

The probability of detection ($P_D$) as a function of time $t$ for a random search against a randomly moving transmit CH whose initial position is uniformly distributed within a region of size $A$ km$^2$ was first derived by Koopman in [42] and is given by

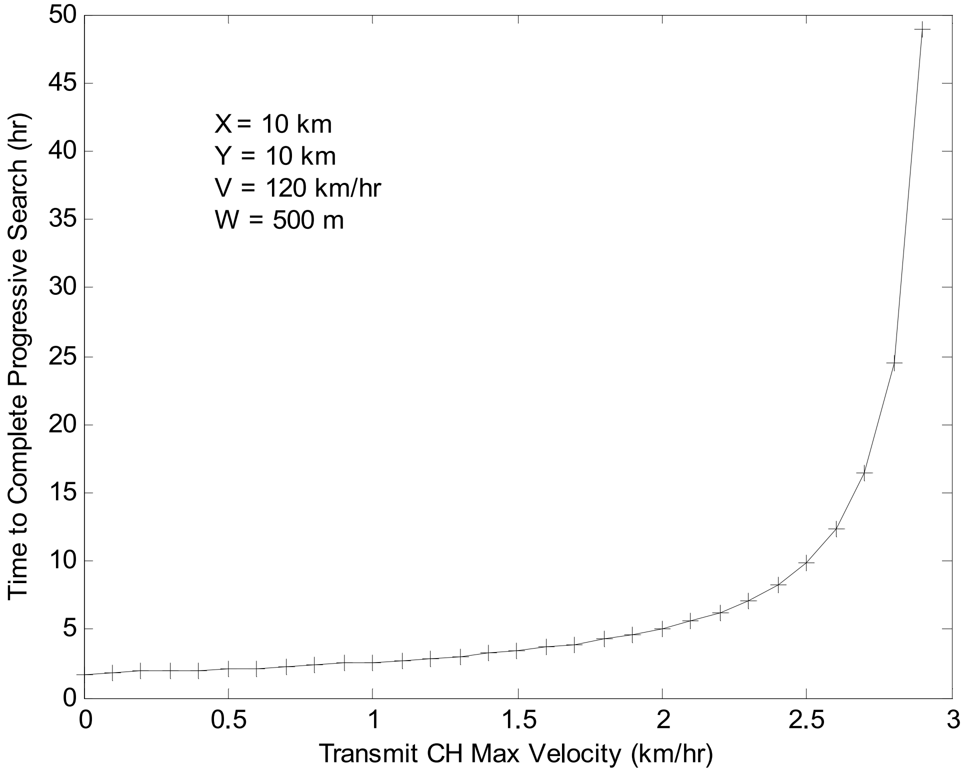


Figure 19. Time to Complete a Progressive Search

$$P_D(t) = 1 - e^{\frac{-WVt}{A}}, \tag{13}$$

where, as before, $W$ is the sweep width and $V$ is the UAV's velocity.

We note from (13) that detection of the transmit CH is never assured when employing a random search, whereas detection is always (eventually) assured with the progressive search (provided the conditions in (10) are satisfied). Thus, the progressive search is clearly desirable in scenarios where it is critical to find the transmit CH.

## D. EXPECTED TIME TO FIND THE BEAM

The progressive search is designed to guarantee detection. Is this more expensive scheme so over-designed to account for the worst-case scenario that it performs poorly under more typical scenarios in which the less costly random search would perform just as well (or better)?

To answer this question, we will compare the expected time to find the transmit CH employing a progressive search to that taken for a random search assuming the transmit CH's initial location is uniformly distributed within the sensor region. Consider that in the progressive search the UAV starts at one end of the sensor region and gradually and methodically progresses forward, with the far end of the sensor region being examined for the first time only at the very end of the search. By contrast, the UAV performing a random search might happen to quickly fly to the position of the transmit CH. Which search will locate a beam more rapidly, on average?

**Theorem**. Suppose a UAV is engaged in a progressive search for a uniformly distributed transmit CH. Let $T_P$ denote the time that the transmit CH is first found by the UAV. An upper bound on the expected value of $T_P$ is given by

$$\mathrm{E}\left[T_P\right]_{UB} = \frac{W}{2V} + \frac{X}{2V} + \left(\frac{X}{V} + \frac{W-S}{V}\right)(M-1)\left\{1 - \frac{M}{2}\left(\frac{W-S}{Y}\right)\right\} , \qquad (14)$$

where $M$, as before, represents the number of sweeps, and is given by (11) as $M = \max\left(\left\lceil \frac{Y-S}{W-S} \right\rceil, 1\right)$, and $S$ is the minimum overlap between sweeps needed to guarantee detection, given by (9) as $S = \frac{(2X+W)\,v}{V+v}$.

**Proof**. We assume that the transmit CH's initial position is uniformly distributed over the sensor region and moves in random directions and with random speeds, up to a maximum speed of $v$. The UAV performs a progressive search of the sensor region, where the overlap between successive sweeps is equal to $S$. The probability of detection as a function of time, $P_D(t)$, will improve as the search progresses and will be equal to unity when the search concludes (at the end of the final sweep). We ask, informally: *What is the worst we can do?* and, more formally: What is a lower bound on $P_D(t)$?

Consider the first sweep. At the end of this sweep, a transmit CH within a region of size $X(W-S)$ will have been detected, and no undetected transmit CH's will be able to enter this region in the future. We can say that the first sweep *sweeps clean* a region of size $X(W-S)$ in the sense that no further effort needs to be explicitly spent in examining this region. Thus, assuming that the transmit CH's position is uniformly distributed within the region, the probability that the CH has been detected at the end of the first sweep is at least equal to $X(W-S)/XY=(W-S)/Y$. All subsequent sweeps except for the final sweep yield the same result: An additional region of size $X(W-S)$ is swept clean, and, at the conclusion of the sweep, the probability of detection has increased by at least $(W-S)/Y$ over the prior sweep. The final sweep differs from the prior sweeps in that some of the UAV's beam may lie outside the sensor region, and the probability of detection will improve by *at most* $X(W-S)$ over the penultimate sweep, rising to one at the conclusion of the last sweep.

To further ensure that our estimate of $P_D(t)$ is indeed a lower bound, we assume that new area covered as the UAV repositions between sweeps does not contribute to the probability of detection; i.e., the new area covered as the UAV repositions is taken into consideration during the analysis of the subsequent sweep. Likewise, we assume that the area covered at the very start of the search, as the perimeter of the UAV's circular beam first enters the sensor region at time $t=0$, until the UAV itself is over the sensor region at $t=W/2V$, does not contribute to the probability of detection. This area is taken into consideration during the analysis of the first sweep.

With these considerations in mind, Figure 20 displays a lower bound for the probability of detection, $P_D$, versus time, $t$, for the progressive search. The initial sweep, which takes time $X/V$, sweeps clean a region of size $X(W-S)$; at the conclusion of the first sweep, the probability that the transmit CH has been detected is $(W-S)/Y$. The UAV then repositions for the second sweep, taking time $(W-S)/V$, and, as mentioned, we assume that repositioning does not contribute to the probability of detection. The UAV then performs the second sweep, again taking time $X/V$, and sweeping clean an additional area equal to $X(W-S)$; the probability of detection at the conclusion of the second sweep is thus $2(W-S)/Y$. The UAV then repositions and sweeps again, in a manner exactly analogous to the second sweep. The UAV performs a total of $M-1$ sweeps prior to the final (terminating) sweep. The probability of detection rises from $(M-1)(W-S)/Y$ to unity during this final sweep.

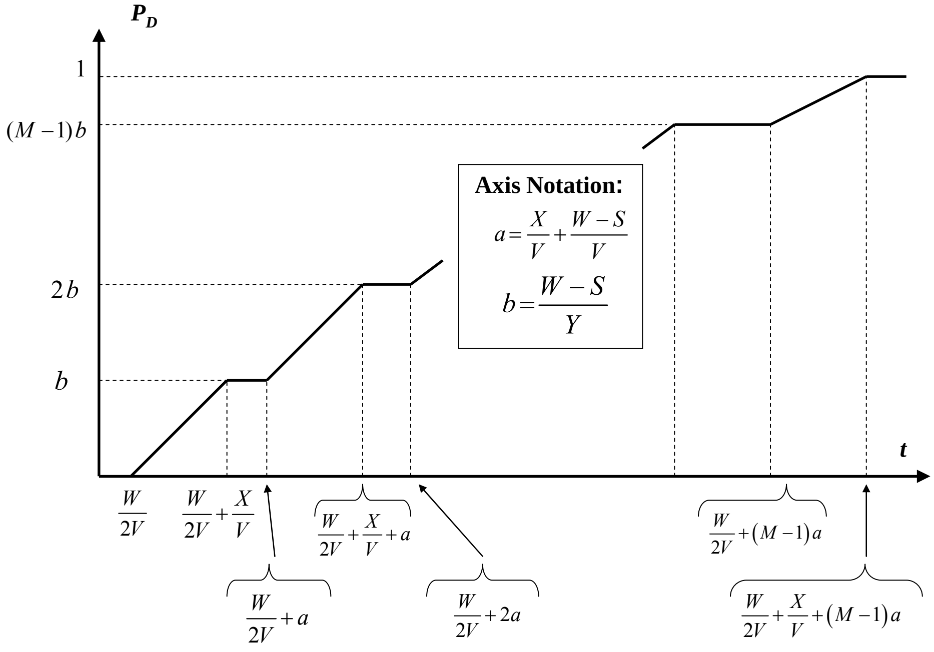


Figure 20. Probability of Detection versus Time for Progressive Search (Lower Bound)

Let the random variable $T_P$ denote the time at which the transmit CH is first detected during the progressive search. Then the probability that $T_P$ is less than some time $t$ is equal to the probability that the transmit CH has been detected by time $t$, which is given by Figure 20; that is, Figure 20 is also a graph of $\mathrm{P}(T_P \le t)$ versus $t$. Since [43]

$$\mathrm{E}[T_P] = \int_0^\infty \left[1-\mathrm{P}(T_P \le t)\right] dt$$

we have merely to determine the area under the graph of $1-P_D$, using the graph of $P_D$ (see Figure 20).

The graph of $1-P_D$ is shown in Figure 21. We divide the area under the curve into four different regions, as labeled in the figure. Note that Region 1 consists of a single rectangle, Region 2 consists of $M-1$ identically sized triangles, Region 3 consists of $M-1$ different-sized rectangles, and Region 4 consists of a single triangle.

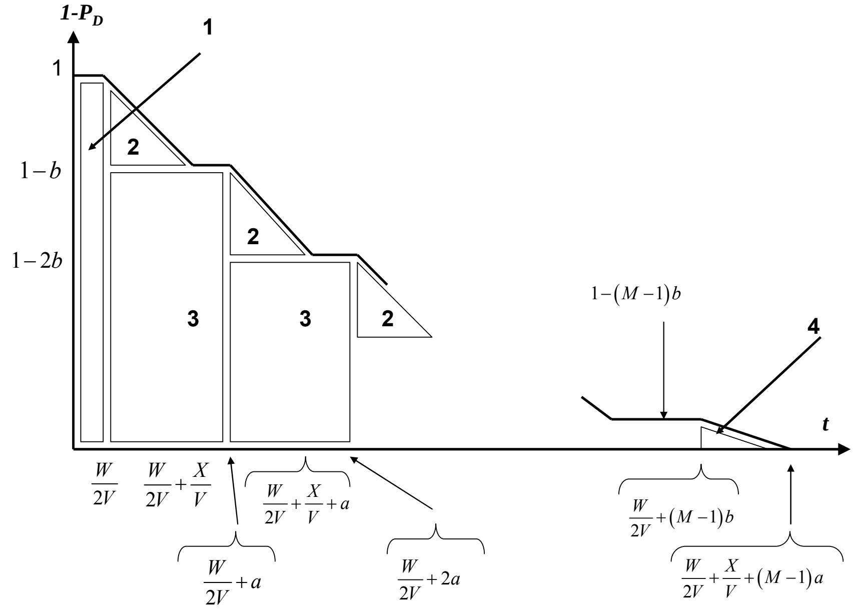


Figure 21. $(1-P_D)$ versus Time for Progressive Search

We compute the corresponding areas as:

Region 1: $\frac{W}{2V}$

Region 2: $\left(\frac{M-1}{2}\right)\left(\frac{X}{V}\right)\left(\frac{W-S}{Y}\right)$

Region 3:
$$\sum_{m=1}^{M-1}\left(\frac{X}{V}+\frac{W-S}{V}\right)\left(1-m\left(\frac{W-S}{Y}\right)\right)$$
$$=\left(\frac{X}{V}+\frac{W-S}{V}\right)(M-1)\left\{1-\frac{M}{2}\left(\frac{W-S}{Y}\right)\right\}$$

Region 4: $\frac{1}{2}\left(\frac{X}{V}\right)\left(1-\frac{(M-1)(W-S)}{Y}\right)$.

Equation (14) results from summing these three areas and simplifying the resulting expression. Recognizing that since our graph of $P_D(t)$ is a lower bound, our calculation of E[$T_P$] yields an upper bound.

**Corollary**. If $M = 1$, i.e., if the UAV is able to complete the search in a single sweep (i.e., $W \geq Y$), then (14) yields E[$T_P$] = $\left(W/2V\right)+\left(X/2V\right)$. This is as expected, since if the transmit CH is uniformly distributed within the search region and only a single sweep (of duration $X/V$) is required, then, on average, the CH will be detected halfway through the search. Note that if $W = Y$, then Figure 21 simply consists of the Region 1 rectangle and the leftmost Region 2 triangle whose area (which yields E[$T_P$]) is $X/2V$. (Alternatively, one can say that in this case Figure 21 consists solely of Region 4 whose area is $\frac{1}{2}\left(\frac{X}{V}\right)\left(1-\frac{(M-1)(W-S)}{Y}\right)$. Substituting $M = 1$ again yields an area of $X/2V$.)

**Theorem**. Suppose a UAV is engaged in a progressive search for a uniformly-distributed transmit CH. Let $T_P$ denote the time that the beam is first found by the UAV. A lower bound on the expected value of $T_P$ is given by

$$\begin{aligned} \mathrm{E}\left[T_P\right]_{LB} &= \mathrm{E}\left[T_P\right]_{UB} - \frac{\pi W^2}{16A}\left(\frac{W+X}{V}\right) \\ &-\left(\frac{X+W-S}{V}\right)(M-1)\left(\frac{S}{Y}\right)-\left(\frac{X}{2V}+\frac{W-S}{V}\right)\frac{W(W-S)(M-1)}{2A}, \end{aligned} \tag{15}$$

where $E\left[T_p\right]_{UB}$ is given by (14).

**Proof**. We sketch the proof. The proof follows the same procedure as the preceding theorem except now we ask, informally: *What is the best we can do?* and, more formally: What is an upper bound on $P_D(t)$?

We upper-bound $P_D(t)$ by examining what would happen if the transmit CH was stationary. The initial sweep examines a region of size $WX$, and the probability that the transmit CH has been detected at the end of the first sweep is $W/Y$. The UAV then repositions for the second sweep, taking time $(W-S)/V$. The second sweep does not contribute as much to the probability of detection as the first sweep since the second sweep overlaps the first by an amount equal to $S$, and only examines an additional area equal to $X(W-S)$; the probability of detection at the conclusion of the second sweep is $W/Y+(W-S)/Y$. The final sweep differs from the prior sweeps in that some of the UAV's beam may lie outside the sensor region, and the probability of detection will rise to one at the conclusion of the last sweep. To further ensure that our estimate of $P_D(t)$ is indeed an upper bound, the new area covered as the UAV repositions is taken into consideration at the conclusion of the preceding sweep. The new area covered by the UAV as it positions for the next sweep is over-overestimated by a rectangle of size $W(W-S)/2$ (see Figure 22), and the incremental probability of detection, equal to $W(W-S)/2A$, is immediately added to the value of $P_D(t)$ at the conclusion of the previous sweep. Likewise, we assume that the area covered at the very start of the search, as the perimeter of the UAV's circular beam first enters the sensor region at time $t=0$, until the UAV itself is over the sensor region at $t=W/2V$, immediately contributes a probability of detection of $\pi(W/2)^2/2A$ at $t=0$. With these considerations in mind, Figure 23 displays an upper bound for the probability of detection, $P_D$, versus time, $t$, for the progressive search. To simplify the axes in the figure, the following notation is used: $a=(X+W-S)/V$, $c=W(W-S)/2A$ and $d=(W-S)/Y$.

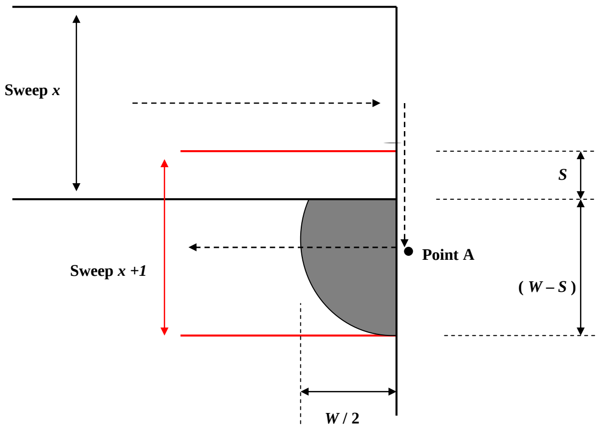


Figure 22. New Area Covered as UAV Repositions for Next Sweep

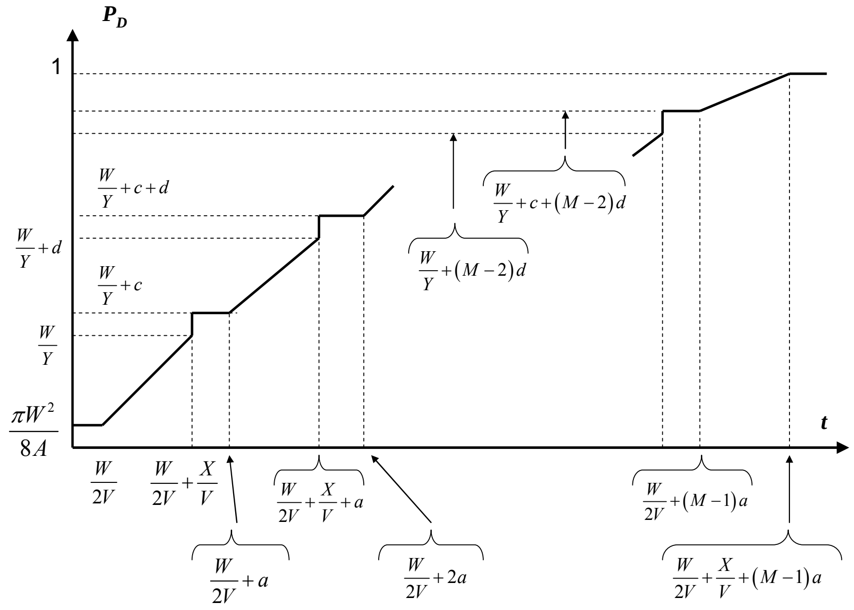


Figure 23. Probability of Detection versus Time for Progressive Search (Upper Bound)

As before, Figure 23 is also a graph of $P(T_P \leq t)$ versus *t*. Determining the area under the graph of $1-P_D$, using the graph of $P_D$ (see Figure 23), provides $E[T_P]_{LB}$. The details of the calculation—tedious but straightforward—are omitted.

**Lemma**. Suppose a UAV is engaged in a search for a uniformly-distributed transmit CH in a region of size $A$. The UAV conducts a random search. Let $T_R$ denote the time at which the CH is first detected by the UAV. The expected value of $T_R$ is given by

$$\mathrm{E}[\ T_R\ ] = \ A/W\,V\ . \tag{16}$$

**Proof**. $T_R$ denotes the value of time at which the transmit CH is first detected during the random search. Then the probability that $T_R$ is less than some time $t$ is equal to the probability that the CH has been detected by time $t$, which is given by (13). Thus, the probability distribution function for $T_R$ is also given by (13):

$$\mathrm{P_D}\left(T_R \le t\right) = 1 - e^{\frac{-W\,V\,t}{A}}\ .$$

The expected value of $T_R$ is calculated as

$$\mathrm{E}\left[T_R\right] = \int_0^\infty \left[1 - \mathrm{P}\left(T_R \le t\right)\right] dt = \int_0^\infty e^{-W\,V\,t/A} = \frac{A}{W\,V}\,.$$

**Theorem**. Consider a search for a transmit CH, uniformly distributed within a region of size $A$. The search is conducted using a UAV. The expected value of the time to find the CH will be lower (i.e., better) using a progressive search as long as the following inequality is satisfied:

$$\frac{W}{2V} + \frac{X}{2V} + \left(\frac{X}{V} + \frac{W-S}{V}\right)(M-1)\left\{1 - \frac{M}{2}\left(\frac{W-S}{Y}\right)\right\} \ < \ \frac{A}{W\,V}\,, \tag{17a}$$

and will be lower using a random search as long as the following inequality is satisfied:

$$\begin{aligned} \mathrm{E}\left[T_P\right]_{UB} - \frac{\pi W^2}{16A}\left(\frac{W+X}{V}\right) - \left(\frac{X+W-S}{V}\right)(M-1)\left(\frac{S}{Y}\right) & \\ -\left(\frac{X}{2V} + \frac{W-S}{V}\right)\frac{W(W-S)(M-1)}{2A} \ > \frac{A}{WV}\ . & \end{aligned} \tag{17b}$$

**Proof**. Follows immediately from (14), (15) and (16).

## E. RELATIONSHIP BETWEEN ANTENNA GAIN, UAV HEIGHT AND SWEEP WIDTH

In this short section, we derive a relationship between the gain of the sensor array's distributed antenna (assumed to be approximately equal to the UAV's antenna gain), the height of the UAV and the UAV's sweep width.

The notion of concentrating the radiated power into a conical beam is illustrated in Figure 24. It can be shown that the relationship between the antenna gain, $G$, and the solid angle, $\Omega$, through which the power is concentrated is given by [44]:

$$G = \frac{4\pi}{\Omega} , \tag{18}$$

where $\Omega$ is in units of steradians. Generally, if a specified section of a sphere's surface has an area $A_{Beam}$, and this section subtends a solid angle of $\Omega$ steradians, then

$$\Omega = \frac{A_{Beam}}{z^2} , \tag{19}$$

where $z$ is the radius of the sphere (a conical section of which is shown in Figure 24). [‡‡] If the antenna gain is large (i.e., $\Omega$ is small), the area of the section of the sphere's surface will be closely approximated by the area of the base of the cone with the same solid angle $\Omega$ and height $z$: $A_{Beam} = \pi\left(W/2\right)^2$, where $W$ is the width of the beam (see Figure 24 and Figure 16a). Substituting this expression into (19) yields $\Omega = \frac{\pi\left(W/2\right)^2}{z^2}$. Substituting this expression into (18) yields a relationship between $W$ and $G$ that will prove useful:

$$W = \frac{4z}{\sqrt{G}} . \tag{20}$$

---

[‡‡] In general the beam area is elliptical, and $A_{beam} = \frac{\pi}{4} W_1 W_2$ where $W_1$ and $W_2$ are the major and minor axes of ellipse. We assume $W_1 = W_2$.

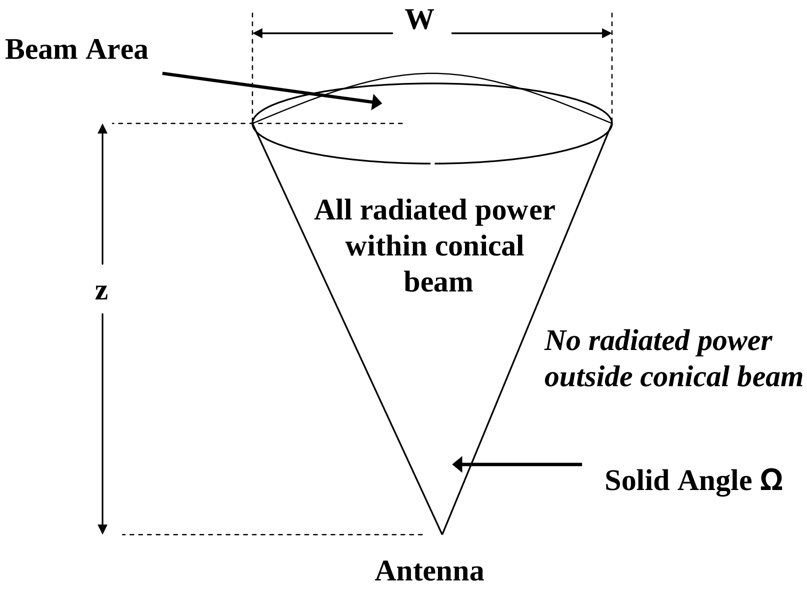


Figure 24. Antenna Radiated Power Contained within Solid Angle Ω

## F. APPLICATIONS

We now present three scenarios that illustrate how real-world practical considerations are made analytically accessible by the foregoing analysis. The following assumptions apply to all three scenarios:

- A sensor network consisting of many sensor nodes is distributed over a 10 km × 10 km region.[§§]
- The maximum communication range between sensor nodes is $d = 25$ m.
- The UAV is a Shadow-200 that will fly over the sensor region at a speed of $V = 120$ km/hr and has a maximum endurance of seven hours [45]. Given that the UAV base is a certain distance from the sensor region, the time that the UAV spends over the sensor region is limited to five hours.
- To minimize the UAV's risk of being shot down, mission planners have decided that the UAV must be kept at an altitude greater than 4000 feet. The UAV altitude is chosen to be $z = 4100$ feet (1250 m).
- To communicate with the UAV at the specified altitude (1250 m), while meeting the required bit error rate for the given modulation scheme, mission planners have calculated that the sensor network must form a

---

[§§] Recall that the UAV assumes nodes are uniformly distributed over the region. They need not (and likely will not) be.

distributed antenna with a gain of $G = 100$. (The details of how this value was determined are not of interest.)

### 1. Scenario 1: The UAV Must (with Certainty) Establish a Link with the Sensor Network

Since we must guarantee link establishment, a progressive search is required (recall that a random search cannot assure link establishment, see (13)). Since $G = 100$ and $z = 1250$ m, we calculate the sweep width using (20) as $W = 500$ m. Using equation (12), we can readily calculate the time to complete the progressive search as a function of the transmit CH maximum velocity $v$. In fact, solving (12) for the given parameters is exactly the scenario that is depicted in Figure 19. Since we must limit the search to five hours, we readily determine (either analytically using (12) or from Figure 19) that the maximum permissible transmit CH velocity is $v = 2$ km/hr, which corresponds to an overlap between successive sweeps equal to 336 m (using (9)). This represents the maximum distance the transmit CH is permitted to move in $\frac{2X + W - S}{V} = 10.08 \text{ min}$. Since, by (8), the distance moved is equal to $(h+1)d$, we find the maximum number of hops separating a newly assigned transmit CH from the previously assigned transmit CH is $h = 12$. Thus, the sensor nodes must be programmed such that:

- The time between shifts in the transmit CH, $T_{hold}$, must be $\geq$ 10.08 minutes. That is, once a cluster (with its associated CH) assumes the role as the sensor network's transmit cluster, it must retain these duties for a period of at least 10.08 minutes before the transmit cluster role is passed on to a different cluster.
- The number of hops separating a newly assigned transmit CH from the previous transmit CH must be $\leq$ 12.

### 2. Scenario 2: The UAV Should Establish the Link with the Sensor Network as Quickly as Possible

Figure 25 displays the expected time to find the transmit CH for both the progressive and random searches as a function of the CH velocity, for the given parameters ($X = 10$ km, $Y = 10$ km, $V = 120$ km/hr and $W = 0.5$ km). These plots are obtained from (14), (15) and (16).

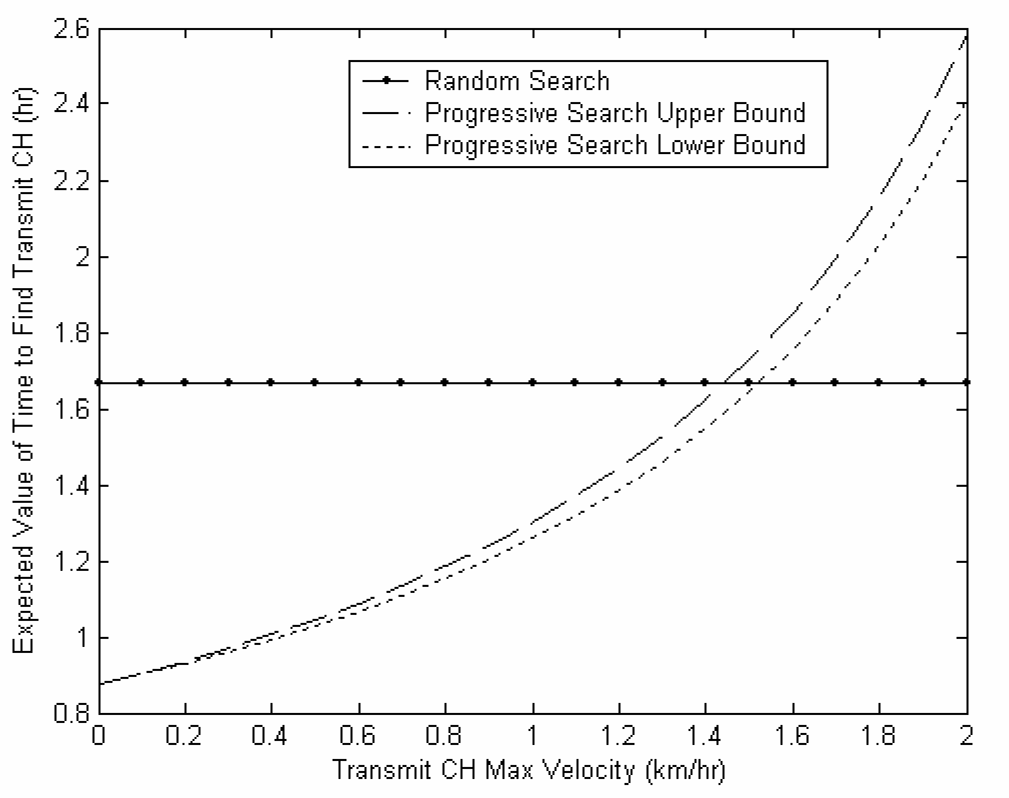


Figure 25. Expected Time to Find the Transmit CH versus CH Velocity

If the sensor network is programmed as in Scenario 1 above, with the transmit CH maximum velocity set to 2 km/hr, Figure 25 would suggest that, if our primary concern is finding the transmit CH as quickly as possible (i.e., we no longer require that a link be established with certainty), we would do better by having a “cheap” UAV search the sensor region randomly. Prior to deployment, a designer, referring to Figure 25, might desire a maximum transmit CH velocity of 1.4 km/hr so that the UAV is certain to find the transmit CH (with probability one) and in a time better than the random search. In this case, using the analysis of this chapter, we readily find that the maximum overlap between successive sweeps is limited to 236.4 meters, in a time of 10.132 min. This would entail programming the sensor nodes so that a transmit CH will retain its duties for a minimum time of 10.132 min and then transfer duties to a new CH at most 8 hops away. So, the tradeoff in designing the sensor nodes to ensure beam alignment with the UAV in a time better (on average) than that which would occur for the random search is a decrease in the distance between successive transmit CH from 12 hops to 8 hops. This restricts the rate at which energy expenditures are distributed throughout the network. (Also, of minor note, the new design requires the transmit CH retain its duties for a slightly longer time).

### 3. Scenario 3: The Sensor Network Can Adjust the Gain, G, of its Distributed Antenna Array

Suppose that the sensor network can adjust the gain of its distributed antenna (by incorporating more sensor nodes into the array). It may wish to do this in order to improve the BER at the UAV's receiving antenna. Increasing the gain has the effect of decreasing the search width $W$ per (20). (Recall that the UAV's gain must be comparable to the sensor network's antenna gain so that when the transmit CH falls within the UAV's transmission beam, the UAV will fall within the sensor network's transmission beam.)

Figure 26 displays the expected time to find the transmit CH for both the progressive and random searches as a function of the transmit CH velocity, for the given parameters ($X$ = 10 km, $Y$ = 10 km and $V$ = 120 km/hr), for three different gains. Note that the curves shown for the progressive search represent the upper bounds (i.e., (14)).

If our intended network was designed with a sensor network gain of 100 and a maximum transmit CH velocity of 1.4 km/hr, then Figure 26 shows that the progressive search is expected to find the transmit CH as fast as the random search and will surely find it by the conclusion of the search. If the gain is increased to 200 with the transmit CH maximum velocity unchanged, Figure 26 shows that now the random search will find the transmit CH faster. If we wish to restore the virtue of the progressive search, the transmit CH velocity would need to be reduced to 1 km/hr. This entails reprogramming the sensor nodes so that a transmit CH will transfer its duties to a new CH at most 5 nodes away (whereas the gain of 100 allows transferring duties to a node 8 nodes away).

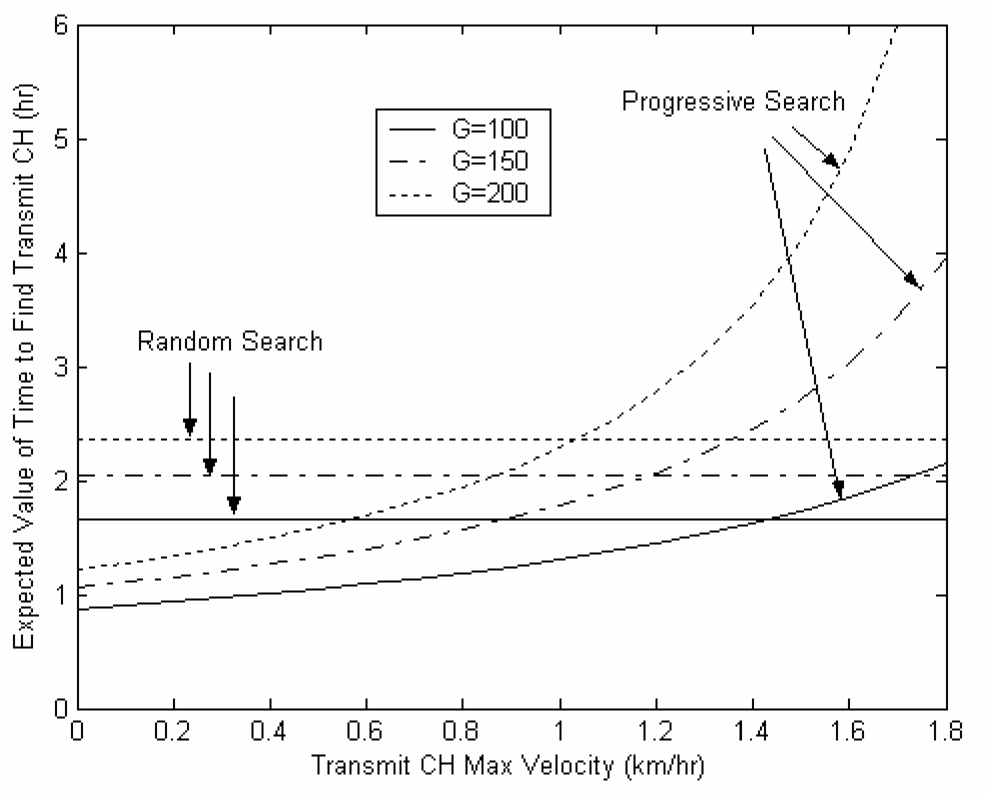


Figure 26. Expected Time to Find the Transmit CH versus CH Velocity

The basic approach described in this chapter greatly reduces the power demand placed on the sensor nodes by reconfiguring the communication and computational burden between the sensor network and an overhead UAV. The UAV is tasked with aligning the sensor network's transmission beam with its own airborne receiver. The sensor network's transmit CH merely calculates a single set of amplitude and phase offsets for the sensor nodes participating in the distributed array, and then "stands down," waiting for the UAV to find it. In this chapter we have described, analyzed and compared the performance of two strategies for reconfiguring the communication and computational burden between a wireless ground-based sensor network and an overhead Unmanned Aerial Vehicle (UAV). Both strategies bring the UAV and the sensor network's transmission beam into alignment while minimizing the energy expended by the sensor network. The performance of the two strategies is compared in terms of probability of beam-UAV alignment as a function of time and the expected time to alignment. We examined the performance tradeoff between the choice of strategy and parameters of the sensor network that affect power conservation. The three example scenarios presented at the close of this chapter illustrated the design tradeoffs that are made analytically accessible by the foregoing analysis presented in this and the previous chapters.

Now that we have dealt with the problems of aligning the transmit CH's beam with the UAV in an energy-conserving manner and have also dealt with the problem of moving the transmit CH around the sensor network in a way that conserves energy while evenly distributing the energy load, we next narrow our focus in the next chapter to a single cluster. How should a cluster form a distributed antenna array in a manner that conserves power?

THIS PAGE INTENTIONALLY LEFT BLANK

# IV. OPTIMIZING THE SIZE OF AN ANTENNA ARRAY‡

Having developed an energy-efficient algorithm to move the transmit cluster to other locations in the sensor network to evenly spread the energy load and extend the lifetime of the network (Chapter II) and having also developed an energy-efficient approach to aligning the transmit cluster's beam with the UAV (Chapter III), we now, in this chapter, examine how the transmit cluster should assemble its antenna array in an energy-efficient fashion.

## A. ANTENNA GAIN AND RADIATION PATTERN

For wireless communication, the electrical signal energy in the transmitter is converted to propagating electromagnetic waves by an associated antenna. As mentioned in Chapter II, successful communication requires that some of the energy that originates at the transmit antenna arrives at the receiver's antenna, where the incident electromagnetic waves are converted back to signals that, hopefully, are smaller-amplitude replicas of those that existed in the transmitter. The electromagnetic field propagates at the speed of light, $c = 3 \text{ x } 10^8$ meters/sec, at a frequency $f$ Hz, and with a wavelength $\lambda$ equal to:

$$\lambda = \frac{c}{f}. \tag{21}$$

Recall from Chapter II that an antenna's *radiation pattern* displays the relative distribution of the radiated power as a function of the direction in space. With the antenna at the origin of a spherical coordinate system, the radiation pattern can be found by measuring the power density at all points on the surface of a sphere at a distance $r$ from the origin (i.e., the antenna). As shown in Figure 6, the variable $r$ ranges from zero to infinity, the angle $\theta$, termed the *elevation angle* ranges from 0° to 180° and the angle $\phi$, termed the *azimuth angle*, ranges from 0° to 360°.

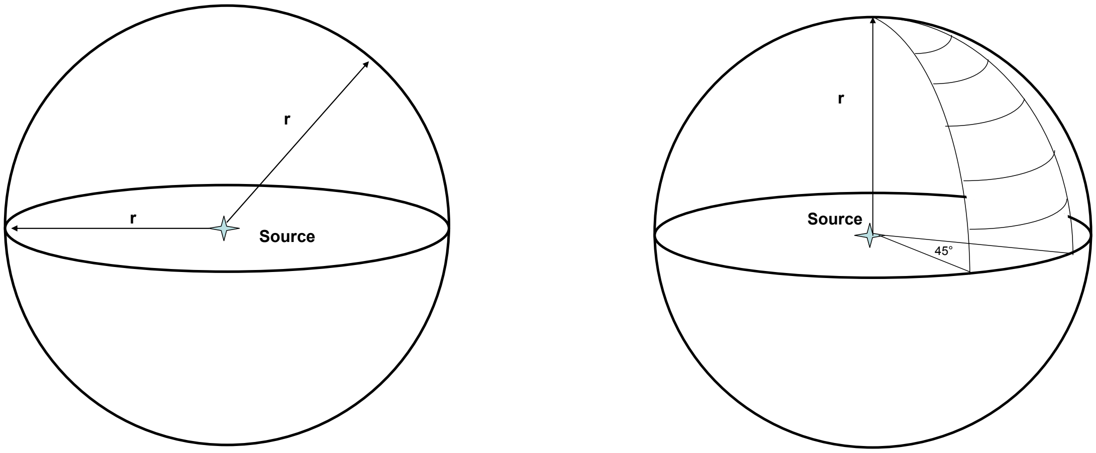
---

‡ The results in this chapter were previously presented at the *40th Annual Asilomar Conference on Signals, Systems, and Computers* [46].

An *isotropic* antenna is one that radiates electromagnetic energy uniformly in all directions, over the entire solid angle of $4\pi$ steradians (Figure 27(a)). For an isotropic antenna, the power density will have the same value on any sphere centered at the origin; that is, since power radiates uniformly in all directions, the power density is the same at any point a distance $r$ from the antenna. If an isotropic antenna is radiating a power of $P_t$ watts, then the power density at any point on the surface of a sphere of radius $r$ centered at the source is

$$\mathrm{P}_r = \frac{P_t}{4\pi r^2} \text{ watts/meter}^2, \tag{22}$$

where $4\pi r^2$ is the surface area of a sphere of radius $r$. Thus, in free space, power density is inversely proportional to the square of the distance of the transmitter.



(a) Isotropic Radiating Source (b) Non-isotropic Radiating Source

Figure 27. Power Density of an Antenna

Suppose, though, that the power can be preferentially radiated in certain directions. In this case, the power density in the preferred directions will be greater than that of an isotropic antenna while the power density in the other directions will be less than that of an isotropic antenna (with both antennas radiating the same total power).

Suppose that we have an antenna that is able to radiate all its power uniformly through the wedge shown in Figure 27(b). Noting that the surface area of the wedge is 1/16 the total surface area of the sphere, the power density will be

$$\mathrm{P}_r^{\,nonisotropic} = \frac{P_t}{\left(\frac{1}{16}\right)4\pi r^2} = \frac{16\,P_t}{4\pi r^2} \text{ watts/meter}^2 \tag{23}$$

at points on the outer surface of the wedge, and zero elsewhere. Comparing equations (22) and (23), we see that at points on the surface of the wedge

$$\frac{\mathrm{P}_r^{\,nonisotropic}}{\mathrm{P}_r} = 16 \ .$$

The non-isotropic antenna in Figure 27(b) is said to have a *gain* of 16 in the wedge through which the antenna radiates (and a gain of zero over the remainder of the sphere). (Recall from Chapter II that we are assuming that dissipative losses in the radiator can be neglected, and thus we do not distinguish between the antenna's directive gain and its power gain).

More formally, the gain, *G*, of an antenna, at a certain point in space, is defined as the ratio of the power density present at that point to that which would be present (at the same point) if an isotropic antenna were radiating the same total power. In the example of Figure 27(b), the gain takes on one of two values: zero or 16. In general, the gain will not be confined to two values but will vary over the surface of a sphere centered at the radiating antenna. Earlier, we defined an antenna's *radiation pattern* as a display of the relative distribution of the radiated power as a function of the direction in space. Now, we see that the gain is equivalent to the radiation pattern defined earlier, provided the radiation pattern is normalized by the power density of an isotropic antenna radiating the same total power.

Returning to Figure 27(b), note that a receiving antenna on the outer surface of the wedge will "see" a power density of $16\,P_t/4\pi r^2$ . The receiving antenna does not know if this power density is due to an isotropic antenna radiating a total power of 16 $P_t$

watts, or a non-isotropic antenna with a gain of 16 radiating a total power of $P_t$ watts. In general, if an antenna with gain *G* radiates $P_t$ watts, the power density at a distance *r* will be

$$P = G\,P_t / 4\pi r^2 \text{ watts/meter}^2 . \tag{24}$$

Thus, an antenna with a gain of 1 and an input power of 10 watts will be as effective as an antenna with a gain of 10 and an input power of 1 watt. Employment of an antenna with a gain of 10 permits a reduction in the required input power by a multiple of 10, while still placing the same amount of power at the receiver. Therefore, if we are interested in minimizing the radiated power, it is highly desirable to use an antenna that manages to concentrate the radiated power within a solid angle less than 4π steradians so that the gain of the antenna is high.

Note that the average gain, $\overline{G}$, over the surface of a sphere (centered at the radiating antenna) must be one, a consequence of the conservation of energy principle. For the isotropic radiator of Figure 27(a), the gain at every point on the surface of the sphere is one, so the average gain is also equal to one. For the non-isotropic antenna of Figure 27(b), the average gain is

$$\overline{G} = \frac{1}{16}(16) + \frac{15}{16}(0) = 1 .$$

Consider an antenna which radiates uniformly through a small section of a spherical surface of area *A*, subtended by the solid angle Ω. The gain of this antenna is

$$G = \frac{P_t / A}{P_t / 4\pi r^2} = \frac{4\pi r^2}{A} \quad . \tag{25}$$

Generally, if a specified section of a sphere's surface has an area *A*, and this section subtends a solid angle of Ω steradians, then, by direct use of (19),

$$\Omega = \frac{A}{r^2} \ , \tag{26}$$

where, again, *r* is the radius of the sphere. Combining equations (25) and (26) we see that

$$G = \frac{4\pi}{\Omega} \ , \tag{27}$$

so, as intuition would expect, the narrower the radiated beam (i.e., the smaller $\Omega$) the larger the antenna gain.[***] Note again that the average gain over an entire sphere centered at the radiating antenna is $\left(\frac{\Omega}{4\pi}\right)G + \left(\frac{4\pi - \Omega}{4\pi}\right)0 = \frac{\Omega\, G}{4\pi} = 1$.

In arriving at (27), note that we have assumed that the antenna radiates a uniform power density within $\Omega$, and has no radiation outside this solid angle. In general, if the angular variation in the radiation around an antenna is $F(\theta,\phi)$, then the directivity, $D(\theta,\phi)$, defined as the ratio of the radiation intensity in a specified direction to the average radiation intensity, is given as [47]

$$D(\theta,\phi) = \frac{\left|F(\theta,\phi)\right|^2}{\frac{1}{4\pi}\iint \left|F(\theta,\phi)\right|^2 d\Omega} \ . \tag{28}$$

Several other points are worth noting briefly. First, we will append a subscript to the gain to differentiate between the gain of a transmitting antenna ($G_t$) and the gain of a receiving antenna ($G_r$). Second, the term "gain" is often used to refer to the value of the gain in the maximum-radiation direction. For example, an antenna with a gain of 2 in some directions and a gain of 1.9 in other directions (and a gain of zero in still other directions) will often be said to have a gain of 2. The intended meaning should be clear from the context. Third, quantities, such as gain and power, are often expressed in decibels (dB): if the gain is *G*, then the gain in decibels is $G_{dB} = 10\log_{10} G$.

We next state a fact from antenna theory that will be useful in the sequel. The derivation of this result requires concepts from electromagnetic radiation theory, the details of which are not of interest. First, recall that the signals in the transmitter are converted to propagating electromagnetic waves by an associated transmitting antenna.

---

[***] (27) is the same as (18), which was first presented in Chapter III without proof so as not to sidetrack the main thrust of the discussion. Now, for the present discussion, the details are relevant.

At the receiving antenna, the incident electromagnetic waves can be measured in terms of a power flux density (in watts/m$^2$), as previously described. From this incident power flux, the receive antenna extracts a useful signal of power $P_r$ watts, and this signal is sent on to the receiver for processing. If the receive antenna sits in a power flux of P watts/m$^2$, and extracts a power of $P_r$ watts, it is useful to consider the receive antenna as having an effective area, $A_r$, such that

$$P_r = \mathrm{P}\, A_r\,. \tag{29}$$

An antenna's effective area can be related to the receive antenna's gain and the wavelength of the incident signal by the formula:

$$A_r = \frac{G_r\,\lambda^2}{4\pi}\,, \tag{30}$$

where $G_r$ is the gain of the receiving antenna and $\lambda$ is the wavelength of the incident signal. For a derivation of (30), see [48].

Having established that it is advantageous for the sensor network to communicate with a UAV, consider Figure 28 which shows a sensor node, with a gain $G_t$, transmitting at a power $P_t$, attempting to communicate with a UAV at an altitude $r$. The gain of the UAV's receiving antenna is $G_r$. Substituting (24) and (30) into (29) (and subscripting the gain in (24) as $G_t$), we have an expression for the useful power extracted by the UAV's receiver:

$$P_r = P_t\, G_t\, G_r \left(\frac{\lambda}{4\pi r}\right)^2 . \tag{31}$$

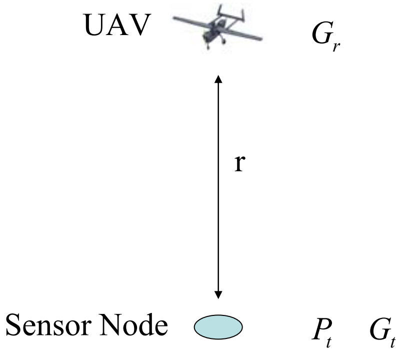


Figure 28. Communication Link Between a Sensor Node and a UAV

Substituting (21) into (31) then yields

$$P_r = P_t G_t G_r \left( \frac{c}{4\pi rf} \right)^2 . \tag{32}$$

We assume that the noise power at the receiver is additive white Gaussian thermal noise, which can be expressed as [49]

$$N = kT_S B , \tag{33}$$

where $k$ is Boltzmann's constant, $1.38 \times 10^{-23}$ watts/K-Hz, $T_S$ is the effective system noise temperature (in degrees K), and $B$ is the receiver bandwidth (in Hz). (The effective system noise temperature, $T_S$, is the sum of the antenna temperature and the effective noise temperature of the receiver.) Combining (32) and (33), the received signal power to noise power ratio at the receiver is given by

$$\frac{P_r}{N} = \frac{P_t G_t G_r}{kT_S B} \left( \frac{c}{4\pi rf} \right)^2 . \tag{34}$$

For digital signaling, we are more interested in the energy per bit ($E_b$) to noise power spectral density ($N_o$) ratio at the receiver, rather than the received power to noise power ratio. If $R_b$ is the data rate of the received digital signal, then $E_b/N_o$ is related to $P_r/N$ by:

$$\frac{E_b}{N_o} = \left( \frac{P_r}{N} \right) \left( \frac{B}{R_b} \right) . \tag{35}$$

Combining (34) and (35), we then have

$$\frac{E_b}{N_o} = \frac{P_t G_t G_r}{kT_S R_b}\left(\frac{c}{4\pi rf}\right)^2 . \tag{36}$$

Finally, (36) is modified by accounting for additional signal degradation and loss factors, and additional noise terms that may arise from sources, such as modulation loss, phase noise, antenna pointing loss, atmospheric loss and noise, imperfect synchronization, adjacent channel interference, and so forth. Representing these additional losses and noise factors as $L_o$, we finally have:

$$\frac{E_b}{N_o} = \frac{P_t G_t G_r}{kT_S R_b L_o}\left(\frac{c}{4\pi rf}\right)^2 . \tag{37}$$

For communication between the sensor node and the UAV to be successful under the prevailing channel conditions, the node's transmit power and gain, $P_t$ and $G_t$, must be sufficient to assure that enough residual energy arrives at the receiving antenna to achieve a specified bit error rate (BER), depending on the modulation scheme. Specifically, the energy per bit to noise power spectral density ratio at the receiver must meet a specified value, where the formula for the energy per bit to noise power spectral density ratio is given by (37).

Now, if the data rate $R_b$ and transmission frequency $f$ are decided in advance, and the BER (and hence the minimum $E_b/N_o$ ) is decided in advance, then for a given UAV at a given elevation in a given channel, we only have the sensor node's transmit power and gain, $P_t$ and $G_t$ , at our disposal to attempt to meet the required $E_b/N_o$ at the UAV.

As previously mentioned in Chapter II, though, sensor nodes are limited in their transmit power. It is likely that, even when the sensor node transmits at maximum power, the energy arriving at the UAV at an elevation of $r$ meters overhead will be insufficient. Even if the sensor node is able to transmit at a sufficient power to close the link with the UAV, it must be recalled that when a sensor's small battery fails, the sensor node is rendered inoperable. Thus, we desire that sensor nodes transmit using the minimal power. So, it would seem we only have the sensor node's gain, $G_t$ , at our disposal to attempt to meet the required $E_b/N_o$ at the UAV.

Unfortunately, the antennas on these small and inexpensive sensor nodes are omnidirectional, with a gain of unity ($G_t = 1$).

So, it would seem, we are at an impasse: we have established that a military sensor network needs to communicate with a UAV, but sensor nodes are unable to communicate with a UAV.

A solution, proposed by [18], is to increase the transmission range between the sensor nodes and the UAV by using distributed beamforming, whereby sensor nodes are grouped into clusters and their transmissions are then coordinated to form a distributed antenna array that directs the beam towards the UAV.

## B. ANTENNA ARRAYS

The gain of an antenna indicates its ability to concentrate power in a certain direction, and an antenna's radiation pattern displays the relative distribution of the radiated power as a function of the direction in space. We have discussed antenna gain in the context of a single antenna element. If multiple antenna elements are used to transmit the same information (with the transmissions from each element differing by only a complex constant; i.e., differing only in magnitude and phase), then the combined group of antenna elements, termed an antenna array, can achieve the same effect as a single antenna with an improved gain.

This improvement in gain when using multiple antenna elements is premised on a simple fact: When two or more electromagnetic waves exist at the same point in space, they interfere with each other; the total electric field at the point is the vector sum of the individual electric field vectors associated with the interfering waves. The waves may interfere constructively (i.e., reinforce each other) or destructively (i.e., tend to cancel each other) depending on the relative amplitudes and relative phase differences between the individual sensors.

Consider Figure 29, which shows two omnidirectional antennas separated by a distance *d*. Suppose that the electric field radiated by Node 1 is $E_o \sin(2\pi ft)$. This electric field propagates outward, and the field at the UAV due to Node 1 is then

$\eta E_o \sin\left(2\pi ft - \frac{2\pi h_1}{\lambda}\right)$, where $\lambda$ is the wavelength of the signal, $\eta$ represents an attenuation factor and $h_1$ is the distance from Node 1 to the UAV. Now, suppose that the electric field radiated by Node 2 is $E_o \sin(2\pi ft + \alpha)$. (Note that for this example, Node 2's transmission has the same signal amplitude, but differs in phase, from that of Node 1.) This electric field similarly propagates outward, and the field at the UAV due to the Node 2 is then $\eta E_o \sin\left(2\pi ft - \frac{2\pi h_2}{\lambda} + \alpha\right)$, where $h_2$ is the distance from the Node 2 to the UAV. (We assume that the differing distances from the UAV to each node are small enough such that the attenuation factor, $\eta$, is the same for both signals.) These two electric fields add at the UAV. It is a simple matter to show that the resultant electric field intensity at the UAV is given by

$$E = 2\eta E_0 \cos\left(\frac{\alpha}{2} - \frac{\pi d \sin\theta}{\lambda}\right), \tag{38}$$

where the angle $\theta$ is shown in Figure 29.

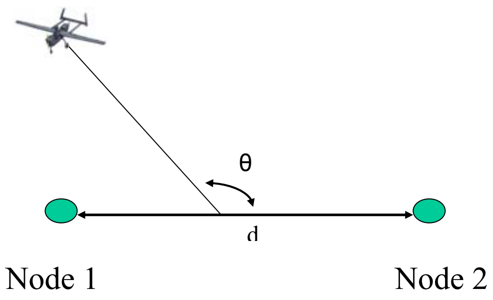
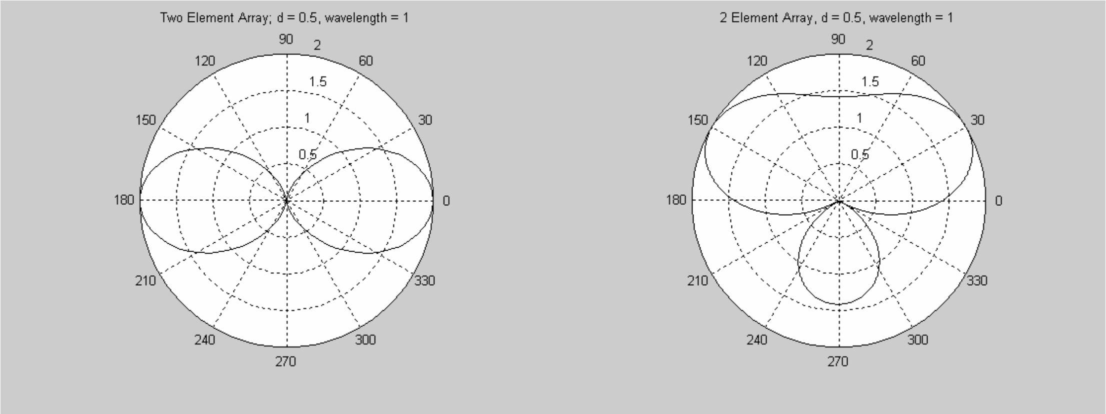


Figure 29. Two-Element Sensor Array

If the transmissions from the two sensor nodes have the same amplitude ($E_o$) and frequency (and hence the same $\lambda$), then the electric field intensity at the UAV as a function of the angle $\theta$ depends on the phase difference between the transmissions ($\alpha$) and the separation of the nodes ($d$). Figure 30(a) shows the radiation pattern that results if the phase difference between the transmissions is zero. Figure 30(b) shows the radiation pattern that results if the phase difference is 90°. (Note that in Figure 30 the two plots are rotated 90° from the scenario shown in Figure 29; the two sensors would be aligned along the vertical axis, and a UAV directly overhead would be at an angle of 0°.)

This example shows that different electric field patterns can result from seemingly small differences in the transmissions from the individual elements comprising the array. Again, the only difference in the scenarios shown in Figure 30 (a) and (b) is a phase shift of 90° in the transmission of one of the elements.



(a) (b)

Figure 30. FElectric Field Pattern for Two-Element Array: (a) $\alpha$ =0 (b) $\alpha = 90°$

The significance of these plots of electric field intensity is the following: the power density at a point in space is proportional to the square of the electric field intensity at that point, and is given by (30): $P = E^2/240\pi$. Thus, by using an array of sensors, we can modify the relative distribution of the radiated power as a function of the direction in space, and, concomitantly, adjust the radiation pattern (and thus, the gain) to suit our needs.

We generalize our discussion from a two-element array to a linear array of $M$ omnidirectional elements, shown in Figure 31. For simplicity, we regard the antenna array as a receiving array, and the transmitter (perhaps a UAV) emits a (complex) signal $s(t)$ modulating a complex carrier $e^{jw_c t}$. We assume that the array is at a distance such that the arriving wavefront is approximately plane, meeting each array element at an elevation angle of $\theta$ with respect to the array normal (i.e., the z-axis). With the reference placed at the origin, the signal wavefront arrives at the $m^{\text{th}}$ element $t_m$ seconds before reaching the origin, with

$$t_m(\theta) = \frac{x_m \sin\theta}{c} \ , \tag{39}$$

where $c$ is the speed of light and $x_m$ is the $x$-coordinate of the $m^{\text{th}}$ element. Now, if we were to measure the output from each antenna element, the measured signal would be the same, except for a phase shift term equal to $e^{j\omega_c t_m(\theta)}$ for element $m$. If we assume that the amplitude of each element response is equal to unity, then the sum of the antenna outputs will give the radiation pattern (the spatial response) of the array [47]. Terming the spatial response as a function of $\theta$ as AF($\theta$), we have:

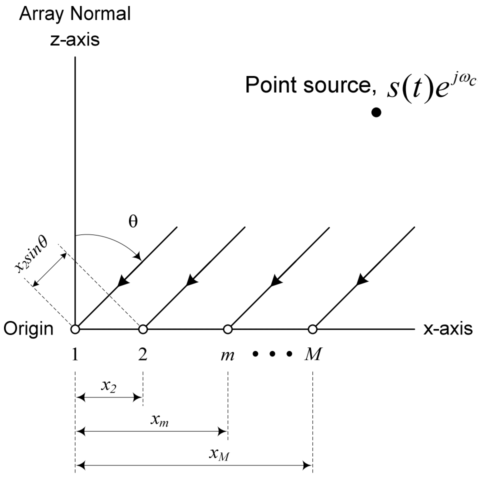


Figure 31. An M-Element Uniform Linear Array[†††]

$$AF(\theta) = \sum_{m=1}^{M} e^{j\omega_c t_m(\theta)} \ . \tag{40}$$

---

[†††] Note that "Uniform" in the context of this figure caption refers to the fact that there is the same amplitude drive for all elements. The inter-element spacing need not be uniform.

Suppose that we multiply the output of each array element (which is $e^{j\omega_c t_m(\theta)}$ for element *m*) by a complex number (termed a *complex weight*), $I_m e^{-j\omega_c t_m(\theta_0)}$ prior to the addition performed in (40). The array factor would then be:

$$AF(\theta) = \sum_{m=1}^{M} I_m e^{-j\omega_c t_m(\theta_o)} e^{j\omega_c t_m(\theta)} . \tag{41}$$

Together, $I_m$ and $e^{-j\omega_c t_m(\theta_0)}$ form what is termed the complex weights, with the phase reversal in the term $e^{-j\omega_c t_m(\theta_0)}$ designed to create a desired maximum value of $AF(\theta)$ at $\theta = \theta_0$. Hence, the main lobe of the radiation pattern points towards $\theta_0$. Equation (41) can be rewritten as

$$AF(\theta) = \sum_{m=1}^{M} I_m e^{-j\beta x_m \sin(\theta_0)} e^{j\beta x_m \sin(\theta)} , \tag{42}$$

where $\beta$ is the wavenumber given by $2\pi / \lambda$. We can more compactly express the array factor as

$$AF(\theta) = \sum_{m=1}^{M} w_m^* e^{j\beta x_m \sin(\theta)} \tag{43}$$

where $w_m = I_m e^{j\beta x_m \sin(\theta_0)}$ is the complex weight applied to the $m^{\text{th}}$ element.

Suppose that we chose complex weights of $w_m = e^{j\beta x_m \sin(\theta_0)}$. Then $AF(\theta)$ will be maximum at $\theta = \theta_0$, and from (43) we have

$$AF(\theta)\Big|_{\theta=\theta_o} = \sum_{m=1}^{M} e^{-j\beta x_m \sin(\theta_0)} e^{j\beta x_m \sin(\theta)} \Bigg|_{\theta=\theta_o} = M . \tag{44}$$

Note that this equation will provide the overall pattern assuming that the individual elements are isotropic radiators. This result will be extended to cover the case of nonisotropic elements at the end of this section.

The maximum gain of a linear array of $M$ isotropic elements spaced uniformly half a wavelength apart has been shown to be [19] [50]:

$$G = M \,. \tag{45}$$

The exact same approach applies for a two-dimensional $M \times N$ omnidirectional array shown in Figure 32. Using the same approach as above, the two-dimensional array factor is given by

$$AF\left(\theta,\phi\right) = \sum_{m=1}^{M}\sum_{n=1}^{N} w_{mn}^{*} e^{j\beta(x_{mn}\sin\theta\cos\phi + y_{mn}\sin\theta\sin\phi)} \tag{46}$$

where $\beta$, as before, is equal to $2\pi/\lambda$, and $w_{mn} = I_{mn}\, e^{j\beta(x_{mn}\sin\theta_o\cos\phi_o + y_{mn}\sin\theta_o\sin\phi_o)}$ is the complex weight assigned to the $(m, n)^{\text{th}}$ element.

An array beamformer works by adjusting the magnitudes and/or phases of the complex weights to form a desired radiation pattern. If a sensor network knows a priori the direction of approach of the UAV, the number of sensor nodes to partake in beamforming and the positions of the participating sensor nodes, then the weights may be selected to point the main lobe towards the UAV.

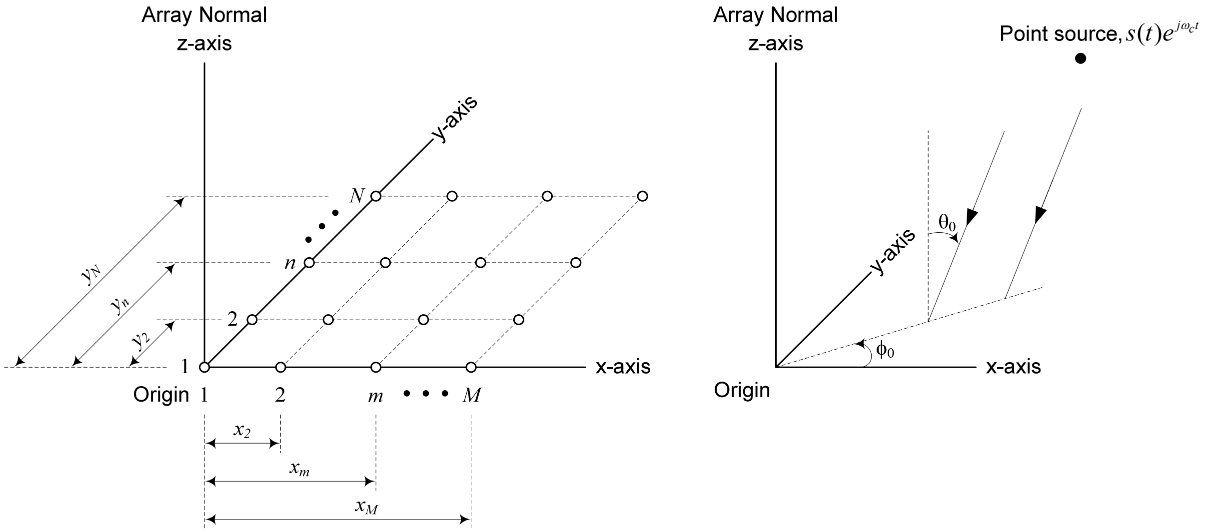


Figure 32. An M×N Planar Array

A conventional beamformer is a delay-and-sum beamformer with complex weights, and is shown in Figure 33, after [20]. The magnitudes and phases are selected so as to direct the beam in a desired direction and may additionally be chosen to produce one or more nulls in the radiation pattern in the direction of known interference signals. A number of techniques are available to determine the complex weights necessary to synthesize a desired radiation pattern using the beamformer of Figure 33. [20]

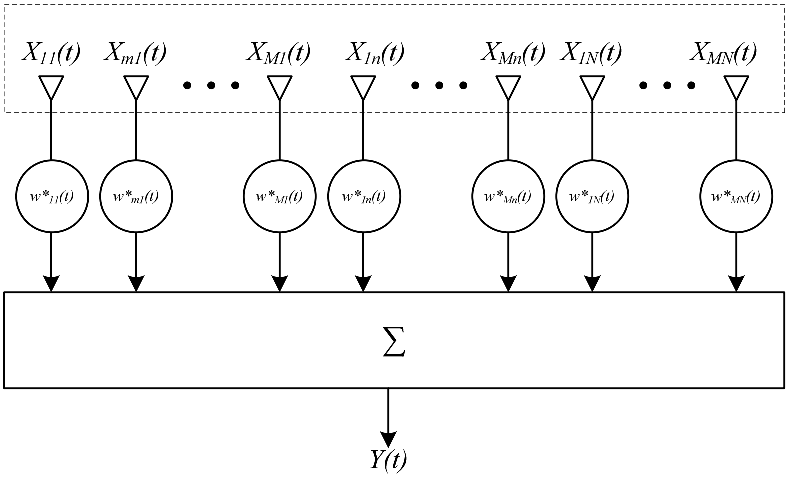


Figure 33. $M \times N$ Narrowband beamformer

As the UAV flies over the sensor network, the elements forming the antenna array will need to steer the beam in an adaptive manner. Figure 34 displays a block diagram for an adaptive beamformer using a least mean squares (LMS) algorithm. This algorithm assumes that the signal transmitted by the UAV, $s(t)$, is known and is used as a reference signal. The complex weights are estimated by subtracting the array output, $Y(t)$ in Figure 34, from this reference signal $r(t)$ (= $s(t)$) and minimizing the mean square error of $r(t)-Y(t)$, to steer the beam in the desired direction. For an overview of adaptive beamforming algorithms, see [20].

To summarize, then, military applications require that sensor nodes be small (so as to be covert) and low-cost (in order to be expendable). Furthermore, military applications presume that the physical hardware of a sensor node will not be serviced in the field, so when a node's battery fails, the whole node (not just the battery) must be discarded. A sensor node's small, one-time-use, low-power battery, coupled with the omnidirectional nature of a sensor node's antenna pattern, places severe restrictions on power management in order to extend the lifetime of the network as well as on a node's communication range. We propose having sensor nodes coordinate their transmissions to form a distributed antenna array that directs a beam towards a UAV, resulting in an increased energy-per-bit to noise power spectral density ratio, $E_b/N_o$, at the UAV (or an increased communication range for a fixed $E_b/N_o$).

Note that the discussion above has considered antenna arrays composed of isotropically radiating elements. More precisely, in discussing the radiation pattern of an array, we have considered the relative positions of the individual elements, but we have not considered that fact that the individual sensor node antennas—being practical radiators—do not radiate isotropically. The radiation pattern of the array will also, obviously, depend on the actual radiation patterns of the individual array elements, a factor we have not heretofore considered.

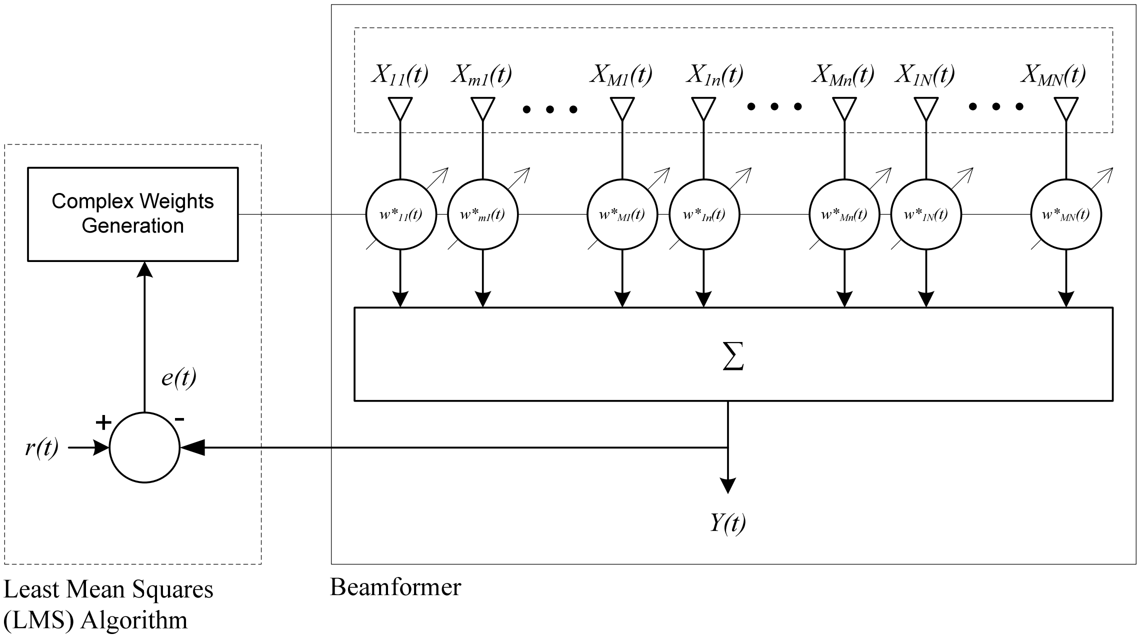


Figure 34. $M \times N$ Adaptive Beamformer

Fortunately, we are able to determine the *actual* array radiation pattern by, essentially, applying a correction factor to the previously determined pattern (i.e., the pattern that was calculated assuming the array was composed of isotropic radiators). Let the actual patterns of the individual array elements be all alike and denoted as $g(\theta,\phi)$, and let the array factor (calculated in the manner described above) be given by $AF(\theta,\phi)$. The *principle of pattern multiplication* states:

> The electric field pattern of an array consisting of similar elements is the product of the pattern of one of the elements (the element pattern) and the pattern of an array of isotropic point sources with the same locations, relative amplitudes and phases as the original array (the array factor). [47]

Using the notation above, the complete pattern of the antenna array, $F(\theta,\phi)$ is thus given by [19]:

$$F(\theta,\phi) = \kappa g(\theta,\phi) AF(\theta,\phi)$$

where $\kappa$ is a normalizing constant.

We note that in the remainder of this dissertation we consider all of the elements to be isotropic radiators. The extension of the results to consider actual radiators (determined by applying the principle of pattern multiplication) is left for future research.

## C. PROPOSED APPROACHES TO POWER MANAGEMENT

Sensor nodes will coordinate their transmissions to perform beamforming, with the aim of minimizing energy expenditures within a sensor node. We next frame the foregoing discussion within its broader overall context: power conservation in sensor networks.

We now consider four approaches to power management in sensor networks. We first look at techniques that rely in some way on the judicious application of power within the physical components of the sensor nodes themselves. Second, we examine methods that reduce power consumption by reducing the amount of data that must be distributed by the network. Third, we explore power-efficient approaches to handling scenarios in which the sensor network becomes congested. Finally, we consider methods for distributing sensor data throughout a sensor network while expending minimal power.

### 1. Adjusting Power to the Sensor Node Components

The most obvious way to conserve power is to turn all of the sensor node's components off. In this case, the sensor node provides limited functionality—perhaps no functionality—but its battery will endure for a long period of time. On the other hand, if all the components of the sensor node are continually powered on, we have full functionality but only for a limited time until the battery is depleted. Thus, the proper approach lies between these two extremes and calls for the judicious application of battery power to the various sensor node components only as needed to accomplish a desired end. Referring to Figure 1, the components of the sensor node that must frequently be powered for normal operation are the sensors, the processor and the transceiver.

We assume that the sensors and processor can each be in one of two states: ON or OFF. (Actual processors may have a number of intermediate states between ON and OFF; we model the processor as a two-state device to simplify the discussion.) Note that when we say a component is OFF, we allow that some minor control and monitoring circuitry might still be active, receiving a small amount of power. For example, when the sensors are in the OFF state, an associated clock and timer might still be powered to periodically wake up the sensors at specified time intervals.

The transceiver, containing the receiver and the transmitter, requires more careful consideration. A power, $P_R$ , is consumed by the receiver whenever it is energized while a power, $P_T$ , is consumed by the transmitter whenever it is energized. During transmission, an additional power, $P_S$, the radiated transmit power of the signal, is consumed. (Note that $P_R$ and $P_T$ are considered fixed quantities, while $P_S$ might be adjustable, in which case it is set as low as possible while still achieving successful communications.) Some researchers contend that for the low radio ranges employed in typical sensor networks, the oscillators and mixers used for up and down conversion dominate the radiated transmit power (i.e., $P_S$ is negligible) and transmission and reception have roughly the same energy cost (i.e., $P_R \approx P_T$)[16]. Others—while agreeing that $P_S$ is negligible for low-range transmissions—contend that $P_R$ is 2 to 3 times greater than $P_T$ since more circuitry is required to receive a signal [51]. In any event, the transceiver can be in one of three states: OFF (some minimal power is still consumed), RECEIVE ONLY ($P_R$ is consumed) or ON, where the latter state implies the ability to transmit as well as receive (either $P_R + P_T$ or $P_R + P_T + P_S$ is consumed). (Note that a TRANSMIT ONLY state is not useful). Thus, a sensor node may assume 12 possible states, categorized in Table 1.

| | **SENSORS** | **PROCESSOR** | **TRANSCEIVER** |
|---|---|---|---|
| ***EXTREME STATES*** | | | |
| **State 1 (Fully Off)** | OFF | OFF | OFF |
| **State 2 (Fully On)** | ON | ON | ON |
| ***SENSORS OFF - AWAITING INSTRUCTIONS*** | | | |
| **State 3** | OFF | OFF | RECEIVE ONLY |
| **State 4** | OFF | ON | RECEIVE ONLY |
| ***VIGILANT SENSOR STATES*** | | | |
| **State 5** | ON | OFF | OFF |
| **State 6** | ON | OFF | RECEIVE ONLY |
| **State 7** | ON | ON | OFF |
| **State 8** | ON | ON | RECEIVE ONLY |
| ***PURE ROUTER STATE*** | | | |
| **State 9** | OFF | ON | ON |
| ***UNUSED STATES*** | | | |
| **State 10** | OFF | ON | OFF |
| **State 11** | OFF | OFF | ON |
| **State 12** | ON | OFF | ON |

Table 1. Sensor Node States

States 1 and 2 represent the two extremes: all components OFF or all components ON, respectively. If the sensors are turned off, as they are in State 1, there exists a distinct possibility of missing an event. Consider a scenario wherein, in an effort to conserve power, nodes are normally off (State 1), but are momentarily woken up (placed in State 2) at specified time intervals, say, every ten minutes, to take a quick "snapshot" of the environment. Such a strategy might be effective if the sensor is intended to measure slowly varying phenomena, such as ambient air temperature. If, on the other hand, a sensor is placed on a roadside to detect passing trucks, the event of interest may transpire in its entirety during the sensor's ten-minute dormant period and thus be missed.

In some applications, it may not be necessary for sensors to continuously sense the environment all the time. In States 3 and 4 the sensors are OFF and the transceiver is RECEIVE ONLY so that the node can receive an instruction to turn the sensors on. As an example of how these states might be of use, consider the scenario depicted in Figure 35, which shows a sensor field deployed to allow soldiers to safely transit through an area. To conserve power, all nodes are in State 3 (sensors and processor OFF, transceiver in RECEIVE ONLY). At a later time, the soldiers, deciding they need to transit via the left portion of the terrain, deploy a UAV, which flies over the transit area, instructing the nodes within this region to shift to State 2 (fully on). Sensor nodes outside the transit area remain in State 3, conserving power.

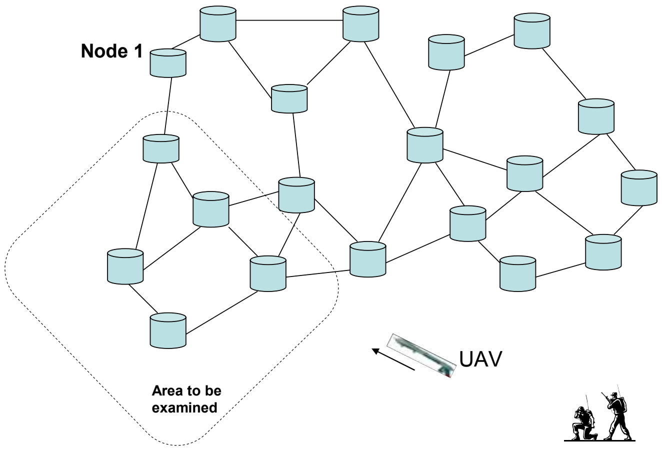


Figure 35. A Deployed Sensor Network

States 5-8 are the vigilant states: the node's sensors are ON, but the node's transceiver is set to either OFF or RECEIVE ONLY (i.e., the transmitter is off). As an example of how these states might be of use, consider again Figure 35, but now suppose all nodes are in State 6 (or 8), the vigilant states wherein a node's transceiver is set to RECEIVE ONLY. If the node labeled Node 1 alerts to a detection, it will shift its transceiver to ON, and send a message to its neighboring nodes to initiate the process of delivering this information to the ultimate end-users. As action is taken to deliver the message, multiple sensor nodes in the network will receive messages and will turn on their transmitter to assist in the delivery process. Note that the sensor nodes *contend* for the channel when they transmit; it is possible that transmissions from different nodes may collide since channel resources are not reserved or assigned in advance. An example of a sensor network wherein the transceiver in each sensor node is maintained in the RECEIVE ONLY state when idle is discussed in [52].

An obvious drawback of States 6 and 8 is that the receivers are continually powered, monitoring the channel at all times, even during extensive idle periods when no communication occurs. In States 5 and 7, on the other hand, power can be conserved by turning the sensor node's transceiver OFF (while still leaving the sensors ON). While saving power, this approach makes communication problematic. Specifically, referring again to Figure 35, but with all nodes in State 5 (or State 7), if Node 1 alerts to a detection, how is it to convey this information to the ultimate end-users if its neighboring nodes—one of whom will be required to route the message—all have their transceivers OFF?

A solution to this problem—allowing sensors to remain continuously vigilant while the transceiver operates at a very low duty-cycle—is to ensure that sensor nodes periodically turn on their transceivers at coordinated times. For example, in the sensor network described in [29], as newly deployed sensor nodes first wake up, they pass messages to discover who their neighbors are. Nodes then agree with their neighbors to fixed transmission and reception schedules, establishing a distributed Time Division Multiple Access (TDMA) scheme. To avoid potential channel conflicts, each transmission and reception schedule employs a randomly-chosen frequency selected from a large pool of available frequencies. Once the schedules are set, the sensor nodes turn off their radios—for reception as well as transmission—when no information is scheduled to be sent or received. The protocol is completely distributed and results in a network topology that is flat. While such schemes may be successfully employed for stable networks, some researchers believe that scheduling schemes (such as distributed TDMA schemes) may prove very inefficient if the network proves to be dynamic.

Further complicating matters, it is worth pausing at this point to note that even if we know that a component, say the transceiver, will not be needed for a certain period of time, we might save power by leaving it ON [51]. As an example, suppose that the transceiver is to be maintained in one of two states: RECEIVE ONLY or ON. If our goal is to conserve power, should we maintain the transceiver in the RECEIVE ONLY state, until it comes time to transmit? To quantify matters, consider that our transceiver has just completed a transmission at time $t = 0$, and we are to decide if our transceiver should be maintained continuously in the ON state (where, say, the power consumed is 200 mW and we can neglect $P_S$), or placed in the RECEIVE ONLY state (where the power consumed is 150 mW) until the next transmission is required. Let the time required to transition between these two states be 500 msec. Suppose that, as events transpire, the next time that a message must be transmitted happens to occur at $t = 1$ sec, and the transmission itself requires 100 msec. If the transceiver was left continuously ON, the total energy expended in transmitting this next message is 220 mJ while if the transmitter were turned off and turned back on when needed, the total energy expended in transmitting the next message is 270 mJ (see Figure 36). Thus, in this scenario, we would have attained an 18.5 percent improvement in energy dissipation by leaving the transmitter continuously ON. The conclusion: Simply turning off an idle component may actually waste energy (since state transitions take time, and power is wasted during transitions), and stochastic information (in this case, statistical information about the time between transmissions) might be employed to intelligently decide when the sensor node should be placed in different states for power conservation. Such computations might be performed by the processor based on observation of events.

Note that, apart from power-conservation considerations, placing the transceiver in the less capable state causes a latency problem; the fact that the transceiver must take time to transition to the more capable state means, of necessity, the sensed information will arrive at the end-user with an additional delay.

The foregoing discussion concerning the transceiver also applies to the processor. The strategy of turning off the processor when it is not actively in use may actually cost us more power and will always incur an additional delay in the time it takes for the sensed data to arrive at the ultimate end-user. The operation of the processor, though, can be adjusted in other ways to conserve power. For instance, even when the processor is left in the ON state, its energy consumption can be reduced by dynamic adjustment of the operating voltage and clock speed to match computational load [53]. Additionally, the processing of sensor data can be done in a manner that trades power against accuracy. Recalling that analog sensor readings must be converted to digital data, the processor can change the number of bits used to represent the analog data and can change the manner in which the digital data are then reconstituted. For example, if power is not a concern, the processor may choose to represent each analog sample by 16 bits (i.e., as one of 65,536 possible values); as power becomes scarce, it may be decided that less accurate 8-bit representations will have to suffice. As a second example, consider a sensor network where sensors are cameras that provide imagery data. If power is not a concern, the processor may process the sensor data using a lossless compression scheme, allowing perfect image reconstruction. As power becomes scarce, the processor may shift to a lossy compression scheme, accepting a less accurate image that requires far fewer bits. As a final example, consider a case where an application requires that a Fast Fourier Transform (FFT) be performed on sensor data. If power is not a concern, the processor may use a 512-point FFT; as power becomes scarce, the processor may shift to using a 128-point FFT [54].

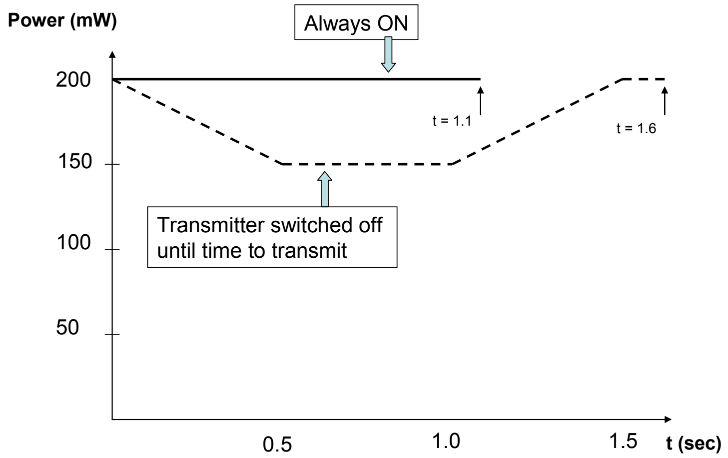


Figure 36. Comparison of Two Different State Transition Schemes

Returning to Table 1, in State 9 the sensor is acting strictly as an intermediate router, supporting one or more traffic flows for other sensor nodes. Note that if the transceiver is ON, the sensor node's processor must be ON as well, and, similarly, it does not make sense to have the processor ON and all other components OFF; thus States 10, 11 and 12 do not occur.

By describing the different node states, we have foreshadowed a few of the many competing concerns that a network designer must address. It should also be clear that the application of the sensor network will often dictate the appropriate state for the sensor nodes. Is it important that the sensor nodes provide a continuous stream of readings? Or, is it appropriate that the communications are triggered only when a detection level exceeds a threshold? Or, is it more appropriate that our sensor nodes not communicate at all, until queried by an end-user (this was the original example associated with Figure 35, where soldiers deployed a UAV to activate a portion of the sensor network).

We close this subsection by considering $P_S$, the radiated transmit power of a transmitted signal (recall that this power exists in addition to $P_T$ , the power consumed by the transmitter whenever it is energized). Clearly, transmissions should only occur when necessary, and then at the minimum power necessary. On the one hand, $P_S$ should be kept as low as possible to conserve energy while, on the other hand, for communication between sensor nodes to be successful under the prevailing channel conditions, $P_S$ must be sufficient to assure that enough residual energy arrives at the receiving antenna to achieve a specified Bit Error Rate (BER).

One way to reduce $P_S$ is to use Forward Error Correction (FEC) coding. For a fixed channel transmission rate, employing an FEC code will reduce the amount of energy required at the receiver to meet a specified BER, which, in turn, means that a lower $P_S$ can be employed at the transmitter. This desirable reduction in the transmission power comes at two costs:

- Increased transmission and reception time. FEC codes add redundant bits to the data, with the number of added bits amounting to a fraction (often ≥ 1) of the number of original data bits. Thus, while the uncoded transmission requires a certain transmission power for a certain duration, the coded transmission requires a lower transmission power, but for a longer duration. This also means that the power that must be applied to energize the receiver and the transmitter ($P_R$ and $P_T$) is applied for a longer duration, and these two power expenditures are not reduced by the presence of coding.
- Increased processor energy. More energy must be used by the processors in the transmitting and receiving sensor nodes in order to decode and encode the data stream.

Consider an uncoded transmission that requires a transmit power of $P_S$ for a time duration $T$. In this case, the energy required for the transmission (assuming the transceivers for both the transmitting and receiving sensor nodes are ON) is

$$E_U = \left(P_S + 2P_T + 2P_R\right)T \ .$$

Suppose that we can use an FEC code that would allow us to reduce the transmission power to $\zeta P_S$ $\left(\zeta < 1\right)$ while increasing the transmission time to $\delta T$ $\left(\delta > 1\right)$. The energy required for this coded transmission is

$$E_C = \left(\zeta P_S + 2P_T + 2P_R\right)\delta T + E_{E/D} \ ,$$

where $E_{E/D}$ represents the additional energy that must be used by the processors in the transmitting and the receiving sensor nodes to encode and decode the data stream. A necessary condition for conserving power by employing this code (i.e., $E_U > E_C$) requires that $\delta\zeta < 1$ and whether power is actually conserved depends on the particular values of the remaining parameters. Generally, coding will be useful for high-power transmissions in which $P_S$ dominates the sum of $P_T$, $P_R$, and the additional energy used by the processors for encoding and decoding. Specifically, given $\delta\zeta < 1$, we will conserve power by employing an FEC code provided the original (uncoded) transmit power satisfies

$$P_S > \frac{2(\delta - 1)}{(1 - \delta\zeta)}(P_T + P_R) + \frac{E_{E/D}}{T(1 - \delta\zeta)}.$$

For a detailed study of the energy efficiency of various coding schemes for a specific sensor node architecture, see [55]. For an analysis of the energy per bit required to encode and decode various BCH (Bose, Ray-Chaudhuri, Hocquenghem) codes, see [56].

### 2. Processing versus Transmission

As a simple example illustrating how processing might be employed to conserve power, consider Figure 37. Five ground-based sensor nodes (labeled 1-5) have been positioned near a highway to detect approaching vehicles. Any detection should be communicated to the tank (labeled 6). (A line joining two nodes indicates that the nodes are within communication range of each other.) Suppose that a truck approaches within the sensing range of Nodes 1, 2 and 3 under two scenarios.

- ***Scenario one***: Node 1 detects the truck, and alerts the tank in a transmission that follows the route 1−4−5−6. Similarly, Nodes 2 and 3 also detect the truck, and their communications follow the routes 2−4−5−6 and 3−4−5−6, respectively. Note that three separate transmissions occur over links 4−5 and 5−6, and a total of nine transmissions take place.
- ***Scenario two***: Node 4 receives the transmissions from Nodes 1, 2 and 3, as before. However, now some processing is done: Node 4 sends a single transmission to Node 5, saying, in essence: “A detection has occurred at Nodes 1, 2 and 3.” Similarly, a single transmission is sent from Node 5 to the tank. In this second scenario, only five transmissions occur.

By extending this simple example, it is noted that, in a network consisting of hundreds or thousands of nodes, where each communication entails numerous hops, the reduction in the number of transmissions can be immense. Note that the savings achieved in the second scenario are not free: Some processing must be done at Node 4, and this processing also expends some of the limited energy that Node 4 possesses. But communication costs much more than data processing. An example previously mentioned is worth reiterating: For a given scenario [29] with a ground-to-ground transmission, transmitting 1,000 bits of data a distance of 100 m expended 3 J of energy, the same amount of energy required for a modest processor to execute 300 million instructions. Another example is provided in [57]: Sending 100 bits a distance of 100 m expends 10 µJ, but executing a single 32-bit instruction takes 0.06 nJ, a factor of 100,000 times less energy. As a general rule, then, we would like to limit the amount of radio transmission at the expense of more processing, and thus it is desirable for processing (such as data collection, aggregation, fusion and compression) to be performed as close to the detection point as possible.

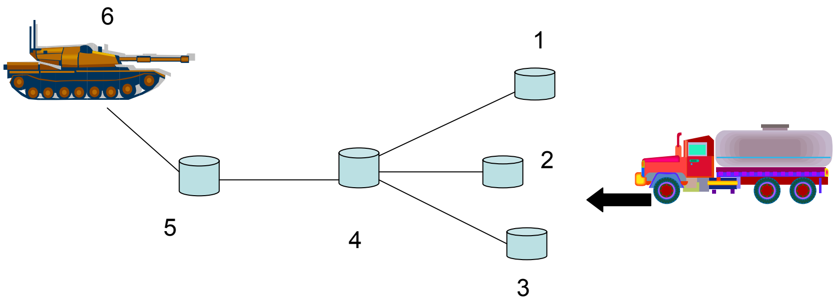


Figure 37. Simple Sensor Network

Processing can also be employed to limit the amount of purely redundant data to be transmitted. Returning to Figure 37, perhaps Node 4 “knows” that Nodes 1, 2 and 3 are very close together, and a detection occurring at any one of the three sensor conveys all the information that is important (e.g., the three sensors were placed by the road for redundancy, in case some sensor nodes were expected to be damaged during deployment). In this case, a single transmission to Node 5, saying, in essence: “A transmission has occurred at Node 1” is all that is needed.

It has been noted [58] that the greatest gains in data aggregation are obtained when the sources are close to each other and far away from the sink; an optimal way to take advantage of data aggregation would be to route data using a minimum Steiner tree containing all source nodes and the sink node. Unfortunately, determining a minimum Steiner tree on a set of nodes is an NP-hard problem [59].

### 3. Congestion Control

Consider Figure 38, which depicts a network of six sensor nodes used by a soldier in the field. Suppose Nodes 1, 4 and 5 each alert to a detection. The information from Node 1 is sent to the soldier via path 1−2−3−4−6−soldier while the information from Node 5 is sent along the path 5−4−6−soldier and the information from Node 4 follows the path 4−6−soldier.

Suppose that the link connecting Nodes 4 and 6 becomes congested, and one of these three data streams must be dropped. Whose data (that initiated by Node 1, Node 4 or Node 5) should be discarded? Power conservation would direct that Node 4's data be dropped because more energy has already been invested in Node 1 and Node 5's data. For instance, consider Node 1: power is expended in transporting its data to Node 2, then to Node 3, and then to Node 4. To discard Node 1's data at Node 4 would mean that all the energy that has already been invested is wasted.

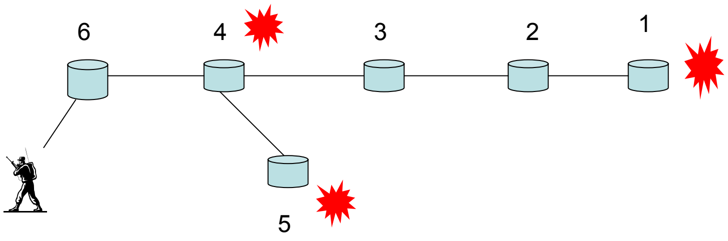


Figure 38. Congestion Control Example

This notion is incorporated into the congestion control scheme studied in [52] in which each sensor node regulates congestion by employing an additive increase multiplicative decrease approach, akin to that used by TCP [60], but with a notable exception: a preference is given to route-through traffic. In the scenario above, when Node 4 detects congestion, it would reduce the transmission rate of its own originating data by a factor $\beta$ (where $0 < \beta < 1$), but would reduce the route-through data by $1.5\beta$, thus giving a preference to the traffic coming from Nodes 1 and 5 over its own traffic. In a like vein, once congestion alleviates, Node 4 will increase the transmission rate of route-through traffic by an additive value $\alpha$ while increasing the transmission rate of its originating data by only $\alpha / 3$ (since, in this example, a total of three nodes are vying for the 4-6 link).

In some applications, this power-conservation technique would be unwise; it is easy to imagine military scenarios in which we would want to bias in favor of transmissions closer to the sink. If, in Figure 38, the sensor nodes are used to detect anthrax, the soldier will be most interested in the information provided by the closest sensor node (Node 4), and will not be altogether pleased with an algorithm that restricts information about his local environs in order to provide information about distant locales. So, it is sometimes the case that a compromise must be reached between risk (in this example, health risk) and reward (power savings).

### 4. Data Delivery and Routing

In Chapter II, we discussed in detail the use of flooding as a routing mechanism in sensor networks. We now describe power control in sensor networks that rely on *point-to-point routing* in which an attempt is made to route data via an efficient path from source to sink, where, for sensor networks of interest here, efficiency is measured in terms of energy conservation. Generally, these algorithms will be significantly more complex than flooding and will entail the exchange of control messages needed to determine the network topology and the state of the network's nodes—information that must then be employed by the routing algorithm, with the results stored by individual nodes. The goal is that these power-consuming activities are more than made up for by the power savings gained later when the actual data is transmitted via a well-chosen route.

Point-to-point routing algorithms can be classified by when the optimal routes are computed: proactively in advance (so that immediate transmission can commence as soon as the need arises) or reactively "on the fly," as the need to transfer data arises.

If routes are precomputed, sensor nodes store results in routing tables, the size of which generally scales with the size of the network. Establishing optimal routing in advance is particularly useful in stable networks in which large amounts of data are transferred with some regularity. In such cases, a big up-front expenditure of energy (i.e., the transmission of control packets and processor computations) will pay constant dividends for a long time. Sensor networks, though, may be highly dynamic. First, sensor nodes may fail due to battery exhaustion. Second, even sensor nodes with sufficient energy reserves may choose to operate their transceivers at a low duty cycle, and thus may cause the pre-selected routes chosen by other nodes to fail. Third, environmental fluctuations may affect link quality. In such cases, ideal routes will only remain ideal for short periods of time. Indeed, the network may precompute routes that become invalid even before they are used, and attempting to keep track of good routes for a dynamic network may require the exchange of a barrage of control information. Even in a stable sensor network, precomputing routes may waste energy: a node may determine in advance the ideal route from it to each of the thousands of other nodes in the stable network, but, as time goes on, it may just so happen that the node never detects an event, and thus never needs to transmit anything. An example of a routing protocol that relies on precomputed routes is Destination-Sequenced Distance Vector (DSDV) [61].

If routes are not computed in advance, then sensor nodes do not need to maintain large routing tables (obviating the need to transmit control data to maintain such tables). Now, though, each specific transmission will require that control information be exchanged and computations be performed to select an efficient route. If sources communicate regularly with specific destinations (and the network is stable), then it would be more efficient to determine a good route once (in advance) rather than invoking a routing algorithm repeatedly. An example of a reactive routing protocol is the Dynamic Source Routing (DSR) protocol [62]. For a recent review of point-to-point routing protocols for ad hoc networks, see [63].

Regardless of whether a point-to-point routing algorithm is proactive or reactive, once it comes time to establish a route, there are several ways that energy and power can be incorporated as metrics into the route selection process.

It would seem clear that we should not route messages through nodes that do not have great energy reserves remaining. Simply using shortest-path routing may consistently route messages through a small number of nodes (that happen to lie on many shortest paths), quickly exhausting them. Consider a scenario wherein sensor nodes periodically exchange (via control messages) information about the state of their respective batteries, i.e., their power remaining. Then, an algorithm can select a route that uses the nodes with the largest power reserves. Note that it would be unwise to simply sum the power available at all nodes on the prospective route, and select the route that has the maximum (summed) power. In Figure 39, Node A has two alternative routes available to send data to Node B. Although the power reserves along Route 1 total 50 units, while those along Route 2 total only 10 units, Route 1 would not be a good choice since sensor Node D (whose reserve is only 1 unit) may have its battery exhausted by taking part in the transmission. It would then not be available for any other purpose such as relaying data or gathering information with its own sensors. The underlying goal of energy conservation must be kept in mind: We wish to maximize the lifetime of the network and its sensor nodes.

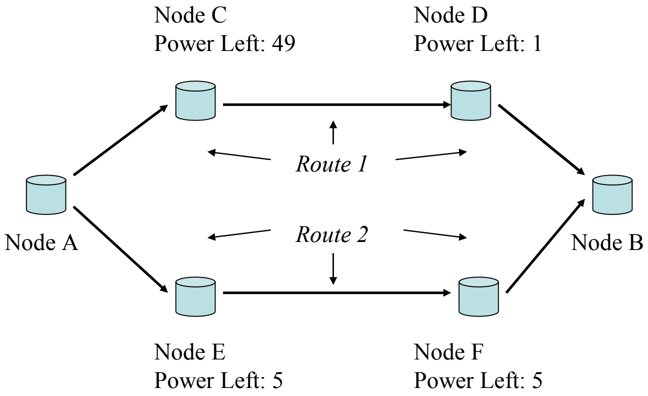


Figure 39. Route Selection in a Simple Network

So, an alternative route selection algorithm would avoid sensor nodes with low energy reserves. One proposal [64]: For each potential route $j$, locate the node which has the minimum remaining power available, and call its remaining power $p_j$. Then select the route that has the maximum $p_j$ value (i.e., we select the route that has the maximum value of the minimum power remaining). This ensures that nodes with low battery reserves are not used for routing, allowing them to remain active as sensors for as long a time as possible.

In selecting the most energy-efficient route, we have (up to this point) only looked at half of the problem: we have considered the power available at sensor nodes. The other major component that must be considered is the energy required to transmit along a route. In Figure 40, sensor Node A has two alternative two-hop paths to choose from to transmit to Node B; based only on the remaining node energy, we would select Route 1 since its relay node has greater energy reserves. But Figure 40 is also annotated with the power required to effectuate transmission over each hop. Since each hop on Route 1 requires 8 units of power, while each hop on Route 2 requires 1 unit of power, Route 1 would be a poor choice in this scenario. A reflexive change in our strategy—say, adopting a route-selection strategy that selects the route that requires the least expenditure of energy—would similarly be flawed [63]. The point: Routing strategies require careful consideration of power available at various nodes as well as the power required to transmit over various links.

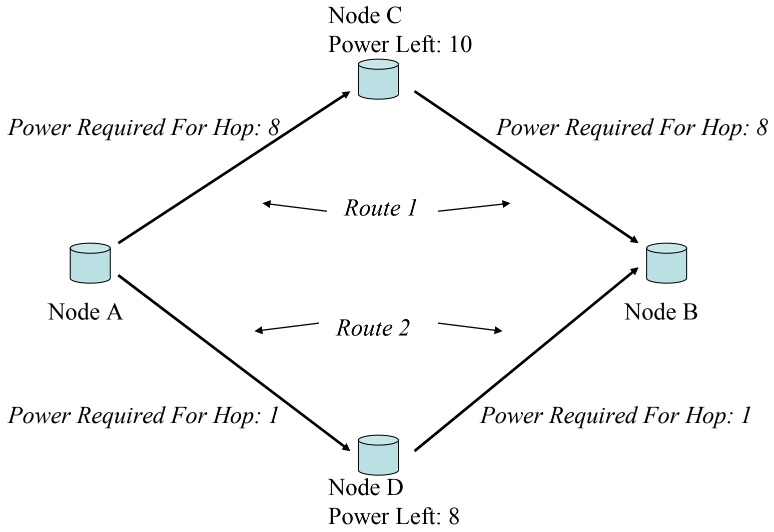


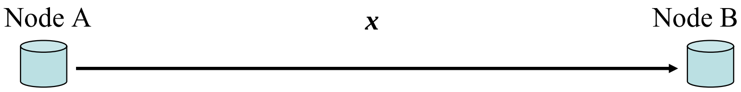
Figure 40. Route Choice Based on Energy Considerations

We close this section by considering a further nuance that has been dealt with in the sensor-network literature: Is it more energy-efficient to send data over one long hop, or to send data over several shorter hops? Assume sensor Node A needs to transmit a block of data to sensor Node B, which is separated by a distance *d*. Node A's transmit power as a function of the distance *d*, $P_S\left(d\right)$, may be estimated using the log-distance path loss model as

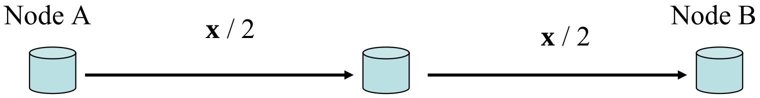
$$P_S\left(d\right)=k\,d^n$$

where *n*, the *path loss exponent*, is equal to two if propagation is in free space, but typically ranges between 2.7 and 5 in outdoor radio environments [15]. The constant *k*, the *path loss coefficient*, accounts for the transmitting and receiving antenna gains, the wavelength of the radiated signal, the power required at the receiver to attain a specified BER, and losses not related to propagation (e.g., filter losses). Suppose that a sensor node is to transmit data to another node, separated by a distance *x*, using a single hop, as shown in Figure 41(a). Assuming the transceiver in each node is ON, the total power consumed by the transceivers is

$$2\,P_T+2\,P_R+k\,x^n\ .$$



(a) Communication by Single Hop



(b) Communication by two hops

Figure 41. Single Hop versus Multihop

Now suppose that an intermediate sensor node, lying on the path between nodes A and B, is to be used as a relay, as shown in Figure 41(b). Now the total power consumed by the transceivers is

$$2\left(2P_T+2P_R+k\left(\frac{x}{2}\right)^n\right) = 2\left(2P_T+2P_R\right)+\frac{k\,x^n}{2^{n-1}}.$$

So, when deciding if sending data over short hops will be more energy efficient than sending data over one long hop, two competing effects must be considered. On the one hand, as we employ more intermediate relay nodes, we must expend additional power to energize the additional transceivers; each additional hop adds $2P_T+2P_R$ to the required power. On the other hand, as we employ intermediate relay nodes (lying near the direct path between the source node and ultimate destination so that the total distance the signal travels remains relatively unchanged), the total power required for transmission decreases; in the simple example above, the power required for transmission was reduced from $k\,x^n$ (one hop of distance $x$) to $\frac{k\,x^n}{2^{n-1}}$ (two hops of total distance $x$).

Note that if it were not for distance-independent overhead (represented by terms such as $P_T$ and $P_R$), to minimize power we would, ideally, use an infinite number of hops over the smallest possible distances [65]. When we do consider $P_T$ and $P_R$, provided we can determine the path loss exponent and coefficient ($n$ and $k$), and if we have the ability to place any number of equally spaced relays on the straight line between transmitter and receiver, then we can calculate the optimal number of hops to use [66]. Of course, in actual sensor networks, it will be very unlikely that potential relay nodes will lie on the direct path between a transmitter and a sink, and, as a consequence, employing relay nodes will necessarily increase the total distance the communication signal must travel. Figure 42 shows two potential communication paths between a transmitting sensor Node A and a sink Node B: a one-hop path (of distance *x*) and an eight-hop path, where each hop is of distance *x*/2. If the path loss exponent is less than three, the multihop path will consume more transmission power even if the distance-independent overhead is negligible. A method for analytically determining, based on the positions of a destination node and a potential relay node, when using the relay node will consume less total power than transmitting directly, is presented in [67].

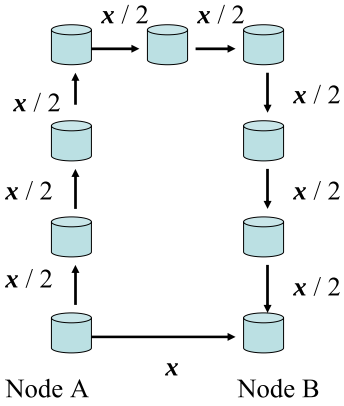


Figure 42. Single-Hop Path versus Eight-Hop Path

A recent algorithm [68] attempts to maximize sensor network lifetime by solving a linear programming problem. The algorithm attempts to maximize the amount of information transfer between sources and destinations by using lowest-cost routing, where the cost of each edge is a combination of the amount of energy required to span the distance and the residual energy levels at the two end nodes. The authors note, however, that their proposed solution is not scalable and may not be suitable for large networks. Another approach [69] suggests routing by choosing the path which expends the minimum power (ignoring the power remaining in each node) until some node reaches a threshold of remaining energy, and at that point the route that has the maximum minimum-power battery. An algorithm that attempts to find a route from source to destination that minimizes the power consumed and maximizes the minimum residual power after message transfer is complete is presented in [70].

Two further points merit brief mention. First, implementing multihop transmission will require a more complex protocol than simple direct transmission, and this increased protocol complexity will require that more energy be consumed by the processors in the sensor nodes that take part. Second, using a multihop path over a more direct path will increase the end-to-end delay in the transmission of the data, both because of the increased overall distance of the multihop path and the various delays (e.g., buffering, processing and queueing) incurred at each relay node.

All of the aforementioned algorithms use the node's addresses for routing. This could be a problem if the network is forced to answer questions such as "Give me data if the temperature is over 30 °C," or "What is the temperature at location at coordinates ($x$, $y$)." In this case, our interest is in the data, and the addresses of the nodes that match this query are not known. Such a query would need to be flooded to every node in the network to determine which nodes satisfy the conditions. In many cases, sensor networks are *data-centric*; that is, we are interested in the data and have no interest in the identity (addresses) of the nodes that provide the data. Protocols have been developed that route data based on their content and user-interests, not the addresses of nodes.

One such algorithm is *directed diffusion*. [71] A potential sink node propagates a message stating its interest in specific data. This interest message propagates to all nodes by flooding. Each node that receives an interest message caches it and records the immediately-preceding neighboring node from which the interest message was received. Once a node detects data that correspond to a cached interest message, it sends the data to the sink using the preceding node. Note that the source may send the data to multiple neighboring nodes if the interest message was received from multiple neighbors (recall that the interest message was propagated by flooding). Each node that receives this data message checks its interest cache and routes the data message in the same manner. The data propagates back to the source along the reverse path followed by the interest message. This node in turn routes the message back, and so forth. Eventually the sink receives the data. Now, since the interest message was flooded, it would be expected that the received data will be received by the originator of the interest message along a multiplicity of routes. The sink notes the neighbor that first provided the data; this node is assumed to be the node on the empirically determined best (i.e., lowest-delay) route. Thus, this node is sent a reinforcement indication (to keep sending messages) while the other nodes are sent suppression messages. This neighbor now does the same action, etc., until a single route from source to sink is established. A key energy conservation technique is that nodes can aggregate data in-network before transmission, minimizing network traffic. For details, see [70].

## D. OPTIMIZING THE SIZE OF AN ANTENNA ARRAY

In light of the foregoing, we now analyze the optimal number of sensor nodes that should participate in a beamforming process in order to minimize the energy expended by nodes in a sensor network.

We briefly review the sensor network employment scenario: After deployment, the sensor nodes awaken and partition themselves into non-overlapping clusters. Each cluster is managed by a fixed single node designated as the clusterhead (CH). A specified cluster, termed the transmit cluster, is selected from among the many clusters to act as the transmission point for the sensor network. Any sensed data originating anywhere in the network are routed to the transmit cluster's CH (hereafter termed the transmit CH), consolidated with any data already collected and prepared for transmission to the UAV. The transmit CH organizes a subset of its cluster nodes into an antenna array and calculates the phase offsets to be applied to the otherwise identical transmissions of the participating nodes in order to form a radiation pattern that concentrates the power into a narrow beam.

As the number of participating nodes, $N$, increases, the gain of the distributed antenna, $G$, can be increased. As noted in (45), under certain conditions the gain of an array of $N$ isotropic elements can be made to equal to $N$. Thus, increasing $N$ for a fixed energy per bit to noise power spectral density ratio ($E_b/N_0$) at the UAV allows for a significant reduction in amount of power required of each node in transmitting the data packet.

Because these duties of the transmit cluster are power-intensive, the role of transmit cluster is shifted to different clusters as time progresses in order to evenly distribute the energy load and extend the lifetime of the network. We presented an algorithm to accomplish this in Chapter II.

For this plan to work, the narrow transmission beam must be directed such that the UAV falls spatially within it. Military applications must presume that the network does not know *a priori* where the UAV is, nor does the UAV know the direction from which the sensor network has aimed its transmission beam. Techniques that an antenna array can employ to estimate the direction of the UAV are complex, and usually task the UAV with transmitting a known reference signal while the antenna array steers a very narrow beam continually in space, analyzes the spatial spectrum as a function of position, and selects the direction that yields the highest power as the direction of the UAV. This requires that the transmit CH continually calculate different sets of amplitudes and phases and transmit these to the participating nodes comprising the array, and these nodes, in turn, will have to continually send their receptions to the transmit CH for analysis. When energy constraints require that transmit cluster be shifted, all relevant data collected up to this point must be passed to the new transmit CH, or the entire process of finding the UAV must begin anew. All these operations involve a large number of radio transmissions among sensor nodes, expending scarce battery resources.

In Chapter III, we proposed that the power-constrained sensor nodes should not be burdened with finding the highly capable UAV; instead, the burden of aligning the transmission beam is shifted to the UAV. The transmit CH organizes nodes into an antenna array and calculates the phase offsets needed for the resultant beam of the desired gain to point straight up (at 0° elevation). These values, calculated only once, are passed to the participating nodes, but the distributed antenna then “stands down,” and the transmit CH functions as an omnidirectional antenna again, waiting for the UAV to find it. The UAV, meanwhile, travels over the region transmitting a known reference signal, pointing its transmission beam straight down. If the transmit CH should fall within the UAV's beam, the transmit CH will then send the data stream (containing the consolidated data gathered by the sensor network) to the elements in the antenna array, and these elements will, in turn, transmit the signal using the precomputed phases. If the antenna array's gain is close to the UAV's antenna gain (and employs a similar conically shaped beam), then whenever the transmit CH falls within the UAV's transmission beam, the UAV will fall within the sensor network's transmission beam.

In summary, then, a subset of sensor nodes is tasked with forming a distributed antenna array and performs only a single set of computations necessary to aim the resulting beam straight up. But, the question remains, what is the optimal number of nodes that should partake in the beamforming process in order to minimize the energy expended by the sensor network? This is the question we presently examine.

### 1. Beamforming Algorithm

The energy per bit to noise power spectral density at the UAV is given by (37), repeated here for convenience:

$$\frac{E_b}{N_o} = \frac{P_t G_t G_r}{kT_S R_b L_o}\left(\frac{c}{4\pi rf}\right)^2 \quad . \tag{47}$$

In this equation, $P_t$ is the total transmitted power of the sensor nodes, $G_r$ is UAV's antenna gain, $d$ is the UAV elevation, $k$ is Boltzmann's constant, $T_S$ is the receiver's effective temperature, $R_b$ is the data rate, $f$ is the frequency, $c = 3\times10^8$ m/sec and $L_o$ represents additional losses. As we increase the number of sensor nodes, $N$, involved in beamforming, the gain $G_t$ of the resulting antenna array increases (and, as mentioned, the gain achievable under certain conditions is $N$). Thus, increasing $N$ for a fixed $E_b/N_o$ at the UAV allows for a significant reduction in each node's transmit power. Since our aim is to minimize the power expended by sensor nodes, it would seem that the more nodes that participate in beamforming, the better. We show that this is not the case, and we specify the optimal number of nodes to minimize the total energy expended.

Consider a linear arrangement of sensor nodes that are to form a distributed antenna array, as shown in Figure 43. To simplify the results that are to follow, we assume that the number of nodes that will participate is an odd number; this is done only to place the clusterhead at the center of the linear array. The extension of our results to arrays with an even number of nodes is straightforward.

To establish the array and to subsequently communicate with the UAV, two different types of communication occur:

- *Inter-node communication*: Nodes must transmit control packets to neighboring sensor nodes to coordinate formation of the antenna array. We assume that all such coordination packets are of a fixed size of $L_1$ bits. Furthermore, we assume that the power required for inter-node communication is a fixed value for all nodes equal to $P_{S1}$ watts. Neighboring nodes are close enough so that $P_{S1}$ is the minimum possible node transmission power.
- *Transmission to the UAV*: Each node in the antenna array will transmit a packet to the UAV, containing the collected sensor data. These transmissions to the UAV will be identical, except for a phase offset. We assume that the size of the packet transmitted to the UAV by each node is equal to $L_2$ bits, and the power used by each node is $P_{S2}$ watts. See Figure 44.

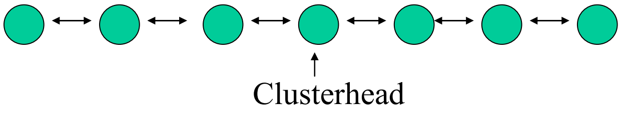


Figure 43. Linear Arrangement of Sensor Nodes

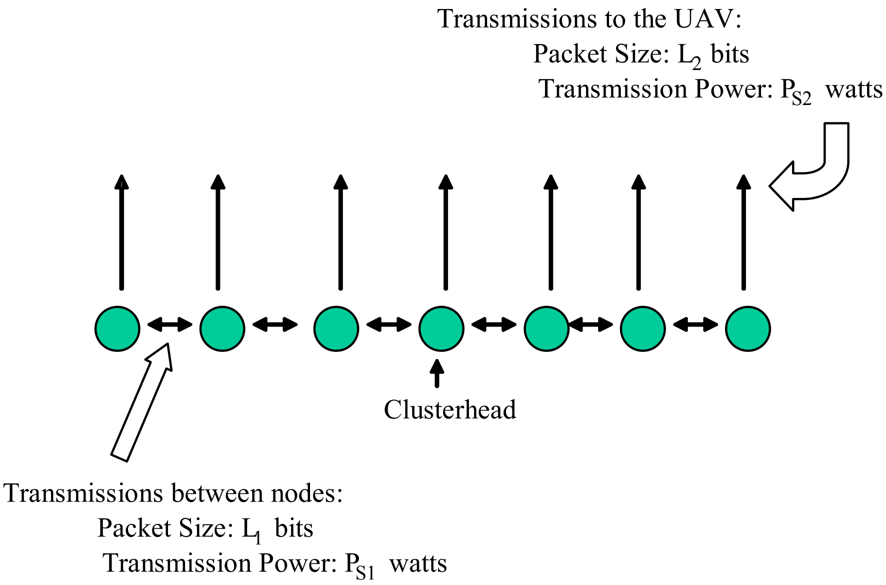


Figure 44. Two Types of Communication

Consider a very simple algorithm for formation of the linear antenna array and subsequent transmission of the sensor data to the UAV:

1. The cluster head requests participation from a total of *N* sensor nodes (including itself).
2. Sensor nodes reply to the request. If insufficient sensor nodes reply affirmatively, the cluster head will query additional nodes.
3. Once *N* nodes have been selected and transmission to the UAV is to occur, the cluster head passes the packet containing the sensor data and the phase offsets to the participating sensor nodes.
4. The participating sensor nodes transmit the packet containing the sensor data to the UAV.

We are interested in estimating the *absolute minimum* energy consumed by the sensor network in implementing this scheme. To calculate the minimum energy, we assume the following:

- There are no packet collisions during the implementation of the algorithm.
- All sensor nodes that are requested to participate reply affirmatively. Thus, in Step 2 above, the cluster head never has to make a second query to find additional nodes willing to participate.

- All energy for processing (such as energy expended in determination of the array phase offsets) is negligible.
- The number of bits representing the phase offsets is negligible compared to the size of the packet containing the sensor data.

### 2. Analysis of Energy Required to Implement the Beamforming Algorithm

The radio transceiver in each sensor node contains a receiver and a transmitter. Recalling our previous notation, a power, $P_R$ , is consumed by the receiver whenever it is energized, while a power, $P_T$, is consumed by the transmitter whenever it is energized. During transmission, an additional power, the radiated transmit power of the signal, is consumed. This additional power is equal to $P_{S1}$ for internode transmissions and $P_{S2}$ for transmissions to the UAV. All transmissions are at a rate of $R$ bits/sec.

To calculate the minimum energy required of the scheme, consider Figure 45. The minimum total energy consumed in a single one-way transmission between adjacent sensor nodes is:

$$E_{\min}^{\text{one-way}} = \left(P_T + P_R + P_{S1}\right)\left(\frac{L_1}{R}\right) . \tag{48}$$

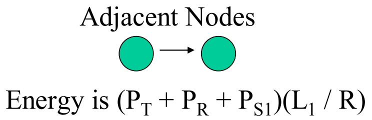


Figure 45. Energy Required for a Single One-Way Inter-node Transmission

In Step 1 of the algorithm, $(N-2)$ transmissions of the type described by (48) occur, where, again, $N$ is the total number of nodes intended for the linear array, including the cluster head. (Note that the initial transmission from the cluster head is passed to its two adjacent neighbors. Thereafter, each node passes the message onward to its neighbor.)

In Step 2 of the algorithm, $(N-1)$ transmissions of the type described by (48) occur, as each node replies affirmatively, and nodes pass collected data to the cluster head. In Step 3 of the algorithm, the total energy consumed is given by

$$E^{\text{setup}} = (N-2)(P_T + P_R + P_{S1})\left(\frac{L_2}{R}\right), \tag{49}$$

as the cluster head passes the packet containing the sensor data to each of its neighbors (Again, note that the initial transmission from the cluster head is passed to its two adjacent neighbors, hence the coefficient is $N-2$ and not $N-1$).

Finally, in Step 4 of the algorithm, the participating sensor nodes transmit the packet containing the sensor data to the UAV, consuming a total energy of:

$$E^{\text{data}} = NP_{S2}\left(\frac{L_2}{R}\right). \tag{50}$$

Combining (48), (49) and (50), and considering the discussion above, the minimum energy consumed by the algorithm is given by:

$$E = (2N-3)(P_T + P_R + P_{S1})\left(\frac{L_1}{R}\right) + (N-2)(P_T + P_R + P_{S1})\left(\frac{L_2}{R}\right) + NP_{S2}\left(\frac{L_2}{R}\right). \tag{51}$$

Now, in (51), the quantities $P_T$, $P_R$, $L_1$, $L_2$ and $R$ are assumed to be fixed and known. As mentioned earlier in this section, we further assume that sensor nodes are close to each other such that the inter-node communication power, $P_{S1}$, is fixed at the minimum transceiver power. This means that the only variables in (51) are $N$, the number of participating nodes, and $P_{S2}$, the power used by each node to transmit to the UAV.

Now, rearranging (47), we have

$$PG_t = \frac{\left(\frac{E_b}{N_o}\right) kTRL_o}{G_r\left(\frac{c}{4\pi df}\right)^2}. \tag{52}$$

In (52), the power $P$ on the left side of the equation refers to the total power transmitted by the array, which is $NP_{S2}$. We further note that the best achievable gain using $N$ nodes is given by (45) as $G_t = N$. Thus, Combining (45) and (52), we calculate the power used by each node to transmit to the UAV as:

$$P_{S2} = \frac{\left(\frac{E_b}{N_o}\right) kTRL_o}{N^2 G_r \left(\frac{c}{4\pi df}\right)^2} \tag{53}$$

Substituting (53) into (51) yields the minimum energy required by the algorithm as a function of $N$, the number of participating nodes:

$$E = N\left\{\left(\frac{P_t + P_r + P_{S1}}{R}\right)(2L_1 + L_2)\right\} - \left(\frac{P_t + P_r + P_{S1}}{R}\right)(3L_1 + 2L_2) + \frac{1}{N}\left\{\frac{L_2\left(\frac{E_b}{N_o}\right) kTL_o}{G_r\left(\frac{c}{4\pi df}\right)^2}\right\} \tag{54}$$

Thus, we see that as $N$ increases, the leftmost term increases and the rightmost term decreases. The middle term in (54) is an artifact of our algorithm, independent of $N$. The leftmost term can be viewed as a required overhead that linearly increases with $N$. The rightmost term, which decreases as $1/N$, represents the transmit power required by each sensor node.

### 3. Examples

We are interested in determining the value of $N$, the number of sensor nodes that participate in beamforming, that minimizes the total energy, as given by (54).

We first consider the commercially available MICAz mote, which, per its specifications, has a minimum transmit power ($P_{S1}$) of $4 \times 10^{-6}$ W, operates at 2.4 GHz, and supports a data rate of 250 kbps. We assume $L_1 = 1000$ bits, $L_2 = 1 \times 10^6$ bits, and $P_T$ and $P_R$ are both 100 $\mu$W. The receiver in this example has a gain of 10 dB, uses QPSK and requires a bit error rate of at least $10^{-5}$ with a gain margin of 10 dB, which, in turn, requires that $E_b/N_o$ be at least 20 dB [72]. The UAV is 6000 feet overhead. Figure 46 shows a plot of (54) for this scenario. We note that the optimal value of $N$ is $N = 21$.

Figure 47 shows a plot of (54) using a different commercially available mote, the MICA2, which has a minimum transmit power ($P_{S1}$) of $1 \times 10^{-6}$ W. Letting $L_1$ = 1000 bits, $L_2 = 1 \times 106$ bits, $P_T = P_R = 100$ $\mu$ watts as before, and other parameters as shown in the figure, we find for this case that the optimal value of $N$ is $N = 7$.

In summary, we have analyzed the optimal number of sensor nodes that should participate in a beamforming process in order to minimize the energy expended by nodes in a sensor network. As we increased the number of sensor nodes involved in beamforming (for a fixed energy per bit to noise power spectral density required at the receiver), the transmit power of each node is significantly reduced, but the overhead in implementing even a simple collaborative algorithm increases. We provided a framework that a system designer can use to analyze the minimum energy expended in a simple beamforming algorithm, which, in turn, specifies the optimal number of nodes to use to minimize the total energy expended, and thus extend the lifetime of a sensor network.

Note that for the specific commercially available motes selected for illustration in Figure 47, the wavelength (given the operating frequency) is approximately 0.33 m. This suggests that for effective beamforming the nodes would need to be exceedingly close to each other, with separations on the order of 15 cm. The implications of this fact must be considered by the designer when choosing the operating frequency to be used.

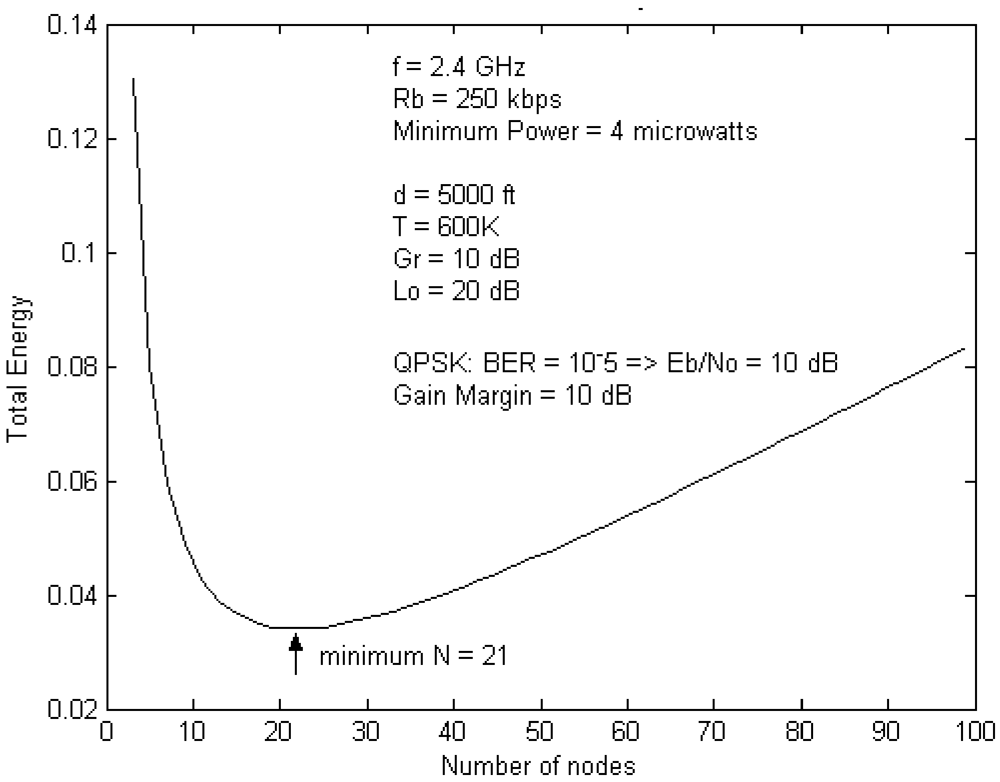


Figure 46. Total Energy versus Number of Nodes for MICAz Mote

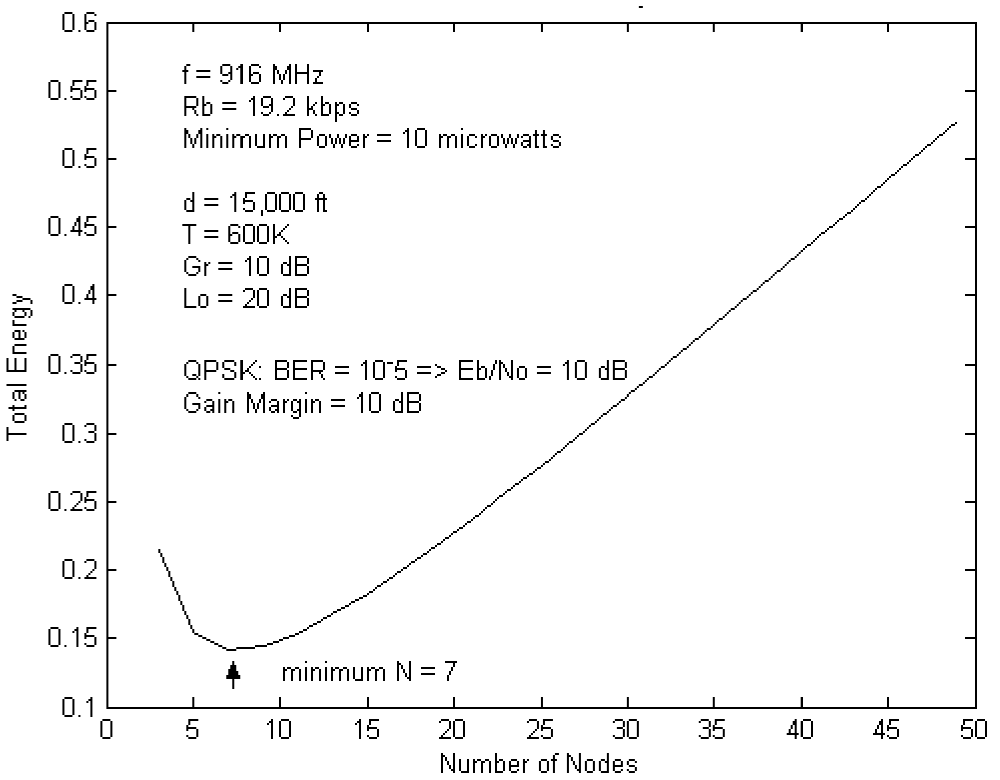


Figure 47. Total Energy versus Number of Nodes for MICA2 Mote

# V. OPERATION IN THE FACE OF RANDOMNESS

In the previous chapters, we developed an energy-efficient algorithm to move the transmit cluster to other locations in the sensor network to evenly spread the energy load and extend the lifetime of the network, and we have also developed an energy-efficient approach to aligning the transmit cluster's beam with the UAV. We then, in Chapter IV, examined how the transmit cluster should assemble its antenna array in an energy-efficient fashion, and we analyzed the optimal number of sensor nodes that should participate in a beamforming process in order to minimize the energy expended by nodes in a sensor network. Now, in this chapter, we analyze our design approach in the face of randomness. We first propose a clustering methodology for sensor networks. We then analyze the probability that an arbitrarily selected sensor node is connected to a specified number of other nodes; i.e., the probability that a communication path exists between a selected node and at least *m* other nodes for the purposes of forming a distributed antenna array. We present an algorithm for the formation of near-linear arrays given random placement of sensor nodes, and analyze the effectiveness of our approach through simulation.

## A. CLUSTERING

As a sensor network grows in size, the performance tends to deteriorate. If the network management and traffic handling responsibilities are fully distributed, then each individual sensor node must maintain network connectivity information, perform routing calculations and updates, and monitor network performance. Such operation—in the presence of node failures and link connectivity variations—requires sophisticated sensor nodes and considerable control traffic overhead since the dynamic nature of the network topology causes inconsistencies in the routing and connectivity information possessed by each node.

Since sensor networks contain a large number of nodes and cover large geographic regions, the nodes must be organized into a network that has some hierarchical organization or reliable structure that is maintained (at least to some degree) under varying network connectivities. This structure should, if possible, be attained in a distributed fashion, without reliance on a centralized controller. The network organization scheme must be designed to cope with the multiple access and collision problems inherent in the broadcast nature of the radio channel, as well as the scarcity of resources present in the wireless scheme.

One method of organizing a large group of mobile users into a network consists of hierarchically partitioning the overall sensor network into subnetworks called clusters, each managed by a distinct node termed the clusterhead (CH). A cluster induces a connected subgraph of the global sensor network and has the property that all nodes in a cluster are within a prescribed distance from the cluster's CH. As a practical matter, a cluster is a subset of sensor nodes in close geographic proximity to each other, whose operations are managed by their cluster's CH. Once the sensor network has been divided into clusters, the cluster organization must be maintained or adjusted as nodes fail.

Clustering a network is beneficial for three reasons:

- Organizing nodes into clusters with local controllers facilitates the use of techniques to resolve channel scheduling conflicts. For example, clusters provide a good framework for the use of efficient medium access techniques within the cluster (e.g., fixed assignment schemes such as TDMA, reservation schemes and polling schemes). Partitioning the network nodes into spatially segregated clusters facilitates the reuse of CDMA codes and TDMA time slots in different clusters.
- The cluster head can be utilized as a local controller for nodes within its jurisdiction. For example, a scheme might require that all nodes send any messages to their cluster head for further routing. A clusterhead could then measure and regulate the transmit power for each node within its jurisdiction. [27]
- A backbone network, comprised of the clusterheads (and, if necessary, additional supporting nodes as well) is available, if desired, for routing of traffic in the network. The backbone links can be designed to support virtual circuits, or the backbone nodes may communicate with each other using fixed assignment or reservation schemes.

Clustering attempts to find a middle ground between vulnerable centralized architectures in which a single station controls the network and high-overhead fully-distributed architectures in which network control is the equal responsibility of all nodes. Clustering should, if possible, be attained in a distributed fashion, without reliance on a centralized controller. A good clustering algorithm should be robust; clusters should be selected such that assignments will not change considerably for a long period of time as the networks topology changes. [73] Otherwise, clusterheads will constantly lose control of their clusters and the scheme will be inefficient. Clustering algorithms should be simple, should generate a minimum of overhead, and should perform network clustering in a distributed manner, without reliance on centralized control or global network knowledge.

In sensor networks, clustering can be used for data aggregation. A CH can collect data from nodes in its cluster, and then process the data (e.g., fuse or summarize) before sending the data onward to other nodes. [74] This reduces the amount of traffic in the network. Note that this function is generally not performed by CHs in an ad hoc network. As suggested in [73], CH selection should incorporate the remaining energy that a sensor has. Specifically, the likelihood of a node becoming a CH should in some sense be proportional to the amount of energy remaining in a node.

### 1. Definitions and Mathematical Preliminaries

We consolidate a number of definitions and mathematical results that will prove useful in the following.

**Definition**: Connectivity Graph

We model the topology of the sensor network as an undirected graph $G = (V, X)$ where $V$ is the set of sensor nodes and $X$ is the set of edges. An edge $e \in E$ exists between two nodes $v_1, v_2 \in V$ if the two nodes can communicate with each other. Two nodes that are adjacent (i.e., have an edge joining then) are called one-hop neighbors. Two nodes that are not adjacent but have a path of length two between them are termed two-hop neighbors, and so forth. The graph topology is time-varying; as sensor nodes deplete their batteries or as environmental conditions change, nodes will disappear and edges will be broken.

**Definition**: Degree of a Node

The degree of a node is the number of one-hop neighbors it has in the connectivity graph. The maximum node degree present in the connectivity graph is denoted as $\Delta$.

**Definition**: Neighbor Adjacency List

A node's neighbor adjacency list is a list of the node's one-hop neighbors.

**Definition**: Dominating Set

A dominating set $D \subset V$ is a set of nodes such that each node $v \in V$ is either a member of $D$ or is one-hop away from at least one member of $D$.

**Definition**: Ordinary Node

The set of nodes that are in the original connectivity graph but are not in $D$ is called the set of ordinary nodes, denoted $O$. Note that $V = D + O$ and $D \cap O = \varnothing$.

**Definition**: Covered Node

An ordinary node is said to be a covered node if it is within one-hop of at least one clusterhead. An ordinary node that is not a neighbor to at least one clusterhead is said to be "uncovered".

**Definition:** Dominating Set Problem.

The dominating set problem, known to be NP-complete, seeks a minimum dominating set in a graph.

**Definition**: The Bounded-Degree Dominating Set Problem

The bounded-degree dominating set problem seeks a minimum dominating set in a graph for which $\Delta$ is known a priori.

**Definition:** An Independent Set

An independent set for a graph $G = (V, E)$ is a subset of nodes $I \subseteq V$ such that no two nodes in $I$ are adjacent in $G$.

**Definition**: The Independent Set Problem

The independent set problem, known to be NP-complete, seeks a largest independent set in a graph. Note that the largest independent set will also be a dominating set.

**Definition**: The Bounded-Degree Independent Set Problem

The bounded-degree independent set problem seeks a maximum cardinality independent set in a graph that has a known $\Delta$.

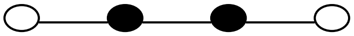
In analyzing an algorithm, the notion of a *performance ratio* is often used as a metric. An algorithm to find a maximum independent set is said to have a performance ratio (or an *approximation ratio*) of $\rho$ if, for any input, the number of nodes in the algorithm's solution is within a factor of $\rho$ of the optimal. That is, $\rho$ gives the factor by which the optimal solution is larger than that produced by the algorithm: $\frac{IS^*}{IS^A} \leq \rho$, where $IS^*$ is the cardinality of the optimal solution (the largest independent set) while $IS^A$ is the cardinality of an independent set determined by an algorithm under examination. Note that $\rho$ is always greater than or equal to one, and may be a function of one or more graph parameters (e.g., the total number of nodes in the network).

Using the best-known algorithm to determine a maximum cardinality independent set, the approximation ratio that can be guaranteed in polynomial time currently stands at $O\left(n/\log^2 n\right)$ [75]. Another paper provides a strong lower bound of $n^{\varepsilon}$ for any $0 < \varepsilon < 1$ [76].

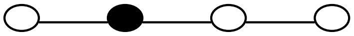
As mentioned, the Bounded-Degree Independent Set Problem seeks a maximum cardinality independent set in a graph that has a maximum node degree of $\Delta$. The problem is known to be NP-complete even for planar graphs of degree bounded by 3 or 3-regular [77]. The first performance ratio established for the problem was Δ [78]; improved algorithms provided ratios of $\frac{\Delta}{2}$ [79] and then $\frac{\Delta+2}{3}$ [80]. The best algorithms known to date provide performance ratios of $\frac{\Delta+3}{5}$ [81] and $\frac{\Delta}{6}+O(1)$ [82] for even Δ, and $\frac{\Delta+3.25}{5}$ [80] and $O\left(\Delta/\log\log\Delta\right)$ [81] for odd Δ. A recent heuristic established a performance ratio of $6/5$ for graphs with $\Delta = 3$ and $7/6$ for cubic graphs [83].

What of performance ratios for algorithms that attempt to find dominating sets? An algorithm with a performance ratio of $\ln n$ (where $n=|V|$) was provided in [84]. The classic greedy set-cover algorithm was shown to have a performance ratio of $\ln n - \ln\ln n + 0.78$ when applied to the dominating set problem [85]. A very important result demonstrated that no polynomial-time algorithm can approximate the optimal dominating set within a ratio of $\frac{1}{2}\log_2 n$ [86]. That is, no polynomial-time algorithm can achieve a performance bound better than $\frac{1}{2}\log_2 n$. If the maximum node degree is bounded by Δ, the simple greedy set cover algorithm was shown to have a performance ratio of $\sum_{i=1}^{\Delta}\frac{1}{i}$ when applied to the bounded-degree minimum dominating set problem [87].

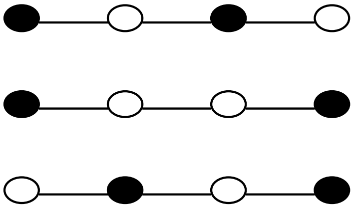
As an example to clarify the foregoing, the darkened nodes in Figure 48(a) represent a dominating set for the graph, but these nodes do not comprise an independent set. The darkened node in Figure 48(b) represents an independent set for the graph, but this single node does not comprise a dominating set. Each of the sets of darkened nodes in Figure 48(c) are both independent and dominating sets for the graph.



(a) DS, not an IS



(b) IS, not a DS



(c) DS and IS

Figure 48. Independent Sets and Dominating Sets

## 2. Clustering Requirements

A set of clusterheads should satisfy the following three requirements:

- *Covering requirement*: Sensor nodes communicate with clusterheads either directly or through other sensor nodes. In selecting the set of clusterheads, we specify that a fraction $p$ ($0 < p \leq 1$) of sensor nodes should be within a distance of $h$ hops from at least one clusterhead. If, for instance, we specify that $h = 1.0$ and $p = 0.9$, then 90% of all users will be within one hop of one or more clusterheads. If, on the other hand, we specify that $h = 1$ and $p = 1$, then all users can communicate directly with at least one clusterhead, and, in graph-theoretic terms, the set of clusterheads form a dominating set for the underlying graph.
- *Minimality requirement*: We desire that the number of nodes selected as clusterheads be minimal in some sense. Obviously, an algorithm that chooses every node as a clusterhead will not be useful. If we specify that $h = 1$, $p = 1$ and, additionally, we require the number of nodes selected as clusterheads be minimal, the set of clusterheads is a minimum dominating set.
- *Robustness requirement*: The CH assignments should not change considerably for a long period of time in the face of node battery depletions and node connectivity variations.

### 3. Clustering Algorithms

Many clustering algorithms in the literature ensure that all nodes in a cluster are one hop away from (and thus can directly communicate with) the cluster head; such algorithms assign each network node one of two possible roles: cluster head or ordinary node, where all ordinary nodes are associated with a distinct cluster head that is one hop away. Thus, in such algorithms, the set of clusterheads forms a dominating set for the underlying connectivity graph. Many algorithms in the literature provide techniques to find satisfactory dominating sets. In this section, we examine several algorithms that provide dominating sets.

#### Clustering Algorithms based on Node-ID Number

The prototypical algorithm based on node-ID number was first presented in [26]. We term this algorithm *ID1*. ID1 is a distributed algorithm in which nodes, using only locally available information, group themselves into clusters such that all nodes in a cluster are one hop away from their cluster head (the cluster heads, thus, forming a dominating set for the network's connectivity graph). The input to the algorithm is the network's connectivity graph. The algorithm requires that all nodes keep global time synchronicity and assumes that each node in the network has a unique node ID-number.

**ID1 Algorithm** The nodes transmit messages in ID number order (lowest to highest) in a TDMA frame. The rule for clusterhead selection is that a node, upon reaching its turn to transmit in the TDMA frame, declares itself a clusterhead if it is not already linked to a previously declared clusterhead. Each node chooses as its own clusterhead the lowest numbered connected node that has declared itself a clusterhead. By the end of this TDMA frame, every node is a member of a cluster and is assigned a unique clusterhead.

Note that clusterheads will never be one-hop neighbors, and, in fact, each clusterhead will be either two or three hops from its nearest neighboring clusterhead(s). ID1 deals with potential changes in the network graph by rerunning the algorithm from scratch periodically. The ID1 algorithm was the basis for the network architecture in recent wireless mobile network designs (e.g., see [28]).

As noted in [88], algorithms based on node degree have some difficulties and disadvantages. First, the algorithm (as presented above) is biased toward selecting nodes with low identification numbers as clusterheads, which may present a problem if these particular nodes are not very capable (this problem can be mitigated by assigning low numbers only to very capable nodes). Second, during clustering, nodes exchange information using TDMA frames, with each node assigned a slot. This requires a global network time standard, and it is difficult to adjust the algorithm in a distributed manner to allow for the addition of a new node. Adding a node requires simultaneous adjustment of the TDMA frame among all nodes in the network. A method for partially mitigating this difficulty is presented in [89].

### Clustering Algorithms based on Node Degree

We now briefly examine network organization algorithms that are based on node degree. These algorithms differ from those in the previous section in that now the clustering and clusterhead selections are dictated by the various degrees of network nodes instead of by the node ID-numbers. The first such algorithm, which we term *D1* was presented in [27].

> **D1 Algorithm** Each node determines through broadcast communication its own degree as well as the degrees of all neighboring nodes. A node is said to be "covered" if it is either a cluster head or it is a one-hop neighbor to a cluster head. (note that this use of the term "covered" differs from our

> earlier use). Initially, when the algorithm begins, all nodes are "uncovered." A node becomes a cluster head if it is "uncovered" and has the highest degree among all its "uncovered" one-hop neighbors. If two uncovered neighboring nodes have the same highest degree value, the node with the lower ID-number becomes the cluster head.

A second clustering algorithm based on node degree was presented in [90] and will be designated as *D2*. We present the centralized version of the algorithm, noting that [89] also provides a distributed version.

> **D2 Algorithm** Find the node in the graph that has maximum degree and select it as a clusterhead. In the event that two or more nodes have the same maximum degree, choose the node with the lowest ID-number. Form a cluster out of this node and all its one-hop neighbors. Now, delete this cluster from the graph—that is, remove all nodes in the cluster and all the edges in and incident to the cluster—and repeat the algorithm with this new modified graph. Note that when we remove a cluster from the graph, the degree of each node that was two hops away from the deleted clusterhead will be reduced. Continue in this fashion until all nodes have been placed in clusters.
>
> The two node degree algorithms D1 and D2 above dynamically adjust the clustering topology in the face of node failures and node connectivity changes.

None of the algorithms make any claim on optimality with regard to determining a minimum dominating set for the underlying network graph, and some communication schemes in the literature have mixed and matched the various schemes. For instance, the design presented in [91] uses the a node ID-number algorithm (ID1) to initially partition the network into clusters, but the dynamic adaptation of the clustering is based solely on node degree.

The clustering algorithms described above were applied to the study of a specific type of network termed a mobile ad hoc network. Thus, it is pertinent to ask the question: Do these algorithms as currently applied (in the domain of mobile ad hoc networks) provide guidance on how to cluster sensor networks?

### 4. Are Sensor Networks MANETS?

A *mobile ad hoc network* (MANET) is a collection of mobile platforms, where each platform is able to serve as a router that can support communications for a number of other users. A MANET is a mobile wireless network that has no pre-existing infrastructure and supports multihop communication: two platforms (e.g., radios or computers) that cannot communicate directly can still exchange information by using other mobile platforms as intermediate relays.

For example, consider the MANET depicted in Figure 49 in which each soldier carries a radio that serves as node in the MANET, and a double-sided arrow between two soldiers indicates that that they can communicate with each other. A data stream from Soldier 1 (S1) to S4 might follow a path that employs the radios of S2 and S3 as intermediate relays. Note that S2 and S3 do not exist for the purpose of supporting communications between S1 and S4; they merely happen to be in certain positions and happen to have sufficient resources (e.g., processing capability, bandwidth, buffer size and power) available such that the underlying protocols choose to use them to enable communications between S1 and S4. In general, S2 and S3 are (and remain) independent agents, free to move in an arbitrary fashion. If S2 moves out of range from S3, the communication path between S1 and S4 will rupture and their radios, employing appropriate protocols might then attempt to reestablish communications by discovering an alternative route (such as S1-S5-S6-S3-S4). Indeed, this is what makes management of MANETs so difficult: radios just happen to be in certain positions for *reasons of their own*, and the protocols that establish communication paths choose appropriately positioned nodes in an *opportunistic* fashion.

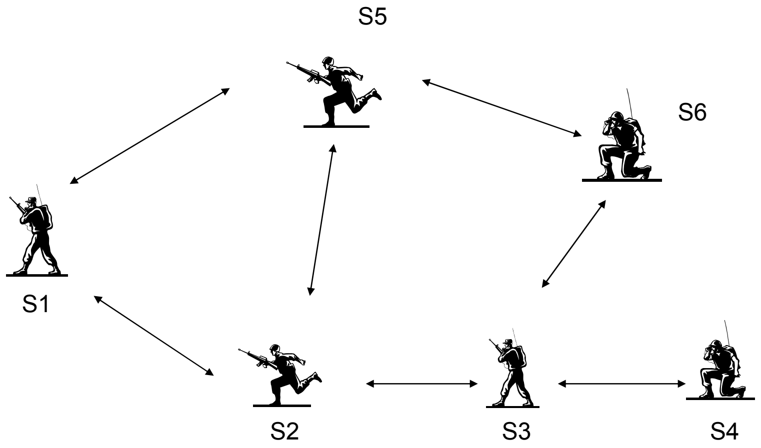


Figure 49. A Mobile Ad hoc Network (MANET)

In a sensor network, as in a MANET, there is no existing infrastructure to support communication except for the sensor nodes themselves, and, upon deployment, the nodes must dynamically organize themselves into a network that can relay the sensed data to the intended user. Similarly, as in a MANET, the sensor network is not static; the sensor nodes may fail (due to battery depletion), or external environmental factors (e.g., rain) may affect the ability of any two given nodes to communicate, and in such scenarios the network must adapt to ensure reliable communication of the sensed data to the end users.

In a loose sense, the MANET shown in Figure 49 might be considered a sensor network, where the soldiers are viewed as extremely capable sensor nodes: The sensors consist of the soldiers' eyes and ears, the processors consist of the soldiers' brains and the transceivers consist of the soldiers' radios.

At first blush, it would seem that a sensor network is a MANET, and the algorithms and techniques that have been developed for MANETs can be transplanted to the domain of sensor networks. We posit that this would be a mistake. There are key differences between sensor networks and a MANETs.

First, sensor nodes do not physically move. One might, at first, argue that this is a distinction without a difference, at least as far as some of the underlying protocols are concerned. For example, a stationary sensor node in a stationary network exhausting its battery is akin to a mobile node in a MANET moving out of range, since in each case the challenge for the remaining nodes is the same: reconfiguring the network so that the remaining nodes can communicate with each other as might be required. And the protocols for reconfiguring networks that have lost a node are the same, regardless of why the node was lost. While this latter argument has some validity, the dynamism in a MANET would be expected to be significantly greater than that present in a sensor network. More dynamism can be expected in a network whose nodes consist of soldiers racing over a battlefield than in a network whose nodes are stationary sensor elements, trying their best to conserve power.

This point leads to the second distinction: A primary consideration in the design and employment of a sensor network is power (i.e., battery) conservation. If we want to classify sensor networks as MANETs, then we would have to further specify that a sensor network is a MANET composed of nodes that are very power-limited.

Third, most MANETs carry voice data (conversations), and the network must ensure that the end-to-end transmission delay is very low. For satisfactory voice conversations, the maximum end-to-end delay (mouth to ear) should be on the order of 100 msec. [92] For sensor networks, the end-to-end delay is not as stringent a consideration; if a sensor detects, say, an intruder, it would not likely matter whether this information was received by the ultimate end-user within 50 msec or 500 msec.

Fourth, some MANETs must contend with high-bandwidth data, such as real-time full-motion video. On the other hand, many of the sensors in a sensor network might collect data that can be summarized in just a few dozen bytes.

Fifth, in the MANET of Figure 49, each of the soldiers might equally well serve as the intended destination, or sink, of a data stream. In a sensor network, on the other hand, most network traffic—while traversing a path that might use intermediate sensor nodes—is usually destined for just a few sinks. Consider the sensor network shown in Figure 50 which displays a 25-node sensor network distributed over a terrain. Nodes 1, 2, 3 and 4 are able to communicate with a more capable long-range radio that can transmit data from the sensor network to, say, a ground-based network (not shown) while Nodes 5 and 6 are able to communicate with a long-range radio that can transmit data to, say, a satellite. Strictly speaking, the two long-range radios constitute the data sinks for the network as all sensor data is ultimately intended to go to one of these sinks. While the presence of only a few potential sinks simplifies the routing problem (since sensor nodes need only be concerned with routing to a few destinations), a very heavy burden is placed on the sensor nodes that are adjacent to the data sinks. Limiting our view to just the sensor network, we can consider Nodes 1, 2, 3, 4, 5 and 6 to be the network sinks in the sense that any data that originate at any of the 25 sensor nodes must ultimately be retransmitted by one of these nodes in order to be delivered to an end-user. (Note that the sensor sink nodes may change; the high-power radios in the network of Figure 50 might be picked up and moved to another location. The important point is that for lengthy periods of time there may be only a few sink nodes.)

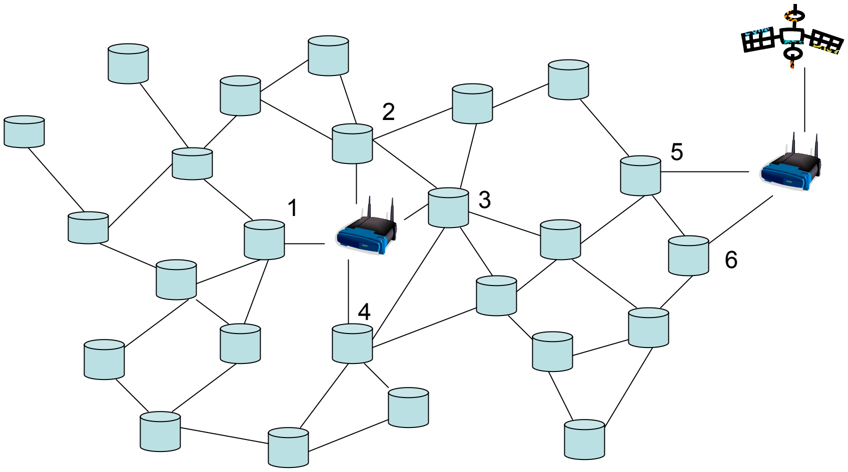


Figure 50. A Deployed Sensor Network

Sixth, in a sensor network there might not even exist an explicit sink. Suppose, for example, the sensor network is left unattended, and a sink node "plugs into" the network every so often to see what is transpiring. For example, consider the scenario of the previous chapters wherein a military sensor network is deployed in an inhospitable region to detect, say, the presence of anthrax. In this case, a UAV might be sent over the region every so often (as opportunities arise) to query the network. In instances when no sink is present, what should the network do with data that arise? Should the network discard all sensed data—potentially important information—if there is no sink present at the time of detection, in the interests of conserving power? This would mean, in effect, that we only use the sensor network when a sink is attached, and we are not interested in any history outside of such time periods. Or, on the other hand, is sensed data so important that it should be stored pending the arrival of a sink? If the answer to the latter question is yes, then where should the data be stored? Only at the node that originally sensed the event, in the interests of conserving power? Or, should we expend power to replicate important data at other nodes, just in case the node that first detected the data should expire (and take with it the data)? These questions don't arise in MANETs since communication is usually interactive, two-way, and real-time.

Seventh, as described at great length in Chapter II, sensor networks often use flooding as a routing algorithm, as opposed to point-to-point routing. Flooding might be appropriate for distributing small packets of a few bytes containing a sensor reading. Flooding would not be appropriate for transferring large blocks of data (e.g., video streams).

### 5. Clustering in Sensor Networks

In light of the foregoing, we provide a qualitative answer to the question: How should clustering be performed in sensor networks?

The first issue we consider is: Should the clustering algorithm be based on node ID (e.g., as in the algorithm ID1 above) or node degree (e.g., as in the algorithms D1 and D2 above)?

There are some qualitative differences between the two types of algorithms (i.e., those based on node ID and those based on node degree) that we can immediately surmise. For instance, it is obvious that the node degree algorithms are much more dependent on the actual underlying graph topology than the algorithms based on node ID-number. For instance, in using the ID1 algorithm, the node with ID-number of 1 will always be selected as a cluster head, regardless of where it is situated in the topology. In fact, in using ID1, identical network topologies may result in radically different clustering structures, just based on how the nodes happen to be assigned their ID-numbers. By contrast, the algorithms based on node degree organize clusters that are much more predicated on the underlying graph structure, although they retain some dependence on node numbering by virtue of how "ties" are broken.

A comparison of ID1 and D1 was carried out in [27], with the comparison based solely on robustness, i.e., on how well the algorithms perform in selecting clusterheads whose assignments do not change considerably in the face of network topology variations. Simulation studies showed that ID1 is more robust (i.e., more stable) than D1—node mobility caused fewer changes in the clustering, where changes account for the number of nodes which change their roles as cluster heads and the number of nodes which switch clusters. The lesser stability of the D1 scheme owes to the fact that if a node loses just a single link due to terminal movement, it may no longer qualify as a cluster head based on its new (lower) connectivity. A node which loses a link in ID1 may still qualify as a cluster head. Other techniques have been developed which attempt to improve robustness after the initial clustering by limiting the scenarios which require nodes to give up their role as clusterhead or take on the role of a clusterhead (see, for example, [93]).

In many MANET applications, the most important criterion in judging a clustering algorithm is stability. Frequent reclustering affects the ability to route traffic with Quality of Service (QoS) guarantees. Against this criterion, clustering based on node-ID number is better.[‡‡‡] In the words of a prominent MANET researcher, “In the presence of mobile nodes, ID-based clustering results in a more stable…architecture than connectivity-based clustering can achieve.” [72].

Does this finding apply to sensor networks as well? Should sensor networks, like MANETs, be clustered using node-ID clustering algorithms such as ID1? We posit that the answer is: *No*. We base this conclusion on the preceding description of the distinctions between sensor networks and MANETs. As noted, the dynamism in a sensor network would be expected to be significantly less than that present in a MANET. More

---

[‡‡‡] It is interesting to note that according to the ISO Web of Science, reference [27] (the paper that first noted that clustering based on node-ID is more stable than clustering based on node degree) has been cited in other journal articles over 150 times, probably qualifying as a “landmark paper.”

significantly, for sensor networks, the end-to-end communication delay is not as stringent a consideration, and, additionally, many of the sensors in a sensor network might collect data that can be summarized in just a few dozen bytes. Put another way, QoS constraints are less severe in the sensor networks that we are considering.

Now, algorithms based on node degree organize clusters that are much more predicated on the underlying graph structure. The clusterheads selected by node-degree algorithms constitute more *sensible* choices, since these nodes will have a large number of adjacent nodes to supervise. Unfortunately, this initial sensible selection of clusterheads is not robust. The clusterhead assignments will need to change as the network topology changes, and this is very bad when attempting to route traffic with QoS guarantees.

Thus, for MANETs, it is better to have bad clustering that is stable (such as that provided by node-ID algorithms) than good clustering that must be changed from time to time (such as that provided by node-degree algorithms). Note that even though the ID1 clustering algorithm is less susceptible to change, it may not provide good clustering to begin with. This is fine for MANETs which would prefer bad clustering that doesn't change frequently.

For sensor networks, which are not as concerned with meeting QoS guarantees, it is more sensible to use good clustering. When reclustering is necessary (e.g., when the battery power of a CH is reduced below a threshold, or when the number of nodes in a cluster is reduced below a threshold), a new set of clusterheads can be selected again based on node degree. A shift from a good set of clusterheads to an alternative set of good clusterheads is not as problematic since we do not need to be concerned with interrupting QoS-provisioned traffic streams.

In summary, based on the lower dynamism of sensor networks and the lack of an overriding need to support QoS constraints, we conclude that clustering should be based on node degree, note on node-ID number.

The second issue we consider is: Should clusterheads be selected to be an independent set?

This second question is motivated by the fact that the MANET clustering algorithms discussed above do not specifically attempt to strive for minimality in the number of clusterheads. Again, the main performance measure of concern in comparing the algorithms is robustness. Robustness, to be sure, is certainly important, but minimality should also be taken into account. Consider, as an example, the clustering algorithm: "All nodes are cluster heads." Obviously this clustering scheme is very robust in the face of node movement since all nodes simply remain cluster heads. Despite the robustness, this suggested clustering scheme is completely useless.

Many clustering algorithms select clusterheads that are not only dominating sets, but also independent sets (i.e., the clusterheads form a dominating independent set). Since these algorithms desire that all nodes be able to reach their clusterhead in a single hop, the need for a dominating set is clear. Why also stipulate the need for an independent set? The independent-set requirement is motivated by the desire that clusterheads be “well-scattered” across the network, yielding a more uniform geographical distribution of CHs. The clustering algorithm enforces “scattering” by insisting that no two clusterheads be one-hop neighbors; i.e., by ensuring that the clusterheads form an independent set.

We next state our second conclusion regarding clustering in sensor networks: Clustering algorithms should not impose the requirement that cluster heads form an independent set.

Consider the network shown in Figure 51 (a). If clustering algorithms are applied based on node degree, the clusterhead selection will be isomorphic to the selection shown in Figure 51(b). Clearly, the clusterhead selection shown in Figure 51(c) is a better selection, even though these clusterheads do not comprise an independent set. Even though the network in Figure 51 was selected to make a dramatic point (i.e., Figure 51(b) versus Figure 51(c)), it is reasonable to conclude, in general, that clustering algorithms that do not impose the independence requirement will tend to organize networks with fewer clusterheads. We offer an intuitive reason for this: Let the set of all possible *dominating* sets for an arbitrary graph be denoted *X*. Let the set of all dominating sets that are also *independent* sets in the same graph be denoted as *Y*. Clearly $Y \subset X$. Let the minimum dominating set for this graph be denoted as $z$. Then, it must be the case that $z \subset X$. It is not necessarily the case that $z \subset Y$. In other words, by stipulating that our dominating set must also meet the additional restriction of being an independent set, we potentially rule out many candidate dominating sets, possibly including the minimum dominating set.

In summary, to seek a set of clusterheads that is minimal, we conclude that clustering algorithms for sensor networks should not seek an independent set.

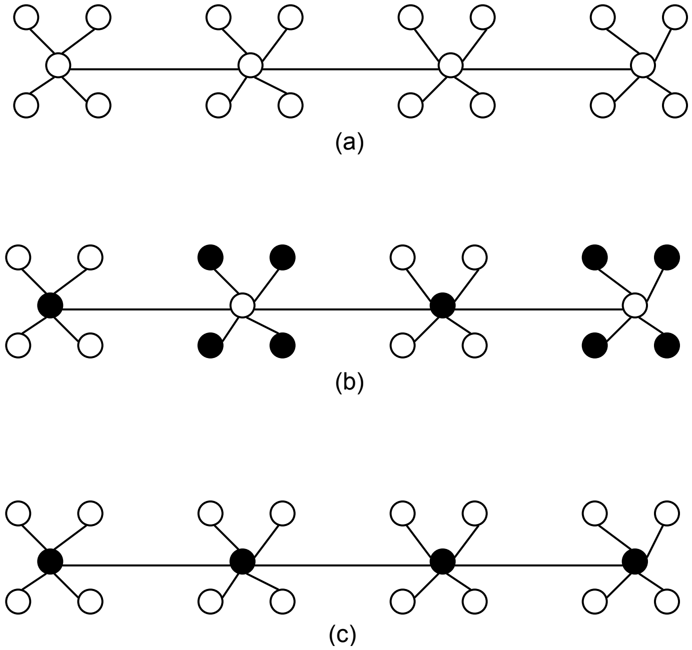


Figure 51. Clusterhead Selections

Having just concluded that clusterheads selection algorithms should not seek an independent set, we next consider a third issue: Should clusterheads be selected such that the choice comprises a dominating set?

Many MANET clustering algorithms in the literature require that every node be within one hop of its clusterhead. This requirement is motivated by the desire to minimize the amount of communication (here represented by the number of hops) required for a node to coordinate with its CH. That is, for ease of control, nodes should be able to reach their CH in a single hop.

In a dominating set, all cluster nodes are either in the dominating set or neighbors to one or more nodes in the dominating set. In a communications network, we may not want to impose a requirement that all nodes be immediate neighbors to the dominating set. For instance, if the network's graph topology has a "tail," a path consisting of degree-two nodes terminating at a degree-one node, we don't want to extend the dominating set out into the "tail" for the sake of the outlying degree-one node. As another example, we may want to synthesize a connected "2-dominating" set wherein all network nodes are either in the dominating set or within 2 hops of the dominating set (instead of forcing all nodes to be immediate neighbors to the dominating set).

It must be recognized that some of the performance bounds in the literature appear, at first, to be very discouraging. For instance, as previously noted, it has been proven that no polynomial-time algorithm can construct a dominating set with an assured performance ratio of $\frac{1}{2}\log_2 n$. Notwithstanding this rather discouraging performance ratio, dominating set algorithms may actually perform quite well for "typical" topologies. The bound of $\frac{1}{2}\log_2 n$ might actually apply to only extreme pathological topologies, and actual algorithm performance might be very good for the vast majority of realistic network topologies. What if probabilistic relaxations are placed on the domination requirement. For instance, instead of requiring that all nodes be within one hop of a node in the dominating set, we may only require that a fraction of the network nodes (e.g., 95%) be within one hop of a node in the dominating set. Or, as another example, we may require that 90% of the nodes be within one hop, 95% be within two hops and 99% be within three hops of node in the dominating set.

Based on the fact that sensor networks likely do not need to meet very stringent QoS requirements, we reach our third conclusion: Clusterhead selection algorithms for sensor networks should not seek sets of nodes that comprise dominating sets.

We summarize our conclusions. Clustering algorithms in sensor networks should seek a set of clusterheads based on node-degree, not node-ID. These clustering algorithms should not stipulate that clusterheads comprise an independent set. Furthermore, these clustering algorithms should not stipulate that clusterheads comprise a dominating set.

We note that our conclusions above are merely qualitative, suggesting an area ripe for further research.

Finally, for completeness, we note that a sensor network design that uses clustering was recently considered in [94]. This sensor network differs from ours in that the network in [93] contains two types of sensor nodes: a large number of ordinary sensor nodes and a smaller number of highly-capable sensor nodes containing more power. These few latter nodes are the only eligible candidate clusterheads. Additionally, in [93] all nodes know the location of the stationary sink *a priori*, and all data transmissions are two-hop: from the sensor node to its CH to the sink (no multihop scenario). Additionally, the MAC problem is not addressed (perfect coordination is assumed) and the sensor network is clocked; every clock cycle every sensor node has one fixed-size packet to transmit. Finally, [93] assumes that all nodes have enough power to always reach some CH one-hop away.

## B. PROBABILITY OF CONNECTIVITY[*]

We next analyze the probability that an arbitrarily selected sensor node in a sensor network is connected to a specified number, *m*, of other sensor nodes; i.e., that a communication path exists between a selected node and at least *m* other nodes. This issue is particularly pertinent for unattended sensor networks that perform cooperative operations (such as beamforming) requiring the participation of at least *m* nodes.

---

[*] The results presented in this section were previously presented at the *40th Annual Hawaii International Conference on System Sciences (HICSS-40)* [95]

Suppose we have a set of sensor nodes, each of which has a communication range of *r*, in the sense that nodes within a distance *r* of each other will be able to directly communicate, but nodes separated by distance greater than *r* will not be able to directly communicate. For example, the two nodes in Figure 52(a) can directly communicate with each other, but the nodes in Figs 52(b) and 52(c) cannot.

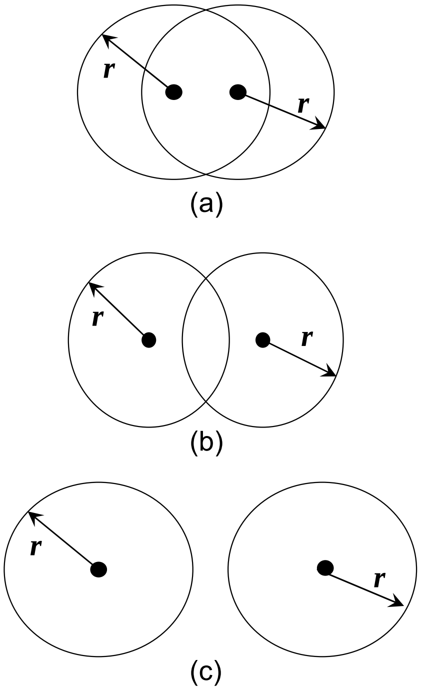


Figure 52. Node Communication Range

Let *n* such nodes be distributed over a finite planar region. As before, we model the resulting communication network as an undirected graph $G = (V, E)$, where $V$ denotes the set of nodes ($|V| = n$) and $E$ denotes the set of edges, where an edge exists between two vertices if the corresponding nodes can directly communicate. From graph theory, a path from node *u* to node *v* is an alternating sequence of distinct vertices and distinct edges, beginning at *u* and ending at *v*, with each edge incident to the two vertices immediately preceding and following it in the sequence [96]. If a path exists between two nodes, the two nodes are said to be connected, and if a path exists between every pair of vertices in *V*, then the graph *G* (and the communication network it serves to model) is said to be connected. Thus, if a communication network is connected, then every node in the network will be able to communicate with every other node, either directly or using other nodes as intermediate relays.

Suppose that n nodes are independently distributed within a region R of size A (square units). The position of each node is uniformly distributed over R; that is, the probability density function is equal to $1/A$ for all points within R and is equal to 0 outside R (see Figure 53). We ask the question: What is the probability that a selected node will be connected to at least m other nodes (where $m \le n-1$)?

Applications in which such a question might arise include the following:

- Consider sensor nodes employed to measure temperature. Individual sensor nodes might not provide very accurate readings; we may want to denote the temperature of a point as the average of the readings of the nearest, say, eight sensor nodes. If we "plug into" an arbitrary node, to what extent can we be assured of being able to obtain readings from seven neighboring (connected) sensor nodes as well?
- Consider sensor nodes employed to measure acoustic signals. By combining the measurements from several sensors, we can estimate the direction of an acoustic source. If, for example, we find that we need to combine the readings of five nodes to obtain an accurate direction estimate, then to what extent can we be assured that an arbitrarily selected node will have at least four connected neighbors to exchange data with?
- Consider the example of Chapters IV where we noted that if multiple sensor nodes coordinate their transmissions—each sending the same signal except for calculated phase and amplitude offsets—the propagating electromagnetic waves will interfere, with the net effect that the total radiated power can be focused in preferred directions. The ability to focus power in certain directions is termed the antenna gain and the participating antennas are collectively termed an antenna array. Suppose that, in the interests of power conservation, we require that six nodes coordinate their transmissions in order to attain a desired antenna gain. To what extent can we be assured that an arbitrarily selected node will be connected to at least five other nodes with whom to form an antenna array?

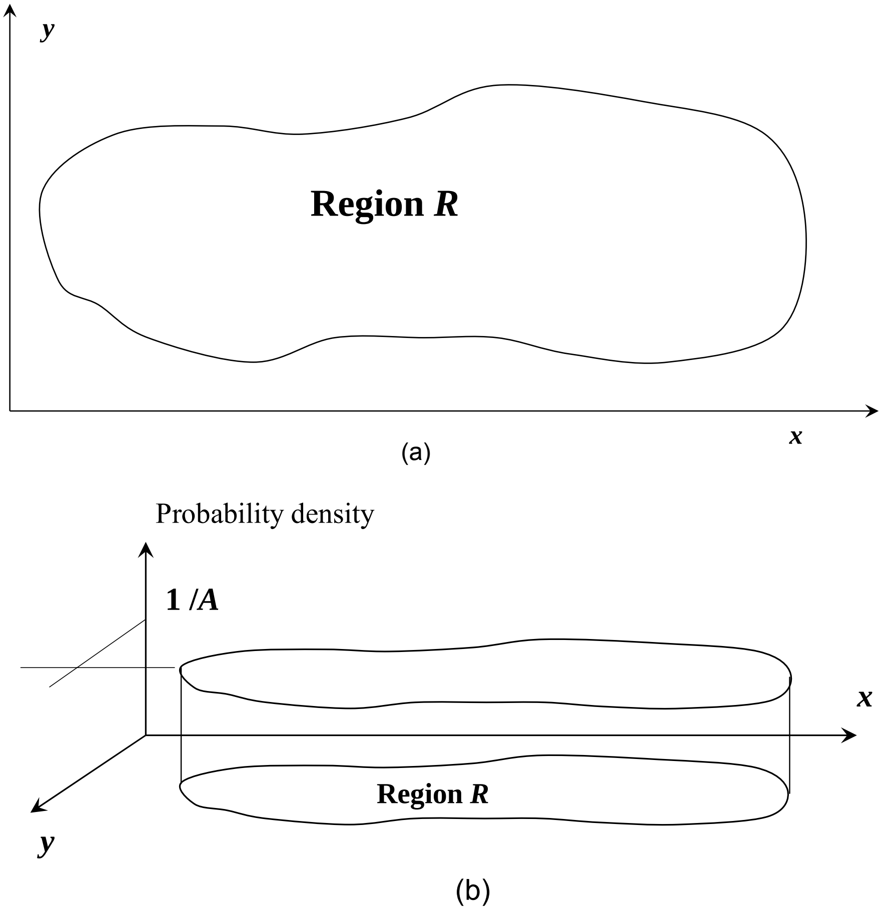


Figure 53. Nodes Distributed Uniformly Over Area

A number of recent papers have addressed the problem of assuring global network connectivity. These recent results offer limited utility for our application for three reasons.

First, some papers assume that the node positions are known with certainty. In [97], for example, nodes are carefully arranged in a grid pattern, and then fail with a specified probability—the randomness arises due to node failure. In our scenario, on the other hand, the randomness arises due to the initial random placement of the nodes.

Second, some papers (for example, [98]) are primarily concerned with ensuring that sensor-equipped nodes *cover* a region of interest, with every point in the region lying within the *sensing radius* of at least one node. It is then noted—as a side effect—

that if the communication range is at least double the sensing range, complete sensor coverage implies that the network is connected. We, on the other hand, are not concerned at all with whether the region is covered; for instance, we are content if the network happens to be arranged such that all the nodes are clumped in a corner of the region, providing sensor coverage for only a small limited subregion.

Third, many papers study connectivity properties that apply asymptotically, providing connectivity assurances as the region size grows to infinity, or as the number of nodes goes to infinity. Such results are of great theoretical interest, but are found lacking in real-world scenarios that involve finite areas and a finite number of nodes. One result [99], for example, establishes conditions to assure connectivity if the region of interest is a disc, and the number of nodes $\rightarrow \infty$. We, on the other hand, are interested in answering questions such as: "How many nodes do I need to deploy in a certain area to assure that an arbitrarily selected node is connected to at least 10 other nodes with a probability of at least 0.8," or, alternatively: "If I deploy 1000 nodes within a certain area, how sure can I be that an arbitrarily selected node is connected to three other nodes?"

We state the main result of this section as a theorem.

**Theorem**. Suppose that $n \geq 2$ nodes are independently distributed over a region of size $A$ (square units), with the position of each node uniformly distributed over the region. Each node has a communication range of $r$. Let $P_c(n,m)$ denote the probability that an arbitrarily selected node (from among the $n$ nodes) is connected to at least $m$ other nodes ($m \leq n-1$). A lower-bound on $P_c(n,m)$ is given by

$$P_c(n,m) > 1 - (1-x)^{n-1} - \sum_{k=1}^{m-1} \binom{n-1}{k} (1-x)^{n-1-k} \left( \frac{r^2}{A} \right)^k \prod_{j=0}^{k-1} \{\pi + 2j\} \tag{55}$$

where $x = \dfrac{\pi r^2}{A}$.

**Proof.** Let $P_d(n,m)$ denote the probability that the selected node is not connected to at least *m* other nodes; $P_d(n,m) = 1 - P_c(n,m)$. We find an upper-bound on $P_d(n,m)$, which provides a lower-bound on $P_c(n,m)$.

Suppose that first, before the nodes are distributed over the area, they are each assigned a unique number in the range [1, *n*]. After the nodes are distributed over the area, a random number is selected from the discrete uniform distribution U[1, *n*] (i.e., any integer between 1 and *n* has an equal probability of being selected). The node with this number is located in the graph and is painted red (the others nodes remaining, say, black). Employing the union bound [100], we have

$$\begin{aligned} P_d(n,m) < \; & \text{prob(the red node is disconnected from all other nodes)} \\ & + \text{prob(the red node is connected to 1 node and no others)} \\ & + \text{prob(the red node is connected to 2 nodes and no others)} \\ & + \\ & \vdots \\ & + \text{prob(the red node is connected to } m-1 \text{ nodes and no others)}. \end{aligned} \tag{56}$$

We next upper-bound each of the terms on the right side of (56).

In the analysis to follow, we ignore "edge-effects." That is, since the area $\pi r^2$ (where *r* is the communication range of a node) is much less than the area of the entire region (denoted as *A*), we will ignore the fact that for nodes near the boundary of the region, some of the area that is within the communication range of a node may be outside the region *A*.

First, consider the probability that the red node is disconnected from all the other nodes. Referring to Figure 54, all other nodes must be outside the circle of radius *r* centered at the red node. Hence

$$\text{prob(the red node is disconnected from all others)} = \left[\frac{A - \pi r^2}{A}\right]^{n-1}. \tag{57}$$

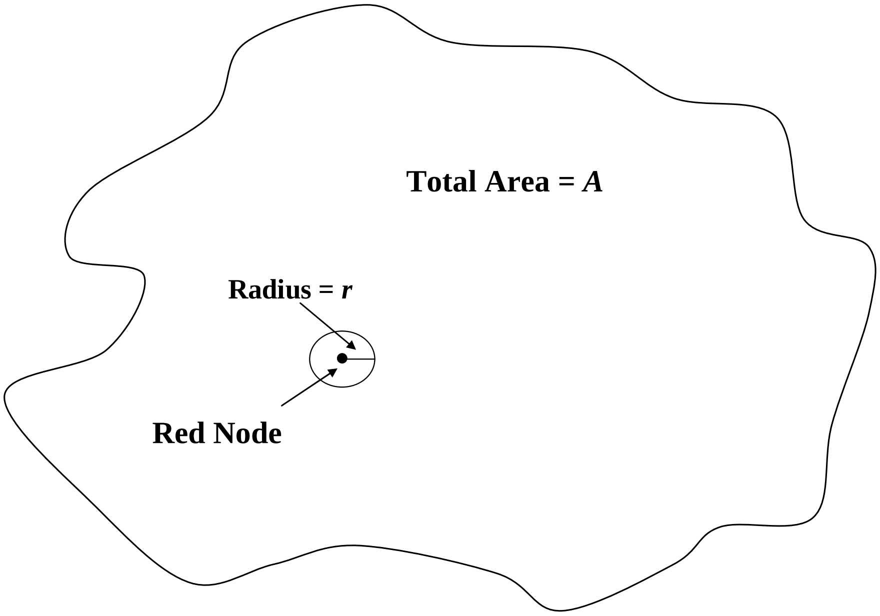


Figure 54. One Node Disconnected From All Others

Next, consider the probability that the red node is connected to one other node and disconnected from all other nodes. The node connected to the red node must lie within the circle of radius $r$ centered at the red node, and the probability of this event is $\pi r^2/A$. The other $n-2$ nodes must lie outside of the two overlapping circles of radius $r$ centered at the red node and the one connected node (Figure 55); the probability of this latter event is $\left[(A-B)/A\right]^{n-2}$, where $B$ is the size of the area contained within the two overlapping circles. Since $\pi r^2 \le B < 2\pi r^2$, and since the one connected node can be chosen from any of the set of all $(n-1)$ nodes, we have

$$\text{prob(the red node is connected to 1 node and disconnected from all others)} < \binom{n-1}{1}\left(\frac{\pi r^2}{A}\right)\left(\frac{A-\pi r^2}{A}\right)^{n-2}. \tag{58}$$

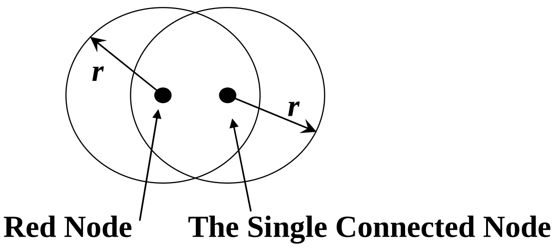
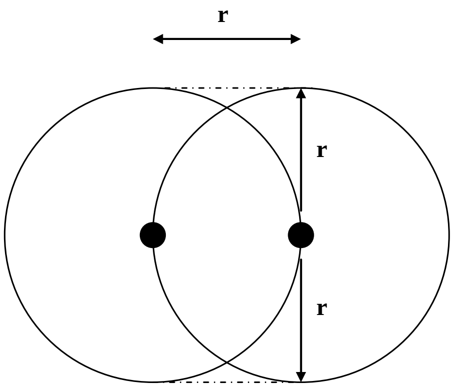


Figure 55. One Node Connected to a Single Node

Next, consider the probability that the red node is connected to two nodes and disconnected from all other nodes. (Recall that it is not necessary that the subgraph induced by these three nodes be the complete graph $K_3$; it may, for example, be $P_3$.) Starting with the red node, the closer node (say X) must lie within the circle of radius *r* centered at the red node, and the remaining node (say Y) must lie within the two overlapping circles centered at the red node and node X (and this latter area is certainly less than $\pi r^2 + 2r^2$; see Figure 56). The remaining $n-3$ nodes must lie outside of the three overlapping circles of radius *r* centered at the three nodes (the red node, node X and node Y); the probability of this latter event is $\left[\left(A-B\right)/A\right]^{n-3}$, where B is the size of the area contained within the three overlapping circles. Since it is certainly the case that $\pi r^2 \le B$, and since we can choose the two connected nodes from any of pool of $n-1$ nodes, we have

prob(the red node is connected to two nodes and disconnected from all others)

$$< \binom{n-1}{2}\left(\frac{\pi r^2}{A}\right)\left(\frac{\pi r^2 + 2r^2}{A}\right)\left(\frac{A-\pi r^2}{A}\right)^{n-3} . \tag{59}$$



The largest possible area of the two overlapping circles is less than the sum of the areas of the two semicircles at each end ($\pi r^2$) plus the area of the rectangle between the two ends ($2r \times r = 2r^2$).

Figure 56. Area Calculation for Two Connected Nodes

In like manner, consider the probability that the red node is connected to three nodes and disconnected from all others. The closest node to the red node must lie within a circle of radius $r$ centered at the red node, the second closest node must lie within an area that is less than $\pi r^2 + 2r^2$ (see Figure 56), and the remaining node must lie within an area that is less than $\pi r^2 + 4r^2$ (see Figure 57). The remaining $n-4$ nodes must lie outside of the four overlapping circles of radius $r$ centered at the four connected nodes; the probability of this latter event is $\left[(A-B)/A\right]^{n-4}$, where B is the size of the area contained within the four overlapping circles. Since, as before, it is certainly the case that $\pi r^2 \le B$,

$$\text{prob(the red node is connected to three nodes and disconnected from all others)}$$
$$< \binom{n-1}{3}\left(\frac{\pi r^2}{A}\right)\left(\frac{\pi r^2 + 2r^2}{A}\right)\left(\frac{\pi r^2 + 4r^2}{A}\right)\left(\frac{A - \pi r^2}{A}\right)^{n-4}. \quad (60)$$

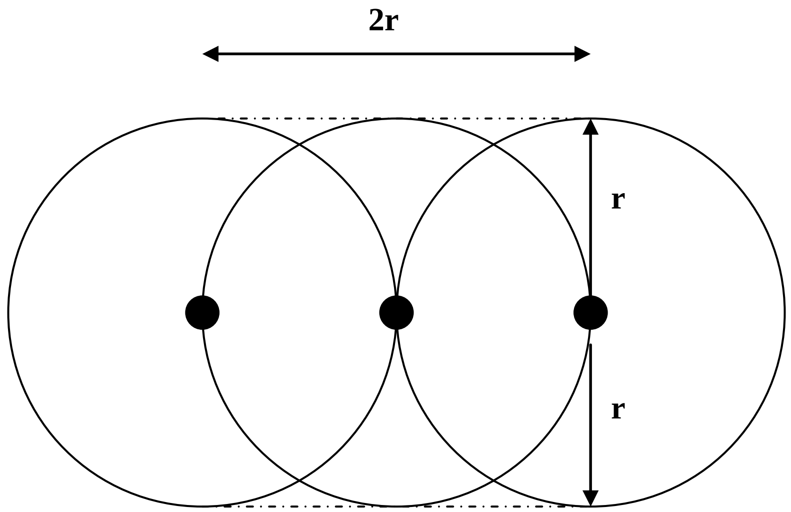


Figure 57. Area Calculation for Three Connected Nodes

In like manner, the probability that the red node is connected to $k$ nodes ($1 \le k \le m-1$) and disconnected from all others is bounded above by

$$\text{prob(red node is connected to } k \text{ nodes and disconnected from all others)} < \binom{n-1}{k}\left(\frac{A-\pi r^2}{A}\right)^{n-(k+1)} \prod_{j=1}^{k} \frac{\pi r^2 + 2(j-1)r^2}{A} \quad . \tag{61}$$

Substituting (57) and (61) into (56) yields

$$P_d(n,m) < \left(\frac{A-\pi r^2}{A}\right)^{n-1} + \sum_{k=1}^{m-1}\binom{n-1}{k}\left(\frac{A-\pi r^2}{A}\right)^{n-(k+1)} \prod_{j=1}^{k} \frac{\pi r^2 + 2(j-1)r^2}{A} \ , \tag{62}$$

and simplifying further yields

$$P_d(n,m) \ < \ \left(\frac{A-\pi r^2}{A}\right)^{n-1} \ + \ \sum_{k=1}^{m-1}\binom{n-1}{k}\left(\frac{A-\pi r^2}{A}\right)^{n-(k+1)} \left(\frac{r^2}{A}\right)^{k} \prod_{j=0}^{k-1}\{\pi + 2j\} . \tag{63}$$

Letting $x = \dfrac{\pi r^2}{A}$, we have

$$P_d(n,m) \ < \ (1-x)^{n-1} + \sum_{k=1}^{m-1}\binom{n-1}{k}(1-x)^{n-(k+1)} \left(\frac{r^2}{A}\right)^{k} \prod_{j=0}^{k-1}\{\pi + 2j\} \ . \tag{64}$$

Thus

$$P_c(n,m) > 1 - \left(1-x\right)^{n-1} - \sum_{k=1}^{m-1} \binom{n-1}{k} \left(1-x\right)^{n-1-k} \left(\frac{r^2}{A}\right)^k \prod_{j=0}^{k-1} \left\{\pi + 2j\right\} . \qquad (65)$$

As an example, Figure 58 shows a plot of $P_c(n,m)$ versus $m$ for the case of $A$ = 100, $r$ = 0.5 and $n$ = 1000. We see from the plot that an arbitrarily selected node is connected to at least two other nodes with a probability of 0.997, at least three other nodes with probability 0.977, and at least four other nodes with probability 0.861. Thus, if an application requires that a given node communicate with at least three other nodes (say, to form a 4-element array antenna), we can be reasonably assured that the required connectivity will exist.

As another example, consider that we have a number of sensor nodes to uniformly deploy over a region of size $A$ = 100, where each sensor node has a communication range of $r$ = 0.5. How many sensor nodes would we need to deploy in order to assure that an arbitrarily selected node is connected to at least four other nodes with a probability of 0.9? Figure 59 shows a plot of the probability that a node is connected to at least four other nodes as a function of the total number of nodes (i.e., Figure 59 is a plot of (65) for a fixed value of $m$). We find that the requirements are met for $n \geq 1064$ nodes.

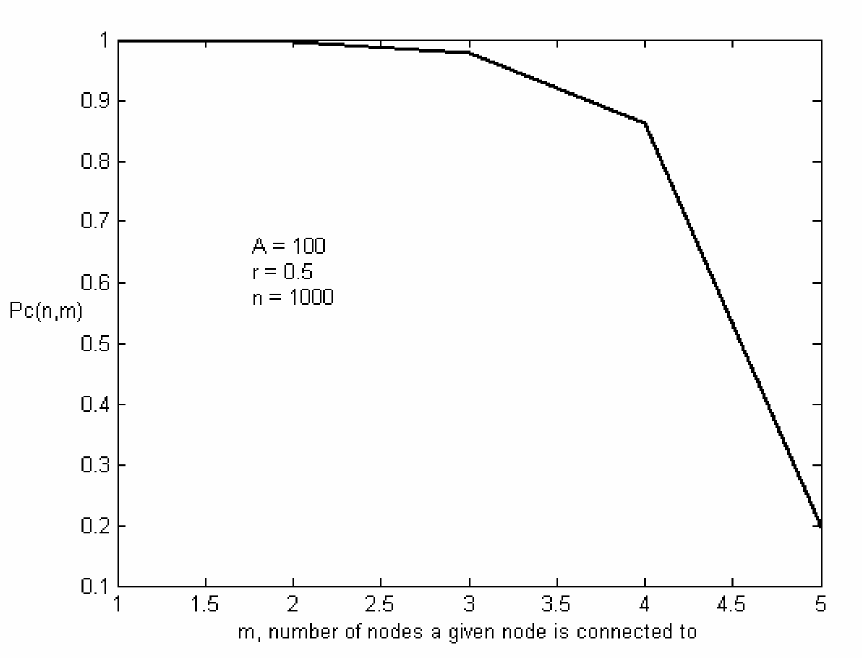


Figure 58. Plot of $P_c(n,m)$ versus $m$

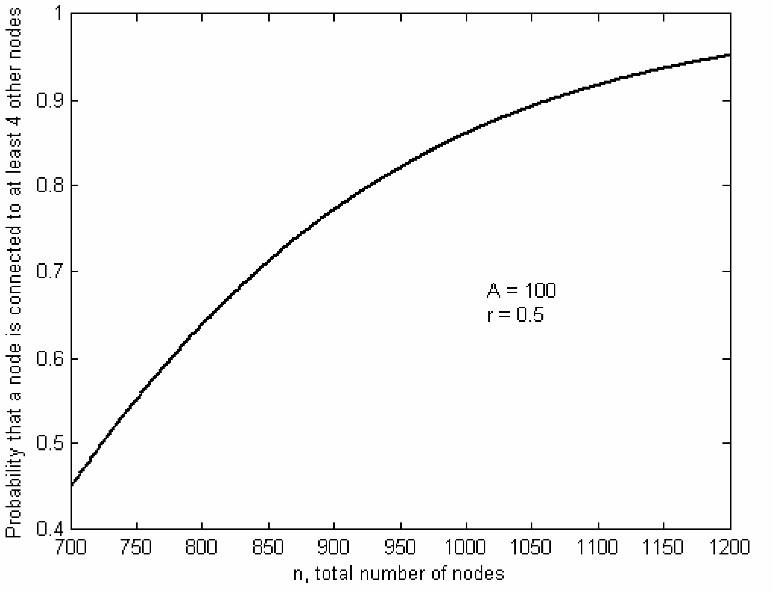


Figure 59. Probability that a Node is Connected to at Least Four Other Nodes

We note from (65) that, for a fixed number of nodes *n*, the probability that a node is connected to at least *m* other nodes decreases as *m* increases (as expected) since additional terms are then incorporated into the summation.

Since (65) is intended to bound a probability distribution function, it is useful to determine when the term on the right-hand side of the inequality behaves as a probability distribution function. For the right-hand side of (65) to be a valid distribution function, it must be the case that $P_c\,(n,\ m)$ is a non-deceasing function of $n$. Instead of determining when the bound on $P_c\,(n\ ,m)$ is a non-decreasing function of $n$, it is somewhat simpler to determine when the bound on $P_d\,(n,\ m)$, as given by (64), is a non-increasing function of $n$.

**Lemma.** $P_d\,(n,\ m)$ is a non-increasing function of $n$ as long as $n > (m-1)\dfrac{A}{\pi r^2}$.

**Proof.** $P_d\,(n,\ m)$ will be a non-increasing function of $n$ if $P_d(n,m) - P_d(n+1,m) > 0$. From (64),

$$\begin{aligned} &P_d(n,m) - P_d(n+1,m) = \\ &\qquad (1-x)^{n-1} - (1-x)^n + \sum_{k=1}^{m-1}\binom{n-1}{k}(1-x)^{n-(k+1)}\,D_k - \sum_{k=1}^{m-1}\binom{n}{k}(1-x)^{n-k}\,D_k \end{aligned} \tag{66}$$

where $D_k = \left(\dfrac{r^2}{A}\right)\prod_{j=0}^{k-1}\{\pi + 2j\}$. Now, since $(1-x) < 1$, it must be the case that $(1-x)^{n-1} - (1-x)^n > 0$. Thus, from (66), $P_d(n,m) - P_d(n+1,m) > 0$ whenever

$$\sum_{k=1}^{m-1}\binom{n-1}{k}(1-x)^{n-(k+1)}\,D_k \geq \sum_{k=1}^{m-1}\binom{n}{k}(1-x)^{n-k}\,D_k \quad . \tag{67}$$

Examining the first terms on both sides of the inequality (i.e., the term that applies for $k = 1$), the first term on the left-hand-side of (67) will be greater than the first term on the right-hand-side of (67) when $(n-1)(1-x)^{n-1} \geq n(1-x)^n$, or when

$$n \geq \frac{A}{\pi r^2} \quad . \tag{68}$$

Examining the second terms on both sides of the inequality (i.e., the term that applies for $k = 2$), the second term on the left-hand-side of (67) will be greater than the second term on the right-hand-side of (67) when $n \geq \dfrac{2A}{\pi r^2}$, which is a more restrictive condition than that given in (68). Continuing in like manner, examining the last terms on both sides of the inequality (i.e., the terms that apply for $k = m-1$), the last term on the left-hand-side of (67) will be greater than the last term on the right-hand-side of (67) when $n \geq \frac{(m-1)A}{\pi r^2}$. This latter condition provides the greatest restriction on $n$.

As an example, considering the scenario depicted in Figure 59, where sensor nodes having a communication range of $r = 0.5$ are deployed over an area of size $A = 100$, $P_c(n,4)$ will be monotonic non-decreasing (and asymptotically approaching one) as long as $n \geq \frac{(m-1)A}{\pi r^2} = 382$.

Thus, we have developed an analytical expression that specifies the number of sensor nodes that must be deployed to ensure that any node is connected to at least $m$ other nodes with a specified probability, assuming that the nodes have a given communication range and are uniformly distributed over a region of given size. Alternatively, our results can be used to determine a lower bound on the probability that an arbitrary node is connected to at least a specified number of other nodes, for a fixed number of total nodes. Using these results, network planners can implement algorithms that require sensor nodes to cooperate with (and hence have the ability to communicate with) a fixed number of other nodes.

## C. FORMATION OF LINEAR ARRAYS WITH RANDOM NODE PLACEMENT

We now present a technique for assembling a subset of sensor nodes into a distributed antenna array useful for beamforming. Recall that our approach entails a small subset of sensor nodes receiving and aggregating information gathered by the network and forming a distributed antenna array, concentrating the radiated transmission into a narrow beam aimed towards the UAV. Although, in general, the relative positions of the elements in a distributed antenna array have an effect on antenna performance, our proposed approach can be employed in scenarios where the individual sensor nodes do not have knowledge of their location within an absolute coordinate system.

The utility of many sensor network applications hinges on the ability of sensor nodes to determine their absolute positions (e.g., latitude and longitude) within some coordinate system—a process termed *localization*. For sensor networks employed in commercial and environmental applications (e.g., [3], [8]), the localization problem is readily solved. For example, scientists positioning sensor nodes on the ground can preprogram the precise geographic position into the node. Similarly, large and expensive nodes that can be serviced in the field can be outfitted with Global Positioning System (GPS) receivers.

Such approaches to solving the localization problem are not applicable to sensor networks employed in a variety of foreseeable military scenarios. Recall that our applications entail a large number of sensor nodes being dropped from an aircraft or unmanned aerial vehicle (UAV), densely covering an area of interest. In such scenarios the nodes cannot have their positions preprogrammed since their final landing locations will not be known *a priori*. Similarly, in other applications, we may wish to deploy sensor networks in dangerous or remote environments, where node batteries cannot be serviced after deployment, and in still other applications we may require that sensor nodes be small (i.e., covert) and low-cost (i.e., expendable), necessitating small low-cost batteries. In scenarios such as these, the strong premium placed on battery power conservation, node cost and node size prohibit the use of onboard GPS receivers.

Many sensor network applications discussed in the literature propose solving the localization problem through *beaconing*. A subset of the network's sensor nodes are permitted to be larger, more expensive, and more capable, and are thus assumed able to precisely determine their own absolute positions (e.g., by using GPS). These nodes, termed *beacon nodes*, then broadcast their position to neighboring nodes. Using information from multiple beacons, other sensor nodes can then, in turn, estimate their own positions. As a coarse approximation, a node may estimate its position as the centroid of all of the beacons it hears [101]. A different approach [102] allows the non-beacon nodes to estimate their positions from knowledge about the presence or absence of communication links: All of the connectivity information is passed to a central computer, which then estimates locations for all the nodes by solving a convex optimization problem (the presence of a connection imposes a proximity constraint). If *r* is the maximum range between nodes, the research presented in [101] shows that the locations of about 10% of the nodes must be known (i.e., must be beacons) in order for the remaining nodes to determine their positions within a mean error of *r*. If the locations of only 1% of the nodes are known, the position estimation algorithm performs as poorly as random guessing.

Distributed schemes that allow nodes to determine their positions if a subset of nodes (beacons) initially have their absolute geographic coordinates are presented in [103], [104], [105]. The basic idea is that the beacon nodes, as before, broadcast their positions. The neighboring nodes measure their distances to the beacon nodes and use multilateration to determine their positions. These nodes then become new beacon nodes, and transmit their positions to their neighbors, with the process repeating in an iterative (fully distributed) fashion. It is shown in [106] that for a 300-node network, with 10% of the nodes as beacons, 40% of the nodes will not resolve their positions (but those that do will resolve their position to an accuracy of 2 cm). An extensive study in [107] determined that localization schemes based solely on ranging require that the average degree of a node in the network be 11-12 in order to achieve even modest success (90% of nodes localized to within 20% of their actual positions with an average mean error in position of 5%).

In summary, providing sensor nodes in military applications with precise positioning information is quite difficult, and applications that depend on precise positioning information can not be relied upon to guarantee performance in critical scenarios. In this section, we propose an approach to enabling communications between a sensor network and an Unmanned Aerial Vehicle (UAV) in scenarios wherein the individual sensor nodes do not have, and do not require, knowledge of their location within an absolute coordinate system. Specifically, we present an approach for formation of a distributed antenna array for beamforming.

We have proposed, in Chapter II, that sensor nodes coordinate their transmissions to form a distributed antenna array that directs a beam towards a UAV, resulting in an increased signal-to-noise ratio (SNR) at the UAV (or an increased communication range for a fixed SNR). A small subset of sensor nodes, termed a *transmit cluster*, receives and aggregates sensor information gathered by the network, consolidating the collected data into a packet. The operations of the transmit cluster are controlled by a distinct node selected as the *transmit clusterhead* (*transmit CH*). For this plan to work, the narrow transmission beam must be directed such that the UAV falls spatially within it. In Chapter III, we proposed that the power-constrained sensor nodes should not be burdened with finding the highly capable UAV; instead, the burden of aligning the transmission beam is shifted to the UAV. The transmit CH organizes nodes into an antenna array and calculates the amplitude and phase offsets needed for the resultant beam of the desired gain to point straight up (at 0° elevation). Note that the precise algorithm for determining the complex weights is not addressed in this dissertation and is left for future research. These values, calculated only once, are passed to the participating nodes, but the distributed antenna then "stands down," waiting for the UAV to find it. But, the question remains, how can a set of nodes form a distributed antenna if they lack information about their relative positions? This is the question we presently examine.

Suppose that we wish to use $M$ nodes to form a linear array, as shown in Figure 31. Note from our development in Chapter IV that computation of the radiation pattern for a linear array requires that the relative positions of the participating nodes be known. Specifically, we must know the inter-element spacings (i.e., the values of $x_2$, $x_3$, ...$x_M$) in Figure 31, and we must be assured that the array elements lie on a line.

Thus, the question to be considered is: How can nodes that lack position information successfully organize themselves into a distributed antenna array? Note that nodes that lack knowledge of their position can determine their relative separations, but can not readily determine their positions; if, for example, four nodes are able to determine that their separation from each neighbor is 1 unit, the nodes will not be able, absent other information, to determine if their physical layout is as shown in Figure 60(a) or Figure 60(b) (or some other configuration). Without knowing their relative positions, the nodes can not determine the power radiation pattern.

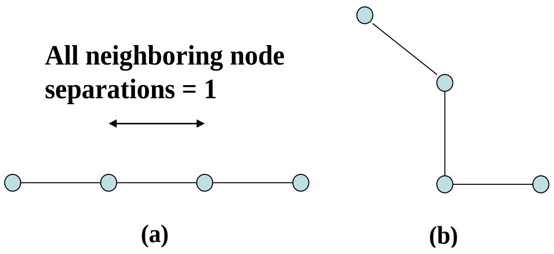


Figure 60. Different Node Layouts

To illustrate our solution approach, consider the three nodes shown in Figure 61(a). The nodes are able to measure their inter-node distances, $d_{12}$, $d_{13}$ and $d_{23}$, as shown. Suppose it is the case that $d_{12} + d_{23} = d_{13}$. Then the actual layout must, in fact, be as shown in Figure 61(b), and these nodes, through simple data exchanges, can determine that they are oriented on a line.

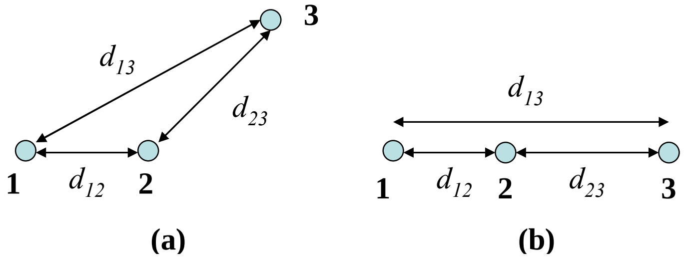


Figure 61. Linearity Condition

Our preliminary algorithm thus operates as follows: A node is selected, based on battery power remaining, to function as a "seed" for "growing" a linear array of the required size. Through local data exchanges, the seed node can first determine the 3-node distributed antenna (containing the seed node) that comes closest to satisfying the linearity condition (Figure 61(b)) while also assuring that the inter-node separations are close to equal. Mathematically, if $S$ is the seed node and $d(x, y)$ represents the distance between nodes $x$ and $y$, we determine the nodes $i$ and $j$ that are neighbors to $S$ that minimize $\alpha \mid d(S, i) + d(S, j) - d(i, j) \mid + \beta \mid d(S, i) - d(S, j) \mid$, where $\alpha$ and $\beta$ are appropriately chosen weights and $\mid x \mid$ denotes the absolute value of $x$. The foregoing calculation assumes that the seed is at the center of the 3-node array; we perform a similar calculation that examines cases in which the seed is an end node of the array. Once the best 3-node array is selected, each of the two end nodes implements a similar algorithm to determine which new node should be added, and the process iterates until an array of the required length is constructed.

This algorithm was implemented in the MATLAB programming language. In our algorithm, the seed node is chosen as that node in the largest connected component possessing the largest node degree (i.e., the most neighbors). Figure 62 shows, for example, the 6-node array (in green) formed when 75 nodes with a communication range of 2 are uniformly distributed over a $10 \times 10$ area. (Node 73 is the seed; $\alpha = \beta = 1$, and only the applicable portion of the network is shown).

Note that this algorithm is highly distributed, with each node, at each iteration, only exchanging data with immediate neighbors. The nodes then form a beam assuming that they are arranged on a straight line (recall that the nodes know their inter-node spacing). Since the nodes are not actually arranged precisely on a straight line (see Figure 62), the performance will be less than what the array "expects."

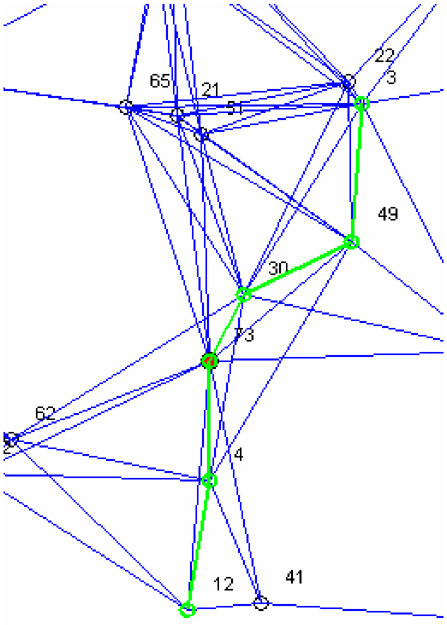


Figure 62. Six-Node Array

As an example, consider the following scenario : 80 sensor nodes are uniformly (randomly) distributed over a 10 unit × 10 unit planar region. Each node has a communication range of 2 units. We desire that nodes form a five-element linear array. Using our algorithm, the five-element array is shown in green in Figure 63 (as before, $\alpha = \beta = 1$, and only the applicable portion of the network is shown).

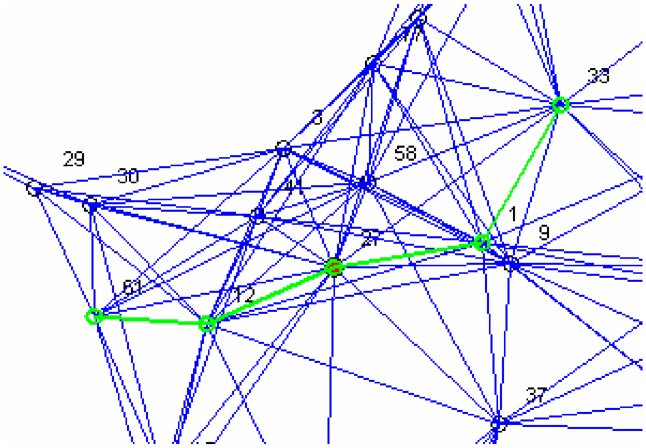


Figure 63. Five-Node Array

Note that the five nodes chosen for the array happen to be 33, 1, 27, 12 and 61. The inter-element spacings are 1.00128, 0.94983, 0.89247 and 0.70744, respectively. (Recall that nodes can determine their separations, but not their relative positions.) Thus, these nodes "believe" that they are the linear array shown in Figure 64. (Compare this to the actual configuration shown in Figure 63.)

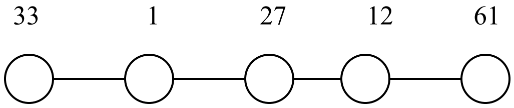


Figure 64. Idealized Linear Array

As an example, suppose the communication frequency was chosen to be 315 MHz. In Figure 65, we compare the linear array of Figure 64 to the five-element linear array with equal inter-element spacings of half of a wavelength; the normalized gain for this latter array is shown in black in Figure 65. The normalized gain for the linear array given the actual spacings (Figure 64) is shown in red. Since the nodes in the sensor network only know their inter-element spacings but not their relative positions, the curve in red would approximate what the sensor network "believes" to be its gain given the operating frequency.

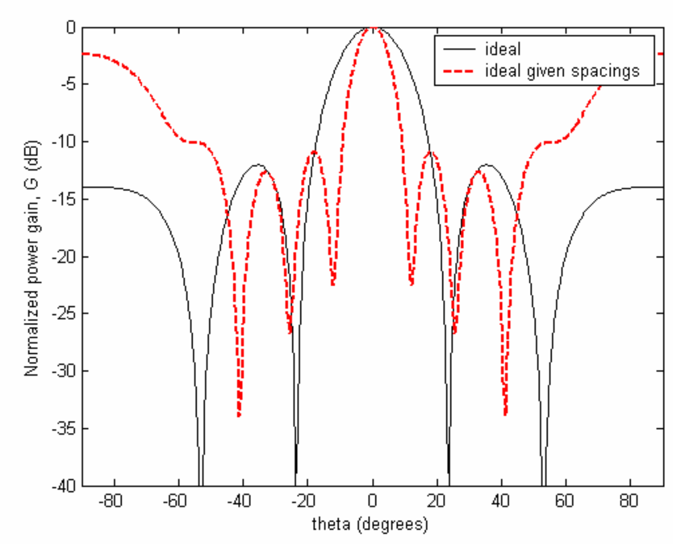


Figure 65. Normalized Gain

In Figure 66, we compare the linear array of Figure 64 to the actual five element array (Figure 63). The normalized power gain for the ideal linear array given the actual spacings (Figure 64) is shown in red (this is the same curve shown in red in Figure 65). The black curve in Figure 66 is the actual radiation pattern given the actual element locations shown in Figure 63. The curve is taken through the plane that runs through a straight-line fit of the sensor node's relative positions. We note that the shape of the main lobe and the 3 dB beamwidth of the main lobe are almost unaffected by the lack of relative position information, but the actual array's sidelobe performance is several dB worse.

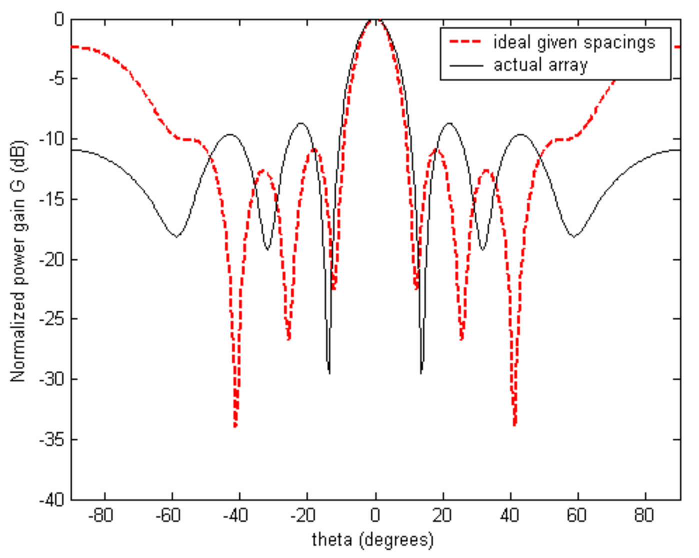


Figure 66. Normalized Gain

It is interesting to note that the normalized gains and 3 dB bandwidths are approximately the same at $\theta$ = 0º for both of the curves shown in Figure 66. That is, at $\theta$ = 0º, the actual array (whose nodes are not on a straight line) has approximately the same power gain as the linear array with the same inter-node spacings. Upon further reflection, it is noted that this is as it should be. Consider the planar array shown in Figure 32. If the UAV is directly overhead ($\theta$ = 0º), then the arriving (planar) waveform arrives at all elements at the same time; their actual positions in the plane become irrelevant. This fact can also be noted from (46): Substituting $\theta$ = 0º into (46) yields the maximum array factor regardless of the array positions, since each term in the summation will evaluate to unity.

Recall that we assume the sensor node antennas each radiate isotropically. If the actual patterns of the individual array elements were known (and all alike), then the principle of pattern multiplication could be applied to determine the actual array pattern.

In this chapter, we have presented a technique for assembling a subset of sensor nodes into a distributed antenna array useful for beamforming. Although, in general, the relative positions of the nodes in a distributed antenna array have a great effect on antenna performance, our proposed approach can be employed in scenarios where the individual sensor nodes do not have knowledge of their location within an absolute coordinate system. Our algorithm seeks an optimal linear array, given the actual random distribution of nodes. Assuming isotropic radiators, we find that the shape of the main lobe and the 3 dB beamwidth of the main lobe are almost unaffected by the lack of knowledge of the relative positions of the array elements, but the actual array's sidelobe performance is worse than would be the case had the nodes been arranged on a line.

THIS PAGE INTENTIONALLY LEFT BLANK

# VI. CONCLUSIONS

In this closing chapter, we present a summary of the research, highlight the significant contributions and suggest topics for future research that will build upon the ideas presented in this dissertation.

## A. SUMMARY OF RESEARCH

We briefly review the sensor network employment scenario: After deployment, the sensor nodes awaken and partition themselves into non-overlapping clusters. Each cluster is managed by a fixed single node designated as the clusterhead (CH). A specified cluster, termed the *transmit cluster*, is selected from among the many clusters to act as the transmission point for the sensor network. Any sensed data originating anywhere in the network are routed to the transmit cluster's CH (termed the *transmit CH*), where it is consolidated with any data already collected and prepared for transmission to the UAV. The transmit CH organizes a subset of its cluster nodes into a distributed antenna array and calculates the phase offsets to be applied to the otherwise identical transmissions of the participating nodes in order to form a radiation pattern that concentrates the power into a narrow beam aimed towards the UAV.

For military applications, sensor nodes should be small (so as to be covert) and inexpensive (in order to be expendable). Furthermore, such applications presume that the hardware of a sensor node will not be serviced in the field; when a sensor's battery fails, the whole sensor node is rendered inoperable. Thus, careful power management is critical in order to extend the lifetime of the sensor network.

In this dissertation, we presented a system-level approach for minimizing the power expended in achieving successful communication between a ground-based sensor network and an overhead UAV.

We divided our problem into a series of layers, starting at the highest level of abstraction. Specifically, we first noted that since the duties of the transmit cluster are power-intensive, the role of transmit cluster would need to be shifted to different clusters as time progressed in order to evenly distribute the energy load and extend the lifetime of the network. We determined methods for moving the distributed antenna about the sensor network in a manner that broadly spread the energy load across the network, while minimizing the energy expended in moving the antenna.

Once this problem was examined, we moved down to the next layer of abstraction and determined how the distributed antenna's beam should be aligned with the UAV in a manner that minimizes the energy expenditure.

Having developed an energy-efficient algorithm to move the distributed antenna to other locations in the sensor network to evenly spread the energy load and extend the lifetime of the network, and having also developed an energy-efficient approach to aligning the distributed antenna's beam with the UAV, we then examined how the sensor network should assemble nodes into a distributed antenna in an energy-efficient manner.

Finally, we analyzed our design approach in the face of randomness by proposing a clustering methodology for sensor networks, analyzing the probability that a cluster will be able to form an antenna array given a random placement of sensor nodes, and presenting an algorithm for the formation of near-linear arrays.

## B. SIGNIFICANT CONTRIBUTIONS

Because the duties of the transmit cluster are power-intensive, the role of transmit cluster must be shifted to different clusters as time progresses in order to evenly distribute the energy load and extend the lifetime of the network. In Chapter II, we presented a method for more uniformly distributing the energy burden across a wireless sensor network communicating with an overhead UAV. We presented an algorithm to reassign the transmit cluster, specifying the time that should elapse between reassignments and the number of hops that should be placed between successive transmit clusters in order to achieve three competing goals: First, we wish to better and more broadly spread the energy load across the sensor network while, second, minimizing the energy expended in moving the transmit cluster, all the while, third, reducing to the extent practicable the time to bring the UAV and the sensor network's beam into alignment. The algorithm thus extends the lifetime of the sensor network while meeting system-level performance objectives.

For successful communication to be achieved, the transmit cluster's narrow transmission beam must be directed such that the UAV falls spatially within it. Military applications must presume that the network does not know a priori where the UAV is, nor does the UAV know the direction from which the sensor network has aimed its transmission beam. Techniques that an antenna array can employ to estimate the direction of the UAV are complex and usually task the UAV with transmitting a known reference signal while the antenna array steers a very narrow beam continually in space, analyzes the spatial spectrum as a function of position, and selects the direction that yields the highest power as the direction of the UAV. This requires that the transmit CH continually calculate different sets of amplitudes and phases and transmit these to the participating nodes comprising the array, and these nodes, in turn, will have to continually send their receptions to the transmit CH for analysis. All these operations involve a large number of radio transmissions among sensor nodes, expending scarce battery resources.

In Chapter III, we proposed that the power-constrained sensor nodes should not be burdened with finding the highly capable UAV; instead, the burden of aligning the transmission beam is shifted to the UAV. The transmit CH organizes nodes into an antenna array and calculates the phase offsets needed for the resultant beam of the desired gain to point straight up (at 0° elevation). These values, calculated only once, are passed to the participating nodes, but the distributed antenna then "stands down," and the transmit CH functions as an omnidirectional antenna again, waiting for the UAV to find it. The UAV, meanwhile, travels over the region transmitting a known reference signal, pointing its transmission beam straight down. If the transmit CH should fall within the UAV's beam, the transmit CH will then send the data stream (containing the consolidated data gathered by the sensor network) to the elements in the antenna array, and these elements will, in turn, transmit the signal using the precomputed phases.

In Chapter III we described and analyzed two alternative strategies that bring the UAV and the sensor network's beam into alignment. Both schemes seek to minimize, to the extent practicable, the energy expended by the sensor network in aligning the beam with the UAV. One strategy employs a very capable UAV and ensures that the beam is aligned with the UAV, also guaranteeing that it will be done as quickly as possible given that alignment is to occur. The second scheme allows the use of a less capable UAV, and, while not guaranteeing alignment, will find the beam more quickly in a variety of scenarios. The performance of the two strategies was compared in terms of probability of beam-UAV alignment as a function of time and the expected time to alignment. We examined the performance tradeoff between the choice of strategy, on the one hand, and those parameters of the sensor network that affect power conservation, on the other hand.

Thus, a subset of sensor nodes is tasked with forming a distributed antenna array, and performs only a single set of computations necessary to aim the resulting beam straight up. In Chapter IV, we analyzed the optimal number of sensor nodes that should participate in a beamforming process in order to minimize the energy expended by nodes in a sensor network. We noted that, as we increase the number of sensor nodes involved in beamforming (for a fixed energy per bit to noise power spectral density required at the receiver), the transmit power of each node is significantly reduced, but the overhead in implementing even a simple collaborative algorithm increases. We provided a framework that a system designer can use to analyze the minimum energy expended in a simple beamforming algorithm, which, in turn, specifies the optimal number of nodes to use to minimize the total energy expended, and thus extend the lifetime of a sensor network.

We then analyzed our design approach in the face of randomness. We proposed a clustering methodology for sensor networks and differentiated sensor networks from mobile ad hoc networks. In Chapter V, we developed an analytical expression that specifies the number of sensor nodes that must be deployed to ensure that any node is connected to at least m other nodes with a specified probability, assuming that the nodes have a given communication range and are uniformly distributed over a region of given size. Alternatively, our results can be used to determine a lower bound on the probability that an arbitrary node is connected to at least a specified number of other nodes, for a fixed number of total nodes. These results provide a tool that network planners can use to implement algorithms that require sensor nodes to cooperate with (and hence have the ability to communicate with) a fixed number of other nodes. Finally, we have presented a technique for assembling a subset of sensor nodes into a distributed antenna array useful for beamforming. Although, in general, the relative positions of the elements in a distributed antenna array have an effect on antenna performance, our proposed approach can be employed in scenarios where the individual sensor nodes do not have knowledge of their location within an absolute coordinate system. Our algorithm seeks an optimal linear array, given the actual random distribution of nodes. We find that the shape of the main lobe and the 3-dB beamwidth of the main lobe are almost unaffected by the lack of knowledge of the relative positions of the array elements, but the actual array's sidelobe performance is worse than would be the case, given if the nodes were arranged on a line.

## C. FUTURE RESEARCH

We suggest topics for further research that would build upon the ideas presented in this dissertation.

In our development, the BER required at the UAV is fixed at some required value. This fixed receiver BER fixes, in turn, the value of $G/h^2$, where $G$ is the gain of the sensor network's distributed antenna, and $h$ is the UAV altitude. So, by requiring a fixed BER, $G$ and $h$ cannot vary independently of each other. Further research would develop a more complex search scheme that recognizes the tradeoffs between BER, sweep width, UAV altitude, probability of UAV shootdown, and the time to complete the search. If the UAV is at a higher altitude, it has a greater sweep width, faster search time and a lower probability of being shot down (all good), but the UAV's receiver will suffer a greater BER. If a UAV is at a lower altitude, it has less sweep width, takes a longer time to complete the search, and stands a greater chance of being shot down (all bad), but the UAV's receiver has an improved BER.

Our proposed algorithm moves the transmit cluster in order to extend the lifetime of the sensor network. The analytical results derived from the mathematical model, while intuitively appealing and understandable, might be verified with a simulation study. Specifically, we note that the energy to move the transmit cluster can be quite high. The energy benefits inherent in our algorithm might be compared to an alternative approach that leaves the location of the transmit cluster fixed. The lifetime of the sensor network under both conditions might then be compared under varying network loads.

In shifting the transmit CH to a new location, the current algorithm to choose the next transmit CH makes a random selection from among those CH's a specified number of hops away. More sophisticated strategies might be considered: First, a node which previously served as a transmit CH might not ordinarily serve a second time and, second, the replying cluster whose CH has the highest remaining energy might be selected as the new transmit CH. Additionally, the analysis of the algorithm to shift the transmit cluster might also be extended to account for link failures and packet collisions.

Proper synchronization is required in order to implement the phase differences required for optimal beamforming. Although we demonstrated and discussed the errors induced by random node positioning, our investigation did not address the difficulty in synchronizing the oscillators in the various sensor nodes. (Also, we did not address an even more basic question: How does a sensor node that lands on the ground determine which way is up (i.e., skyward)? ) Further research would also attempt to incorporate representative element patterns into the analysis of the array pattern (we assumed isotropic radiators).

We presented an algorithm that seeks an optimal linear array, given the actual random distribution of sensor nodes in a plane. Our algorithm, which limits a node's communication to only its neighbors, is focused on conserving battery power. Further research would explore whether investing more battery power (e.g., allowing nodes to coordinate actions with nodes that are several hops away) leads to the construction of significantly improved arrays, and whether such power investments are worthwhile. Further research would also explore extending these algorithms to seek optimal planar arrays.

Finally, these ideas could be extended to include the possibility that nodes have the capability of repositioning themselves after initial deployment. Such movement might occur based on a node's perception of its surroundings and capabilities, or be directed by external commanders far from the scene.

# LIST OF REFERENCES

[1] D. D'Agostino, “Sense and Sensibility; Sensor Networks are not Quite Enterprise-ready, but it's not too Early to Start Evaluating this Important Technology,” *CIO Insight*, 1(47), 61, 2004.

[2] R. Szewczyk, E. Osterweil, J. Polastre, M. Hamilton, A. Mainwaring and D. Estrin, “Habitat Monitoring with Sensor Networks,” *Communications of the ACM,* vol. 47, no. 6, pp. 34-40, 2004.

[3] R.C. Johnson, “Futuristic Factories Make Mesh,” *Electronic Engineering Times,* (1333), 1, 2004.

[4] R. Allan, “Wireless Sensors Land Anywhere and Everywhere,” *Electronic Design,* vol. 53, no. 16, pp. 65-70, 2005.

[5] A. Ricadela, “Sensors Everywhere,” *InformationWeek,* no. 1023, p. 32, 2005.

[6] W.J. Broad, “A Web of Sensors, Taking Earth's Pulse,” *New York Times,* p. F.1, May 10, 2005.

[7] C.Y. Chong and S. Kumar, “Sensor Networks: Evolution, Opportunities, and Challenges,” *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1247-1256, August, 2003.

[8] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, “A Survey on Sensor Networks,” *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102-114, 2002.

[9] B.A. Forouzan*, Data Communications and Networking, 4th ed.*, McGraw Hill, New York, 2007.

[10] P.J. Vincent, M. Tummala and J. McEachen, “A New Method for Distributing Power Usage across a Sensor Network,” *Proceedings of the Third Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON 2006)*, pp. 518-526, 2006.

[11] J.W. Lowe, “Sizing up the Mighty Mote,” *GeoSpatial Solutions,* vol. 14, no. 5, pp. 42-45, 2004.

[12] J. Hill, M. Horton, R. Kling and L. Krishnamurthy, “The Platforms Enabling Wireless Sensor Networks,” *Communications of the ACM, vol. 47, no.* 6, pp. 41-46, 2004.

[13] J. Edwards, “Sensors Working Overtime,” *CFO,* p. 1, August 2005.

[14] L.W. Couch, *Digital and Analog Communication Systems* (3rd ed.), Collier Macmillan, New York, 1990.

[15] T. S. Rappaport, *Wireless Communications : Principles and Practice* (2nd ed.), Prentice Hall, New Jersey, 2002.

[16] G. Pottie and W. Kaiser, “Wireless Integrated Network Sensors,” *Communications of the ACM,* vol. 43, no. 5, pp. 51-58, 2000.

[17] S. Tilak, N.B. Abu-Ghazaleh and W. Heinzelman, “A Taxonomy of Wireless Micro-sensor Network Models,” *Mobile Computing and Communications Review,* vol. 6, no. 2, pp. 28-36, 2002.

[18] M. Tummala, C. Wai and P. Vincent, “Distributed Beamforming in Wireless Sensor Networks,” *Proceedings of the 39th IEEE Asilomar Conf. on Signals, Systems and Computers*, pp. 793-797, 2005.

[19] L.V. Blake, Antennas, John Wiley and Sons, NY, 1966.

[20] J. Litva and T.K. Lo, *Digital Beamforming in Wireless Communications*, Artech House, Norwood, MA, 1996.

[21] L.C. Godara, “Application of Antenna Arrays to Mobile Communications, Part II: Beamforming and Direction-of-Arrival Considerations,” *Proceedings of the IEEE*, vol. 85, no. 8, pp. 1195 – 1245, 1997.

[22] D. Bertsekas and R. Gallager, *Data Networks*, Prentice Hall, N.J., 1992.

[23] J. Kulik, W. Heinzelman and H. Balakrishan, “Negotiation-Based Protocols for Disseminating Information in Wireless Sensor Networks, *Wireless Networks*, vol. 8, pp. 169-185, 2002.

[24] Y.C. Tseng, S.Z. Ni, Y.S. Chen and J.P. Sheu, “The Broadcast Storm Problem in a Mobile Ad Hoc Network,” *Wireless Networks*, vol. 8, pp. 153-167, 2002.

[25] W. Stallings, *High-speed Networks and Internets : Performance and Quality of Service* (2nd ed.), Prentice Hall, Upper Saddle River, N.J. 2002.

[26] D.J. Baker, A. Ephremides and J.A. Flynn, “The Design and Simulation of a Mobile Radio Network with Distributed Control,” *IEEE Journal on Selected Areas in Communications*, vol. SAC-2, no. 1, pp. 226-237, January 1984.

[27] M. Gerla and J.T. Tsai, “Multicluster, Mobile, Multimedia Radio Network,” *Wireless Networks*, vol. 1, no. 3, pp. 255-265, 1995.

[28] C.R. Lin and M. Gerla, “Adaptive Clustering for Mobile Wireless Networks,” *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 7, pp. 1265-1275, September 1997.

[29] K. Sohrabi, J. Gao, V. Ailawadhi and G.J. Pottie, “Protocols for self-organization of a wireless sensor network,” *IEEE Personal Communications, 7*(5), 16-27, 2000.

[30] D. Niculescu, “Communication Paradigms for Sensor Networks,” *IEEE Communications Magazine*, March 2005, pp. 116-122.

[31] S. Keshav, *An Engineering Approach to Computer Networking*, Addison-Wesley, MA, 1997.

[32] D. West, *Introduction to Graph Theory*, Prentice Hall, N.J., 2001.

[33] P.J.Vincent, M.Tummala and J. McEachen, “An Energy-efficient Approach for Information Transfer from Distributed Wireless Sensor Systems,” *Proceedings of the 2006 IEEE International Conference on Systems of Systems Engineering*, pp. 100-105, 2006.

[34] B.O. Koopman, *Search and Screening*, Pergamon Press, New York, 1980.

[35] L.D. Stone, *Theory of Optimal Search*, Academic Press, New York, 1975.

[36] S. Benkoski, M. Monticino and J. Weisinger, “A Survey of the Search Theory Literature,” *Naval Research Logistics*, Vol. 38, pp. 469-494, 1991.

[37] V.J. Lumelsky, S. Mukhopadhyay and K. Sun, “Dynamic Path Planning in Sensor-Based Terrain Acquisition,” *IEEE Transactions on Robotics and Automation*, vol. 6, no. 4, pp. 462-472, 1990.

[38] S. Hert, S. Tiwari and V. Lumelsky, “A Terrain-Covering Algorithm for an AUV,” *Autonomous Robots*, vol. 3, no. 2, pp. 91-199, 1996.

[39] V. Ablavsky and M. Snorrason, “Optimal Search for a Moving Target: A Geometric Approach,” *Proceedings of the AIAA Guidance, Navigation and Control Conference*, American Institute of Aeronautics and Astronautics, 2000.

[40] J. Kurabayashi, J. Ota, T. Arai and E. Yoshida, “Cooperative Sweeping by Multiple Mobile Robots,” *Proceedings of the IEEE International Conference on Robotics and Automation*, IEEE Computer Society Press, pp. 1744-1749, 1996.

[41] H. Choset, “Coverage for Robotics-A Survey of Recent Results,” *Annals of Mathematics and Artificial Intelligence*, Vol. 31, pp. 113-126, 2001.

[42] B.O. Koopman, “The Theory of Search II, Target Detection,” *Operations Research*, vol. 4, no. 5, pp. 503-531, 1956.

[43] H.M. Taylor and S. Karlin, *An Introduction to Stochastic Modeling*, Academic Press, California, 1994.

[44] G.G Skitek and S.V. Marshall, *Electromagnetic Concepts and Applications*, Prentice Hall, Englewood Cliffs, N.J., 1982.

[45] Department of Defense Office of the Secretary of Defense Memorandum, U*nmanned Aerial Vehicles Roadmap 2005-2030*, dated August 4, 2005. Available at http://www.acq.osd.mil/usd/Roadmap%20Final2.pdf (link checked December 10, 2006).

[46] P. Vincent, M. Tummala and J. McEachen, “Optimizing the Size of an Antenna Array,” *Proceedings of the 40th Annual Asilomar Conference on Signals, Systems and Computers*, 2006.

[47] W.L. Stutzman and G.A. Thiele, “Antenna Theory and Design,” John Wiley and Sons, NY, 1998.

[48] S. Ramo, J.R. Whinnery and T. Van Duzer, *Fields and waves in communication electronics* (2nd ed.), Wiley, New York, 1984.

[49] B. Sklar, *Digital Communications* (2nd ed.), Prentice Hall, 2001.

[50] B.D. Steinberg, *Principles of Aperture and Array System Design*, John Wiley and Sons, NY, 1976.

[51] E. Shih, S. Cho, N. Ickes, R. Min, A. Sinha, A. Wang and A. Chandrakasan, “Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks,” Proceedings of the *7th annual ACM international conference on mobile computing and networking (MobiCom’01)*, pp. 272-286, 2001.

[52] A. Woo and D.E. Culler, “A transmission control scheme for media access in sensor networks,” *Proceedings of the 7th annual ACM international conference on mobile computing and networking (MobiCom’01),* pp. 221-235, 2001.

[53] A. Sinha and A. Chandrakasan, “Dynamic power management in wireless sensor networks,” *IEEE Design and Test of Computers,* vol. 18, no. 2, pp. 62-74, 2001.

[54] A. Wang and A.P. Chandrakasan, "Energy-aware architectures for a real-valued FFT implementation," *Proceedings of the 2003 IEEE International Symposium of Low Power Electronics and Design (ISLPED'03)*, pp. 360-365, 2003.

[55] E. Shih, S. Cho, F.S. Lee, B.H. Calhoun and A. Chandrakasan, "Design considerations for energy-efficient radios in wireless microsensor networks," *Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology,* vol. 37, no. 1, pp. 77-94, 2004.

[56] R. Min, M. Bhardwaj, S. Cho, E. Shih, A. Sinha, et al., "Low-Power Wireless Sensor Networks," *Proceedings of the 14th IEEE International Conference on VLSI Design*, pp. 205-210, 2001.

[57] D. Niculescu, "Communication Paradigms for Sensor Networks," *IEEE Communications Magazine*, March 2005, pp. 116-122.

[58] B. Krishnamachari, D. Estrin and S. Wicker, "The Impact of Data Aggregation in Wireless Sensor Networks," *Proceedings of the 1st International Workshop on Distributed Event Based Systems*, 2002.

[59] M.R. Garey and D.S. Johnson, "Computers and Intractability, A Guide to the Theory of NP-Completeness," W.H. Freeman and Company, 1979.

[60] A.S. Tanenbaum, *Computer networks* (4th ed.), Prentice Hall PTR, 2003.

[61] C.E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," *ACM SIGCOMM Computer Communication Review*, vol. 24, no. 4, pp. 234-244, 1994.

[62] D.B. Johnson, D.A. Maltz and J. Broch, "DSR: The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks," in *Ad Hoc Networking* (C.E. Perkins, ed.), Chapter 5, pp. 139-172, 2001.

[63] M. Abolhasan, T. Wysocki and E. Dutkiewicz, "A Review of Routing Protocols for Mobile Ad Hoc Networks," *Ad Hoc Networks*, vol. 2, no. 1, January 2004.

[64] S. Singh, M. Woo and C.S. Raghavendra, "Power-Aware Routing in Mobile Ad Hoc Networks," *Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM'98)*, pp. 181-190, 1998.

[65] J.M. Rabaey, M.J. Ammer, J.L.J. da Silva, D. Patel and S. Roundy, "PicoRadio supports ad hoc ultra-low power wireless networking," *Computer,* vol. 33, no. 7, pp. 42-48, 2000.

[66] B.H. Calhoun, D.C. Daly, N. Verma, D.F. Finchelstein, D.D. Wentzloff, A. Wang, et al., “Design considerations for ultra-low energy wireless microsensor nodes,” *IEEE Transactions on Computers,* vol. 54, no. 6, pp. 727-740, 2005.

[67] V. Rodoplu and .H. Meng, “Minimum Energy Mobile Wireless Networks,” *IEEE Journal on Selected Areas of Communications*, vol. 17, no. 8, pp. 1333-1344, August 1999.

[68] J.H. Chang and L. Tassiulas, “Maximum Lifetime Routing in Wireless Sensor Networks,” *IEEE/ACM Transactions on Networking*, vol. 12, no. 4, pp. 609-619, August 2004.

[69] C.K. Toh, “Maximum Battery Life Routing to Support Ubiquitous Mobile Computing in Wireless Ad Hoc Networks,” IEEE Communications Magazine, pp. 138-147, June 2001.

[70] Q. Li, J. Aslam and D. Rus, “Online Power-aware Routing in Wireless Ad-Hoc Networks,” *Proceedings of the IEEE/ACM International Conference on Mobile Computing and Networking (MOBICOM 2001),* pp. 97-107, 2001.

[71] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann and F. Silva, “Directed Diffusion for Wireless Sensor Networks,” *IEEE/ACM Transactions on Networking*, vol. 11, no. 1, pp. 2-16, February 2003.

[72] J.G. Proakis, *Digital Communications* (4th ed.), McGraw Hill, 2001.

[73] M. Streenstrup, “Cluster-Based Networks,” in *Ad Hoc Networking* (C.E. Perkins, ed.), Chapter 4, pp. 75-138, 2001.

[74] D. Estrin, R. Govindan, J. Heidemann and S. Kumar, “Next century challenges: Scalable coordination in sensor networks,” *Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM*, 263-270, 1999.

[75] R. Boppana and M. Halldorsson, “Approximating Independent Sets by Excluding Subgraphs,” *BIT*, Vol. 32, pp. 180-196, 1992.

[76] J. Hastad, “Clique is Hard to Approximate,” *Proceedings of the 37th Annual Symposium on Foundations of Computer Science*, pp. 627-636, 1996.

[77] P. Berman and T. Fujito, “On Approximation Properties of the Independent Set Problem for Low Degree Graphs,” *Theory of Computing Systems*, Vol. 32, pp. 115-132, 1999.

[78] L. Lovasz, "Three Short Proofs in Graph Theory," *Journal of Combinatorial Theory*, Series B, Vol. 19, pp. 269-271, 1975.

[79] D.S. Hochbaum, "Efficient Bounds for the Stable Set, Vertex Cover and Set Packing Problems," *Discrete Applied Mathematics*, Vol. 6, pp. 243-254, 1983.

[80] M.M. Halldorsson and J. Radhakrishnan, "Greed is Good: Approximating Independent Sets in Sparse and Bounded-Degree Graphs," *Algorithmica*, Vol. 18, pp. 143-163, 1997.

[81] P. Berman and M. Furer, "Approximating Maximum Independent Set in Bounded Degree Graphs," *Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*," pp. 365-371, 1994.

[82] M.M. Halldorsson and J. Radhakrishnan, "Improved Approximations of Independent Sets in Bounded-Degree Graphs," *Nordic Journal of Computing*, Vol. 1, No. 4, pp. 475-492, 1994.

[83] P. Berman and T. Fujito, "On Approximation Properties of the Independent Set Problem for Low Degree Graphs," Theory of Computing Systems, Vol. 32, pp. 115-132, 1999.

[84] U. Feige, "A Threshold of ln n For Approximating Set Cover," *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing*, pp. 314-318, 1996.

[85] P. Slavik, "A Tight Analysis of the Greedy Algorithm For Set Cover," *Journal of Algorithms*, vol. 25, no. 2, pp. 237-254, 1997.

[86] C. Lund and M. Yannakakis, "On the Hardness of Approximating Minimization Problems," *Journal of the Association for Computing Machinery*, Vol. 41, No. 5, pp. 960-981, 1994.

[87] T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, Introduction to Algorithms, 2nd edition, MIT Press and McGraw-Hill, 2004.

[88] D.J. Baker and A. Ephremides, "The Architectural Organization of a Mobile Radio Network via a Distributed Algorithm," *IEEE Transactions on Communications*, vol. COM-29, no. 11, pp. 1694-1701, November 1981.

[89] A. Ephremides, J.E. Wieselthier and D.J. Baker, "A Design Concept for Reliable Mobile Radio Networks with Frequency Hopping Signaling," *Proceedings of the IEEE*, vol. 75, no. 1, pp. 56-73, January 1987.

[90] C.R. Lin and M. Gerla, "Multimedia Transport in Multihop Dynamic Packet Radio Networks," *Proceedings of the 1995 International Conference on Network Protocols*, pp. 209-216, November 1995.

[91] C.R. Lin and M. Gerla, "A Distributed Architecture for Multimedia in Dynamic Wireless Networks," *Proceedings of the IEEE Globecom'95*, pp. 1468-1472, November 1995.

[92] A. P. Markopoulou, F.A. Tobagi and M.J. Karam, "Assessing the quality of voice communications over internet backbones," *IEEE/ACM Transactions on Networking*, 11(5), 747-760, 2003.

[93] C.C. Chiang, et al., "Routing in Clustered Multihop Mobile Wireless Networks With Fading Channel," Proceedings of the IEEE Singapore International Conference on Networks, pp. 197-211, 1996.

[94] E.J. Duarte-Melo and M. Liu, "Analysis of Energy Consumption and Lifetime of Heterogeneous Wireless Sensor Networks," *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM'02)*, vol. 1, pp. 21-25, 2002.

[95] P. Vincent, M. Tummala and J. McEachen, "Connectivity in Sensor Networks," *Proceedings of the 40th Annual Hawaii International Conference on System Sciences (HICSS-40)*, 2007.

[96] F. Harary, *Graph Theory*, Addison-Wesley, 1969.

[97] S. Shakkottai, R. Srikant and N. Shroff, "Unreliable Sensor Grids: Coverage, Connectivity and Diameter," *Proceedings of INFOCOM 2003, the Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 2, pp. 1073 – 1083, 2003.

[98] H. Zhang and J. Hou, "On the Upper Bound of α-lifetime for Large Sensor Networks," *ACM Transactions on Sensor Networks*, vol. 1, no. 2, November 2005, pp. 272-300.

[99] P. Gupta and P. Kumar, "Critical Power for Asymptotic Connectivity," *Proceedings of the 37th IEEE Conference on Decision and Control*, pp. 1106-1110, 1998.

[100] A. Leon-Garcia, *Probability and Random Processes for Electrical Engineering*, Addison-Wesley, 1994.

[101] N. Bulusu, J. Heidemann and D. Estrin, "GPS-less Low-Cost Outdoor Localization for Very Small Devices," *IEEE Personal Communications*, October 2000, pp. 28-34.

[102] L. Doherty, K. Pister and L. Ghaoui, "Convex Position Estimation in Wireless sensor Networks," *20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2000)*, vol. 3, pp. 1655-1663, 2001.

[103] D. Niculescu and B. Nath, "Ad Hoc Positioning System (APS)," *Proceedings of the IEEE Global Telecommunications Conference GLOBECOM '01*, vol. 5, pp. 2926-2931, 2001.

[104] S. Meguerdichian, S. Slijepcevic, V. Karayan and M. Potkonjak, "Localized Algorithms in Wireless Ad-Hoc Networks: Location Discovery and Sensor Exposure," *Proceedings of the 2nd ACM International Symposium on Mobile Ad-Hoc Networking and Computing (MOBIHOC 2001)*, pp. 106-116, 2001.

[105] D. Niculescu and B. Nath, "Ad Hoc Positioning System (APS) Using AOA," *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communication Societies, INFOCOM 2003*, vol. 3, pp. 1734-1743, 2003.

[106] A. Savvides, C. Han and M.B. Srivastava, "Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors," *Proceedings of the 7th International Conference on Mobile Computing and Networking (MOBICOM)*, pp. 166-179, 2001.

[107] K.K. Chintalapudi, A. Dhariwal, R. Govindan and G. Sukhatme, "Ad-Hoc Localization Using Ranging and Sectoring," *23rd Annual Joint Conference of the IEEE Computer and Communication Societies (INFOCOM 2004)*, vol. 4, pp. 2662-2672, 2004.

THIS PAGE INTENTIONALLY LEFT BLANK

# INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Fort Belvoir, VA

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, CA

3. Bernie Schneider
   US SOCOM
   Tampa, FL

4. Martin Kruger
   Office of Naval Research
   Washington, D.C.

5. George Hinckley
   VARDI
   Quantico, VA

6. Michael Nierman
   SPAWAR
   Charleston, SC

7. Jamie Carson
   SRC
   Charleston, SC

8. Richard Wylly
   SRC
   Charleston, SC

9. Prof. Murali Tummala
   Naval Postgraduate School
   Monterey, CA

10. Prof. John McEachen
    Naval Postgraduate School
    Monterey, CA

11. Prof. Frank Kragh
Naval Postgraduate School
Monterey, CA

12. Prof. Roberto Cristi
Naval Postgraduate School
Monterey, CA

13. Professor Craig Rasmussen
Naval Postgraduate School
Monterey, CA

14. T. Owens Walker
Naval Postgraduate School
Monterey, CA

15. Patrick Vincent
US Naval Academy
Annapolis, MD