

Analysis Approaches for Predicting Performance of Wireless Sensor Networks

A Dissertation

Presented to

the faculty of the School of Engineering and Applied Science

University of Virginia

In Partial Fulfillment

of the requirements for the Degree

Doctor of Philosophy

Computer Science

by

Ting Yan

August 2006

© Copyright August 2006

Ting Yan

All rights reserved

Approvals

This dissertation is submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
Computer Science

Ting Yan

Approved:

John A. Stankovic (Advisor)

Tarek F. Abdelzaher (Chair)

Sang H. Son

Gang Tao

Gabriel Robins

Accepted by the School of Engineering and Applied Science:

James H. Aylor (Dean)

August 2006

Abstract

In recent years, Wireless Sensor Networks (WSNs) as a new information gathering paradigm have attracted much research interest because of its broad applications in smart buildings, military surveillance, environmental, medical and ecological monitoring, among many others.

The success of wireless sensor network applications is due to WSN's strong capability of pervasive monitoring and wireless communications. However, some key constraints of wireless sensor networks, such as power consumption, sensing ability and communication bandwidth are not likely to be relaxed in the foreseeable future. Therefore, system designers must carefully design protocols and schemes and adjust parameters to balance various aspects of performance under these stringent constraints. Most published work takes best-effort approaches in the design and relies on trial and error real-world testing to find the best combination of parameters. These approaches have proven to be time and resource consuming, and can cause performance degradation due to environmental fluctuations. Therefore, it is of vital importance to quantitatively analyze the effects of system and protocol parameters on various aspects of wireless sensor networks, and to design protocols or select parameters based on the analysis in order to have predictable performance.

This dissertation takes the latter approach to explore the possibility of predicting performance based on quantitative analysis and design with guarantees in three key aspects of wireless sensor network performance, and describes how to integrate wireless sensor network analysis into a well known design tool.

The first aspect is sensing coverage, which is critical for many applications. Some applications need even higher degree of coverage for reliable monitoring of critical areas. The degree of coverage requirement needs to be balanced with the requirements on the system lifetime and energy

consumption. The work presented in the dissertation in this area proposes an energy efficient sensing coverage protocol. What distinguishes this approach with most other published approaches is that it provides the flexibility for system designers to enforce various degrees of sensing coverage with proven guarantee.

The second aspect is tracking performance for a duty-cycle based wireless sensor network system. The tradeoff between energy consumption and tracking performance, such as expected detection delay and detection probability, is thoroughly analyzed in the dissertation. This provides a theoretical foundation for choosing system and protocol parameters and predicting tracking performance based on these parameters.

The third aspect is the communication capacity for periodic streams. The dissertation presents a theoretical analysis which answers whether the rates of a specified set of communication streams can be satisfied by a network with a certain bandwidth. This analysis also provides clear understanding of the effects of communication ranges and interference ranges on the achievable rates.

Finally, the dissertation presents the efforts of incorporating these results into Eclipse plug-ins based on AADL, a standardized analysis and design language tool, to help system designers choose appropriate system and protocol parameters to satisfy the performance requirements.

The overall result of the dissertation is significant progress towards an analysis capability for wireless sensor networks. In addition, the methodologies applied in the analytical approaches presented in the dissertation serve as good examples for future extension to more aspects of wireless sensor network performance for similar analysis.

Acknowledgments

I thank my advisor, Dr. John A. Stankovic, for his guidance and support for the last five years. I always admire his energy, intelligence and diligence. He is not only a great advisor, but also a role model of mine.

I feel so fortunate that I had the opportunity to work in such a great group. Dr. Tarek F. Abdelzaher and Dr. Sang H. Son are always extremely supportive and inspiring, and they help me a lot on the dissertation work. I also get a lot of help from Tian He, Lin Gu, Gang Zhou, Jingbin Zhang, Qing Cao, Liqian Luo, Radu Stoleru, Leo Selavo, Anthony Wood, Pascal Vicaire, Brian Blum, Qiuhua Cao, and many others. They are great friends as well as good researchers, and I learned a lot from them.

I am very grateful for all the time and efforts of the other two committee members: Dr. Gabriel Robins and Dr. Gang Tao. My colleagues at Eaton Corporation are very supportive of my work on the dissertation in the last two months, which I appreciate a lot. I also thank Dr. Chenyang Lu and Octav Chipara from Washington University in St. Louis and Yu Gu from University of Minnesota for their contributions to the dissertation work.

Last but not least, I would like to thank my family. Without the enormous support from my wife, this dissertation would not have been possible. The love from my parents and brother, although thousands of miles away, has always been the momentum that pushes me to pursue knowledge and truth. I dedicate the dissertation to all of them.

Contents

1	Introduction	1
1.1	Introducing Wireless Sensor Networks	1
1.2	Problem Statement	5
1.3	Contributions	6
1.4	Outline of the Dissertation	9
2	State of the Art	10
2.1	State of the Art for Sensing Coverage and Power Management	10
2.2	State of the Art for Low Duty-Cycle Surveillance	13
2.3	State of the Art for Communication Capacity Analysis	15
3	Flexible Sensing Coverage	17
3.1	Introducing the Energy Efficient Sensing Coverage Problem	17
3.2	Basic Energy Efficient Sensing Coverage Protocol Design	18
3.3	Enhancement with Differentiated Coverage	34
3.4	Other Extensions	36
3.5	Dealing with Reality	36
3.6	Evaluation	39
3.7	Conclusions	56
4	Low Duty-Cycle Surveillance	57
4.1	Introducing the Low Duty-Cycle Surveillance Problem	57

<i>Contents</i>	viii
4.2 Theoretical Derivation	59
4.3 Evaluation	75
4.4 Dealing with Reality	78
4.5 Conclusions	87
5 Capacity Analysis for Periodic Streams	88
5.1 Introducing the Capacity Analysis Problem for Periodic Streams	88
5.2 Analysis of a Single Stream	92
5.3 Analysis of Multiple Streams	100
5.4 Dealing with Reality	107
5.5 Simulations	108
5.6 Implications for Designers	113
5.7 Conclusions	115
6 Integration with AADL	116
6.1 Introducing AADL and OSATE	116
6.2 Integrating the Analysis Approaches into OSATE	117
6.3 Benefits and Issues	120
7 Conclusions and Future Work	123
7.1 Contributions Revisited	123
7.2 Potential Extensions	125
Bibliography	128

List of Figures

3.1	Node Working Schedule	21
3.2	Sensing Coverage to Single Grid Point X	22
3.3	Calculating node schedules for a single grid point	23
3.4	The Process of Schedule Integration on a node	25
3.5	Grid Size Granularity Issues	27
3.6	A Wakeup Example for Rescheduling	29
3.7	Communication Connectivity	32
3.8	Polygon partitioned into 4 chains	37
3.9	Two corners of rectangle on polygon	37
3.10	Process for finding the largest inscribed rectangle in a polygon	37
3.11	The method used to extend the sensing range	38
3.12	Understanding of Sensor Coverage by Neighboring Nodes	40
3.13	Total Energy Consumption per Unit of Time	41
3.14	Single Node Energy Consumption: Average and Standard Deviation	42
3.15	Half-Life of the Network	43
3.16	Sensing Coverage Over Time	44
3.17	Three Types of Grids	45
3.18	Energy Consumed for Grid Deployments	46
3.19	Half-Life for Grid Deployments	47
3.20	Coverage Over Time for Grid Deployments	47
3.21	Total Energy Consumed for Differentiated Surveillance	48

3.22	Actual Degree of Coverage for Differentiated Surveillance	48
3.23	Total Energy Consumption per Unit of Time(Target Size=2)	49
3.24	Single Node Energy Consumption: Average and Standard Deviation(Target Size=2)	50
3.25	Total Energy Consumption Vs. Target Size(Node Density= $2/r*r$)	50
3.26	Half-Life of the Network(Target Size=2)	51
3.27	Half-Life of the Network Vs. Target Size(Node Density= $2/r*r$)	51
3.28	Sensing coverage over time ($1/r*r$)	52
3.29	Sensing coverage over time ($2/r*r$)	52
3.30	Sensing coverage over time ($3/r*r$)	53
3.31	Sensing coverage over time ($4/r*r$)	53
3.32	Half-Life of the Network	54
3.33	Total Energy Consumed for Reference Point Scattering	54
3.34	Half-Life for Reference Point Scattering	55
3.35	Coverage Over Time for Reference Point Scattering	55
4.1	Duty Cycle of a Particular Node O	61
4.2	Target Detection Scenarios	62
4.3	Target Detection Example	65
4.4	Fast Target Detection	65
4.5	Fast Target Detection Illustration	67
4.6	Type I target Detection	68
4.7	Regions A and B	68
4.8	Detection Probability or Expected Delay for Four Types of Targets	74
4.9	Stationary Detection Delay Distribution	75
4.10	Average Stationary Detection Delay	76
4.11	Mobile Detection Delay Distribution	77
4.12	Average Mobile Detection Delay	77
4.13	Range with Directions	79

4.14	Ranges of Two Nodes	79
4.15	Range Approximations	80
4.16	Realistic Sensing Effect	83
4.17	Robustness of Theoretical Predictions	83
4.18	Impact of Deployment on Expected Detection Delay	85
4.19	Detection Delay for Targets Moving Along Circular Curves with Speed 5m/s	86
4.20	Detection Delay for Targets Moving Along Circular Curves with Speed 50m/s	87
5.1	Communication and interference ranges	91
5.2	Interference between $s.l(0, 1)$ and $s.l(2, 3)$	93
5.3	Link convoying when $R_i = R_c$	93
5.4	Interference when $R_i = 2R_c$	94
5.5	Convoying in a low density network	96
5.6	Two streams with data rates $W/6$	101
5.7	When unconvoying is necessary	102
5.8	Relay region and interference region	104
5.9	10% Percentile of k -hop distances	109
5.10	Curve Fitting for 5% Percentile of k -hop distances	110
5.11	Least-Square Fitting Functions for $d_m(d, k, 95\%)$	110
5.12	Distribution of achievable rates	111
5.13	Schedulable percentage for two parallel streams	112
5.14	Schedulable percentage for three streams	113
5.15	Impact of routes on feasibility	114
6.1	AADL Plug-In Screenshot	118
6.2	Specifying Property Set	119
6.3	Specifying Component Types	120
6.4	Specifying Component Instances	121
6.5	Surveillance Analysis Results	121

List of Symbols

R	Sensing range
Ref	Reference point
T_{front}	Time before the reference point
T_{end}	Time after the reference point
d	Density
β	Duty cycle
A_{system}	System area
T	Scheduling period
t_{start}	Starting time
t_d	Detection delay
r	Sensing range
λ	Average number of nodes in sensing range
cdf	Cumulative distribution function
s	Stream
$s_i.r$	Data rate of stream i
$s_i.p$	Scheduling period of stream i
$s_i.N$	Number of hops of stream i
$s_i.l(j, j + 1)$	The j th hop of stream i
$s_i.dist$	The source-sink distance of stream i
R_c	Communication range
R_i	Interference range

W	Bandwidth
$H(s, k)$	Minimum k hop distance for stream s
$K(s)$	Minimum integer that satisfies $H(s, K(s)) > R_i$
g	Grid size
$I_H(s)$	Potential interference stream set (high density)
$I_L(s)$	Potential interference stream set (low density)
U_c	Convoyed stream set
U_u	Unconvoyed stream set

Chapter 1

Introduction

1.1 Introducing Wireless Sensor Networks

With the technological development of microsenors, wireless communication and pervasive computing, wireless sensor networks (WSNs) have emerged as a powerful paradigm for information collection and processing. In the last few years, wireless sensor networks [5] have been utilized in broad applications in areas such as military surveillance, ecological and enviromental monitoring, smart buildings, industrial and medical automation, among many others. Several prototype implementations have shown that quite a few significant design goals are achievable in real systems.

In a wireless sensor network, sensing nodes with limited power, computation, communication and storage resources cooperate to fulfill monitoring and tracking functionalities. Compared with conventional sensors, wireless sensor networks have the following advantages. Since the relevant technologies have become technologically and economically feasible, people want to gather much more information from more places in the physical world, which was either impossible due to technological difficulties or formidable due to high cost, in terms of money and human power. Decreasing form factors and costs of microsenors make deployment of hundreds even thousands of sensors much more feasible than conventional sensors that are too cumbersome and expensive. Many conventional sensors send data to data acquisition (DAQ) modules in personal computers or workstations, and typically the sensors, personal computers and workstations are interconnected in a wired manner for data collection, aggregation and processing. When the number of sensors

deployed in a certain area increases exponentially, this approach obsoletes. A more appropriate approach is to connect the sensors with low cost microcontrollers and to send and receive information through wireless links. This approach is made feasible by decrease in costs, form factors and power consumption of microcontrollers and radio transceivers.

Based on the sensing ranges of quite a few commonly used types of sensors, a deployment density of one node per one hundred square meters is typical. To cover an area of one square kilometer, 10,000 nodes are needed. Currently a typical sensor node costs around \$100 per piece in large volume, which makes the total cost around one million dollars. Based on past experiences, we can project that in the near future, the price will drop significantly due to the exponentially decreasing trend of the prices of microcontrollers and radio transceivers, and the factor of large scale production. This will make large scale wireless sensor network applications more economically feasible.

A typical wireless sensor network is comprised of tens, hundreds, or even thousands of sensor nodes. Typically each sensor node is composed of a microcontroller, a radio transceiver, one or more micro sensors, power source and other components. The microcontroller samples the micro sensors, send the data, either with or without processing, through radio links to the locations where the information is needed. Due to the limited radio range and the relatively larger target areas, in many cases a multiple hop ad hoc wireless network is formed for the information transmission. We define the devices that gather the information from the wireless sensor networks as base stations. There may be one or more base stations for a wireless sensor network. The base stations may be static or mobile. However, for many applications, the sensor nodes themselves are not moving, either due to the scenario requirements, or technical or economical hindrance.

Although wireless sensor networks provide unprecedented power and convenience compared with conventional sensors, there are a few constraints that limit their performance. Gordon Moore predicted that “the number of transistors and resistors on a chip doubles every 18 months”, which has been proven to be true with minor modifications for more than 40 years. The cost of equivalent computing power has also been decreasing exponentially. We can reasonably envision that the computing power for a single chip or for the same cost will continue increasing with a fast speed.

However, other performance aspects that are critical for wireless sensor networks do not develop with the same speed.

Battery power is one example. Although some wireless sensor networks do operate on industrial or residential alternating current electrical power systems, in many scenarios it is impossible for wireless sensor network systems to leverage wired power supply. Solar cells do not prove to be a universally feasible approach for power supply, either. In many applications, battery power is still the only option. Many scenarios require the sensor nodes to be physically unattended after deployment for their whole lifetime, especially for military surveillance systems. For these scenarios, battery recharging is not feasible, either. On the other hand, historical trends show that battery power per unit weight evolves with a much slow pace compared with the exponential increase in computational power per chip and the exponential decrease in cost for the same computational power. Considering the form factor for each sensor node, we cannot project a dramatic increase in the battery power for a sensor node. This constraint is a key issue that limits the lifetime of a wireless sensor network. Bearing this in mind, it is common practice for designers to reduce as much unnecessary power consumption as possible. Energy efficient schemes for sensing and communication thus are a very extensively studied area for wireless sensor networks.

The second key constraint is sensing capability. For a certain category of targets, the sensing range, or the furthest distance for a target to be detectable by a certain type of sensor, is limited by laws of physics. Longer sensing ranges always need more energy consumption, higher cost, or larger form factors. This factor limits the sensing ranges for sensors, which are unlikely to increase drastically, either.

Communication capability is another constraint. Communication range is decided by the transmitting power, radio attenuation, receiver sensitivity and other factors. Transmitting power is limited by the power supply. Radio attenuation and receiver sensitivity are limited by laws of electromagnetism. Therefore, communication range is not likely to increase significantly, either. Typical communication range based on the form factors and power supply for wireless sensor networks is below 100 feet when the sensor nodes are placed on the ground. If the designers want to cover a larger area, multiple hop communication must be applied. Communication bandwidth is also lim-

ited, too. Due to government regulations, bandwidth available for wireless sensor networks is very scarce and shared with other technologies. For example, the popular 2.4GHz bandwidth is also used by Bluetooth and ZigBee.

In summary, power supply, sensing range and communication capabilities constitute three main constraints for wireless sensor networks. These constraints are not likely to be relaxed significantly in the foreseeable future. To develop an efficient wireless sensor network that satisfies the performance requirements, system designers have to take these factors into account and understand the intrinsic relationship among them and their corresponding effects on the system performance.

We take MicaZ [20], a popular wireless sensor node, as an example. If we keep the power of the microcontroller, sensing modules and/or the radio module on a MicaZ node on all the time, 2 AA batteries can support its operation for up to a week. Most applications require a much longer lifetime than that. To reduce the average energy consumption over time, it is necessary to put the microcontroller, sensing modules and/or radio transceiver to power-saving modes. This approach certainly sacrifices the sensing and/or communication performance.

Therefore, system designers have to balance among system lifetime and sensing and/or communication performance, and other performance metrics under the economical and physical constraints. In principle, there are three categories of approaches that can achieve this goal. For approaches in the first category, best-effort protocols and schemes are designed, and the system and protocol parameters are choosed based on trial and error tests. Most published approaches belong to this category. For approaches in the second category, new protocols and schemes are designed with performance guarantees based on analytical proofs. For the third category, quantitative analysis is performed on existing, well-known protocols and schemes so that the performance is predictable. In this dissertation, we focus on the latter two categories of approaches for the sake of predictable performance. Details on the advantages of the latter two categories of approaches over the first one are discussed in Section 1.3.

1.2 Problem Statement

In this dissertation, we identify three major aspects of functionality of wireless sensor networks: sensing coverage, tracking and periodic communication. As the name “wireless sensor network” implies, sensing is one of the most important functionalities. Related to sensing, tracking is another important functionality for many wireless sensor network applications, especially for applications such as military surveillance, urban warfare and homeland security. Our focus is on the protocol design and analysis of energy-efficient sensing and tracking schemes with performance guarantees. Another important functionality is communication, since a main advantage of wireless sensor networks over conventional sensors is that the sensory data can be gathered through wireless links. In a typical routing sensing scenario, communication in the network can be divided into two categories: periodic communication and sporadic communication. In the dissertation we study the analytical capacity bounds of wireless sensor networks for periodic communication.

1.2.1 Flexible Sensing Coverage

Sensing coverage is one of the most important functionalities for wireless sensor networks. Many wireless sensor network applications require that the whole target area where the network is deployed to be completely covered all the time. In some critical applications such as military surveillance, some part of the target area needs a higher degree of coverage so that even a sensor node fails, the part of the target area is still fully covered all the time. Meanwhile, the lifetime requirements often prevent the nodes to be working all the time so that duty-cycling needs to be applied to reduce energy consumption and therefore prolong the network lifetime. The problem we manage to solve is to provide a scheduling scheme that can satisfy the requirement of differentiated degrees of coverage with analytical guarantees, while minimizing the energy consumption per unit of time.

1.2.2 Low Duty-Cycle Tracking

Since we can deploy a large number of microsensors with the capability of communicating with each other, tracking is a naturally attractive application area for wireless sensor networks. Mean-

while, a wireless sensor network aimed at a long lifetime has to save as much energy as possible. The problem we manage to solve in this aspect is the quantitative relationship between duty-cycling and tracking performance, such as detection probability and detection delay.

1.2.3 Capacity Analysis of Periodic Streams

Another important functionality for wireless sensor networks is to send the periodic sensing information from sensors, which we call sources, back to where it is needed, which we call sinks. For quality of service, a certain data rate has to be maintained for each source-sink pair. On the other hand, the communication bandwidth is a key limitation to the achievable rates. For large scale networks, the information has to be relayed through multiple hops. This requirement and potential interference add into complexity of the relationship between network bandwidth, locations of streams and achievable rates. This dissertation tries to answer the question whether a set of streams can achieve their specified rates given the bandwidth limitation.

1.3 Contributions

Since wireless sensor networks have attracted significant research attention, much research work has been published to improve performance of wireless sensor networks. The next chapter introduces related work in details.

Compared with related work, the contributions of my work can be categorized to two aspects of impacts: theoretical impacts and practical impacts. They are described as follows.

1.3.1 Theoretical Impacts

Although there have been many pieces of work on improving the performance metrics of wireless sensor networks, the quantitative characterization of fundamental limitations on several key aspects of performance remains an open research problem. As an example, although the lifetime of a wireless sensor node can be estimated through dividing the capacity of batteries (typically specified in milliampere hours) by the average energy consumption (typically in milliwatts) per unit of time,

the impact of duty-cycling, a typical energy conservation technique, on the sensing and tracking performance was not thoroughly explored in a quantitative way until the work described in this dissertation. Here is a summary of theoretical contributions of this work:

First, the work presented in this dissertation on flexible sensing coverage of wireless sensor networks is the first to provide a theoretically guaranteed degree of coverage over a certain target with high energy efficiency.

Second, the work presented in this dissertation on duty-cycle based tracking is the first to provide a thorough analysis of the impact of duty-cycling on the tracking performance. With the results from this work, the tradeoff between system lifetime and tracking performance is quantitatively characterized.

Third, the work presented in this dissertation on capacity analysis of periodic streams can characterize the relationship between the parameters of network communication and the feasibility of a specific set of periodic communication streams. This is a significant improvement over the asymptotic bounds proposed earlier than this work.

Furthermore, these pieces of work can serve as a foundation for further exploration of the theoretical constraints of wireless sensor network performance due to scientific and technological limitations.

1.3.2 Practical Impacts

The work presented in this dissertation also has significant practical impacts for real-world system designers. During the design of a practical wireless sensor network system, it is always a difficult job to balance various aspects of system performance, such as energy consumption/system lifetime, sensing coverage, tracking performance, communication rates and others. Before this work, the selections of system and protocol parameters have been mostly conducted in a trial and error fashion. The designers choose a set of parameters based on guessing or past experiences, test the system to see if the performance metrics are satisfied, and iterate this process until the requirements are met. These approaches are a significant drawback for the wireless sensor network development. First of all, it is very costly in terms of manpower to conduct this kind of experiments, especially when the

system scales to hundreds or even thousands of nodes. Second of all, the potential combinations of system and protocol parameters are enormous. As an example, the VigilNet system [30] has more than twenty very important system and protocol parameters among other less important ones. Even if each parameter takes two possible values, which is a great underestimation, the number of possible combinations is in the order of magnitude of millions, which is practically impossible to try out all. Therefore, it depends on expert knowledge, which is hard to document and transfer, to obtain the optimal or even suboptimal combination of parameters. Last but not least, environmental factors such as temperature fluctuations may affect (in many cases towards the negative direction) the system performance. It is almost impossible to capture all environmental situations in experiments. Considering these factors, it is very important to have analytical results for practical system building. The work presented in this dissertation serves this purpose in three aspects.

First, the flexible sensing coverage scheme provides a practical solution for differentiated surveillance requirements with proven guarantees. It helps the wireless sensor network system designers to select system and protocol parameters based on various requirements on system lifetime and coverage.

Second, the analysis work on duty-cycle based tracking system helps the system designers thoroughly understand the relationship between duty cycles and achievable tracking performance. This makes selecting parameters such as duty cycles, sensing range and deployment density as easy as applying a mathematical formula.

Third, the analysis of communication capacity for periodic streams provides a viable approach for system designers to decide if the network bandwidth assigned for periodic communication is adequate to satisfy the rate specifications of these streams. This is of vital importance to guarantee the feasibility of periodic communication streams. The approach is constructive, in the sense that if it gives a positive answer, the schedules can be easily derived from the analysis.

The author also shows in this dissertation the efforts to transfer these analysis approaches into Eclipse AADL plug-ins in OSATE. These plug-ins help the system designers to select the system and protocol parameters to satisfy the performance requirements in a very straightforward way.

1.4 Outline of the Dissertation

The rest of this dissertation is organized as follows. Chapter 2 is a detailed survey of the state-of-the-art of research work on the three aspects of functionalities for wireless sensor networks we explore in this dissertation. Chapter 3 presents the energy-efficient flexible sensing coverage scheme with guarantee of degrees of coverage. Chapter 4 presents the detailed tracking performance analysis for duty-cycle based wireless sensor network systems. Chapter 5 presents the capacity and feasibility analysis of wireless sensor networks for periodic streams. Efforts of integrating these analysis approaches into a AADL-based software tool are presented in Chapter 6. Chapter 7 concludes the dissertation and proposes potential research extensions.

Chapter 2

State of the Art

Publications on related work are introduced in this chapter. Section 2.1 describes the state of the art for sensing coverage and power management in general. Section 2.2 describes related work on tracking. Most work in these fields has a best-effort flavor and lacks adequate analytical guarantee for performance predictability. Our work presented in this dissertation manages to provide predictable performance based on protocol design and analysis. Section 2.3 describes related work on communication analysis of wireless sensor networks. Although quite a few publications have appeared in the last few years, many of them only provide asymptotic bounds, thus from them the question whether a specific set of communication requirements can be satisfied in a wireless sensor network with a certain bandwidth specification cannot be answered. Our work in this area differs from them in the sense that our analysis is able to answer these kinds of questions.

2.1 State of the Art for Sensing Coverage and Power Management

A set of research issues needs to be addressed before surveillance-based applications such as military tracking and environmental monitoring using low cost sensor nodes such as Berkeley motes become technically feasible and economically practical. Recently, several schemes are proposed to address the sensing coverage problem in sensor networks. In [51], full surveillance coverage is support by a node-scheduling scheme based on off-duty eligibility rules, which allows nodes to turn themselves off as long as the neighboring nodes can cover the area for them. This rule guarantees

100% sensing coverage as long as no void exists. However, this rule underestimates the area that the neighbor nodes can cover, which leads to excess energy consumption. The coordinated sleep algorithm proposed in [35] improves the aforementioned algorithm by providing continuous coverage and compares the approach with a random sleep algorithm. In [60], surveillance coverage is achieved by a probing mechanism. In this solution, after a sleeping node wakes up, it broadcasts a probing message within a certain range and waits for a reply. If no reply is received within a timeout, it takes the responsibility of surveillance until it depletes its energy. In this solution, the probing range and wakeup rate can be adjusted to affect the degree of coverage indirectly. However, this probing-based approach has no guarantee on sensing coverage and blind points can occur. Co-Grid [55] is a distributed protocol for sensing coverage maintenance based on coordinating fusion groups on virtual grids. It maintains probabilistic detection guarantees based on probabilistic detection models and data fusion among multiple nodes. Our solution is different from these solutions in the sense that it can not only guarantee sensing coverage to a certain geographic area, but it can also adapt the degree of coverage to that area, up to the limitation imposed by the number of sensor nodes present.

Energy balancing is another research issue addressed in this dissertation. [51] and [59] only consider the metric in terms of the total amount of energy consumed regardless of the distribution of the energy among the nodes. We argue that unbalanced energy dissipation causes some nodes to die much faster than others do, therefore, the half-life of the network is dramatically reduced in the unbalanced approach. Research has addressed the energy balance issue from different aspects of sensor networks. SPEED [31] balances the traffic by non-deterministically forwarding the packet through multiple routes. GAF [57] performs leader rotation among the nodes inside a virtual grid, in order to balance energy consumption.

Research on network topology control such as ASCENT [16], SPAN [18], LEACH [33] and GAF [57] addresses the problem of providing communication coverage within an energy conservation context. These have a similar flavor as the surveillance coverage problem. For example, LEACH [33] partitions a network into clusters and randomly rotates the cluster leader in order to evenly distribute the energy consumption among the sensors. SPAN [18] is another randomized

algorithm where nodes make local decisions on whether to sleep or to join a backbone network in order to reduce energy consumption. The major difference between those and this work is that communication coverage considers only connectivity between the nodes. In contrast, surveillance (sensing) coverage addresses the coverage problem to every physical point in the terrain. As a result, new scheduling is required to force some nodes stay awake for surveillance purposes even though they are not participating in data forwarding.

It also helps network communication to provide a full sensing coverage. The impact of full sensing coverage of a sensor network on greedy forwarding algorithms is studied in [54]. An upper bound of network dilation is derived so that the sum of lengths of transmission hops along the routing paths is bounded, which leads to high communication efficiency. Geometric analysis is conducted on the relationship between sensing coverage and communication connectivity in [56]. A Coverage Configuration Protocol (CCP) is proposed for energy-efficient sensing coverage and is integrated with SPAN for connectivity. Compared with CCP, which requires broadcasts of join/withdraw signals for each state rotation, the schedule of each node in our approach is generated at the beginning of the network operation and does not require more communication at each rotation. In addition, our protocol provides more flexible control over degree of coverage for the system.

Besides the aforementioned work in communication and sensing coverage that conserves energy, work in energy conservation for general sensor networks has been considered at various levels of the communication stack. From the bottom to the top, special hardware [45] is designed with multiple energy dissipation settings. MAC layer protocols developed for energy savings mostly take advantage of overhearing and scheduling to allow nodes to sleep while they are not transmitting or receiving messages [26, 33]. At the network and routing layers, solutions are diversified. Data placement schemes [12] minimize energy along the transmission path through data caching. In [48], communication range is adjusted dynamically based on the node density to conserve energy consumed in transmission. MFR [36] uses a minimal hop path to reduce the total number of transmissions. [58] sets routes according to the energy remaining at nodes along that path, and [31] uses mechanisms to save energy through the distribution of messages among various paths from

source to destination. At the application layer, the protocols incorporate routing semantics to form groups and rotate leadership responsibilities, allowing non-leader nodes to sleep and conserve their energy [18]. Finally, data aggregation techniques inside the network and application layers also provide energy conservation features in both application independent [28] and application dependent fashions [37,42]. All of these protocols tackle the energy issue from different aspects of sensor networks, thus we consider them complementary to our work.

Research has also been conducted to address QoS issues to support differentiated services for sensor networks. SWAN [4] uses feedback information from the MAC layer to regulate the transmission rate of non-real-time traffic in order to sustain real-time traffic. RAP [14] uses velocity monotonic scheduling to prioritize real-time traffic and enforces such prioritization through a differentiated MAC Layer. In [11], the delivery ratios of packets at different priority levels are affected by the forwarding probability at intermediate nodes. To date, to the best of our knowledge, no algorithm has been specifically designed to address how to provide differentiated surveillance for sensor networks. Due to the surveillance nature of most sensor network applications, we deem that this service is essential.

2.2 State of the Art for Low Duty-Cycle Surveillance

The advantages of wireless sensor networks over conventional sensors in cost, form factor and convenience of deployment and communication make them ideal for both active and passive tracking applications. Therefore, wireless sensor networks have been applied to various kinds of tracking applications, including military, ecological and medical surveillance systems.

VigilNet [30] is an energy efficient military surveillance system built upon XSM motes. Each sensing node is equipped with acoustic, magnetic and passive infrared sensors. The sensor network system gives real-time information of locations of the targets. The system also classify targets into four categories: human beings without weapon, human beings with weapon, small vehicles and large vehicles based on their unique sensing signatures. The prototype system shows the feasibility and strong benefits of using wireless sensor networks to track and classify targets in a set of real-

world testings consisting of more than 200 sensor nodes.

Similar approaches are presented in [13,44]. Although design and implementation details differ significant, the basic idea behind all three approaches is to form a logical group for each physical target based on the proximity of the sensors that detect the target.

In [8] it is shown that even minimalist sensors that provide only noisy binary information can be used to provide reasonable tracking performance when a filtering algorithm is used. They also present pioneering work in conducting analysis to derive the theoretical limitations of such sensors on tracking performance.

As in sensing coverage, work/sleep schedules, or duty-cycles attract research attention in the area of power-conserving tracking applications, too. As an example, the VigilNet [30] system elects sentry nodes which are awake during a rotation cycle, and other nodes are non-sentry nodes which are asleep for most of the time for power efficiency. Sentry nodes are elected in a particular manner so that an almost full sensing coverage is still provided so that the probability of a target being missed is minimized.

The full sensing coverage model is very suitable for areas where continuous vigilance is required, but consumes considerable energy. There exists a lower bound on the minimal energy consumption if the requirement of full sensing coverage is to be fulfilled. To guarantee a longer sensor network lifetime, in certain situations, the designer may wish to keep partial sensing coverage in exchange of longer product lifetime.

Recent research efforts [25,49,34,15] has turned attention to the study of target tracking in the context of partial sensing coverage. In [25], the authors propose the metric of quality of surveillance, which is defined as the average traveled distance of the target before it is detected. Obviously this is quite consistent with the model in our dissertation, where the average detection delay rather than distance is used. The authors further explore various approaches to minimize this metric, while our work is more concerned with thorough understanding of its quantification. The work in [49] initiates analysis of type II target detection. However, our work is considerably different in that we are more interested in deriving closed-form expressions for all four scenarios, whereas [49] only considers fast targets and does not give closed-form formulas for detection delay or stealth

distance as we do in the dissertation. Another interesting paper is [34], which considered various random and coordinated scheduling approaches used majorly for partial sensing coverage. In [15], a near-optimal approach that provides work/sleep schedules to sensor nodes is proposed to provide a bounded delay sensing coverage, and the guaranteed bounded delay is close to the theoretical lower bound. However, none of these papers derived the closed form results as in our work presented in the dissertation, and therefore, we believe our work is a necessary complement towards thorough understanding of detection delay performance.

2.3 State of the Art for Communication Capacity Analysis

Research on capacity of wireless networks has been active since the seminal work by Gupta and Kumar [27]. In [27], the authors derive the information theoretical asymptotic bounds for obtainable throughput of wireless networks under various models. For random deployment and random traffic patterns, the per node achievable throughput is $\Theta(W/\sqrt{n \log n})$, in which W is the bandwidth and n is the number of nodes. While for optimal deployment and optimal traffic patterns, the per node achievable throughput is $\Theta(W/\sqrt{n})$. These results are true for both protocol model and physical model.

The analysis is improved in [3] and the bounds in [27] is bracketed by a factor of $\sqrt{8}$. The analysis is also extended in quite a few papers for various settings. Among them, [7] extends the results in [27] to accommodate more general communication model and mobility model, and points out that as the number of nodes approaches infinity, the desired per node throughput is achievable only if the average number of hops between sources and sinks does not grow indefinitely with the node number. [10] derives the per node throughput in a grid sensor network, gives guidance of optimal routing scheme to achieve the capacity bounds and uses a couple of node failure models to investigate the robustness of the network. It is discovered in [41] that the throughput bounds can be significantly improved if a communication infrastructure is provided so that the ad hoc nodes can utilize the infrastructure through gateways. In [17], the authors compare the multiple hop approach with alternatives such as beamforming, and prove that multiple hop communication

can provide order-optimal asymptotic performance compared with beamforming and overperforms beamforming under certain circumstances.

Although these results provide general asymptotic bounds for wireless networks, criteria to decide whether a network is able to satisfy a certain set of communication requirements or not are not provided. Our approach is essentially different from work presented in the above publications, because we provide a criterion for determining whether a set of periodic communication requirements can be scheduled within a certain bandwidth, instead of an asymptotic bound. Asymptotic bounds do not account for the detailed communication patterns. The constants in the bounds are often unknown. Therefore, such bounds may not be used for feasibility analysis.

In [40], achievable rates are characterized for a multi-hop wireless network given a connectivity and interference graph. In contrast, our approach does not necessarily rely on explicit knowledge of a node's location or its communication/interference neighbors.

The impact of interference on wireless network performance is studied in [38]. It provides an approach to obtain the maximum achievable throughput if the exact locations of the nodes are known with a conflict graph for interference modeling. Our approach does not require such details, provides a less time consuming analysis and helps designers understand the relationship between system parameters and their constraints on the achievable performance.

A real-time schedulability bound is derived in [2]. Under this sufficient condition real-time data can be successfully transferred with their deadlines met. Our work focuses on deriving the relationship between system parameters and achievable rates, instead of deadlines.

The capacity of ad hoc wireless networks for 802.11 MAC is investigated in [43]. Both simulations and first-principle analysis are used to study the interaction between 802.11 and ad hoc forwarding and the scalability of per-node throughput. Our work provides a more general bound without specific constraints on MAC protocols.

Chapter 3

Flexible Sensing Coverage

This chapter presents the energy efficient flexible sensing coverage protocol that provides various degrees of coverage. Section 3.1 introduces the problem we solve in details. The basic protocol design and the proof of its performance guarantee are described in Section 3.2. The enhanced protocol with differentiated coverage is presented in Section 3.3. Section 3.4 presents further extensions to the basic and enhanced protocols. The performance of the various protocols and extensions described in this chapter is evaluated based on extensive simulations in Section 3.6. Section 3.7 concludes the chapter.

3.1 Introducing the Energy Efficient Sensing Coverage Problem

Low-cost deployment is one acclaimed advantage of sensor networks, which imply that the resources available to individual nodes are severely limited. Limited processor bandwidth and small memory are two arguable constraints in sensor networks, which will disappear with the development of fabrication techniques. However, the energy constraint is unlikely to be relieved quickly due to slow progress in developing battery capacity. Moreover, the untended nature of sensor nodes and hazardous sensing environments preclude battery replacement as a feasible solution. On the other hand, the surveillance nature of sensor network applications requires a long lifetime; therefore, it is a very important research issue to provide a form of energy-efficient surveillance service for a geographic area.

Previous research focuses on how to provide full or partial sensing coverage in the context of energy conservation. In such an approach, nodes are put into a dormant state as long as their neighbors can provide sensing coverage for them. These solutions regard the sensing coverage to a certain geographic area as binary, either it provides coverage or not. However, we argue that, in most scenarios such as battlefields, there are certain geographic sections such as the general command center that are much more security-sensitive than others. Based on the fact that individual sensor nodes are not reliable and subject to failure and single sensing readings can be easily distorted by background noise and cause false alarms, it is simply not sufficient to rely on a single sensor to safeguard a critical area. In this case, it is desired to provide higher degree of coverage in which multiple sensors monitor the same location at the same time in order to obtain high confidence in detection. On the other hand, it is overkill and energy consuming to support the same high degree of coverage for some non-critical area. Based on such observations, this chapter leverages previous solutions and addresses the problem of providing a differentiated surveillance service for sensor networks. By differentiated surveillance, we mean providing different degrees of sensing coverage for a sensor network according to different requirements. Two major goals are achieved in this chapter. First, our new sensing coverage algorithm outperforms other state-of-the-art solutions in terms of energy conservation, energy balance and communication overhead when it runs in non-differentiated mode. Second, this chapter is the first to propose differentiated surveillance for sensor networks with minimal overhead. Furthermore, this chapter proposes three additional enhancements, taking the size/shape of targets, the unbalanced initial energy capacity of individual nodes, and reference distribution into account to enhance the energy conservation and balance of the sensor network further.

3.2 Basic Energy Efficient Sensing Coverage Protocol Design

In this section and section 3.3, we introduce the design of an adaptive sensing coverage scheme for sensor networks. The basic design without differentiation is introduced first in section 3.2. This is then the basis for the extension for differentiated surveillance to support scenarios in which a partial

coverage ($<100\%$) is sufficient or a high degree of coverage ($>100\%$) is desired (section 3.3).

3.2.1 Design Goals

The basic design goal for this work is to provide energy efficient sensing coverage for a geographic area covered by sensor nodes. Though it has been addressed in previous work [51, 59], our work distinguishes itself from previous solutions in following sub-goals that we achieve.

- Reduce total amount of energy consumed.
- Reduce energy variation among nodes.
- Reduce communication overhead in establishing nodes' work schedules.
- Support a new extension to the service for differentiated surveillance with small overhead.
- Provide communication connectivity as an auxiliary benefit.

In the remaining parts of the chapter, we support these claims with analytical analysis and simulation results.

3.2.2 Assumptions

First, we assume that each node knows its own location [29] and nodes are not moving. The node location information does not need to be precise when we are using conservative sensing ranges to decide the schedules (see 3.2.4). These are common assumptions for many sensor network applications. For simplicity and convenience of protocol description in the rest of the chapter, we refer to the sensing area of a node as a circle with a nominal radius r centered at the location of the node itself. We later show that we can also deal with irregular and/or non-uniform sensing areas as long as the neighboring nodes are aware of each other's sensing area. In the protocol description, we deploy the sensor nodes in a two-dimensional Euclidean plane. However, the protocol can be extended to a three-dimensional space or a curved surface without much difficulty. We also assume that the neighboring nodes should be roughly time synchronized on the order of seconds, which can be easily achieved by current millisecond-level synchronization solutions such as found in [23].

Also, the tolerance for small synchronization skews (see 3.2.4.2) allows the re-synchronization process to happen even less frequently so that the cost can be neglected.

The last assumption we make is that nodes can directly communicate with the neighboring nodes within a radius larger than $2r$ (r is nominal sensing radius). This is a typical case in all systems we experienced. For example, MICA 2 [21] has a open-space communication range of about 500 feet, while the sensing range is within a hundred feet even with long-range motion sensors. Actually, this is an optional assumption. The protocol works so long as the nodes are able to communicate directly or indirectly with each other within the distance of $2r$ and the communication range needs not to be regular. We make this assumption for simplicity in the protocol description and to avoid routing overhead for any two nodes that can sense a common area. This assumption also helps provide a connectivity guarantee when full sensing coverage is achieved.

3.2.3 Basic Design without Differentiation

Each node in the sensor network is either in sleeping mode or in working mode. Our goal for the basic design without differentiation is to have as many nodes as possible go to sleep to save energy and extend the lifetime of the sensor network while guaranteeing 100% sensing coverage to the target area.

3.2.3.1 A Node's Working Schedule

The lifetime of a sensor network is composed of an initialization phase and a sensing phase. During the initialization phase, each sensor node finds its own position [29] and synchronizes [23] time with neighboring nodes. After that, nodes enter into a sensing phase and start to sense environmental events. The sensing phase of nodes is divided into rounds with equal duration and are synchronized. Each node establishes a working schedule through our algorithm, which tells it when to sleep and when to work for each round. When a node goes to sleep, its sensing, communication and computation components can all be asleep and only a timer needs to work and wake up all components according to its schedule.

We denote the duration of each round as T . The schedule for a node is determined by a tuple with four parameters: $(T, Ref, T_{front}, T_{end})$. As shown in Figure 3.1, Ref is a random time reference point chosen by a node within $[0, T)$. T_{front} is the duration of time prior to the reference point Ref , and T_{end} is the duration of time after reference point Ref . We describe how to decide these parameters in section 3.2.3.2. For any given tuple $(T, Ref, T_{front}, T_{end})$, a node executes the following working schedule:

A node wakes up at time $(T \times i + Ref - T_{front})$ and goes to sleep at time $(T \times i + Ref + T_{end})$.

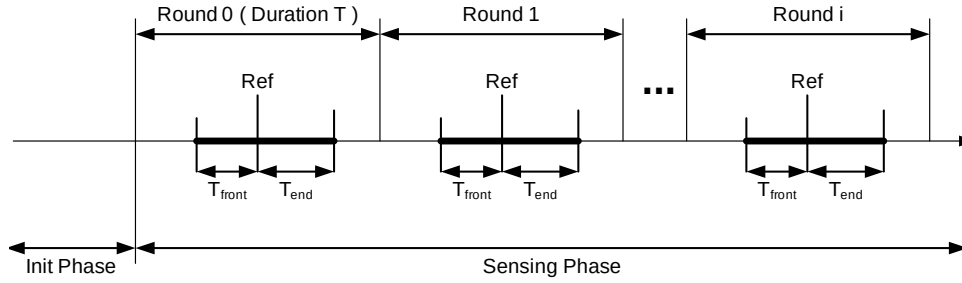


Figure 3.1: Node Working Schedule

It should be noted that the following two rules hold for the schedule:

1. The tuple $(T, Ref, T_{front}, T_{end})$ for each node is chosen during the initialization phase and does not change during the whole sensing phase, unless rescheduling is required for fault-tolerance purposes.
2. Time duration T_{front} and T_{end} should be in the interval $[0, T)$ and furthermore, the sum of T_{front} and T_{end} , which is the time duration for a node to work during each round, should be less than or equal to T .

3.2.3.2 Setting Tuple Parameters $(T, Ref, T_{front}, T_{end})$

In our approach, we cover the target area with a virtual square grid. We describe how to choose the unit grid size later in section 3.2.4.1. Now we describe the protocol that decides the schedule for each sensor node to guarantee that each grid point in the target area is covered by at least one

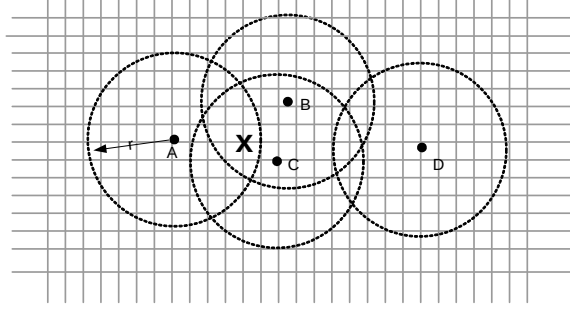


Figure 3.2: Sensing Coverage to Single Grid Point X

working node at any time and with minimum energy consumption. Here we only consider the energy consumed by the sensing function, so the energy consumed by a node is proportional to its working time for each round, which is $(T_{front} + T_{end})$. Parameter T , which is the duration of each round, is pre-determined and kept constant across all the nodes inside the sensor network. We discuss how to set this parameter in more detail in section 3.2.4.2.

After the nodes complete localization and time synchronization, they each broadcast a beacon (a random jitter in transmission time is added to avoid collision). This beacon has the information of its location and a randomly chosen time reference point Ref . The reference time should be uniformly chosen among $[0, T)$ and follows an identical independent distribution (iid). According to the assumption of direct communication within $2r$, each node within the distance of $2r$ of the sending node should receive the beacon. For example in Figure 3.2, nodes A, B and C hear each other. After every node beacons the time reference point Ref , each node maintains a neighbor table with the information of locations and time reference points of its neighbors within $2r$'s distance. For example, node A keeps information about nodes B, C and itself in a table. D's beacon is ignored since it is outside the $2r$'s distance of A.

Before discussing the sensing coverage of the entire area, we first focus on how to provide sensing coverage for a single grid point. Here we consider node A and grid point x shown in Figure 3.2. Obviously, all the sensor nodes whose sensing areas cover grid point x are in the $2r$'s distance of node A. Therefore, node A has the information of their locations and time reference points in its neighbor table. With such information node A sorts those time reference points in ascending order and lays out all N reference times including its own reference time point (see

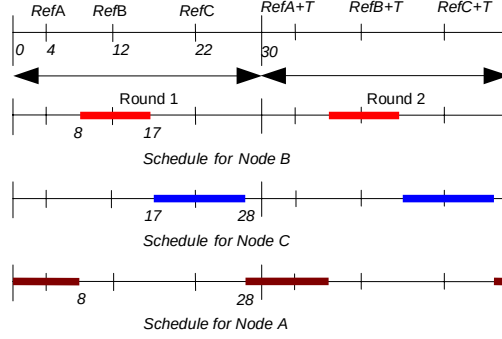


Figure 3.3: Calculating node schedules for a single grid point

Figure 3.3). (Suppose that the total number of sensor nodes which can cover grid point x is N).

Without loss of generality, suppose node A has the i th reference time $Ref(i)$, node B has the $(i-1)$ th reference time $Ref(i-1)$ and node C has the $(i+1)$ th reference time $Ref(i+1)$. Then the time interval T_{front} and the time interval T_{end} for grid point x are set by our algorithm to $[Ref(i) - Ref(i-1)]/2$ to $[Ref(i+1) - Ref(i)]/2$, respectively. There are two special cases for the T_{end} and T_{front} calculations:

1) If node A has the smallest reference time among all N nodes that can cover grid point X , then the time interval T_{front} is $(T + Ref(1) - Ref(N))/2$ and the time interval T_{end} is $(Ref(2) - Ref(1))/2$.

2) If node A has the largest reference time among all N nodes that can cover grid point X , then the time interval T_{front} is $(Ref(N) - Ref(N-1))/2$ and the time interval T_{end} is $(T + Ref(1) - Ref(N))/2$.

To understand the procedure better, we explain how to decide node schedules for a certain grid point x through an example shown in Figure 3.3. Suppose the duration of each round T is 30 minutes and only nodes A, B and C can provide coverage to grid point x . Nodes A, B and C choose reference time Ref values 4, 12 and 22, respectively. According to our algorithm, node B sets the T_{front} value to $(12-4)/2 = 4$ and set the T_{end} value to $(22-12)/2 = 5$ to cover the half gap between the reference time points. Consequently, the parameter tuple $(T, Ref, T_{front}, T_{end})$ for B is $(30, 12, 4, 5)$, which means node B stays awake for $(4+5)$ minutes out of the 30 minutes period (4 minutes before and 5 minutes after time reference point 12). Similarly, the parameter tuples for A and C are $(30, 4, 6, 4)$ and $(30, 22, 5, 6)$.

It should be noted that such a schedule for a single grid point is optimal in terms of energy

consumption, since the total awake time for all three nodes is equal to the duration of the round (30 minutes in the example). Any other schedule, which has less total awake time, introduces a blind time on that grid point. Since there is no overlap among the node's working schedules for grid point x and total coverage time equals the duration of the round, it is easy to conclude that at any time, grid point x is covered by at least one node's schedule.

The same procedure is carried out for every grid point. In the end, every node has a set of schedules for all the grid points it can cover. Since it is impossible for a node to execute multiple schedules simultaneously, it is necessary to create a single integrated schedule. Now we describe how a node combines this set of schedules for grid points into a single schedule for the area it can cover.

After each node calculates its schedules for all grid points it can cover, it creates an integrated schedule $(T, Ref, T_{front}, T_{end})$. The integrated T_{front} of node A is the largest one among all T_{front} time intervals of node A for all the grid points that node A can cover. Similarly, the integrated time interval T_{end} of node A is the largest one among the T_{end} time intervals of node A for all grid points that node A can cover. Therefore, the integrated schedule of each node is the union of its schedules for all the grid points it can cover. Previously, we have concluded that a given grid point is covered fully in time by at least one neighboring node's schedule. Thus with the calculated integrated schedules of all the sensor nodes, at any time any grid point is covered by at least one awoken sensor node. To clarify this integration process further, Figure 3.4 shows how to calculate the integrated schedule on a node according to its schedules for a set of grid points.

Note that many sensor networks are deployed with a reasonable density so that when every node is working, the whole target area is sensed by the sensing nodes without blind holes and our approach guarantees a 100% sensing coverage at any time during the sensing phase. Another noteworthy point is that, in order to prevent the nodes at the edge of the network from working all the time, when we specify the target area, we need to exclude the edge of the network. Then with enough density, it is highly probable that any grid point in the specified target area can be sensed by at least two nodes when all nodes are awake.

One might argue that intuitively such integration through the union of schedules for each grid

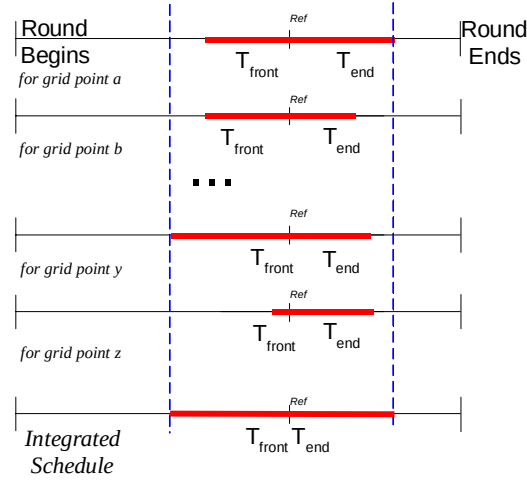


Figure 3.4: The Process of Schedule Integration on a node

point would lead to a pessimistic result with very long working duration. However, this is not the case. In fact, a node usually has the same schedule for grid points near to each other, which does not increase the length of integrated T_{front} and T_{end} when the union operation takes place. This has been shown in the simulation.

Our approach has several major advantages for sensor network deployment:

1. Communication overhead in our approach is minimized. For the approach described above, each node needs only to beacon one message without consideration of collisions. Compared to approaches that a node beacons each time it wakes up [59] or for each round [51], the energy consumed by our approach's communication is much less.
2. The random time reference point with uniform distribution contributes to the energy balance among the nodes. An optimization on this (discussed later) can enhance this energy-balancing feature even more.
3. Due to the optimal energy consumption property for a single grid point, the total energy consumption of our approach increases very slowly with the increase of node density. As a result, the system lifetime increases proportionally with the increase in node density. We show this in the evaluation section.

Although the length of working time for each node may differ due to randomness, the protocol is still fair in the sense that no node is given priority to work longer or shorter.

It can be easily seen from the above description that even when at the initialization phase some node is not discovered by its neighbors, once there are enough nodes to cover the target area, the 100% sensing coverage is still guaranteed. However, it may cause other nodes to work unnecessarily longer.

The protocol just described provides a guarantee of 100% sensing coverage at any time for the target area. Acknowledging that in some situations, full coverage by single sensor nodes is not enough to obtain high confidence in detection, our basic design can be easily extended to provide a guaranteed degree of coverage for a geographic area with very small overhead (see section 3.3).

3.2.4 Design Issues

This section completes our approach with several design issues concerning grid granularity, clock skew, irregular sensing patterns and node failures.

3.2.4.1 Dealing with Possible Blind Points in Space due to the Large Granularity of the Grid Size

One may argue that in our protocol description, there can be small sensing holes because we only guarantee grid point coverage, instead of a grid area. Here we explain why this approach still makes sense. We denote the unit grid size as d . We also define a conservative sensing area of a sensor node as a smaller concentric circle than its actual sensing area in order to handle this problem and other issues.

From Figure 3.5, it can be easily seen that the target area is fully covered by the sensor nodes' actual sensing areas as long as 1) each grid point is inside the conservative sensing area of at least one sensor node, and 2) the difference between the actual sensing radius r_{actual} and the conservative sensing radius $r_{conservative}$ is greater than $d/\sqrt{2}$. Then the problem of covering the whole target area with actual sensing areas reduces to that of covering the grid points with conservative sensing areas. With such observations, we decide the upper bound of the grid size according to the following rules.

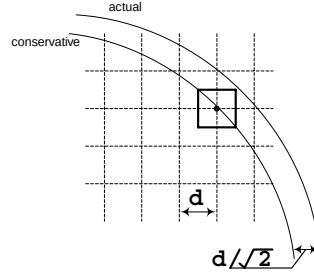


Figure 3.5: Grid Size Granularity Issues

1. If we use the actual sensing radius to calculate the nodes' working schedules, the grid size d should be smaller than the size of the targets we want to detect, so that these targets do not fit in the gap among the grid points.
2. If we use a conservative sensing radius, the grid size d should be smaller than $\sqrt{2}(r_{actual} - r_{conservative})$.
3. At the same time, a lower bound for the grid size should be set with consideration of computational cost (see 3.2.7.1).
4. Other factors like the imprecision of location information and irregularity of the sensing areas should also be considered when we decide the conservative sensing radius.

In the remaining protocol description, we do not distinguish between concepts of grid point coverage and area coverage.

3.2.4.2 Dealing with Possible Blind Points in Time due to Synchronization Skew

Due to the precision limits of synchronization algorithms for real sensor network systems, there is always some synchronization skew. When nodes try to trigger some events at the same time, the actual time the event is triggered on each node is different due to the synchronization skew. Our protocol is based on synchronization, so we must compensate for the synchronization skew to make the schedules seamless. We define the maximum skew of the whole sensor network as T_s . In order to bridge the gaps caused by the synchronization skew, we only need to extend the schedule of each

node for $T_s/2$ in both directions. This compensation affects the performance of the protocol. In order to minimize the impact of this performance degradation, it is favorable to increase the time duration T for each round. The reason is that when T_s is fixed, the longer a round is, the lower is the compensation overhead compared to the duration of a round or the schedules for a round. For example, suppose $T_{front}+T_{end}$ is 30 minutes, if we allow synchronization skew to be 1 second, the performance is degraded less than 0.1%. However, T should not be very large, which may lead to unbalanced energy consumption.

3.2.4.3 Dealing with Possible Blind Points in Space due to Irregularity in Sensing Patterns

In the description of the protocol, we assume that the sensing area of each node is uniformly a regular circle, and the nodes are deployed in a two-dimensional Euclidean plane. However, if we review the protocol description, it is clear that these assumptions are not mandatory. We define a node's sensing neighbors as the nodes that can sense at least a sub-area of its sensing area. Our protocol works as long as the sensing neighbors are aware of each other's sensing area, through pre-existing knowledge or communication with small overhead, no matter where the nodes are deployed and whether the sensing areas of nodes are regular or uniform. Even without precise information about sensing areas or node locations, we can use a conservative sensing range to make sure of sensing coverage.

3.2.4.4 Reschedule for Fault Tolerance

In this subsection, we propose a mechanism to provide fault tolerance with the basic protocol that guarantees a 100% coverage. We assume any two nodes do not fail at the same time and when some node fails, it just stops working silently. After each node calculates its integrated schedule, it broadcast this schedule to its $2r$ neighbors. When a node is working, it sends a heartbeat signal periodically to its working neighbors within the $2r$ range. Each working node knows the schedules of its $2r$ neighbors and expects heartbeat signals from the working ones among such neighbors. After a node detects the failure of one of its neighbors through heartbeat timeout, it wakes up all its neighbors that are also the $2r$ neighbors of the failed node. The wakeup mechanism can be

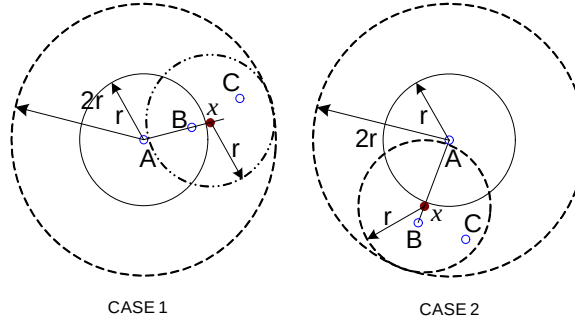


Figure 3.6: A Wakeup Example for Rescheduling

achieved through either a hardware wakeup circuit [24] or a software solution [32]. Then these $2r$ neighbors recalculate the schedules with the knowledge that the problematic node failed. With the new schedules, the target area is guaranteed to be fully covered by working nodes' sensing areas. In exchange for the fault tolerance feature, the communication overhead increases significantly due to the heartbeat signals.

Figure 3.6 gives a simple proof to show that all sleeping $2r$ neighbors of a failed node A can be awakened by at least one working node that is also node A 's $2r$ neighbor. There are two cases: 1) suppose a sleeping node B is inside the sensing area of node A . We extend the line segment that connects A and B . A point x on the extended line segment is very close, but outside of node A 's sensing area. There must be some working node C that is sensing point x . It can be easily seen that node C is in the $2r$ neighborhood of node A , and the distance of C and B is less than $2r$ so that node C can wake up node B . 2) Suppose a sleeping node B is outside of node A 's sensing area, we connect A and B with a line segment. We choose point x so that x is on the line segment and is very close, but outside of A 's sensing coverage. There must be some working node C that covers point x . It can be easily seen that C is in the $2r$ neighborhood of A and the distance of C and B is less than $2r$ so that C can wake up B .

3.2.5 Optimizations and Extensions

This section discusses optimizations and extensions built upon our basic design. These techniques achieve better performance in exchange for more beacons required.

3.2.5.1 Second Pass Optimization for Lower Energy Consumption

Each node's integrated schedule is the super set of its schedules for many grid points. It is possible that the integrated schedules are more than sufficient to provide the coverage guarantee. Therefore, we make a second pass optimization to reduce this kind of redundancy. In order to simplify the description, we only describe the optimization for a 100% degree of coverage. After a sensor node calculates the integrated schedule, it sends this integrated schedule to its neighbors within the distance of $2r$. When some node finds out that it is the one with the longest schedule, which has the largest $T_{front} + T_{end}$ value compared to its $2r$ neighbors, it recalculates a new schedule. The new schedule shrinks T_{front} and T_{end} values as much as possible, while still guaranteeing 100% coverage for all the grid points the node is able to sense. Then it beacons the new schedule to its neighbors within $2r$'s distance. Each node that has the longest schedule among the nodes in the $2r$ range that have not updated their schedules recalculates the schedule and beacons its new schedule. When there is a conflict of beacons, the conflicting nodes make use of a random back-off scheme to send the beacons again and avoid conflicts. In the evaluation section, we show that the system performance improves with this optimization.

3.2.5.2 Multi-Round Extension for Energy Balance

A sensor network's lifetime is affected by both the total energy consumption of all the sensor nodes and the variation of energy consumption per unit time by each node. In a sensor network that has higher variance in energy consumption, some nodes run out of energy much earlier than others do. Then blind holes appear in the target area, or another scheduling process should be applied to eliminate the blind holes. Given the same total energy consumed, the less is the variation of energy consumption per unit time, the longer the network lasts with full sensing coverage.

Energy consumption variance in our protocol can be attributed to at least two reasons. First, due to the randomness of node deployment, some nodes may have fewer neighbors than others in the range of $2r$, so they have to work longer than others do. This unbalance is intrinsic for a certain deployment and little can be done to deal with this problem. The second reason is due to the randomness of reference time. As an example, if several sensor nodes can all sense a grid point,

and their reference times are selected very close to each other, there must be an extraordinarily long schedule of one of the nodes for this grid point compared to others. Here we propose an extension to smooth the energy consumption variation caused by the randomness of reference time selection. Instead of calculating a single schedule, we calculate $\mathbf{M}$ schedules for each node according to $\mathbf{M}$ independently selected random reference times Ref for each node. At the initialization phase, each node beacons its $\mathbf{M}$ reference times Ref to its $2r$ neighbors. The i th (i is between 1 and $\mathbf{M}$) schedule is calculated with the i th reference time of each node. Then at each round in the sensing phase, the nodes choose one schedule consecutively. We show in the evaluation section that this extension decreases the variation in energy consumption.

3.2.5.3 Enhancement For Different Initial Node Energy

In the previous sections, we assume that all nodes have the same initial energy, however, this is not always the case in the real applications. Therefore, we propose a solution to deal with this issue.

First, intuitively, we need to modify the equations for deciding T_{front} and T_{end} . In the previous design, the time interval T_{front} and the time interval T_{end} for grid point x are set to $[Ref(i) - Ref(i-1)]/2$ and $[Ref(i+1) - Ref(i)]/2$, respectively, based on the assumption that all nodes have same initial energy and each node should work one-half time between two reference points. In the case of different initial node energy, we should consider this factor when deciding T_{front} and T_{end} . We modify the existing formulas into $[Ref(i) - Ref(i-1)]/[(Initial\ Energy\ for\ i)/(Initial\ Energy\ for\ i-1)]$ and $[Ref(i+1) - Ref(i)]/[(Initial\ Energy\ for\ i)/(Initial\ Energy\ for\ i + Initial\ Energy\ for\ i+1)]$, respectively. By changing the formulas, for each grid point, the amount of time that each node works is according to the portion of its energy to the total energy of two neighboring nodes, and therefore balances the energy consumptions among nodes for a grid point.

However, we found that modifying the equations for T_{front} and T_{end} alone cannot ensure the extension of network lifetime, because the integrated T_{front} and T_{end} for a node are the largest ones among all T_{front} and T_{end} , respectively. If on a specific grid point two low energy capacity nodes are neighbors, their T_{front} or T_{end} benefits little from the revised formulas discussed above, according to the algorithm for calculating integrated T_{front} and T_{end} , the optimized T_{front} and T_{end} then may

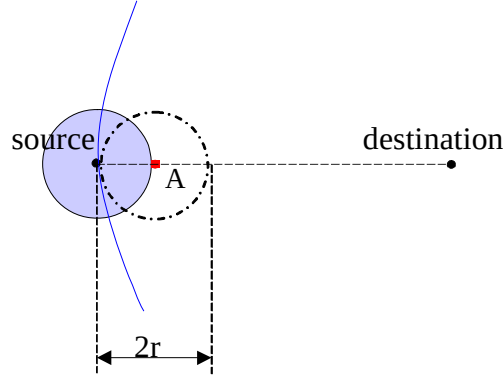


Figure 3.7: Communication Connectivity

be shaded by those un-optimized T_{front} and T_{end} , resulting small balancing gain in node energy consumption. Therefore we propose, instead of selecting reference points for nodes completely randomly, we separate the domain for reference points by the node density of network, and generate random reference points for high energy capacity nodes and low energy capacity nodes in certain intervals so as to increase the chance that the reference points of nodes with a significant energy difference are neighboring to each other.

For example, when the node density is 4 per r^2 , we assign the reference points for high energy capacity nodes to be the random numbers in the first and third quarter of round time, while setting low capacity ones to the second and fourth. So when we decide T_{front} and T_{end} for a grid point, the chance that high capacity ones are next to low capacity ones is much higher than what we can obtain by completely selecting reference points randomly. In the evaluation section, we show this extension improves the lifetime of network by 18% in case that nodes in the network have different initial energy.

3.2.6 Effect on Communication Connectivity

Sensing coverage and communication connectivity are two major targets for energy-efficient approaches for sensor networks. For quite a few routing protocols like [31], the communication connectivity is based on the existence of a next hop closer to the destination. Say we want to send a message from a source to a destination and the source cannot reach the destination in one hop

(Figure 3.7) in a sensor network whose 100% sensing coverage is guaranteed with our protocol. Let us consider the location A that is very close to, but out of the sensing circle of the source node and on the line segment connecting the source and the destination. In order to cover this location, there must be another node in the dashed circle with the radius r . We can see that any node inside the dashed circle is closer to the destination than the source node so that it can serve as the next hop. Thus, the existence of a next hop closer to the destination is guaranteed when a 100% sensing coverage is provided, which is a good feature that supports communication connectivity.

3.2.7 Protocol Analysis

This section provides the theoretical analysis of our algorithm for a better understanding of the performance issues.

3.2.7.1 Complexity Analysis

The major complexity involved in the proposed protocol is the computational complexity. In the following discussion, we analyze the steps the protocol takes to calculate the integrated schedule for a single node.

The first step is to calculate the schedules for each grid point. We know that the number of grid points in the node's sensing area is about $\pi r^2/d^2$, and suppose that a single grid point can be sensed by N nearby nodes on average, which depends on node density. For each grid point, a node should decide if it is in the sensing area of each node in the $2r$ range, which involves $4N\pi r^2/d^2$ distance computations. For each grid point, all these nodes' reference times are sorted, which takes $CN \log N$ basic steps, where C is a small constant. The schedule for a grid point is decided in negligible time. Last, it takes $2\pi r^2$ comparison to get the integrated schedule for this node. For instance, in a certain sensor network, we choose d as $r/10$ and N as 10, the basic steps it takes to calculate the integrated schedule is in the order of magnitude of ten thousand. In a typical hardware platform [21], the energy consumed for tens of thousands of basic calculations is in the same order of magnitude of transmitting a single bit, so it is negligible compared to communication overhead.

The memory space occupied for calculating the integrated schedule is composed of 1) $4N$ entries on average for the neighborhood table, 2) N memory units for sorting the reference times on average and 3) $2\pi r^2$ memory units for storing the schedules for grid points. Typically, the memory usage of the protocol on each node is in the order of magnitude of Kbytes, which is suitable for resource bounded sensor nodes like [21].

3.2.7.2 Communication Overhead Analysis

In the basic design, each node only beacons once (without consideration of collisions) in the initialization phase, disseminating the information about its reference time and location. During the sensing phase, there is no need to send extra messages for coordination. In the second round optimization, each node beacons three messages indicating the reference time, the integrated schedule and the updated schedule. Typically, as mentioned, the energy consumption for sending or receiving a single bit is on the same order of magnitude as running thousands of instructions. In real sensor networks, the retransmissions used to deal with contention make the energy consumption for sending a message even higher. Therefore, the very few number of messages transmitted in our protocol is preferable for minimizing energy consumption and extending system lifetime. Any extra energy consumed by complicated computation is negligible compared to the energy saved by decreasing communication overhead.

3.3 Enhancement with Differentiated Coverage

It is easy to modify our basic protocol to deal with differentiated sensing coverage when there are different requirements, statically or dynamically. In the basic algorithm, the middle point between two reference times is selected as the start/end point of a schedule for a grid point. That is to say, T_{front} and T_{end} are chosen as half of the intervals between two reference times, respectively. If we want to adjust the degree of sensing coverage to an arbitrary degree α , the only change in the protocol required is to extend or shrink T_{front} and T_{end} proportionally, i.e., T_{front} and T_{end} should be multiplied by α . It can be easily seen that with all the schedules for each grid point multiplied

by α , the aggregate schedule of a sensor node is also multiplied by α . There is one constraint for the extension: if the sum of the intervals multiplied by α is greater than the round duration T , we should shrink the multiplier so that $T_{front} + T_{end}$ does not exceed the round interval.

More formally, a node's working schedule for differentiated coverage is determined by the tuple $(T, Ref, T_{front}, T_{end}, \alpha)$, and a node executes the following working schedule:

A node wakes up at time $(T \times i + Ref - T_{front} \times \alpha)$ and it goes to sleep at time $(T \times i + Ref + T_{end} \times \alpha)$

The intended degree of coverage α provides a guarantee that the average number of working nodes that sense each grid point in each round is at least α , up to the limitation imposed by the number of sensor nodes present. Specifically, when α is 2, at any time, each grid point is sensed by at least 2 working nodes. When any single node fails, the target area is still fully covered. For an α value other than 1 and 2, the number of working nodes that sense a specific grid point is only guaranteed in the average sense. This feature is of great importance for a highly secured area. In fact, this differentiation mechanism can be generalized to the situations where α is smaller than 1. By doing so, our solution provides partial coverage in time to the target area and further reduces the total energy consumed.

It should be noted that this differentiation mechanism can be applied with very small overhead. After the sensor network is deployed, the sensor nodes are programmed so that any grid point of the target area is covered. When an emergency event happens, sensor nodes in the whole sensor network or part of the network should raise the degree of coverage immediately. For example, when a sensor network used for surveillance detects an intruder, more nodes should be awakened to achieve a more precise and reliable tracking of the intruder. The obvious advantage of our scheme is that no rescheduling through beaconing is required. One can disseminate the degree of coverage α to the target area and nodes simply expand their schedules according to this parameter to provide higher degree of coverage without further beacon exchange.

3.4 Other Extensions

In the basic protocol, the nodes select their reference points in a totally random and independent manner. Since intuitively the ideal scheduling results should end up with the scenario in which nodes in a neighborhood have short and relatively uniform working schedules, their reference points should also be distributed in a relatively uniform way. When the distance between two nodes is short, and these reference points are very close to each other, the schedules may be longer than necessary due to the schedule imbalance, therefore the randomness of choosing the reference points may hurt the total energy consumption, too.

Based on the above observations, we propose an enhanced design named “reference point scattering” to more evenly scatter the reference points. The idea of reference point scattering is rather simple. At the initialization phase, each node selects its reference point based on the reference points of other nodes in its circular neighborhood with radius R_{rs} that have already broadcast their beacons. For instance, if a node has received the reference points from k other nodes: $Ref_1, Ref_2, \dots$, and Ref_k , sorted in ascending order, then a period $[Ref_1, Ref_1 + T)$ is divided by the reference points into k gaps, $[Ref_1, Ref_2), \dots, [Ref_k, Ref_1 + T)$. The node chooses the maximum gap among them and selects the middle point as its own reference point and broadcasts it.

One issue here is the selection the radius of neighborhood R_{rs} . We show the impact of R_{rs} on performance through simulations in Section 3.6.

3.5 Dealing with Reality

In the basic protocol, we assume targets are one-dimensional points for the sake of simplicity. However, this assumption, unfortunately, imposes an unnecessary requirement that the area should be fully covered. Since in reality targets are not points, if we consider the target size and shape, we can guarantee the detection with a partial coverage, which leads to a better energy conservation. In the enhancement described in this section, we assume that the type of targets are known a priori, i.e., the minimal target size is one of the inputs into our coverage algorithm. We also assume that the shape of sensed targets can be described by arbitrary polygons. The requirement for the partial

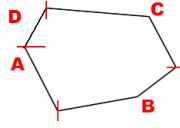


Figure 3.8: Polygon partitioned into 4 chains

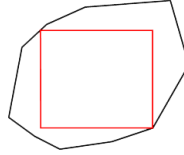


Figure 3.9: Two corners of rectangle on polygon

coverage enhancement is to ensure that (static or dynamic) targets do not fall into any sensing hole in the networks.

In the basic protocol, nodes are scheduled to cover the grid points. If the gaps between the grid points are very large, a target denoted by a polygon can exist within the gap, hence can not be detected. If the gaps between grid points are very small, we consume excessive energy. Ideally, we need to find the largest inscribed rectangle of a polygon and use this rectangle to build the grid. Finding inscribed rectangle is a typical problem in the field of computational geometry, where several solutions using the prune-and-search technique have been proposed [6,39,22]. To illustrate the idea without getting into too much detail, we give an example here. As shown in Figure 3.8, first, we identify the minimum X_{min} , Y_{min} and maximum X_{max} , Y_{max} values among the vertexes of the polygon. The vertexes with these values are called extreme vertexes (e.g., vertexes with a crossing-bar in Figure 3.8). Second, by breaking at extreme vertexes, we decompose the boundary of a polygon into four pieces (e.g., A, B, C, D as shown in Figure 3.8). Third, we consider two cases where an inscribed rectangle has either two (Figure 3.9) or three corners (Figure 3.10) on the boundary of this polygon. We can obtain the largest inscribed rectangle by searching the possible

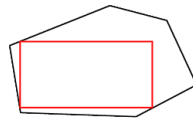


Figure 3.10: Process for finding the largest inscribed rectangle in a polygon

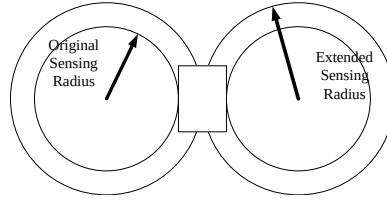


Figure 3.11: The method used to extend the sensing range

corner positions within divided pieces. As for the two-corner case, one of the corners must be located at a polygon vertex as shown in Figure 3.9. We omit the searching method for the more complex three-corner case, which can be found at [6]. By finding the largest inscribed rectangle, we are sure that as long as the four corners of the inscribed rectangle could be covered by the nodes, then the polygon could also be covered. As proven in the [6, 39, 22], there could be multiple largest inscribed rectangles for a polygon. In that case, we choose an inscribed rectangle that has the smallest difference between the longer and shorter sides to maximize the sensing range and grid size discussed below.

To make minimal change to the basic protocol, instead of changing the full coverage algorithm to a partial coverage algorithm, we *virtually* extend the physical sensing range by one half of the shorter side of the rectangle in our coverage algorithm, as shown in Figure 3.11. From Figure 3.11, we can see that even if the target is between the sensing circles of nodes, it still can be detected. We note that this enhancement is very simple. A change of the sensing range parameter according to the target size is all we need. This enhancement is also very flexible. If the assumption about the target size changes, all we need to do is to re-calculate the schedule with a different sensing range. This enhancement is orthogonal to other aspects of our protocol. For example, the differentiation scheme can be still applied together with this enhancement, offering more flexibility to the system design.

3.6 Evaluation

3.6.1 Baselines for Comparison

To enable a detailed evaluation, theoretical baselines and baselines from recent publications [51,59] are considered in our evaluation.

3.6.1.1 Theoretical Upper Bound and Lower Bound

Here we give the upper bound and lower bound of energy consumption per unit time for a 100% coverage of a target area. We use them as baselines. The upper bound is trivial and happens when all the sensor nodes are working all the time. For different hardware platforms, the energy consumed by a working node per unit time varies significantly. For generality, we denote it as one unit of energy consumption U_e . So the upper bound of energy consumption by a sensor network composed of N nodes is trivially NU_e . The lower bound of energy consumption of a sensor network differs according to the deployment of the sensor nodes. However, we are able to obtain a theoretical lower bound although it may not be reachable by a certain sensor network. The lower bound is based on the following theorem citewill01:

When we cover a unit area with equivalent circles, the lower bound of the number of circles used is $2/\sqrt{27}r^2$. The corresponding lower bound for energy consumption is $2AU_e/\sqrt{27}r^2$, where A is the size of the target area. Details of this theorem can be found in [53].

3.6.1.2 Comparison to Sponsored Coverage Scheme

[51] and [59] are the first works to address the sensing coverage problem in sensor networks. [59] proposes a mechanism based on probing for energy-efficient sensing coverage for sensor networks. In the mechanism, sleeping nodes wake up according to a dynamically changing wakeup rate. When a node wakes up, it broadcasts a probing message. If there is any working node in a radius of r centered at the wakeup node, the working node broadcasts a reply and the awakened node receives it. If the awakened node does not receive any reply in a certain amount of time, the node switches into working mode. The node in working mode continuously senses its sensing area until it fails

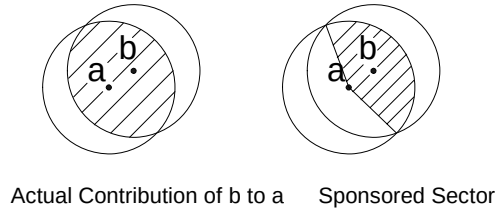


Figure 3.12: Understanding of Sensor Coverage by Neighboring Nodes

or runs out of energy. [51] shows that such a probing-based mechanism is not able to ensure the original sensing coverage and avoid blind holes. It also points out that although by decreasing the probing area, the probing based mechanism can minimize the area of blind holes; it keeps more nodes working compared to the sponsor coverage-based scheme in [51]. Hence, more energy consumption is required in [59]. Therefore, in the evaluation section we only compare our protocol with [51].

[51] is referred to as the sponsored coverage scheme in this chapter. It presents a node-scheduling mechanism based on an eligibility rule. At the beginning of each round, each node calculates its eligibility for going to sleep. A back-off mechanism is used to avoid sensing holes caused by simultaneous actions of multiple nodes. Our approach has two advantages over this one. First, the sensor nodes only need to send a few messages for scheduling in the initialization phase, while the scheme in paper [51] requires broadcasting to the neighbors at the beginning of each round. Although the exact results depend on the system settings, it can be easily concluded that the per-round based broadcast messages cause more overhead than ours. Second, when calculating eligibility for going to sleep, the contribution of a neighbor node's sensing area to the sensing area of the node itself is simplified to a "sponsored sector", which is much smaller than the actual contribution, especially when the neighbor node is very close to the node itself (Figure 3.12). So compared with our protocol, [51] needs more nodes to be awake; therefore, more energy consumption occurs.

In the rest of this section, we demonstrate the improved performance generated by our scheme in terms of 1) total amount of energy consumed, 2) energy variation among nodes, 3) half-life of the network and 4) sensing coverage over time. Moreover, we also demonstrate how we optimize the performance by considering the target size and unbalanced initial power that are not supported

by previous schemes.

In the evaluation we do not include communication cost due to data transfer because it is highly application specific. Also for some applications as intruder detection, data transfer only happens when some rare events are issued. For such systems, energy spent on data transfer is relatively insignificant.

3.6.2 Simulation Configuration

We run our basic protocol and extensions on a special purpose simulator. In our simulation, the sensor nodes are distributed in a $160\text{m} \times 160\text{m}$ square field. The sensing range is 10m and the communication range is 25m. The sensor nodes are deployed with a uniform distribution into the square field, unless otherwise stated. For our protocol, the target area is the $140\text{m} \times 140\text{m}$ square in the center of the square field to prevent the nodes at the edge from working all the time. We only do statistics on the central $100\text{m} \times 100\text{m}$ field to eliminate the edge effect. All experiments are repeated 10 times with different random seeds and different node deployments. The 95% confidence intervals of the results are about 5~10% of the means.

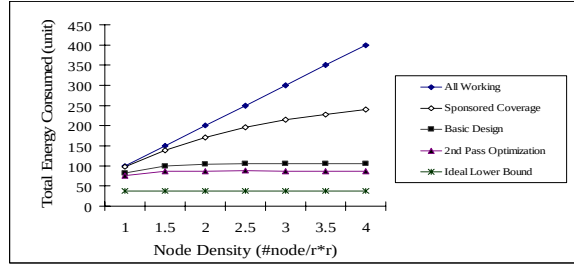


Figure 3.13: Total Energy Consumption per Unit of Time

3.6.3 Total Energy Conservation for the Basic Design

In this experiment, we investigate the energy conservation performance in term of total energy consumed per unit of time. We collect results from our basic design, the second pass optimization of our design and [51]’s sponsored coverage scheme. We also compare the simulation results with the lower bound and the upper bounds.

From Figure 3.13, we can see that our protocol consumes much less energy than [51]’s sponsored coverage scheme. This is due to the fact that in the sponsored sector based scheme, the contribution of one working node to another node’s sensing coverage is not fully considered. To be more specific, the sponsored sector is much less than the actual contribution (see 3.6.1.2). While in our protocol, the full contribution is considered so the redundancy is reduced. Another distinguished feature that our protocol provides is that with the increase of node density, the total energy consumed converges to about $105U_e$. From this observation, we use $1.05/r^2d$ as an estimate for the average duty cycle and apply this formula in the AADL analysis tool described in Chapter 6. In contrast, the sponsored sector based scheme suffers when density increases. This is because the closer nodes are to each other, the more severe an underestimation of sensor coverage by neighboring nodes is. As a result, our scheme outperforms sponsored coverage scheme by as much as a 50% reduction in energy consumption when the density is about 3 nodes per r^2 . Our scheme ensures that the system lifetime increases nearly proportionally with the node density of the sensor network, which is desirable for long-lifetime surveillance applications.

3.6.4 Balancing the Energy Consumption

In this simulation, we investigate performance of energy balance. We measure the standard deviation of energy consumed by each node in our basic design and in the multiple round extension with $M=10$. Results are compared with the sponsored coverage scheme.

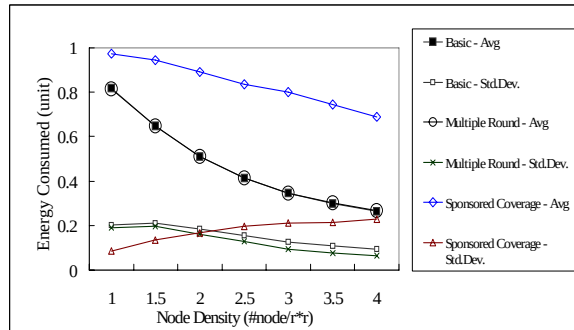


Figure 3.14: Single Node Energy Consumption: Average and Standard Deviation

Figure 3.14 shows that in both the sponsored coverage scheme and our scheme, the average

energy consumption for a single node reduces when the network density increases. However, our scheme reduces at a much faster rate than the sector-based scheme. When the node density is reasonably high ($> 2 \text{ node}/r^2$), our protocol outperforms the sponsored sector based scheme with regards to energy consumption balance among nodes. From Figure 3.14, we can also see that the multiple round extension can effectively reduce a certain amount of the variation in energy consumption.

3.6.5 Half-Life of the Network for the Basic Design

In order to measure the system lifetime extension due to our protocol, we define the half-life of a sensor network as the time from the beginning of the deployment until exactly half of the sensor nodes are still alive. We assume that the lifetime of a node when always working is 1000 minutes. The round duration we used in simulation is 10 minutes.

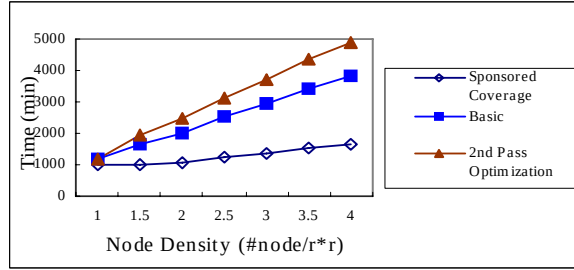


Figure 3.15: Half-Life of the Network

We see from the Figure 3.15 that the distinguishing feature of our approach is that the system half-life increases nearly linearly as the node density increases, while the sponsored approach increases slowly when the node density increases. For example, our approach increases the half-life of the network by 130% when node density is 4 per r^2 . There are two reasons contributing to this phenomenon. First, the sponsored coverage approach consumes more energy on average than our approach. Second, the standard deviation among nodes in the sponsored approach increases significantly as node density increases (Figure 3.15), which leads to some nodes dying faster than others. From the linearity of the simulation results of half-life for our basic protocol, we can estimate the system half-life as $T_{hl}r^2d$, in which T_{hl} is the system half-life without duty-cycling. We also use

this observation in the AADL analysis tool described in Chapter 6 to estimate the system half-life.

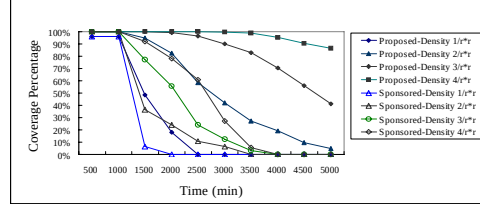


Figure 3.16: Sensing Coverage Over Time

Cost is another key constraint for utilizing wireless sensor networks in surveillance applications. Although deploying more nodes into the system can extend the system half-life based on the above observation, cost is also increased. Assume that the target area is a rectangle of $100m$ by $100m$, the sensing range r is $10m$ and the node density is 1 per r^2 , the total number of nodes is 100. A typical sensor node costs \$100 per piece. The total cost of all the sensor nodes is around 10,000 dollars to provide a 100% sensing coverage over the target area of $100m$ by $100m$. The lifetime of a typical sensor node is around two weeks if four AA batteries are used as the power source and the nodes never go to sleep. The resulting half-life of the system is also about two weeks. If the designers are willing to deploy the nodes with a density of 4 per r^2 , the system half-life is extended to about two months, although the total cost of all the sensor nodes is around 40,000 dollars. Therefore, the system designers need to balance the tradeoff between the cost and the half-life of the system properly. The formula in the last paragraph provides a convenient way for the designers to make such decisions.

3.6.6 Coverage over Time for the Basic Design

One of the most important things that the designers of sensor networks are interested in is, after a certain amount of time from when the network is deployed, how much of the target area is covered by the working sensors. In order to answer this question, we simulate the percentage of coverage after certain time intervals. The percentage of coverage takes care of the coverage in both time and space. For each grid point in the target area, we choose several sampling times during a round. If for some (grid point, time) pair, the grid point is sensed by some working node, we call this pair to

be "valid". We count the ratio of the valid pairs among all the pairs with both our proposed protocol and the sponsored sector based scheme described in [51]. Different curves shown in Figure 3.16 are obtained under different node density. The curves with the hollow markers are the results from the sponsored coverage scheme and the curves with the solid markers are the results from our scheme.

From the simulation results, we see that over time with the same node density, the target area is covered more with our protocol than with the sponsored coverage scheme. For example, at density of 2 per r^2 , it take about 5000 minutes for our scheme to degrade to zero sensing coverage, while the sponsored coverage scheme degrades to zero sensing coverage in just 3500 minutes.

3.6.7 Comparison with Regular Deployments

The above simulation results show the efficiency of the basic protocol when the nodes are deployed with a uniform random distribution. In this case, the variation among energy consumption is due to two random factors. First, the random selection of reference points contributes to the difference of energy consumption among nodes. Second, the randomness in the deployment is another reason of the variance. It is therefore interesting to explore through simulations how the latter factor, i.e., the deployment randomness affects the energy consumption variance among nodes. It is a well-known result that the most compact circular coverage of a two dimensional plane happens when the centers of circles are deployed on the grid points of a hexagonal grid. This suggests that a regular deployment of nodes may result in lower total energy consumption. We verify this hypothesis through simulations, too.

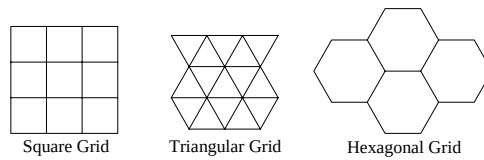


Figure 3.17: Three Types of Grids

We deploy the nodes according to three types of grids with various densities and run the simulations. The three types of grids are square grid, triangular grid and hexagonal grid, as shown in Figure 3.17. For each density and grid type, we calculate the corresponding distance between two

neighboring grid points and use it for simulation deployments. The means and standard deviations of energy consumption are obtained from the simulations and shown in Figure 3.18.

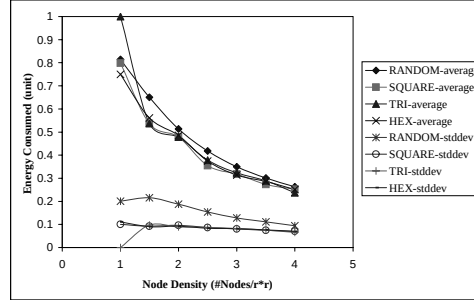


Figure 3.18: Energy Consumed for Grid Deployments

From Figure 3.18, we can see that the average values of energy consumption for regular deployments are almost always lower than those for random deployments when the densities are the same. This proves our earlier hypothesis. The only exception is for the triangular grid deployment when the density is 1 node per r^2 . The reason is that the distance between two neighboring nodes is $\sqrt{2/\sqrt{3}}r$, which is greater than the sensing range r . Each node has to be awake all the time to guarantee the full sensing coverage, because the small area around a node can only be covered by the node itself. The standard deviations of the energy consumption for regular deployments are significantly smaller than those for random deployments. This result indicates that the randomness in deployment does affect the variance of energy consumption among nodes significantly. Another interesting observation is that both the average values and standard deviations of energy consumption are very close for all three kinds of grid deployments. The only exception is for triangular deployment when the node density is 1 per r^2 . With the density increasing, the difference of the standard deviations between random deployments and regular deployments are decreasing.

We also compare the half-lives of the networks for random and regular deployments. The results are shown in Figure 3.19. From the figure, we can see that in general the networks with grid deployments have longer half-lives than randomly deployed networks when the densities are the same. However, which grid type gives better half-lives depends on the node density.

We compare sensing coverage over time for the various deployments. The node density we use is 2 per r^2 . The results are shown in Figure 3.20. From Figure 3.19 we know that at this

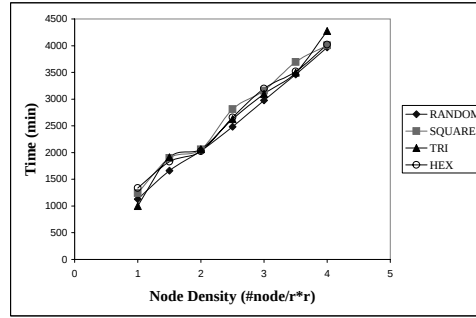


Figure 3.19: Half-Life for Grid Deployments

density, all four kinds of deployment have almost the same system half-life, about 2000 minutes. Figure 3.20 is very interesting: compared with the random deployment, regular grid deployments give a significantly higher sensing coverage (almost 100%) before the system half-life. After that, the coverage provided by regular deployments decreases much abruptly than that by the random deployment. After 3000 minutes, the coverage provided by regular deployments are smaller than that provided by the random deployment. The reason is that for regular deployments, the variation of energy consumption among nodes is significantly smaller than the random deployment case. Therefore, for regular deployments, the time when nodes are depleted of energy is more uniform, which causes more nodes to run out of energy soon after the system half-life.

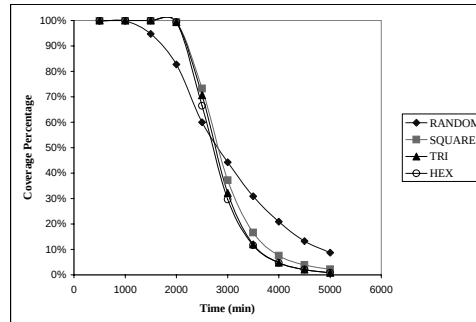


Figure 3.20: Coverage Over Time for Grid Deployments

3.6.8 Total Energy Consumed for Differentiated Surveillance

Unlike previous schemes, our basic design can be easily extended to provide differentiated surveillance by proportionally increasing the T_{front} and T_{end} interval values. Figure 3.21 shows the simu-

lation results of differentiated coverage with two different node densities.

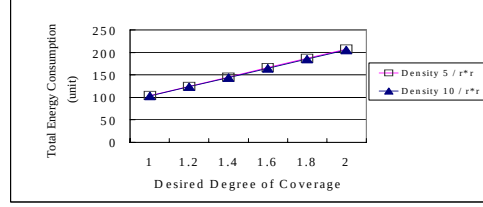


Figure 3.21: Total Energy Consumed for Differentiated Surveillance

From the results, we conclude that 1) the energy consumed per unit time increases linearly when the parameter of desired degree of coverage α increases and 2) the energy consumed under different node densities changes very little when the node densities are high enough. This is shown in Figure 3.21 where the two curves nearly overlap each other. This independence between the physical node density and actual energy consumed is quite desirable.

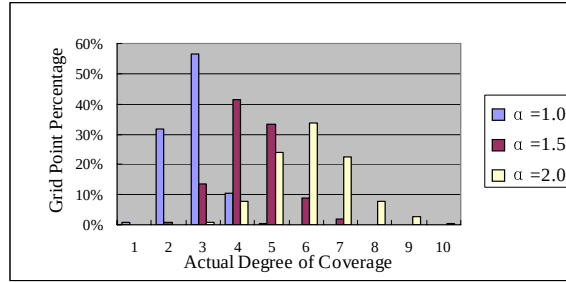


Figure 3.22: Actual Degree of Coverage for Differentiated Surveillance

3.6.9 Actual Degree of Coverage for Differentiated Surveillance

In this simulation, we set the node density to be 5 per r^2 , and sample the actual degree of coverage for 100×100 grid points in the center. In addition to the proof from section 3.3, Figure 3.22 shows that our scheme can guarantee a degree of coverage to the desired value α for the target area. For example, if α is 2, then we guarantee that all grid points are covered by more than two sensor nodes and a single failure node does not affect the full coverage. However, the guaranteed α is the minimum possible degree of coverage for each grid point. Due to the protocol's redundancy, most grid points are covered by a higher degree than the desired degree of coverage α . Figure 3.22

shows the distribution of actual degree of coverage for the grid points when we apply different desired degrees of coverage α . For example when $\alpha = 2$, we guarantee that no grid point has an actual degree of coverage of less than 2. We acknowledge that the current solution only guarantees a sensor coverage lower bound and future work will continue to investigate the possibility to provide a sensor coverage upper bound for better energy conservation.

3.6.10 Energy Consumption for Considering Target Size

In this experiment, we study the improved energy conservation when considering the target sizes. Figure 3.23 shows the simulation results of total energy consumed per unit time for four different node densities. Figure 3.24 denotes the average energy consumption and standard deviation for a single node. And Figure 3.25 demonstrates the impact of target sizes on total energy consumption. The unit of target sizes is the sensing range r .

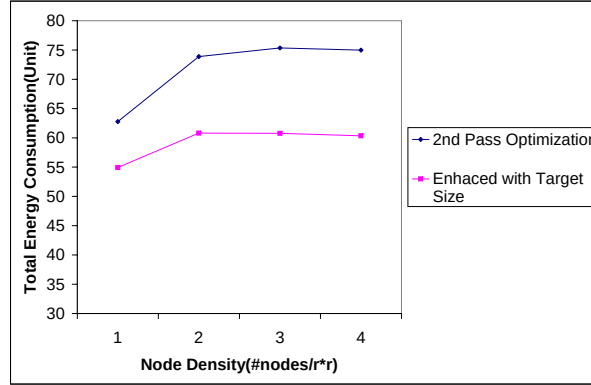


Figure 3.23: Total Energy Consumption per Unit of Time(Target Size=2)

From the results, we can conclude that this enhanced scheme outperforms the 2nd-pass optimization for all node densities, and as node density increases, the ratio of conserved energy also increases. At node density of 4 nodes per r^2 and target size of 2, the total energy consumed and average node energy consumption of enhanced scheme are less than 81% of the 2nd pass optimization scheme, while the standard deviation for single node energy consumption also reduces. And as target size increases, the total energy consumption drops almost linearly. The reason for these phenomena is that the new scheme increases grid size based on target size and virtually extends

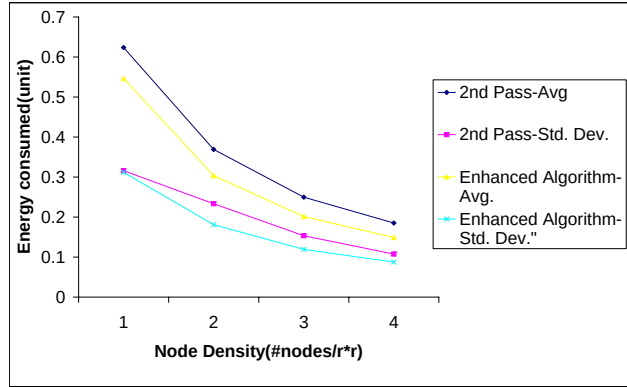


Figure 3.24: Single Node Energy Consumption: Average and Standard Deviation (Target Size=2)

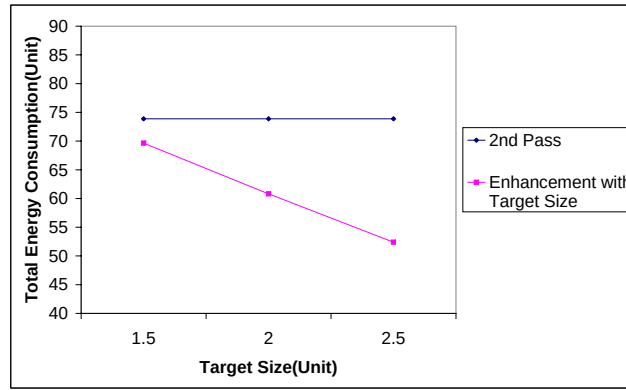


Figure 3.25: Total Energy Consumption Vs. Target Size (Node Density=2/r*r)

the sensing range. Therefore, as node density increases, the chance that a single node has more neighboring nodes increases dramatically, and consequently leads to less per-node and total energy consumption.

3.6.11 Half-Life of the Network for Considering Target Size

To further measure system lifetime extension with the scheme that assigns grid size based on target size, we compared the half-life of enhanced scheme with original 2nd pass optimization scheme under different node densities.

From Figure 3.26, we can see that the half-life of both new and original algorithm grow almost linearly, but with different slopes. The new scheme has a relatively larger slope, which implies a longer half-life for all node densities. Moreover, the rise of target size is also accompanied by the

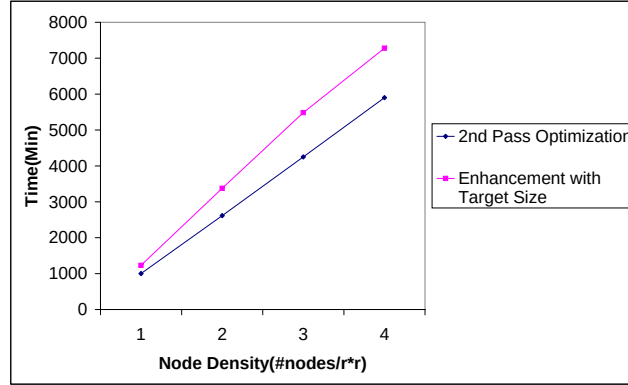


Figure 3.26: Half-Life of the Network (Target Size=2)

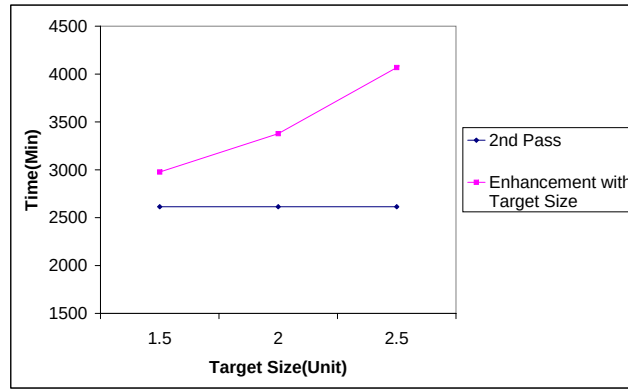


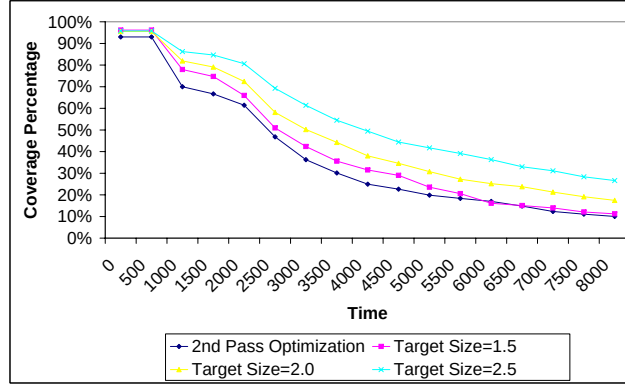
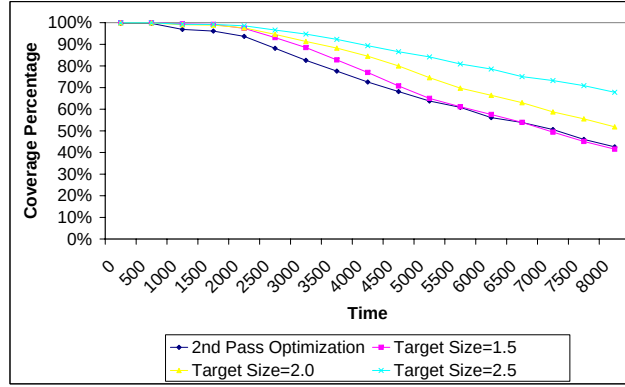
Figure 3.27: Half-Life of the Network Vs. Target Size (Node Density=2/r*r)

linear improvement on system half-life, which is demonstrated in Figure 3.27. In the simulation, at target size of 2, the new scheme has approximately 1.2 times longer half-life time than the 2nd pass optimization scheme, due to the reduced average node energy consumption.

3.6.12 Coverage Over Time for Considering Target Size

In real applications, the coverage over time is one of the most important issues that designers of sensor networks need to consider. In Figures 3.28, 3.29, 3.30 and 3.31, we show the coverage over time with four different node densities.

The four figures show that the new scheme always has better coverage than the original design under different node densities. During the whole network lifetime, the gap between the coverage in the new algorithm and the original algorithm is increasing over time. This is due to the relatively

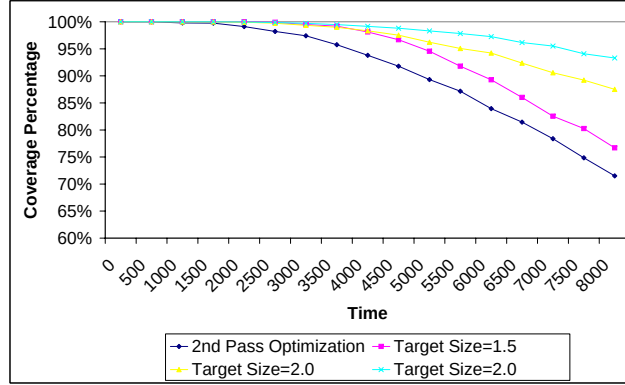
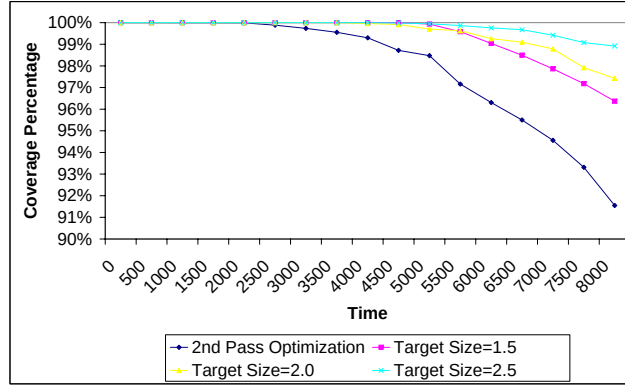
Figure 3.28: Sensing coverage over time ($1/r \cdot r$)Figure 3.29: Sensing coverage over time ($2/r \cdot r$)

lower standard deviation for average node energy consumption in the enhanced algorithm.

3.6.13 Half-Life for Different Initial Node Energy

In this experiment, we randomly assign half nodes with the initial energy of a 500-minute lifetime capacity and the other half with a 1000-minute lifetime capacity.

In Figure 3.32 we can see that the algorithm enhancement when the system has different initial energy leads to a longer half life of the network than the basic design under all node densities. And as node density increases, the benefits gained by the enhancement become more significant. For instance, at the node density of 4 per r^2 , the enhancement outperforms the basic design by 18% on the network half-life. The reason for this result is that by optimizing reference points and formulas for calculating T_{front} and T_{end} , the reference points for nodes that have high energy capacity now

Figure 3.30: Sensing coverage over time ($3/r \cdot r$)Figure 3.31: Sensing coverage over time ($4/r \cdot r$)

are around the ones have low energy capacity, therefore, maximizing the energy balance among nodes.

3.6.14 Energy Consumption for Reference Point Scattering

In this subsection we simulate the average values and standard deviations of energy consumption of nodes with various densities when the reference point scattering enhancement is applied. In the simulation, we choose various values for the neighborhood radius R_{rs} and explore the impact of the R_{rs} values on the performance. We compare the results in Figure 3.33 with the basic design performance. From the figure, we can see that the average energy consumption when $R_{rs} = 0.5r$ is lower than that of the basic design. The average value is even lower when $R_{rs} = 1.0r$. However, when $R_{rs} = 1.5r$, the average energy consumption increases and when $R_{rs} = 2.0r$, the average

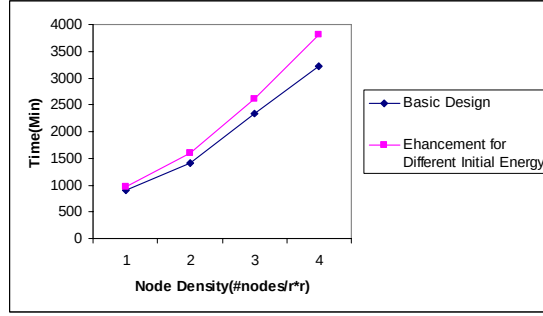


Figure 3.32: Half-Life of the Network

value is greater than that of the basic design. This shows that when $R_{rs} = 1.0r$, the average energy consumption reaches its optimum value among those with other R_{rs} values and the basic design. The reason that $R_{rs} = 2.0r$ does not give a better result is that when the distance between two nodes is close to two times the sensing range, the overlapping coverage area is very small, therefore it is not helpful to consider one reference point into the decision of another. The improvement on the variance of the energy consumption is not significant.

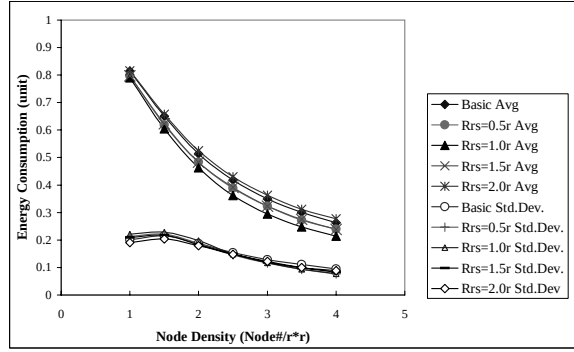


Figure 3.33: Total Energy Consumed for Reference Point Scattering

3.6.15 Half-Life for Reference Point Scattering

With the same simulation setting as the above, we obtain the half-lives of the system. From the results shown in Figure 3.34, we can see that when $R_{rs} = 1.0r$, the system half-life increases over that of the basic design by as much as 25%. When $R_{rs} = 0.5r$ and $R_{rs} = 1.5r$, there is still a gain in the system half-life, but not as much as that of $R_{rs} = 2.0r$, the system half-life is shorter than that of

the basic design. These results are consistent with the results of simulations for the average energy consumption.

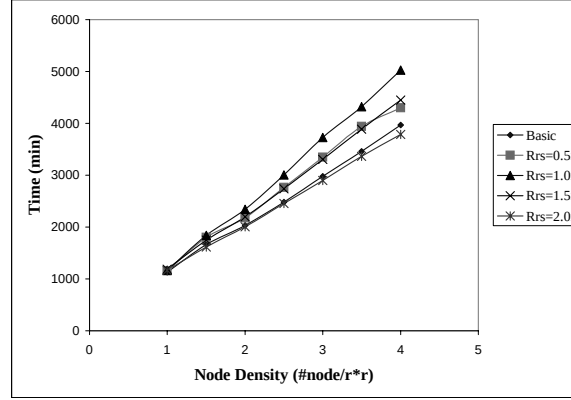


Figure 3.34: Half-Life for Reference Point Scattering

3.6.16 Coverage Over Time for Reference Point Scattering

In this set of simulations, we choose the node density as 2 per r^2 , and the other settings are the same as the above. We obtain the sensing coverage over time and show the results in Figure 3.35. From the figure, we can see that coverage for the case that $R_{rs} = 1.0r$ is significantly greater than that of the basic design. Coverage for the cases that $R_{rs} = 0.5r$ and $R_{rs} = 1.5r$ is better than that of the basic design before 4000 minutes, but worse after that. When $R_{rs} = 2.0r$, the coverage is always worse than that of the basic design. Again, these results are consistent with the results of simulations for the average energy consumption.

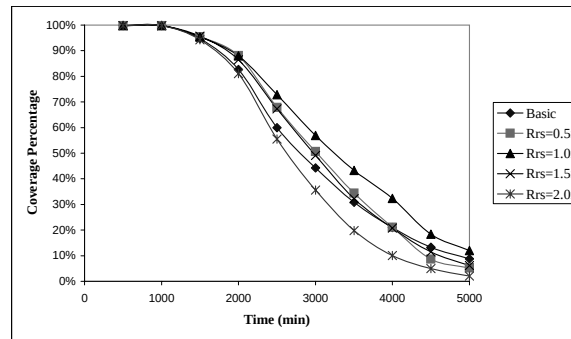


Figure 3.35: Coverage Over Time for Reference Point Scattering

From the above three sets of simulations, we can conclude that $1.0r$, or the sensing range itself, is a good choice for R_{rs} when reference point scattering is applied.

3.7 Conclusions

In this chapter, we introduce an adaptable sensing coverage mechanism for differentiated surveillance in sensor networks. Unlike previous schemes, our scheme guarantees not only full sensing coverage to a certain geographic area, but also the degree of coverage up to the limit imposed by the density of nodes available. The distinguishing advantages of our scheme are a much longer network half-life through energy conservation and balancing among sensor nodes, and a small communication overhead required to establish a working duty schedule among nodes. Several optimizations and extensions are proposed to provide even better performance in terms of network lifetime and energy balance. And with the enhancement considering the target size and shape, we could gain more benefits in energy conservation. With the enhancement considering the initial energy discrepancy among nodes, we achieve a better energy balance among sensor nodes, hence a longer network half-life. Simulations show that our protocol accomplishes differentiated surveillance with low energy consumption and outperforms other state-of-art schemes by as much as a 50% reduction in energy consumption and as much as a 130% increase in the half-life of the network.

Chapter 4

Low Duty-Cycle Surveillance

This chapter presents work on performance analysis of low duty-cycle surveillance. Section 4.1 introduce the problem we solve. Section 4.2 gives detailed derivation of tracking analysis results for both stationary and mobile targets. The theoretical analysis is validated by simulations in Section 4.3. Section 4.4 explores the impacts of various realistic issues, such as irregular sensing models and more realistic deployment strategies, on the surveillance performance. Section 4.5 concludes the chapter.

4.1 Introducing the Low Duty-Cycle Surveillance Problem

A broad range of current sensor network applications involve surveillance. One common goal for such applications is reliable detection of targets with minimal energy consumption. Although maintaining full sensing coverage guarantees immediate response to intruding targets, sometimes it is not favorable due to its high energy consumption. Therefore, designers are willing to sacrifice surveillance quality in exchange for prolonged system lifetime. The challenge is to obtain the analytical relationship that depicts the exact tradeoff between surveillance quality and system parameters in large-scale sensor networks. In this chapter, we characterize surveillance quality by average detection delay and detection probability of intruding targets. For system parameters, we are mainly concerned with duty cycle and node density. This knowledge answers the question of whether a system with a set of parameters is capable of achieving its surveillance goals.

In this chapter, we establish the relationship between system parameters and surveillance attributes. Our closed-form results apply to both stationary and moving targets, and are verified through extensive simulations. Throughout this chapter, we adopt the model of unsynchronized duty-cycle scheduling for individual nodes. In this model, nodes sleep and wake-up periodically. Nodes agree on the length of the duty cycle period and the percentage of time they are awake within each duty cycle. However, the wakeup times are not synchronized among nodes. We call this model, *random duty-cycle scheduling*. There are two reasons for this choice. First, random scheduling is probably the easiest to implement in sensor networks since it requires no coordination among nodes. Coordination among nodes takes additional energy and may be severely impaired by clock drifts. Second, many surveillance scenarios pose the requirement of stealthiness (i.e. that nodes minimize their electronic signatures). Coordinating node schedules in an ad hoc network inevitably involves some message exchange which creates a detectable electronic signature, especially for messages that are periodically beamed to adjust clock drifts for time synchronization. In contrast, random scheduling does not require communication. Each node simply sets its own duty-cycle schedule according to the agreed-upon wakeup ratio, and starts surveillance. No extra messages need to be exchanged before a potential target appears. Random, unsynchronized scheme is also much more resilient to clock drifts. Based on these two considerations, we focus this chapter only on random duty-cycle scheduling.

We make two major contributions of the work presented in this chapter. First, for the first time, we obtain closed-form results to quantify the relationship between surveillance attributes and system parameters. The advantage is that we can now answer a variety of important questions without the need for simulation. For example, to decrease the average detection delay for a potential target by half, how much should we increase the node density? Another interesting question is, given the width of the system area and other system parameters, what is the probability that a target with a maximum speed of 20 mph is detected by the sensor nodes? In this chapter, we demonstrate that these questions, along with many others, can be answered analytically based on our results. Second, we explore the impact of realistic issues on the applicability of theoretical results. We propose a realistic (irregular) model of sensing based on empirical measurements. This model is

then incorporated into our simulation to prove the robustness of our analytical predictions. The results show that even under the irregular sensing model, the analytical closed-form results are still quite accurate. We also study factors such as other deployment strategies.

4.2 Theoretical Derivation

4.2.1 Notations and Assumptions

In this section, we present the notations and assumptions for our derivations. First of all, we assume that nodes are independently and identically distributed conforming to a uniform distribution. Other deployment strategies are discussed in Section 4.4.2. Density d is defined as the total number of nodes divided by area in which the system is deployed, A_{system} . In our analysis, we adopt a simple sensing model in which a point is covered by a node if and only if the distance between them is less than or equal to the sensing range R of the node. For simplicity, we first assume that all nodes have the same sensing range in all directions. We shall address sensing irregularity in Section 4.4.1. Each node is assumed to have a scheduling period T and a duty cycle ratio β , $0 \leq \beta \leq 1$, that defines the percentage of time the node is awake. Each node chooses its wakeup point t_{start} independently and uniformly within $[0, T)$, wakes up for a period of time βT , and then goes back to sleep until $T + t_{start}$. Finally, we assume that for all practical purposes the entire system area is covered when all nodes are awake.

Each target in the area moves in a straight line at a constant speed v in our analysis. We shall address movements along curves in Section 4.4.3. We assume that all targets are point targets so that their physical sizes can be neglected. Observe, however, that larger targets can still be analyzed by increasing the sensing range used in the analysis by the diameter of the target to account for the larger sensory signature.

4.2.2 Stationary Target Sensing Analysis

In this section, we analyze the expected value and probability distribution of detection delay t_d for a stationary, persistent target (e.g., a localized fire).

First, according to our assumptions, since nodes are deployed with a uniform distribution, the number of nodes within an area of πR^2 conforms to a binomial distribution $B(A_{system}d, \pi R^2/A_{system})$. For a sensor network with a reasonable scale, $A_{system}d$, the total number of nodes, is a large number, and $(A_{system}d)(\pi R^2/A_{system}) = \pi R^2d$ is a constant, denoted as λ . With these conditions, the binomial distribution can be approximated by a Poisson distribution with parameter λ . Observe that $\lambda = \pi R^2d$ is the average number of nodes within a sensing range. The probability that there are n nodes covering an arbitrary geometric point is $\frac{\lambda^n}{n!}e^{-\lambda}$, $n = 0, 1, 2, \dots$. The probability that no node covers the point (i.e., $n = 0$) is $e^{-\lambda}$. To make this probability smaller than 1%, λ should be at least $-\ln 0.01 \approx 4.6$. Observe that the detection delay for points that are not covered is infinite, which makes the expected detection delay for the whole area infinitely large. To avoid this problem, we take into account only those points that are covered by at least one node. Our calculations above indicate that as long as there are more than 4.6 nodes on average within a sensing range, the probability that a point is not covered is less than 1%, which is negligible.

In the rest of the chapter, we extensively use two general results from theory of probability. First, if the probability of an event A occurring in a single experiment is p , and if n experiments are conducted, the probability of event A occurring at least once in the series of experiments is:

$$P = 1 - (1 - p)^n \quad (4.1)$$

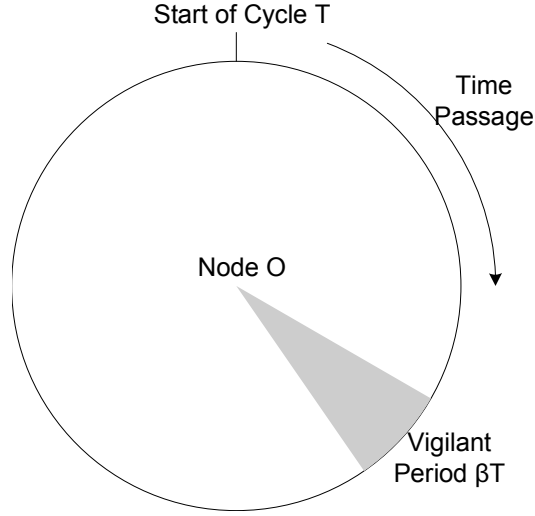
If n conforms to a certain probability distribution instead of being a constant value, the probability becomes:

$$P = E(1 - (1 - p)^n) \quad (4.2)$$

In particular, if n , the *number of experiments* conforms to a Poisson Distribution with parameter λ , the probability of event A occurring at least once in the series of experiments is:

$$P = 1 - e^{-p\lambda} \quad (4.3)$$

In the analysis, we first calculate the probability of detection when the target is covered by

Figure 4.1: Duty Cycle of a Particular Node O

one node in a certain area. Since the number of nodes in a certain area conforms to a Poisson distribution, we get the probability that at least one node covers the target.

Another important result relating to our derivation is Proposition 11.6 in [46]: Let X be a nonnegative random variable, then $E(X) = \int_0^\infty P(X > x)dx$.

Since the cumulative distribution function (CDF) of X $cdf(x) = P(X \leq x)$, we have

$$E(X) = \int_0^\infty (1 - cdf(x))dx \quad (4.4)$$

We will use this fact to derive the average detection delay for both stationary and mobile target detection, after we get the probability that the target is detected within a certain period of time.

Now consider an arbitrary point in the area to be monitored. Suppose node O is the only node covering this point. The duty cycle of node O is shown in Figure 4.1. Denote the random variable corresponding to the detection delay at this point as t_d . Since the node has a probability β of being awake, we have $P(t_d = 0) = \beta$.

The probability density function (PDF) of t_d $f(\tau)$, where $0 < \tau \leq T - \beta T$, conforms to a uniform distribution, so $f(\tau) = \frac{1}{T}$, as long as the target can arrive uniformly anywhere within the duty-cycle. Therefore, when there is only one node covering the point, the cumulative probability distribution

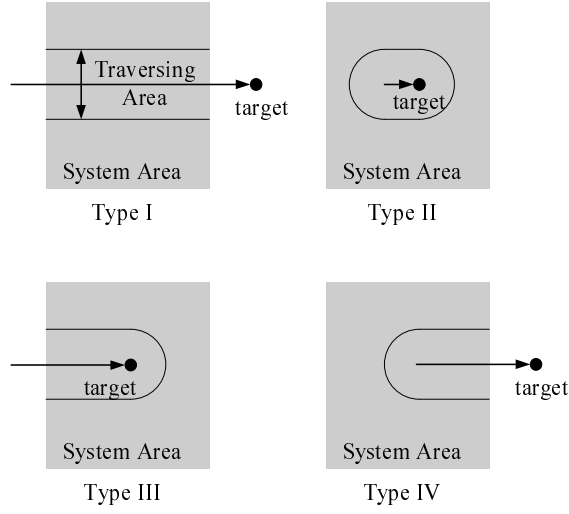


Figure 4.2: Target Detection Scenarios

for the detection delay is:

$$F_{t_d}(\tau) = P(t_d \leq \tau) = \beta + \frac{\tau}{T}, \quad (4.5)$$

where $0 \leq \tau \leq T - \beta T$. When there are n nodes covering the point, we consider the detection of each node as an experiment. Let event A correspond to the fact that the target is detected within an interval time no larger than t_d . Substituting for the single event probability p in Equation (4.3) from Equation (4.5) we get the probability of event A :

$$P(A) = 1 - e^{(-\frac{\lambda\tau}{T} - \beta\lambda)} \quad (4.6)$$

Since we focus only on those points that are covered, the CDF of detection delay for such points is $P(A)/1 - e^{-\lambda}$, or:

$$F_{t_d}(\tau) = \frac{1 - e^{(-\frac{\lambda\tau}{T} - \beta\lambda)}}{1 - e^{-\lambda}}, 0 \leq \tau \leq T - \beta T \quad (4.7)$$

where $e^{-\lambda}$ is the probability of voids.

Second, according to Equation (4.4), the expected value of the delay t_d is:

$$E(t_d) = \int_0^\infty P(t_d > t) dt = \int_0^\infty \frac{e^{-\lambda(\frac{t}{T} + \beta)}}{1 - e^{-\lambda}} dt = \frac{T}{\lambda} e^{-\beta\lambda} / (1 - e^{-\lambda}) \quad (4.8)$$

In particular, when the duty cycle β approaches 0 and $1 - e^{-\lambda}$ approaches 1, Equation (4.8) turns into T/λ . Therefore, we conclude that when the duty cycle is sufficiently small and the density is sufficiently large, the average detection delay is inversely proportional to node density. Therefore, when a very small duty cycle is used, to decrease the expected detection delay, the node density needs to be increased in an inversely proportional manner.

4.2.3 Mobile Target Sensing Analysis

In this section, we analyze the detection delay for mobile targets. We consider a target that moves at velocity v along a straight path, and consider four detection scenarios, as shown in Figure 4.2. We categorize these scenarios based on whether or not the start or end point of the target path is inside the system area. A type I target is a target whose start and end points are both outside the monitored area. A type II target is a target whose start and end points are both inside the monitored area. A type III target starts from outside the monitored area but ends inside the monitored area. A type IV target starts from inside the monitored area but ends outside the monitored area. For target scenarios of type I and type IV (where the end point is outside the monitored area), we are interested in the probability that the target is detected by at least one node before it leaves the area. For target scenarios of type II or type III (where the end point is inside the monitored area), we are interested in the expected time and distance the target travels before it is detected. The system area is assumed to be rectangular, in which nodes are randomly deployed. The target is detected within a radius R (or diameter $2R$) which constitutes the width of the traversing areas shown in Figure 4.2. We focus on type I and type II scenarios, and outline the results for the other two types.

We demonstrate a more detailed model for type II target detection in Figure 4.3 (type I model is similar). Consider the traversing area S consisting of the rectangle and two half-circles in the figure. For a potential target that travels from point A to point B (where distance $AB = L$), only

nodes located within this area can detect the target, for example, node M , which has an intersection length of l with the target's moving track. Therefore, we call this area the detection area. Since all nodes are deployed conforming to a uniform distribution, the number of nodes in the detection area also conforms to Poisson Distribution approximately. According to Equation (4.3) mentioned above, in order to find the probability of detection, we only need to find the detection probability when there is only one node within this area.

Now consider node M . Since it has an intersection length with the target track, the potential target takes time l/v to pass its sensing area. Therefore, the target appears to be a temporary event with a lifetime of l/v to node M , which means it has a probability of $\min(\beta + \frac{l}{vT}, 1)$ to be detected by node M . Considering that node M could be anywhere within the detection area, we now analyze the average detection probability for node M to detect the target.

Notice that $\beta + \frac{l}{vT}$ can be at most 1, and if it is, the target will definitely be detected when it passes. This fact categorizes potential targets into two types, fast and slow. For fast targets, the expression $\beta + \frac{l}{vT}$ is always smaller than or equal to 1. Therefore, we can obtain the expectation of detection delay directly by integrating over the detection area. On the other hand, if v is small enough, $\beta + \frac{l}{vT}$ can become larger than 1, thus, we have to partition the detection area first before the integral. Observe that the maximal intersection length between the target path and a sensor's range is $2R$. Therefore, if target velocity $v \geq \frac{2R}{(1-\beta)T}$, then $\beta + \frac{l}{vT}$ is always smaller than or equal to 1. We use $\frac{2R}{(1-\beta)T}$ as the threshold between fast and slow targets. Here we give the definitions for fast targets and slow targets:

Definition: a *fast target* is one that may traverse a node's sensing range along its diameter without being detected.

Definition: a *slow target* is one that is always detected if it traverses a node's sensing range along its diameter.

For fast targets, we can apply $\beta + \frac{l}{vT}$ as the detection probability for a single node without exceptions. For slow targets, we have to identify the locations of nodes where $\beta + \frac{l}{vT}$ is greater than 1, and use 1 instead of $\beta + \frac{l}{vT}$ for these locations as the detection probability for a single node, so the calculation is more complicated.

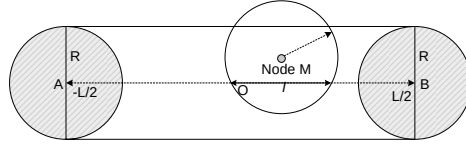


Figure 4.3: Target Detection Example

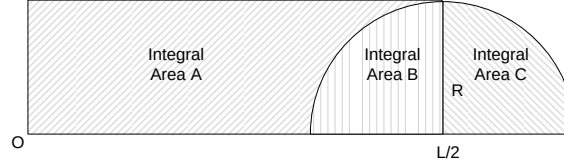


Figure 4.4: Fast Target Detection

4.2.3.1 Detection Analysis for Fast Targets

For a fast target we can express the probability that the target is detected given there is only one node in the detection area, S , as follows:

$$P = \int_S \left(\beta + \frac{l}{vT} \right) ds / S = \beta + \frac{\int_S l ds}{vTS} \quad (4.9)$$

Thus, we only need to calculate $\int_S l ds$ and S .

Because of symmetry, we only need to calculate the integral $\int_S l ds$ within the first quadrant, as shown in Figure 4.4.

For a type I target, we need to do the integral over area A and B , while for a type II target, we need to do the integral over A , B and C .

Fast Type I Target Analysis

For a type I target, the area under consideration consists of area A and B . Therefore, we have:

$$\int_{S_{A+B}} l ds / S_{A+B} = \frac{\phi_A + \phi_B}{LR/2} \quad (4.10)$$

while

$$\phi_{A+B} = \int_0^R dy \int_0^{\frac{L}{2}} 2\sqrt{R^2 - y^2} dx = \frac{\pi R^2 L}{4} \quad (4.11)$$

Thus, we get the overall probability as:

$$P = \beta + \frac{\pi R^2 L}{2RLvT} = \beta + \frac{\pi R}{2vT} \quad (4.12)$$

Fast Type II Target Analysis

Similar to the type I target analysis, we have:

$$\int_{S_{A+B+C}} l ds / S_{A+B+C} = \frac{\phi_A + \phi_B + \phi_C}{LR/2 + \pi R^2/4} \quad (4.13)$$

Therefore,

$$\phi_A = \int_0^R dy \int_0^{\frac{L}{2} - \sqrt{R^2 - y^2}} 2\sqrt{R^2 - y^2} dx = \frac{\pi R^2 L}{4} - \frac{4R^3}{3}. \quad (4.14)$$

$$\phi_B = \int_0^R dy \int_{\frac{L}{2} - \sqrt{R^2 - y^2}}^{\frac{L}{2}} \left(\frac{L}{2} + \sqrt{R^2 - y^2} - x \right) dx = R^3. \quad (4.15)$$

$$\phi_C = \int_0^R dy \int_{\frac{L}{2}}^{\frac{L}{2} + \sqrt{R^2 - y^2}} \left(\frac{L}{2} + \sqrt{R^2 - y^2} - x \right) dx = \frac{R^3}{3}. \quad (4.16)$$

Thus, we get the overall probability as:

$$P = \beta + \frac{\pi R^2 L}{(2RL + \pi R^2)vT} = \beta + \frac{\pi RL}{(2L + \pi R)vT} \quad (4.17)$$

Note that for the case $L < 2r$, the derivation is a little different. However, the result remains the same and for simplicity we omit the details of derivation.

We also have a more intuitive explanation for the calculation of $\int_S l ds$. This expression is an integral of a line segment of length l on the target's locus for each point p in the traversing area S .

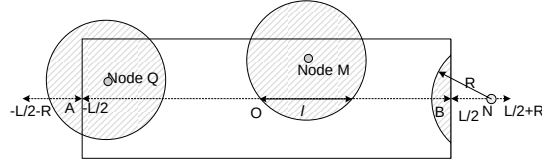


Figure 4.5: Fast Target Detection Illustration

Note that a point p' belongs to the line segment if and only if its distance from p is smaller or equal to R , the sensing range. Therefore, the integral $\int_S l ds$ is equivalent to the integral $\int_L s dl$, where L is the locus of the target, and s is the area in which each point has a shorter distance than R to dl . For type II targets, s is simply πR^2 and therefore $\int_S l ds$ is simply $\pi R^2 L$, which verifies Equation (4.17). It is a little more complicated for type I targets since s can be smaller than πR^2 . For example, for point N in Figure 4.5, s is a half-lens. The expression then becomes the integral of the overlapping area of the circular disk with radius R and the traversing area over L . We can think it as a circular disk virtually moving from $-L/2 - R$ to $L/2 + R$ and calculate the accumulation of overlapping areas. To make the calculation simpler, equivalently, we can also imagine a fixed circular disk and virtually move the traversing area and calculate the accumulation overlapping areas. This also gives a result of $\pi R^2 L$, which verifies Equation (4.12).

4.2.3.2 Detection Analysis for Slow Targets

Now consider $v < \frac{2R}{(1-\beta)T}$. The main difference in this case is that it is possible that for certain node positions (x, y) , $l(x, y) > (1 - \beta)vT$, therefore $p(x, y)$ is 1 instead of $\beta + l(x, y)/vT$. Suppose there are two partitions, U and V . In U , $p(x, y) = \beta + l(x, y)/vT$, and in V , $p(x, y) = 1$. Therefore, we have:

$$\begin{aligned}
 P &= \frac{\int_{S_U} (\beta + \frac{l}{vT}) ds + \int_{S_V} 1 ds}{S_{A+B}} = \frac{\int_{S_{U+V}} (\beta + \frac{l}{vT}) ds + \int_{S_V} (1 - \beta - \frac{l}{vT}) ds}{S_{U+V}} \\
 &= \beta + \frac{\int_{S_{U+V}} l ds}{(S_{U+V})vT} + \frac{S_V(1 - \beta) - \int_{S_V} l ds/vT}{S_{U+V}}
 \end{aligned} \tag{4.18}$$

Obviously, the term $\beta + \frac{\int_{S_{U+V}} l ds}{(S_{A+B})vT}$ is exactly what we have obtained for fast object analysis. So

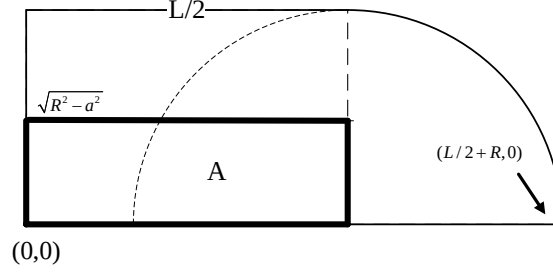


Figure 4.6: Type I target Detection

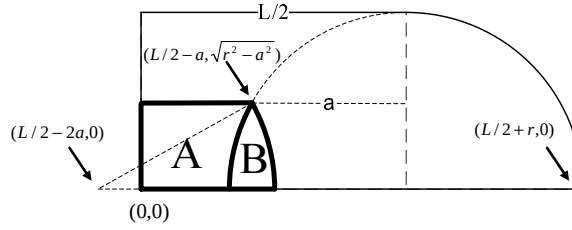


Figure 4.7: Regions A and B

now we need to calculate $S_V(1 - \beta)$ and $\int_{S_V} l$. We classify the analysis into two cases according to type I and type II targets.

Slow Type I target Analysis

In this type, the partition where $p(x, y) = 1$ is the rectangle area A in Figure 4.6. We define another variable a such that $\beta + 2a/vT = 1$. The height of the rectangle is $\sqrt{R^2 - a^2}$.

Observing that $\frac{S_V(1 - \beta) - \int_{S_V} l ds/vT}{S_U + V} = \frac{\int_{S_V} [(1 - \beta)vT - l(x, y)] dxdy}{S_U + VvT}$. Therefore, we have:

$$\int_V [(1 - \beta)vT - l(x, y)] dxdy = \int_A [2a - l(x, y)] dxdy \quad (4.19)$$

in which

$$\begin{aligned} \iint_A [2a - l(x, y)] dxdy &= \int_0^{\sqrt{R^2 - a^2}} \int_0^{\frac{L}{2}} (2a - 2\sqrt{R^2 - y^2}) dxdy \\ &= \int_0^{\sqrt{R^2 - a^2}} [aL - L\sqrt{R^2 - y^2}] dy = aL\sqrt{R^2 - a^2} - L\left(\frac{a\sqrt{R^2 - a^2}}{2} + \frac{R^2}{2} \sin^{-1} \frac{\sqrt{R^2 - a^2}}{R}\right) \\ &= [aL\sqrt{R^2 - a^2} - LR^2 \sin^{-1} \frac{\sqrt{R^2 - a^2}}{R}] / 2 \end{aligned} \quad (4.20)$$

Therefore we obtain:

$$\iint_A [2a - l(x, y)] dx dy = Lk(R, a)/4 \quad (4.21)$$

where

$$k(R, a) = 2a\sqrt{R^2 - a^2} - 2R^2 \cos^{-1}\left(\frac{a}{R}\right) \quad (4.22)$$

Finally, we get when $L \geq 2R$, $v < \frac{2R}{(1-\beta)T}$, for a type I target,

$$P = \beta + \frac{\pi R^2 + k(R, a)}{2RvT} \quad (4.23)$$

Slow Type II target Analysis

In this type, the partition where $p(x, y) = 1$ is less regular. We still define a such that $\beta + 2a/vT = 1$. Observe that Figure 4.7 plots the region where $p(x, y) = 1$, which includes A and B , whose boundaries are formed by line $y = \sqrt{r^2 - a^2}$ and two circles which centered at $(L/2 - 2a, 0)$ and $(L/2, 0)$, respectively. For $(x, y) \in A \cup B$, $p(x, y) = 1$.

Similarly, we have:

$$\int_V [(1 - \beta)vT - l(x, y)] dx dy = \int_{A+B} [2a - l(x, y)] dx dy \quad (4.24)$$

in which

$$\begin{aligned} \iint_A [2a - l(x, y)] dx dy &= \int_0^{\sqrt{R^2 - a^2}} \int_0^{\frac{L}{2} - \sqrt{R^2 - y^2}} (2a - 2\sqrt{R^2 - y^2}) dx dy \\ &= \int_0^{\sqrt{R^2 - a^2}} \left(\frac{L}{2} - \sqrt{R^2 - y^2}\right) (2a - 2\sqrt{R^2 - y^2}) dy \end{aligned} \quad (4.25)$$

$$\begin{aligned}
& \iint_B [2a - l(x, y)] dx dy = \int_0^{\sqrt{R^2 - a^2}} \int_{\frac{L}{2} - \sqrt{R^2 - y^2}}^{\frac{L}{2} - 2a + \sqrt{R^2 - y^2}} [2a - (\sqrt{R^2 - y^2} - \frac{L}{2} - x)] dx dy \\
& = \int_0^{\sqrt{R^2 - a^2}} [(2a - \sqrt{R^2 - y^2} - \frac{L}{2})(2\sqrt{R^2 - y^2} - 2a) \\
& + 2a^2 - aL - 2a\sqrt{R^2 - y^2} + L\sqrt{x^2 - y^2}] dy
\end{aligned} \tag{4.26}$$

Therefore we can obtain:

$$\iint_{A+B} [2a - l(x, y)] dx dy = (L - 2a)k(R, a)/4 \tag{4.27}$$

where

$$k(R, a) = 2a\sqrt{R^2 - a^2} - 2R^2 \cos^{-1}\left(\frac{a}{R}\right) \tag{4.28}$$

Finally, we get when $L \geq 2R$, $v < \frac{2R}{(1-\beta)T}$, for a type II target,

$$P = \beta + \frac{\pi R^2 L + (L - 2a)k(R, a)}{(2RL + \pi R^2)vT} \tag{4.29}$$

Indeed, in Figure 4.7, we only plotted the case where $2a > R$. For $2a < R$, the derivation is similar, and the results are the same.

When $a = 0$, $k(R, a) = -\pi R^2$. When $a = R$, $k(R, a) = 0$. We also have

$$\frac{\partial k(R, a)}{\partial a} = 4\sqrt{R^2 - a^2} \geq 0, \tag{4.30}$$

so $k(R, a)$ is a monotonically non-decreasing function of a . Interestingly, the area of B is exactly $-k(R, a)/2$.

Denote $m(R, \beta)$ as the function substituting a with $(1 - \beta)vT/2$ in $k(R, a)$, then

$$m(R, \beta) = (1 - \beta)vT \sqrt{R^2 - \frac{(1 - \beta)^2 v^2 T^2}{4}} - 2R^2 \cos^{-1}\left[\frac{(1 - \beta)vT}{2R}\right] \tag{4.31}$$

$m(R, \beta)$ is a monotonically non-increasing function of β . When $\beta = 1 - \frac{2R}{vT}$, $m(R, \beta) = 0$. When $\beta = 1$, $m(R, \beta) = -\pi R^2$.

Additionally, for a slow type II target, if the travel distance L is less than $(1 - \beta)vT$, the detection probability for a node located within the detection area cannot be larger than 1. Therefore, we should treat this period of time in the same way as a fast target. Therefore, we can revise the final probability as:

$$P = \beta + \frac{\pi R^2 L + \min((L - 2a)k(R, a), 0)}{(2RL + \pi R^2)vT} \quad (4.32)$$

So far, we have finished the derivation of $P(L, v)$ for both slow and fast objects. Next we find the detection probability for a type I target, and average detection delay for a type II target.

Average Detection Delay and Probability

First, consider fast targets. According to Equation 4.3, for a fast type I target, with a deployment width of L , we obtain the detection probability as follows:

$$P_{\text{detection}}(v) = 1 - e^{-2RLdP} = 1 - e^{-2RLd(\beta + \frac{\pi R}{2vT})} \quad (4.33)$$

while for slow targets, the result is:

$$P_{\text{detection}}(v) = 1 - e^{-2RLdP} = 1 - e^{-2RLd(\beta + \frac{\pi R^2 + k(R, a)}{2RvT})} \quad (4.34)$$

For fast type II targets, the cumulative distribution function of the detection delay t_d is:

$$F_{t_d}(t) = P(t_d \leq t) = 1 - e^{-d(2Rvt + \pi R^2)(\beta + \frac{\pi Rvt}{(2vt + \pi R^2)vT})} \quad (4.35)$$

While for slow targets, the function is:

$$F_{t_d}(t) = P(t_d \leq t) = 1 - e^{-d(2Rvt + \pi R^2)(\beta + \frac{\pi R^2 vt + \min(0, (vt - 2a)k(R, a))}{(2Rvt + \pi R^2)vT})} \quad (4.36)$$

For fast targets, the expected detection delay is

$$E(T_d) = \int_0^\infty e^{-\beta\pi R^2 d - \nu t(2R\beta + \frac{\pi R^2}{vT})} dt = \frac{e^{-\beta\pi R^2 d}}{(2R\beta v + \frac{\pi R^2}{T})d} \quad (4.37)$$

Similarly, when $\nu < \frac{2R}{(1-\beta)T}$, the expected detection delay is¹:

$$\begin{aligned} E(T_d) &= \int_0^{2a} e^{-\beta\pi R^2 d - \nu t(2R\beta + \frac{\pi R^2}{vT})} dt + \int_{2a}^\infty e^{-\beta\pi R^2 d + \frac{2ak(R,a)d}{vT} - \nu t(2R\beta + \frac{\pi R^2 + k(R,a)}{vT})} dt \\ &= \frac{e^{-\beta\pi R^2 d}}{(2R\beta v + \frac{\pi R^2}{T})d} \left[1 - \frac{m(R, \beta) e^{-(2R\beta vT + \pi R^2)(1-\beta)d}}{2R\beta vT + \pi R^2 + m(R, \beta)} \right] \end{aligned} \quad (4.38)$$

4.2.3.3 Summary

So far, we have explained in detail the derivation for type I and type II targets in this chapter. We also have finished the derivation for the detection delay and probability of type III and IV targets, using similar derivation techniques. We only outline our final results for type III and type IV targets, as follows:

Expected detection delay for fast type III targets:

$$E(T_d) = \frac{e^{-\beta\pi R^2 d/2}}{(2R\beta v + \frac{\pi R^2}{T})d} \quad (4.39)$$

Expected detection delay for slow type III targets:

$$E(T_d) = \frac{e^{-\beta\pi R^2 d/2}}{(2R\beta v + \frac{\pi R^2}{T})d} \left[1 - \frac{m(R, \beta) e^{-(2R\beta vT + \pi R^2)(1-\beta)d}}{4R\beta vT + 2\pi R^2 + m(R, \beta)} \right] \quad (4.40)$$

Detection probability for fast type IV targets:

$$P = 1 - e^{-(2RL + \pi R^2/2)d(\beta + \frac{\pi RL}{(2L + \pi R/2)vT})} \quad (4.41)$$

¹ Observe that if $\nu = 0$ (this is equivalent to a stationary target), we obtain the expected delay to be ∞ . This is because we do not exclude voids in the mobile target model, and the existence of voids leads to infinitely large expected detection delay in stationary target detection.

Detection probability for slow type IV targets:

$$P = 1 - e^{-(2RL + \pi R^2/2)d(\beta + \frac{\pi R^2 L + \min((L-a)k(R,a),0)}{(2RL + \pi R^2/2)vT})} \quad (4.42)$$

In Figure 4.8, we show the analysis results for all four types of mobile targets in a table.

4.2.3.4 Discussion

We now discuss the implications of our analytical results. First, we assume that almost all of the area is covered, that is, $1 - e^{-\lambda} \approx 1$, thus, $\lambda \geq 4.6$ according to our earlier discussion. In real applications, it is often desirable to deploy an adequate number of nodes to minimize the number and size of voids so that this assumption holds. Second, in order to save energy, we assume that β approaches 0, that is, the waking period is sufficiently small compared with the total scheduling period. Based on these two assumptions, we have several interesting observations.

Observation 1. For fast type I target detection, Equation (4.33) can be simplified to $1 - e^{-\frac{\lambda}{vT}L}$. Therefore, in order to obtain 99% detection probability, we have $L \geq \frac{4.6vT}{\lambda}$. Assume $\lambda \approx 4.6$, $L \geq vT$. The result is quite intuitive: in order to almost certainly catch an intruding target, the deployment width can be no smaller than the product of the target velocity and the scheduling period.

Observation 2. For a fast type II target, Equation (4.35) can be simplified to $1 - e^{-\frac{\lambda}{T}}$. This means that for a target that starts from within the system area, the probability of being caught increases exponentially. Also, to make this probability larger than 99%, $t \approx T$, given that λ is 4.6. This result is also confirmed in our simulations.

Observation 3. For a type II target, the detection delay can be simplified as T/λ if β approaches 0. This result is the same as the stationary detection delay, which means that as β approaches 0, regardless of the movement pattern of the target, the detection delay is approximately constant, determined only by the scheduling period and node density.

Type I (Probability)	Fast	$1 - e^{-2RLd(\beta + \frac{\pi R}{2vT})}$
	Slow	$1 - e^{-2RLd(\beta + \frac{\pi R^2 + m(R, \beta)}{2RvT})}$
Type II (Delay)	Fast	$\frac{e^{-\beta\pi R^2 d}}{(2R\beta v + \frac{\pi R^2}{T})d}$
	Slow	$\frac{e^{-\beta\pi R^2 d}}{(2R\beta v + \frac{\pi R^2}{T})d} \left[1 - \frac{m(R, \beta)e^{-(2R\beta vT + \pi R^2)(1-\beta)d}}{2R\beta vT + \pi R^2 + m(R, \beta)} \right]$
Type III (Delay)	Fast	$\frac{e^{-\beta\pi R^2 d / 2}}{(2R\beta v + \frac{\pi R^2}{T})d}$
	Slow	$\frac{e^{-\beta\pi R^2 d / 2}}{(2R\beta v + \frac{\pi R^2}{T})d} \left[1 - \frac{m(R, \beta)e^{-(2R\beta vT + \pi R^2)(1-\beta)d / 2}}{2R\beta vT + \pi R^2 + m(R, \beta)} \right]$
Type IV (Probability)	Fast	$1 - e^{-(2RL + \frac{\pi R^2}{2})d(\beta + \frac{\pi RL}{(2L + \frac{\pi R}{2})vT})}$
	Slow	$1 - e^{-(2RL + \frac{\pi R^2}{2})d(\beta + \frac{\pi R^2 L + \min((L-a)m(R, \beta), 0)}{(2RL + \frac{\pi R^2}{2})vT})}$
$m(R, \beta) = (1 - \beta)vT \sqrt{R^2 - \frac{(1 - \beta)^2 v^2 T^2}{4}} - 2R^2 \cos^{-1}[\frac{(1 - \beta)vT}{2R}]$		

Figure 4.8: Detection Probability or Expected Delay for Four Types of Targets

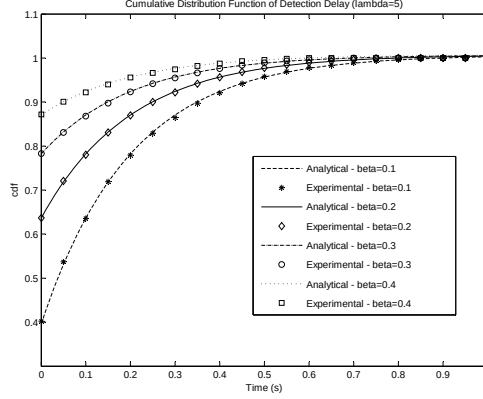


Figure 4.9: Stationary Detection Delay Distribution

4.3 Evaluation

We now demonstrate that the derivation results are consistent with simulation results under perfect circular range assumptions. We show the experimental delay distributions and average detection delays for both stationary and mobile targets through simulations, and then compare them with the results from the analysis in the last section. In the first set of simulations of stationary targets, locations of nodes are generated conforming to a uniform random distribution over a unit area with size $100m \times 100m$, without loss of generality. The period T is chosen to be $1s$. The waking points of the nodes are generated according to a uniform distribution over $[0s, 1s]$. The sensing range for circular model is $10m$. We choose a point $(50, 50)$ and generate 10,000 rounds of random target locations to run the simulations. The parameter $\lambda = \pi r^2 d$ is set to 5. We record the the number of experiments where the detection delay is smaller than or equal to $0s, 0.05s, 0.10s, \dots, 0.90s$, and $0.95s$. We then compare these frequencies with the cumulative distribution functions (CDF's) obtained in the analysis section. The simulation results are shown in Figure 4.9. The lines are obtained from the analytical formulas and the marks are obtained from simulations. The same convention is used for the rest of this section. From the figure we can see that our analysis and simulation results match well.

In the second set of simulations, we choose various λ values (5, 7, 9 and 11) and β values (0, 0.1, 0.2, ..., 1), and run simulations to gather the average detection delays. Other settings are the

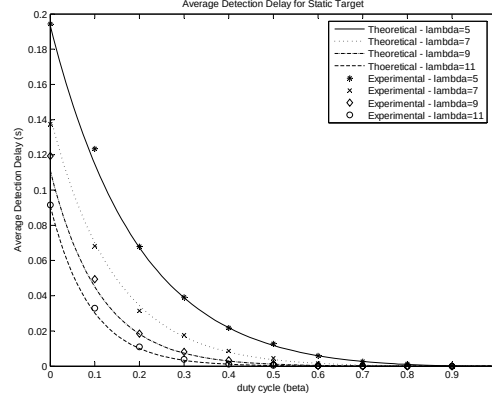


Figure 4.10: Average Stationary Detection Delay

same as the previous set. We compare the average detection delays with the theoretical expected detection delays obtained in the analysis section and plot Figure 4.10. The figure shows that the average delays are very close to the analytical expected delays.

The settings for simulation verification with mobile target tracking are the same as the stationary setting, except that now the target has a velocity. We only consider type II target detection in this section, and other types can be similarly verified. The simulation result is shown in Figure 4.11. Observe that velocity $2.5m/s$ means a slow target, according to our threshold. In particular, its cumulative detection probability has a relatively long tail, compared to the distribution of targets with higher velocity, implying that for slower targets, there is a higher chance for the target to remain undetected after the scheduling cycle T . Again, we observe that our analysis and the simulation results match well.

In the next set of simulations, we choose various λ values (5 and 10), velocity ($5m/s$ and $50m/s$), and β values (0, 0.1, 0.2, ..., 1), and run simulations to gather the average detection delays. Other settings are the same as in the previous set. We compare the average detection delays with the theoretical expected detection delays obtained in the analysis section and plot figure 4.12. The figure shows that the average delays are very close to the analytical expected delays.

The simulation results demonstrate the viability of using our results for both stationary and mobile targets to predict the system performance. For example, suppose that we have a deployment

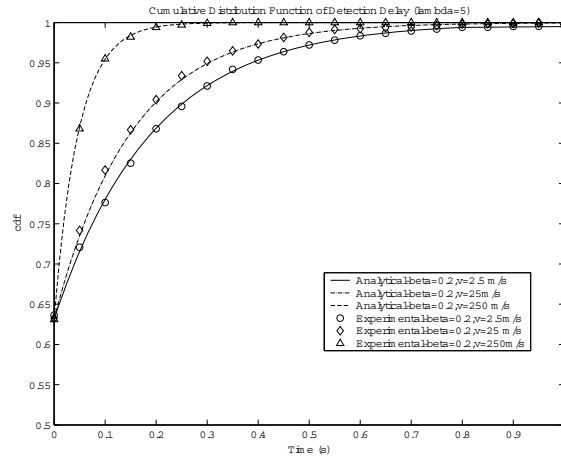


Figure 4.11: Mobile Detection Delay Distribution

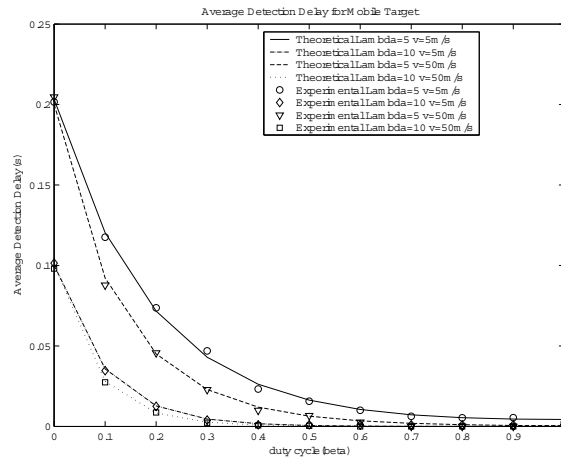


Figure 4.12: Average Mobile Detection Delay

of $\lambda = 5$ and $T = 1s$, in order to make sure that the expected detection delay to be no larger than $0.1s$, we can calculate that β must be at least 0.14 . The simulation results confirm this calculation (Figure 4.10).

4.4 Dealing with Reality

In this section, we explore the impacts of various realistic issues on the theoretical analysis results. In Section 4.4.1, we study the impact of irregular sensing models. In Section 4.4.2, we study the impact of other deployment schemes. We also show that we can estimate the detection delays when the target is moving along a circular curve using the straight line analytical results through simulation in Section 4.4.3.

4.4.1 Dealing With Irregular Sensing Models

4.4.1.1 Sensing Irregularity In Real Applications

An example application of random duty-cycle sensor networks is a surveillance system that tracks trespassers. A person is detected by motion sensors. Once such an event is triggered, the detecting node sends a wakeup command to its neighbors to track the intruder. These command constitute messages with a preamble that is longer than the duty-cycle period T . Thus, all neighbors eventually wake up and receive the message. It alerts them to remain awake and track, possibly waking up their neighbors as well. Subsequent detections from multiple sensor nodes eventually reconstruct the target's path. Other types of sensors, such as magnetic ones, can be applied to tell if the person carries a weapon.

We simulate the aforementioned application and check whether simulation results match analytical predictions. However, our discussions so far have assumed a perfectly circular sensing model. This assumption makes the analysis tractable, and leads to closed-form results. Real sensing devices used in this application do not have such a perfect sensing area. In experiments, we use passive infrared sensors (PIR sensors) as motion sensors, and identify the causes of irregular sensing range. We then propose a realistic sensing model for PIR sensors. We integrate this model into

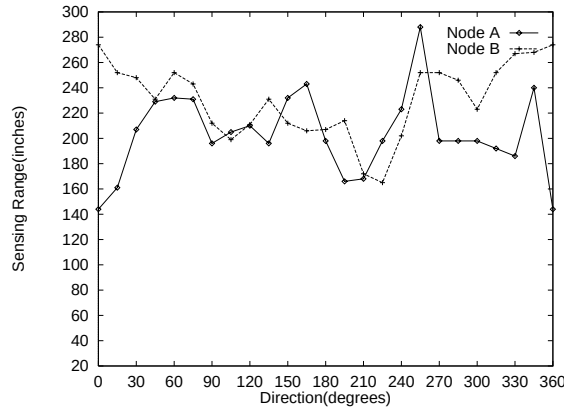


Figure 4.13: Range with Directions

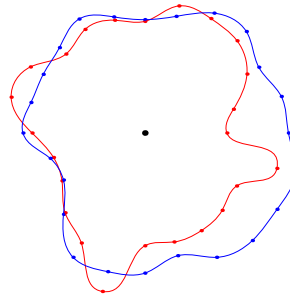


Figure 4.14: Ranges of Two Nodes

our simulations and show that the predictions based on our analysis are quite robust under realistic sensing irregularities.

We have PIR sensors installed on the ExScal XSM motes from OSU and CrossBow [19]). At present, four motion sensors are integrated on a single ExScal Mote to provide a 360 degree surveillance range (each individual motion sensor can only handle a 90 degree area). We carried a series of experiments on ExScal nodes. More specifications of ExScal nodes can be found at [1].

4.4.1.2 Experimental Results

In principle, PIR sensors can detect anything that generates an infrared field disturbance, such as vehicles, persons, etc. We are interested in the ability of PIR sensors to detect walking persons, as demanded by the application. The central metric is sensing range of PIRs. In individual tests, we let a person walk by at different distances and angles relative to the sensing node, and measure

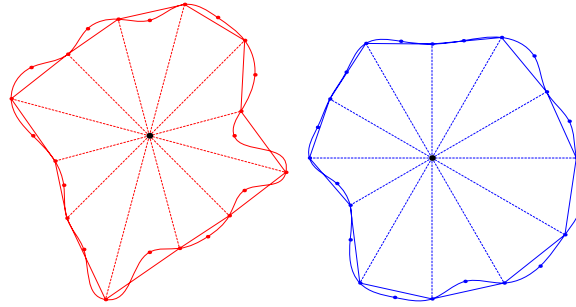


Figure 4.15: Range Approximations

the maximal range of detection. All experiments are repeated in an open parking lot. We program the node with a simple frequency analysis procedure to report events. The procedure relies on an adaptively adjusted threshold to compare the current readings. The main purpose of this adaptive threshold is to avoid false alarms introduced by weather changes, since this change can produce infrared noise that may trigger the sensor. More specifically, our procedure monitors the average readings of the filtered signal within a moving window. If the average reading observed is much smaller than the current threshold, it decreases the threshold by taking a weighted average. On the other hand, if the average energy is close to or larger than the threshold for a certain period, the sensor decides that the weather is noisy, and increases the threshold. Practically we are able to filter out almost all false alarms using this technique. We note that this technique also filters out those slight disturbances that may, in fact, be caused by the target. Thus, the results proposed below present effective sensing range in slightly noisy environments, which are different from precisely controlled environments.

We measured the sensing range from 24 equally divided directions. At each direction, a person moves with different distances from the sensor node. If the fluctuation of the filtered sensor reading exceeds a predefined noise threshold due to the nearby motion, the measured point is within the sensing range. If the motion does not cause a fluctuation greater than the threshold, the measured point is out of the sensing range. In the experiments we found out that the sensitivity of the sensors changes dramatically at the edge of the sensing range. The motion can always be detected in the sensing range and the motion 10 inches beyond the sensing range never triggers the sensors. Therefore, the precision of the range measurement is always within 10 inches (the range itself

being hundreds of inches). The experimental results for two representative nodes are shown in Figure 4.13. Obviously, with four sensors, the range is far from being circular. A more intuitive illustration is shown in Figure 4.14, where the sensing range in each direction is plotted to scale.

Based on the experimental results of multiple tests, we have the following observations regarding the PIR sensing capability. First, the sensing range of one node is not isotropic, that is, the node exhibits different ranges in different directions. Second, the boundary of the sensing range is delineated quite sharply, with predictably no detection when the range is exceeded by about 3-7%. Third, we can consider the variation of ranges relatively continuous. We do observe sudden changes in the sensitivity of some nodes, but this is very uncommon. Therefore, a model may consider connecting sensitivity ranges in different directions using continuous curves. Fourth, the sensing range distribution in different directions roughly conforms to a Normal Distribution. This conclusion is based on statistical analysis of our experimental data using a Kolmogorov-Smirnov Test [52]. We concluded from our experimental data set that the sensing range in one direction can be approximated with a Normal Distribution with an expectation of 217 inches and a standard deviation of 32 inches.

4.4.1.3 Realistic Sensing Model for PIR sensors

In this section, we present a realistic sensing model for PIR nodes. This model is designed to reflect the three key observations: non-isotropic range, continuity and normality. First, the model determines sensing ranges for a set of equally-spaced directions. Each range can either be specified based on actual measurements or obtained from a representative distribution, for example, the normal distribution $N(217, 32^2)$. Next, sensing ranges in all other possible directions are determined based on an interpolation method. For simplicity, we use linear interpolation to specify the boundary of the sensing area. As an example, the approximation of the two representative nodes based on this model are shown in Figure 4.15. Clearly, our simplified model pretty accurately reflects the fluctuations of sensing ranges in different directions.

4.4.1.4 Additional Notes

Despite the fact that we proposed a realistic sensing model based on experimental results for PIR sensor, we have simplified many other aspects of the application. For the detection part, we assumed that an individual node's duty cycle is so large that the sensor warmup time is negligible. We also assumed that the existence of targets within the sensing range can be determined in sufficiently short time. For the target part, we also do not address more complex movements of the target. We acknowledge that introduction of any of these factors will considerably increase the complexity of the analysis. However, our analysis is still innovative and useful in the sense that it reveals quantitative relationships between system parameters and performance. Such understanding will considerably advance our knowledge of the underlying principles of the design and deployment of sensor networks for low power surveillance.

4.4.1.5 Robustness of Theoretical Predictions to Realistic Sensing Model

We now incorporate the realistic sensing range model into simulations to test the robustness of the performance predictions to sensing irregularity. We scaled the normal distribution $N(217, 32^2)$ to $N(10, 1.47^2)$ to fit the simulation setting. We use the same formulas as the previous idealistic experiments. The CDF's of detection delay under different parameter settings are shown in Figure 4.16. To illustrate the difference, we also plot the error relative to theoretical predictions in Figure 4.17.

We have two observations regarding these results. First, for our simulation settings, sensing irregularity has a very small effect when the standard deviation of sensing range is within 20% of the mean value, as we observed for the PIR sensors and applied in the realistic sensing model for simulations. One primary reason is that even though the detection ranges do vary with different directions, the overall degree of coverage for the area remains almost the same, approximated by λ . Therefore, the overall detection delay distribution is almost not affected. Second, we observe that the maximal error relative to theoretical predictions is no more than 1.2%. As an example to show its implications, suppose that we have a set of system parameters that guarantees that 99% of intruding targets are detected within a certain time. Then, the actual detection rate should be no less

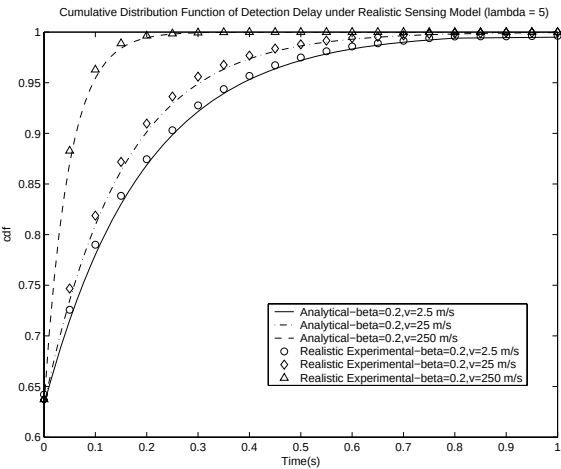


Figure 4.16: Realistic Sensing Effect

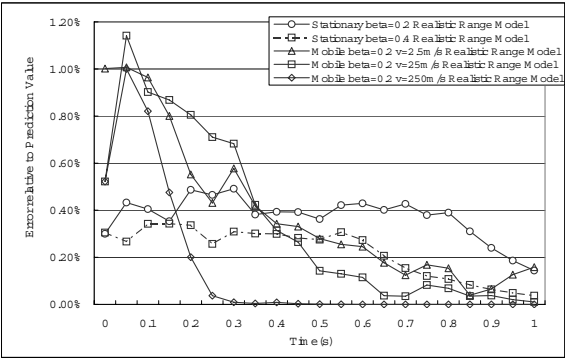


Figure 4.17: Robustness of Theoretical Predictions

than 97% with the existence of irregularity. Based on this observation, we conclude that our model is quite robust to realistic sensing conditions.

At last, we acknowledge that so far, our modeling of the realistic sensing model is only concerned with PIR sensors. Other types of sensors may well exhibit different characteristics, therefore, may have varied effect on the detection performance. However, we envision that with a relatively large system with considerable density, the effect of sensing irregularity will be considerably limited, leading to improvements in the accuracy of the aforementioned results.

4.4.2 Dealing With More Regular Deployments

In reality, it is hard to enforce a perfect uniformly random distribution. More regular deployments are more common in real applications. For example, rectangular grids are often used for simplicity and ease of localization, among other reasons. Some energy saving protocols also make the distribution of nodes that are awake more regular. For example, the sentry service in VigilNet [30] enforces a minimum distance between any pair of sentry nodes. Even if the nodes are deployed randomly at the beginning, after the sentry service is activated and nonsentry nodes are put into sleep, the distribution of sentry nodes that monitor the system area are much more regular.

To study the impact of these factors, we run simulations to compare the expected delays for random deployment from theoretical analysis and those for grid deployment and for sentries.

The simulation settings are as follows. The sensing range is selected as 10 meters. The density used is 0.01 nodes per square meter for grid and random deployments. For the sentry scenario, the density is 0.04 nodes and we use a sentry range of 6.8 meters, which results in an equivalent sentry density of 0.01 nodes per square meter. The duty cycle is selected from 0.1, 0.2, ..., to 0.9, and the period is 1 second. The target speed is chosen as 5 meters per second and 50 meters per second. Other simulation settings are the same as those used in Section 4.3.

Figure 4.18 shows the simulation results. From the figure, we can see that for the same settings and the same target speed, the expected delays for grid deployment and sentry selection protocol are significantly shorter than that for random deployment, which can be derived with our theoretical analysis. The expected delay for grid deployment is slightly shorter than that for the sentry scenario.

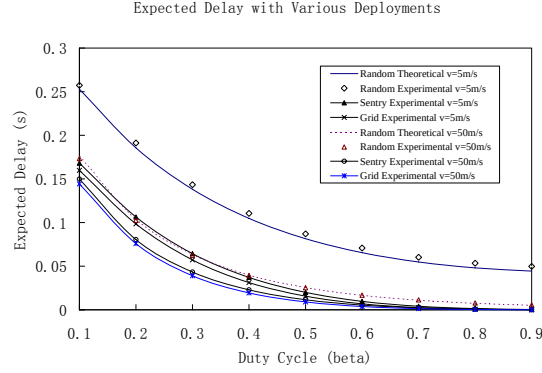


Figure 4.18: Impact of Deployment on Expected Detection Delay

The overall tendency is that with the same density, the more regular the deployment is, the shorter the expected delays are.

Here we give an analytical explanation for this observation. To obtain the expected detection delay for more regular deployments, we can first use the same formula in Equation 4.2, $P = E(1 - (1 - p)^n)$, to get the probability P , which is the probability of the target being detected by at least one sensor node after it moves for a certain distance into the system. Note that p , or the probability of the target being detected by one sensor node is the same as that for random deployment. The only difference is that n , the number of nodes, does not conform to a Poisson distribution in more regular deployments. Note that $1 - (1 - p)^n$ is a strictly concave function of n , therefore $E(1 - (1 - p)^n) \leq 1 - (1 - p)^{E(n)}$, and the left side equals the right side if and only if n is a constant. Given the same $E(n)$, or the node density, the more scattered the distribution of n is, the smaller the value of $E(1 - (1 - p)^n)$ is. Since the random node locations are more scattered than grids or sentries, random deployment has a smaller $E(1 - (1 - p)^n)$ value, or the detection probability after the target moves for a certain distance is smaller than that for grid deployment or the sentry scenario. Therefore, the grid deployment and the sentry scenario result in shorter expected detection delays. Among these two, grid deployment is more regular, so its expected detection delays are the shortest.

From the simulation results and the analytical explanations, we conclude that the analytical results for the expected delay for random deployment can be used as an upper bound for more regular deployment schemes when the density and other parameters are the same. This conclusion

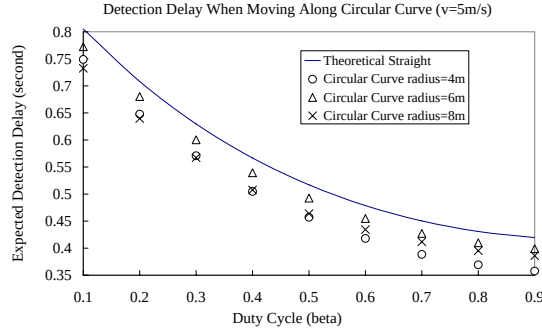


Figure 4.19: Detection Delay for Targets Moving Along Circular Curves with Speed 5m/s

is very important for real applications, since in many cases the deployments are more regular than a uniformly random distribution.

4.4.3 Targets Moving Along Circular Curves

In reality, the targets do not always move along straight lines. Since the number of all possible traces is infinite and irregular traces are hard to model and analyze, we use circular curves with various radii as examples and run simulations to see how closely the expected detection delays from the simulations and the mathematical results based on analysis of straight traces match.

The simulation settings are the same as the previous set of simulations except that the targets move along circular curves with radii of 4 meters, 6 meters and 8 meters, and the density is 0.005 nodes per square meter. The simulation results are shown in Figures 4.19 and 4.20, for target speeds of 5 meters per second and 50 meters per second, respectively.

From the simulation results, we can see that all the detection delays we obtain from the simulations are shorter than the analytical results for straight movements with the same settings, although the differences are not significant, especially for targets with higher speeds. Therefore, the analytical results for targets moving along a straight line can be used as a close upper bounds for targets moving along circular curves.

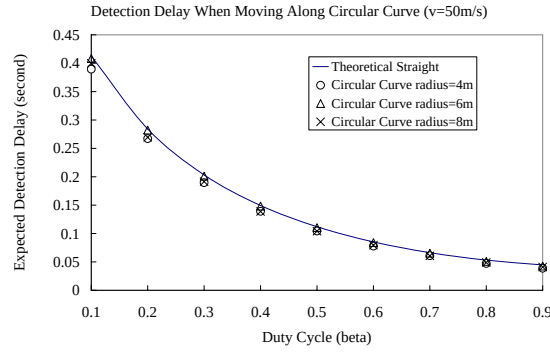


Figure 4.20: Detection Delay for Targets Moving Along Circular Curves with Speed 50m/s

4.5 Conclusions

This chapter is the first to derive closed-form formulas for the distribution and expectation of detection delay for both stationary and mobile targets, and is also the first to propose and evaluate a realistic sensing model for sensor networks, to the best of the authors' knowledge. Extensive simulations are conducted and the results show that they match the analytical derivations very well. Although the basic analysis is based on random deployment, our simulations show that the quantitative results for expected detection delays can be used as an upper bound for more realistic and regular deployments. This result is also explained analytically through mathematical derivations. Simulations also show that the analytical results for targets moving along straight lines can also be used as estimates for the tracking performance of targets moving along curves. The analytical results are highly important for designers of energy efficient sensor networks for monitoring and tracking applications. Designers can apply these formulas to predict the detection performance without costly deployment and testing. Based on these formulas, they can make decisions on key system or protocol parameters, such as the network density and the duty cycle, according to the detection requirements of the system. Therefore, this work is a major contribution towards a thorough understanding of the relationship between system and protocol parameters and achievable detection performance metrics.

Chapter 5

Capacity Analysis for Periodic Streams

This chapter proposes a feasibility and capacity analysis for periodic streams in wireless sensor networks. Section 5.1 introduce the capacity analysis problem. Section 5.2 conducts analysis of a single stream as a basis for more complicated analysis. Section 5.3 extends the results for a single stream to the case of multiple streams. The reality issue of selecting an appropriate interference range is addressed in Section 5.4. Simulation results are presented in Section 5.5 to complement the analysis derivations. Section 5.6 presents implications of the analysis for improve communication performance fore wireless sensor networks. The chapter is concluded in Section 5.7.

5.1 Introducing the Capacity Analysis Problem for Periodic Streams

As hardware evolves, applications that require high performance will become common. For example, an “Ears on the Ground” application monitors the surrounding sounds and transmits them back to base stations. Even more, in the foreseeable future end-users will require WSNs to perform even more demanding tasks such as providing video feedback [47]. Such applications are very challenging to develop because they require high bandwidth utilization. New protocols must be designed to guarantee such levels of utilization.

The network traffic generated by the aforementioned applications may be divided into two categories: aperiodic traffic such as one-time system configuration messages and sporadic event reports; and periodic traffic, such as audio and video information that is generated periodically. When the

bulk of the traffic is periodic, the overall system performance can benefit from a careful resource allocation of periodic traffic. Such a schedule may avoid the unnecessary delays caused by randomness in typical carrier sensing schemes. In contrast, the aperiodic traffic may be better scheduled using CSMA/CA schemes. To achieve good overall performance, we propose a mixed solution where the total bandwidth is divided between the two categories of traffic. To prevent network overload, we identify the amount of bandwidth required for supporting all periodic streams. The remainder of the bandwidth is allocated to the aperiodic traffic, which may use a conventional carrier sensing scheme. This can be easily enforced by dedicating a timeslot for periodic traffic in each period and the remainder to aperiodic traffic. In this chapter, we conduct a capacity analysis on the feasibility of the network to support multi-hop periodic streams with a specific amount of bandwidth reserved, and obtain the relationship between system communication parameters and feasible stream rates. Note that even if the traffic is not strictly periodic, i.e., with fluctuations, the average data rates can be used to conduct the analysis given the nodes have enough memory to buffer the data, because the scheduling scheme we provide with the analysis can regulate the traffic.

For many monitoring applications, the network generates data periodically from one or more locations. The data need to be sent to sinks specified by designers. Designers may have a few locations of interest and want to receive data from them periodically. It is common that reports from sensors in a small neighborhood are aggregated at some predefined locations. Designers may want to receive periodic aggregated data from these locations, and it may take multiple hops for the data to arrive at the sinks. We define a *stream* as a periodic transmission of data from a certain source to a certain sink. Note that this model can also be applied to a common scenario where sources send data towards a single remote congregation point. Each “branch” between a source and the congregation point can be thought as a stream. For streams multi-hop routing is often required because of the limitation of communication range. The delivery feasibility of the streams is affected by a set of factors, including the network topology, the communication model and the interference model. This section discusses the impact of these factors on stream feasibility. This knowledge is of vital importance for system designers to decide on the key system parameters or to adjust the stream data rates in order to deliver the streams successfully to their sinks. The approach also helps

designers understand intrinsic limitations of communication in multi-hop wireless ad hoc networks for periodic streams.

5.1.1 Models and Assumptions

The stream model is given by a set of stream specifications s_i , $i = 0, 1, \dots, M$. A stream s_i has a source location $s_i.src$ and a sink location $s_i.sink$, and the distance between them is $s_i.dist$. The nodes at the source and sink locations are named the source node and sink node, respectively. A stream s_i has a period $s_i.p$. For ease of analysis we use the least common multiple (LCM) for all the periods p as a global period. Another important parameter is the number of bits $s_i.r$ that needs to be sent from the source to the sink for stream s_i for each unit of time. Here we do not exclude communication overhead bits from the raw data. We denote one-hop directed links from the source node to the sink node in stream s_i by $s_i.l(0, 1)$, $s_i.l(1, 2)$, ..., $s_i.l(N_i - 1, N_i)$, assuming it takes N_i hops for the stream to reach the sink from the source. We call the intermediate nodes relays and number them from 1 to $N_i - 1$. We also call the source node as relay 0 for convenience.

We study two types of deployment of wireless ad hoc sensor networks. For the first type, nodes in a sensor network are deployed randomly and uniformly into a two-dimensional system area. For the second type, they are deployed in a grid. We denote the *grid unit size* by g , defined as the length of the edges of grid squares.

A transmitter successfully delivers a message to a receiver if and only if the distance between them is smaller than or equal to communication range R_c and there are no other transmitters in the circular disk of radius R_i (interference range) centered at the receiver. Actually, this condition is sufficient for avoiding the three types of collision [9]: “hidden terminal”, “direct interference” and “self interference”. We also assume that $R_i \geq R_c$. In order to deal with irregularities in R_c [61] and guarantee reliable transmissions we choose a conservative communication range R_c so that if the distance is smaller than or equal to it and there is no interference, the probability that the transmission is successful approaches 1. In order to accommodate the random fluctuation of interference we choose an aggressive interference range R_i [62] so that the effect of a node interfering with another node is negligible if the distance between them is greater than R_i . Figure 5.1 shows the selection of

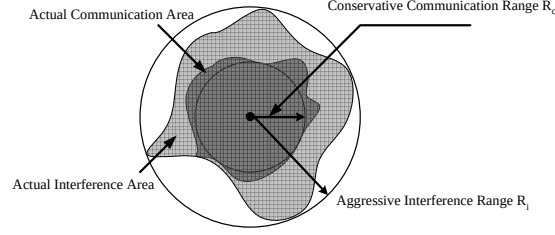


Figure 5.1: Communication and interference ranges

R_i and R_c . It is obvious that the slowest link in a stream decides the overall data rate. Our analysis also shows that a large R_i/R_c value makes the link rates lower, so a conservative communication range and an aggressive interference range can better characterize the effect of the ranges on the achievable rates than the average ranges.

We assume that each communication link in the system has the same reserved bandwidth W . The bandwidth reservation can be achieved in the following way. All nodes toggle between a periodic traffic mode and a sporadic traffic mode periodically and synchronously. (Here we assume time synchronization among nodes.) More specifically, all nodes adopt the same period T , and for βT of the period the whole system is in the periodic traffic mode and the nodes transmit data according to the schedule we obtain in the analysis; for the remaining $(1 - \beta)T$ the whole system is in the sporadic traffic mode and a conventional carrier sensing scheme can be used. Therefore the effective bandwidth W for the periodic traffic mode is just β times the total bandwidth. From now on we only consider and the effective bandwidth and what happens in the periodic traffic mode. We do not consider the issue of lossy links, which can be filtered out by blacklisting [50].

One of the key strengths of our work is that we do not need detailed information of the network topology, or the location of each individual node, which is hard and undesirable to obtain before the network is finally deployed. We do need to know the node density d . However, from the following analysis we see that if the node density is high enough, increasing the node density does not improve the feasible stream rates. We first consider a very high density case for simplicity and to establish the basics of the analysis. Then we extend it to a more realistic lower density case.

The selection of the routing mechanism also affects the analysis result. For simplicity we use a straightforward approach which chooses the node closest to the sink as the next hop. Later in

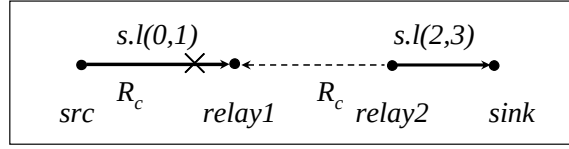
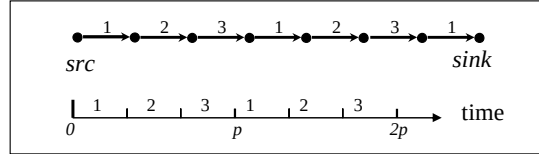
Section 5.6 we discuss the effect of routing schemes on the feasibility and gives an optimistic upper bound for all routing schemes. We do not consider data aggregation along the routes of a stream, so the data rate $s_i.r$ is uniform for all hops.

5.2 Analysis of a Single Stream

In this subsection, we first analyze the scenario where the network has a very high density and there is only one periodic stream s in the system. The high density is assumed to guarantee that forwarding from the source to the sink can occur in a straight line. In other words, the distances from the relay nodes to the straight line connecting the source node and the sink node are negligible, when we always choose the node closest to the sink node in the sender's communication range R_c as the relay node. The distance from relay i to relay $i + 1$, $i = 0, \dots, N - 1$, is very close to R_c because of the greedy routing scheme. Although the very high density assumption is not always realistic, it helps understanding the major factors affecting the achievable rates of the network. Later we consider the complications caused by lower density and inter-stream interference.

5.2.1 A Single Stream with High Density

[17] obtains the following result for the maximum achievable rate of a single stream: “If the link capacity is L , then consecutive levels (or hops in our notation) can share time equally to avoid mutual interference and still ensure that data flows towards the observer at an average rate of $\frac{L}{3}$.” Note that this is true if and only if the number of hops of the stream is greater than or equal to 3. The difference between R_i and R_c is also ignored. For better understanding, we give some explanation of the statement, although this is not our own discovery (see [17]). First, a link is called an *active* link when data is transmitted on it. We consider a stream with three hops as in Figure 5.2. When the source is transmitting, i.e., $s.l(0, 1)$ is active, according to the aforementioned interference model, $s.l(1, 2)$ cannot be active, because relay 1 is receiving data. Similarly $s.l(1, 2)$ and $s.l(2, 3)$ cannot be active simultaneously, either. When $s.l(0, 1)$ is active, $s.l(2, 3)$ cannot be active either, because relay 1 is in the interference range of relay 2. Therefore, at any time only one of the three links can

Figure 5.2: Interference between $s.l(0,1)$ and $s.l(2,3)$ Figure 5.3: Link conveying when $R_i = R_c$

be active, and the maximum overall rate is $W/3$ (or $L/3$ in the notation of [17]). It can be easily proven that the maximum overall rate is W for a single stream with 1 hop and $W/2$ for that with 2 hops. For a single stream with more than 3 hops, Figure 5.3 shows why $L/3$ (or $W/3$ in our notation) is the maximum achievable rate. From the figure we can see that two links $s.l(i, i+1)$ and $s.l(j, j+1)$ can share the same transmission time slot as long as $j-i$ is a multiplier of 3. Therefore a data rate of $W/3$ is achievable. From the above discussion we know that even for three hops the bandwidth cannot support a data rate more than $W/3$. Therefore, $W/3$ is the maximum achievable overall data rate.

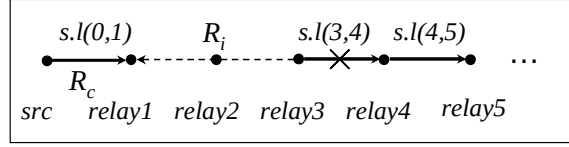
Now we extend the result to the case that $R_i > R_c$.

Theorem 5.1. *The maximum feasible data rate for a single stream s in a very dense wireless sensor network when the interference range $R_i \geq R_c$ is*

$$\begin{cases} \frac{W}{N}, & N < \lfloor \frac{R_i}{R_c} \rfloor + 2 \\ \frac{W}{\lfloor \frac{R_i}{R_c} \rfloor + 2}, & N \geq \lfloor \frac{R_i}{R_c} \rfloor + 2, \end{cases} \quad (5.1)$$

where the hop number $N = \lceil s.dist / R_i \rceil$.

Proof. Let $|i-j|$ be the number of hops between relay i and relay j . According to the high density assumption, the distance between relay i and relay j is $|i-j| \cdot R_c$. Let $k = \lfloor R_i / R_c \rfloor$. Since $R_i \geq kR_c$, the interference range of relay i covers relays $i-k, \dots, i-1$, if they exist. Therefore, relays

Figure 5.4: Interference when $R_i = 2R_c$

$i - k - 1, \dots, i - 2$ cannot transmit at the same time when relay i is transmitting data. Since relay i is transmitting data, it cannot receive data at the same time according to our interference model. Relay $i - l$ can transmit data at the same time with relay i without interfering with each other, if $l > k + 1$. Thus the sufficient and necessary condition that relay i and relay j can transmit data simultaneously is $|i - j| > k + 1$. For the case $N \geq k + 2$, from the above discussion we know that any two relays from relay 0 to relay $k + 1$ cannot successfully transmit data simultaneously, so the bandwidth has to be divided by the transmissions from these relays. Without consideration of other relays, the maximum overall data rate is $W/(k + 2)$ because to maximize it we need to balance the data rates from the relays. We categorize the relays into $k + 2$ groups, and relay i and j belong to the same group if and only if $i \equiv j \pmod{k + 1}$. Any two relays in the same group may transmit data simultaneously, because the two transmissions do not interfere with each other. With the “convoying” described here we conclude that the maximum feasible overall data rate is $W/(k + 2)$. In the case $N < k + 2$, it is easy to prove the maximum feasible overall data rate is W/N . ■

We also give an intuitive explanation for the case $R_i = 2R_c$. From Figure 5.4 we can see that $s.l(0,1)$ and $s.l(3,4)$ cannot be active at the same time due to the increased interference range. Actually, any two links $s.l(i, i + 1)$ and $s.l(j, j + 1)$ can be active at the same time if and only if $|i - j| \geq 4$. We can arrange the links so that $s.l(a, a + 1)$ and $s.l(b, b + 1)$ are active at the same time if and only if $a \equiv b \pmod{4}$. Therefore, for more than three hops, $W/4$ is actually achievable. We call this arrangement of links *link convoying*. For example, in this case $s.l(0,1)$, $s.l(4,5)$ and so on may take the same timeslots in a period p since they do not interfere with each other, therefore “convoyed”. It is worth noticing that this theorem provides a sufficient and necessary condition for the overall achievable rate.

5.2.2 A Single Stream with Lower Density

The high density assumption used above is not realistic, but the above analysis serves as the basis for the following extension. In many realistic cases densities are lower, so the relay nodes may deviate from the straight line connecting the source and the sink, and the one-hop distances are shorter than R_c .

First we consider the simple case where the interference range R_i is the same as the communication range R_c . In this case, the maximum feasible rate of $W/3$ obtained in [17] still holds if a next hop closer to the sink is always available. The reason for this constraint is that when proving Theorem 5.1 we need to guarantee that if $|i - j| > k + 1$, $s.l(i, i + 1)$ and $s.l(j, j + 1)$ do not interfere with each other. Here $k = \lfloor R_i/R_c \rfloor = 1$. Assume that two links $s.l(i, i + 1)$ and $s.l(j, j + 1)$ interfere with each other and $j = i + 3$. Then the distance between relay $i + 1$ and relay j must be smaller than or equal to the interference range R_i , which is exactly the same as the communication range R_c . So relay j can be reached in one hop from relay $i + 1$ and according to the routing scheme relay j is closer to the sink than relay $i + 2$ is, which contradicts the routing scheme. Therefore, the assumption does not hold and if $|i - j| > 2$, $s.l(i, i + 1)$ and $s.l(j, j + 1)$ do not interfere with each other. Figure 5.5 shows this situation.

The following theorem shows the relationship between the network density and the probability that a relay closer to the sink exists. Note that it is fundamental to algorithms based on geographic forwarding.

Theorem 5.2. *The probability that a relay closer to a sink exists for a wireless sensor network with a randomly uniform deployment is in the interval*

$$(1 - e^{-(\frac{2}{3} - \frac{\sqrt{3}}{2\pi})D}, 1 - e^{-\frac{D}{2}}), \quad (5.2)$$

where D is the average node number in πR_c^2 .

Proof. For a wireless sensor network with many nodes, the distribution of the number of nodes in a certain area can be approximated as a Poisson distribution with the parameter $\lambda = Ad$, where A is the area and d is the density. Denote the node that needs to transmit or relay data as the sender.

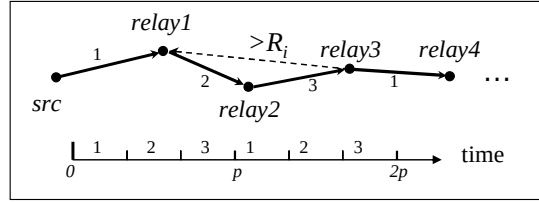


Figure 5.5: Convoing in a low density network

A node qualifies for a next hop if and only if it is both in the communication range of the sender and closer to the sink node than the sender. So a qualified relay exists if and only if that there exists a node in the intersection area of two circular disks. The first is one with radius R_c and centered at the sender. The second disk is centered at the sink and its radius is the distance between the sink and the sender. Note that the distance between the sink and the sender must be greater than R_c , otherwise there is no need for a relay node. The intersection area is monotonically increasing when the distance between the sink and the sender increases. When the distance approaches R_c , the area approaches $2 \cdot (\sqrt{3}R_c^2/4) + 4 \cdot (\pi R_c^2/6 - \sqrt{3}R_c^2/4)$, which is $(\frac{2}{3}\pi - \frac{\sqrt{3}}{2})R_c^2$. When the distance approaches ∞ , the area approaches $\frac{\pi}{2}R_c^2$. According to the probability mass function of the Poisson distribution, the probability that there is no node in area A is $e^{-\lambda} = e^{-dA}$. Substituting the upper and lower bounds of the intersection area, we get the interval shown in the theorem. ■

From (5.2), the interval is $(1 - e^{-1.23R_c^2d}, 1 - e^{-1.57R_c^2d})$ approximately. In order to get a 99% probability that a valid next hop exists, the node density should be at least $3.7/R_c^2$, because $1 - e^{-1.57 \cdot 3.7} > 1 - e^{-1.23 \cdot 3.7} \approx 0.99$.

The case $R_i > R_c$ with lower density is more complicated than that with the high density assumption. We need to modify the results from Theorem 5.1. The reason is that the distance from relay m to relay n is smaller than $|m - n|R_c$. For a certain stream s , denote the minimum distance between any relay pair n and $n + k$, $n \geq 0$ and $n + k \leq N$, by $H(s, k)$. Then we have a modified version of Theorem 5.1 for the lower density case:

Theorem 5.3. *The maximum feasible data rate for a single stream s in a wireless sensor network*

when the interference range R_i is greater than or equal to the communication range R_c is

$$\begin{cases} \frac{W}{N}, & N < K(s) + 1 \\ \frac{W}{K(s)+1}, & N \geq K(s) + 1, \end{cases} \quad (5.3)$$

in which $K(s)$ is defined as the minimum integer that satisfies $H(s, K(s)) > R_i$.

Proof. The proof is similar to the proof of Theorem 5.1. First we consider the case where $N \geq K(s) + 1$. Consider relay i and relay j . Without loss of generality, assume that $j > i$. Relay i and relay j can send data at the same time if $j - i > K(s)$ since the distance between relay j and relay i 's receiver relay $i + 1$ is greater than or equal to $H(s, K(s))$, which is greater than R_i . On the contrary, for any $k \leq K(s)$, according to the definition of $K(s)$ and $H(s, K(s))$, there must be a relay i in the stream so that the distance between its receiver relay $i + 1$ and relay $i + k$ is smaller than or equal to R_i , so relay i and relay $i + k$ cannot be active at the same time. Therefore the maximum achievable rate cannot exceed $W / [K(s) + 1]$. We group the relays into $K(s) + 1$ groups. Relay i and relay j are in the same group if and only if $i \equiv j \pmod{K(s) + 1}$. Then with similar arguments we used in the proof of Theorem 5.1 we can prove this theorem when $N \geq K(s) + 1$. The case where $N < K(s) + 1$ is easy to prove. ■

Note that this theorem provides a sufficient and necessary condition for the feasibility of one stream. Theorem 5.1 is a specific case of this theorem. It is impossible to get an exact hop number for the low density case in a randomly distributed network, but its distribution can be found according to simulation. The following discusses how to apply this theorem to sensor networks with a grid deployment or a random deployment.

(a) The Case of Grid Deployment

For a WSN with a square grid deployment we are able to get a guaranteed lower bound for $H(s, k)$, if we slightly modify the greedy routing scheme in the following way. We define the straight line segment connecting a source location and its sink location as a *source-sink line*. Here we describe the way we choose the relay for a node, if it cannot reach the sink node in one hop. First we choose all the edges (with length g) that intersect or overlap the stream's source-sink line.

For each intersecting edge, choose the vertex closer to the intersection point. For each overlapping edge, choose both vertices. Among them eliminate all those out of the communication range of the sender. Among all the remaining nodes we choose as the relay the node whose projection point on the source-sink line is the closest to the sink location. It can be easily seen that the distance from each relay to the source-sink line is smaller than or equal to $g/2$. Therefore, with this routing scheme we can focus on the area A_r in which any point's distance to the source-sink line is smaller than or equal to $\sqrt{2}g/2$.

Theorem 5.4. *For a wireless sensor network with grid deployment with grid unit size g , if we use the routing scheme described above, the distance between relay m and relay n is at least*

$$|m - n|(\sqrt{R_c^2 - 2g^2} - \sqrt{2}g), \quad (5.4)$$

assuming $R_c > \sqrt{5}g$.

Proof. If the source-sink line intersects one edge of the square grid directly after another edge, these two edges must be two sides of a rectangle with four sides of equal length g . If we always choose the vertex with a closer distance to the intersection point, the maximum distance between two consecutive vertices must be smaller than or equal to $\sqrt{2}g$. If the source-sink line overlaps edges, the distance between any two consecutive vertices is g .

A valid relay must be in the intersection area of two circular disks. The first is centered at the sender that needs a relay, with radius R_c . The second is centered at the sink location, and the radius is equal to the distance between the sink location and the sender that needs a relay. The radius should be greater than $R_c - \sqrt{2}g/2$, considering that the maximum distance from the sink location and the sink node is $\sqrt{2}g/2$. Name the projection point of the sender on the source-sink line as P_s . It can be easily seen that each node in A_r is in the intersection area of the two circular disks, if its projection point is closer to the sink location than P_s , and the distance from its projection point to P_s is among the interval between $R_c - \sqrt{2}g/2 - \sqrt{(R_c - \sqrt{2}g/2)^2 - g^2/2}$ and $\sqrt{R_c^2 - (\sqrt{2}g)^2}$. The length of the interval is a monotonically increasing function of R_c , and when $R_c = \sqrt{5}g$, the interval is longer than $\sqrt{2}g$. So when $R_c \geq \sqrt{5}g$, the interval is longer than $\sqrt{2}g$. From the first paragraph

of the proof, there must be a valid relay node whose projection point is in this interval. And the distance from this node's projection point to P_s must be greater than $\sqrt{R_c^2 - 2g^2} - \sqrt{2}g$. This can be easily extended to the distance between projection points from any relay pair m and n : it must be greater than or equal to $|m - n|(\sqrt{R_c^2 - 2g^2} - \sqrt{2}g)$. The distance between the two relays must be at least the distance between the projection points of them. ■

From Theorem 5.3 and Theorem 5.4, we can obtain the lower bound of the achievable overall data rate for a single stream in a wireless sensor network with grid deployment:

COROLLARY 1. *In a wireless sensor network with grid deployment, when the interference range R_i is greater than or equal to the communication range R_c , the maximum feasible overall data rate for a single stream s is at least*

$$\begin{cases} \frac{W}{N}, & N < K_g(s) + 1 \\ \frac{W}{K_g(s) + 1}, & N \geq K_g(s) + 1, \end{cases} \quad (5.5)$$

in which

$$K_g(s) = \lfloor \frac{R_i}{\sqrt{R_c^2 - 2g^2} - \sqrt{2}g} \rfloor + 1, \quad (5.6)$$

if $R_c > \sqrt{5}g$.

Note that due to the possible underestimate of $H(s, k)$ in Theorem 5.4, the result is pessimistic.

(b) The Case of Random Deployment

Due to the randomness of the deployment, only a statistically estimated lower bound of $H(s, k)$ can be provided. We define a function $d_m(d, k, pr)$, as the $1 - pr$ percentile of the distance between any relay pair n and $n + k$ when the node density is d , and pr is a probability chosen close to 1, such as 90% and 95%. That is to say, when the node density is d , the probability that $H(s, k) \geq d_m(d, k, pr)$ is pr . The values of the function with a different d, k, pr tuple can be obtained from simulation. We may use the simulated $d_m(d, k, pr)$ value and Theorem 5.3 to obtain a probabilistic achievable overall data rate for a single stream in a randomly deployed WSN. $d_m(d, k, pr)$ with a very high pr value (e.g., 99%) can give a rather safe, but pessimistic achievable overall data rate.

5.3 Analysis of Multiple Streams

In this subsection we study the data rate bound when there are two or more streams in the network. The key difference is that the interference among different streams is taken into account. Similar to the above, we start from the simplified case where the node density of the network is very high, then we extend the results to a more realistic case where the density is lower.

In order to consider the spatial relationship among streams, we have the following definitions. If any link in one stream potentially interferes with or is interfered by a link in another stream, we say the two streams *potentially interfere* with each other. Define the *potential interference stream set* of stream s as the set of all streams that potentially interfere with stream s and s itself.

5.3.1 Multiple Streams with High Density

First we have the following theorem:

Theorem 5.5. *For a very dense wireless sensor network, the specified data rates are achievable if for any stream s ,*

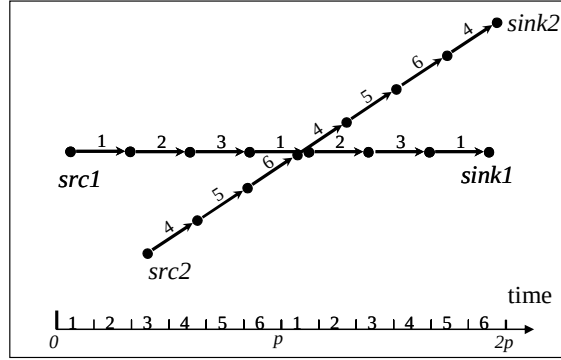
$$\sum_{s_i \in I_H(s)} s_i \cdot r \cdot s_i \cdot k \leq W, \quad (5.7)$$

where $I_H(s)$ denotes the potential interference stream set of stream s , and

$$s_i \cdot k = \begin{cases} s_i \cdot N, & s_i \cdot N < \lfloor \frac{R_i}{R_c} \rfloor + 2 \\ \lfloor \frac{R_i}{R_c} \rfloor + 2, & s_i \cdot N \geq \lfloor \frac{R_i}{R_c} \rfloor + 2, \end{cases} \quad (5.8)$$

. where $s_i \cdot N = \lceil s_i \cdot \text{dist} / R_i \rceil$.

Proof. Prove by contradiction. Assume stream s cannot be scheduled, the reason can only be that other streams in $I_H(s)$ use so much time in p that the remaining empty timeslots are not enough for stream s . In one period p , stream s_i uses time $p \cdot s_i \cdot k \cdot s_i \cdot r / W$. Therefore $\sum_{s_i \in I_H(s)} p \cdot s_i \cdot k \cdot s_i \cdot r / W > p$. This is not true according to the inequality in the theorem. So the assumption is wrong and all streams in the system can be scheduled. ■

Figure 5.6: Two streams with data rates $W/6$

We also give a scheduling scheme to show the feasibility once the condition in the theorem is satisfied. The streams are put into the system one by one. Stream s tries to reserve timeslots of duration $p \cdot s.r \cdot s.k / W$ for itself in period p . Each stream reserves timeslots as close to the beginning of the period p as possible, as long as no timeslots for two streams overlap with each other if they potentially interfere with each other. Since $\sum_{s_i \in I_H(s)} s_i \cdot r \cdot s_i \cdot k \leq W$, then $\sum_{s_i \in I_H(s)} p \cdot s.r \cdot s.k / W \leq p$. This means that this reservation is always feasible. For stream s , the timeslot for each link can be arranged based on Theorem 5.1 within $p \cdot s.r \cdot s.k / W$. Note that the same technique called “link conveying” as defined in Theorem 5.1 is applied here. Figure 5.6 shows how two streams are scheduled when both have a data rate of $W/6$.

The criterion is sufficient, but not necessary. Here is an example that the above criterion gives a decision of non-feasibility, but in reality the stream set is schedulable. Suppose we have three streams s_1 , s_2 and s_3 , and each stream has a data rate of $W/6$. Assume that $R_i = R_c$ and each source takes more than three hops to reach the destination. Each stream interferes with both other streams. However, the three intersection points are far away from each other ($\gg R_i$). With the above procedure, after we put s_1 and s_2 into the system, we are not able to add s_3 into the system, because we cannot allocate enough time slots for the sole use of s_3 . While in reality, s_3 may arrange the schedules of its links to make them schedulable, but it cannot maintain the “link conveying” and the bandwidth the stream as a whole uses is more than three times the data rate. Figure 5.7 shows the necessity of “unconveying”. Although the unconveyed streams as a whole take more timeslots than conveyed streams, the former can make better use of the spatial relationship among streams

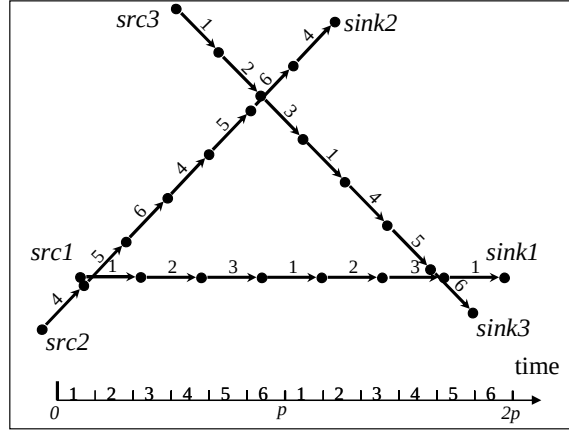


Figure 5.7: When unconvoying is necessary

when the convoying approach fails to give a schedulable result. Considering this, we propose a less pessimistic analysis approach as shown in Algorithm 1.

The streams need to be sorted in descending order of their rates. Then the algorithm tentatively schedules the streams one by one with the convoyed approach as described in the paragraph after Theorem 5.5. If a stream can be schedule this way, it is scheduled and added into U_c , the convoyed stream set; otherwise, it is added into U_u , the unconvoyed stream set without being scheduled. After that, the links in U_u are added into the system on a per-link basis. Each link takes its timeslots as early in the period p as possible, as long as they do not overlap with any link it may interfere or be interfered by. If any of them fails to obtain their timeslots, that is, the total duration of the free timeslots is smaller than the links requirement ($p \cdot s_i \cdot r / W$), the stream set is not feasible. If all of them succeed in obtain their timeslots, the stream set is feasible.

5.3.2 Multiple Streams with Lower Density

The difficulty for analysis of multiple streams in wireless sensor networks of lower densities lies not only in the fact that the k hop distance between relays is smaller than kR_c , but also in the possibility that the route of a stream may deviate from the source-sink line significantly. This makes it hard to tell if two streams interfere with each other or not without complete knowledge of the network topology, therefore complicating the analysis.

Algorithm 1 Multiple Stream Analysis with High Density**Input:** $U = \{s_1, s_2, \dots, s_m\}$, in which $s_1.r \geq s_2.r \geq \dots \geq s_m.r$ **Output:** TRUE/FALSE, feasibility $U_u = \emptyset; U_c = \emptyset; L_u = \emptyset;$ **for** $i=1$ to m **do** **if** $\text{try_add_convoyed}(s_i, U_c)$ **then** $\text{schedule_convoyed_links}(s_i, U_c);$ $U_c+ = s_i;$ **else** $U_u+ = s_i;$ **end if****end for****for each** $s_i \in U_u$ **do** **for each link** $s_i.l(j, j+1)$ **in** s_i **do** **if** $\text{add_unconvoyed_link}(s_i.l(j, j+1), U_c, L_u)$ **then** $L_u+ = s_i.l(j, j+1);$ **else**

return FALSE;

end if **end for****end for**

return TRUE;

For networks with grid deployments, we may apply the modified routing scheme used in Theorem 5.4, so that the distances from all the relays to the source-sink line are definitely smaller than $\sqrt{2}g/2$.

For networks with randomly uniform deployments, we may apply a similar approach which only chooses the relays among the nodes whose distances to the source-sink line are smaller than R_c . Simulations show that this modification does not affect routing performance significantly. With this approach we are able to guarantee that all relays are limited to the narrow area around the source-sink line, which we call the *possible relay region* of the stream. Expanding the area by R_i , we get the *possible interference region* of the stream. They are shown in Figure 5.8. We define the two regions for grid deployment similarly.

From the above discussion we know that the relay nodes of a stream may appear anywhere in the possible relay region of the stream, and any node in the possible interference region of the stream may be interfered by the stream. So we define that two streams *potentially interfere* with each other

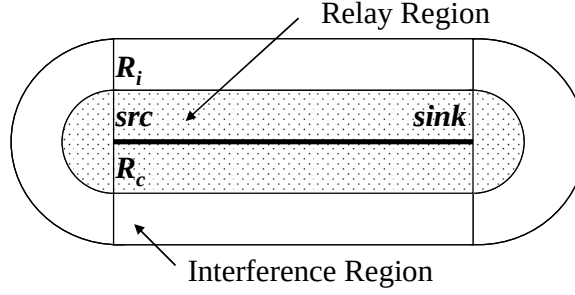


Figure 5.8: Relay region and interference region

when there are common areas for one stream's possible interference region and another stream's possible relay region. Similarly we define a *potential interference stream set* $I_L(s)$ of stream s based on it. Note that it is different from $I_H(s)$ in the high density case due to the uncertainty of relay locations of networks with a low density.

With these modifications we get a pessimistic feasibility condition of multiple streams for lower density networks similar to Theorem 5.5:

Theorem 5.6. *For a sensor network, the specified data rates are achievable if for any stream s ,*

$$\sum_{s_i \in I_L(s)} s_i \cdot r \cdot s_i \cdot k \leq W, \quad (5.9)$$

where

$$s_i \cdot k = \begin{cases} s_i \cdot N, & s_i \cdot N < K(s_i) + 1 \\ K(s_i) + 1, & s_i \cdot N \geq K(s_i) + 1, \end{cases} \quad (5.10)$$

where the definition of function $K(\cdot)$ is the same as in Theorem 5.3.

Proof. Prove by contradiction. Assume that stream s cannot be scheduled. The reason can only be that streams in $I_L(s)$ use so much time in p that the remaining empty slots are not enough for stream s . Divide the period p into time slots $p \cdot s_i \cdot r \cdot s_i \cdot k / W$, which is doable according to the inequality in the theorem. Give stream s_i , $s_i \in I_L(s)$, a time slot of $p \cdot s_i \cdot r \cdot s_i \cdot k / W$. Because of the separation of the time slots, there is no inter-stream interference. According to Theorem 5.3, time slot $p \cdot s_i \cdot r \cdot s_i \cdot k$ is enough for scheduling stream s_i for each $s_i \in I(s)$ without intra-stream interference. So the

assumption is wrong and all streams in the system can be scheduled. ■

The bound or probabilistic distribution of $K(s_i)$ for grid or random distribution can be obtained in the same way as in Subsection 5.2.2. The scheduling scheme is similar as that described in the paragraph after Theorem 5.5. Note that this is a sufficient, but not necessary condition. The pessimism comes from both link conveying and the pessimistic selection of the relay and interference regions.

We propose a less pessimistic feasibility test for the low density case similar as that for the high density case. Since the individual links cannot be located without the complete topology information, we consider all possible links in the system. First we cover the relay area of stream s with a virtual grid, in which the grid unit size should be at least at the same order of magnitude as the precision of R_i and R_c . Then each pair of grid points that are within the distance of R_c is considered the locations of a possible link $s.pl$. The test is shown in Algorithm 2.

The first step is trying to put streams into the convoyed stream set U_c according to a similar scheduling scheme after Theorem 5.5. Streams that cannot be put into U_c are put into U_u . In the next step each possible link is tested one by one. For $s_i.pl$, if it potentially interfere with or is interfered by links in another stream s_j in U_c , the maximum time that links in s_j occupy in period p is $p \cdot s_j.k \cdot s_j.r / W$; if $s_j \in U_u$, this value is $p \cdot s_j.r \cdot \max(s_j.N, 2s_j.k) / W$ due to the unconvoyed feature of s_j (detailed proof omitted). Then the effect of other links in stream s_i itself that may interfere with or be interfered by $s_i.pl$ is considered. The maximum time in period p these links (including $s_i.pl$) take is $p \cdot s_i.k \cdot s_i.r / W$ for $s_i \in U_c$, or $p \cdot s_i.r \cdot \max(s_i.N, 2s_i.k) / W$ for $s_i \in U_u$. If the sum of these items is greater than period p , the stream set is not schedulable. If for any possible link the sum is smaller than or equal to p , the stream set is schedulable. Note that it is also a sufficient condition.

Since the analysis is pessimistic partially due to the selection of relay and interference ranges, we can get better results after we have the complete knowledge of the network topology and know the locations of relays. Then we are able to decide if any of two links interfere with each other according to their exact sender and receiver locations. Based on this information we devise a pure link based scheduling scheme. The streams are first sorted in descending order of rates and then

Algorithm 2 Multiple Stream Analysis with Lower Density

Input: $U = \{s_1, s_2, \dots, s_m\}$, in which $s_1.r \geq s_2.r \geq \dots \geq s_m.r$ **Output:** TRUE/FALSE, feasibility

```

 $U_u = \emptyset; U_c = \emptyset;$ 
for  $i=1$  to  $m$  do
  if  $\text{try\_add\_convoyed\_low}(s_i, U_c)$  then
     $U_c += s_i;$ 
  else
     $U_u += s_i;$ 
  end if
end for
for  $i=1$  to  $m$  do
  for each possible link  $s_i.pl$  in  $s_i$  do
     $t_{occupied} = 0;$ 
    for  $j=1$  to  $m$  do
      if  $\text{interferes}(s_i.pl, s_j)$  then
        if  $s_j \in U_c$  then
           $t_{occupied} += p \cdot s_j.r \cdot s_j.k / W;$ 
        else
           $t_{occupied} += p \cdot s_j.r \cdot \max(s_j.N, 2s_j.k) / W;$ 
        end if
      end if
    end for
    if  $t_{occupied} > p$  then
      return FALSE;
    end if
  end for
end for
return TRUE;

```

scheduled according to this order. For each stream, the links are scheduled according to the order from the source to the sink. When link $s_i.l(m, m+1)$ is scheduled, it tries to take time slots with a total duration of $p \cdot s_i.r/W$ as early as possible in period p as long as it does not overlap with the time slots of any link that has already been scheduled and may interfere with it or be interfered by it. If any link cannot find enough time slots for its share in period p , the scheduling fails. Otherwise it succeeds. In Section 5.5 we implement the algorithm in a simulator and study how pessimistic our analysis is.

5.4 Dealing with Reality

The use of an “interference range” assumes that the impact of the transmission of nodes outside the interference range of a node on its reception is negligible. In reality this assumption does not always hold. The issue can be seen more clearly according to the “physical model” from [27]: a receiver j can successfully receive packets from sender i if and only if

$$\frac{\frac{P_i}{|X_i - X_j|^\alpha}}{N + \sum_{k \in \Gamma, k \neq i} \frac{P_k}{|X_k - X_j|^\alpha}} \geq \beta,$$

in which P_i and P_k are the transmission powers of nodes i and k , correspondingly, X_i , X_j and X_k are the locations of nodes i , j and k , correspondingly, N is the ambient noise, and Γ is the set of nodes that are transmitting at the same time as node i does, α and β are parameters obtained from experiments. Basically, the left hand of the inequality is the Signal to Interference plus Noise Ratio (SINR), and β is an SINR threshold of successful transmission. This model better captures the realistic characteristics of radio propagation and reception than the protocol model. From this formula, we can see that even if all nodes in Γ except node i are outside a predefined interference range of node j , as long as there are enough nodes transmitting, the condition may not hold any more, and the assumption behind the concept of interference range is not applicable either.

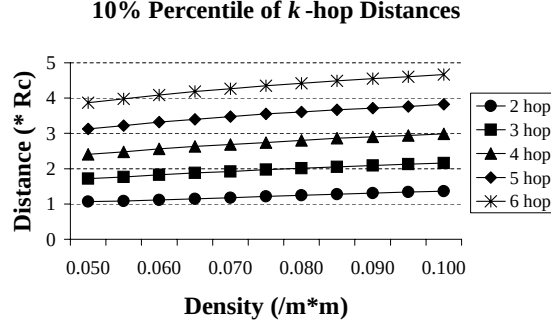
Although this scenario is rare in reality, a realistic analytical tool must take this possibility into account. We can apply a simple approach to check if the condition in the “physical model” still holds if we select a certain interference range. The rational behind the check is that if an interference

range is valid, for each receiving nodes at each time instance, the total interference created by all nodes that are transmitting in the resulting schedule outside the selected interference range is small enough, so that the condition in the physical model still holds.

The check for the analysis of multiple streams in a low density network works as follows. Before the check can be applied, the values of P_i , P_j , P_k , N , α , β need to be obtained, either from site survey and experiments, or from the configuration profile. $|X_i - X_j|$ uses R_c as a pessimistic estimate. We need to test each point as a potential receiver in the virtual grid of each stream's relay region as proposed in the last section. First the intra-stream interference is considered. The maximum number of nodes that are transmitting at the same time in the same stream as the transmitter that is sending data to the receiver in stream i is $\lceil s_i \cdot N / s_i \cdot k \rceil - 1$ ($s_i \cdot k$ is defined in Theorem 5.3), and the lower bound of the distance from an interfering node to the receiver is the selected R_i . Then the inter-stream interference is considered. For any other stream j , the maximum number of nodes that are transmitting at the same time in stream j is $\lceil s_j \cdot N / s_j \cdot k \rceil - 1$, and the lower bound of the distance from an interfering node is the interference range, or the shortest distance from the receiver to the relay region of stream j , whichever is larger. After that, we apply the numbers and distance bounds into the left hand of the inequality in the condition of the physical model and see if it is smaller than β . Note that this calculation is pessimistic. If the inequality is valid for all points in all streams, the selection of interference range is valid. Otherwise, it is possible that the transmissions outside the selected interference range may generate enough interference so that the schedule generated from the analysis does not work. In this case, a larger interference range needs to be selected. This check may be iterated for a few times until an appropriate interference range is selected.

5.5 Simulations

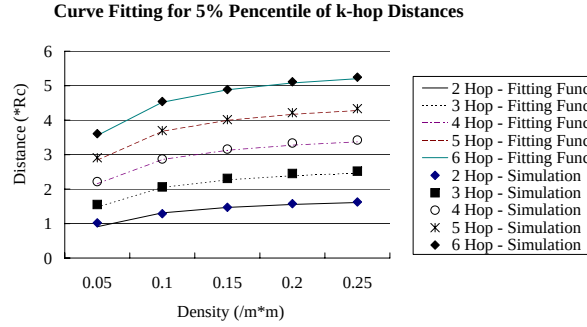
The simulation studies presented in this section complement the theoretical analysis on two aspects. First we show the impact of low densities on achievable rates and how to use simulations to estimate $d_m(d, k, pr)$, which plays a key role in the analysis for randomly deployed networks with low densities. Then we show through simulations how the network density and the distance among streams

Figure 5.9: 10% Percentile of k -hop distances

or stream intersection points affect the feasibility of streams when the traffic load is high. The simulation results on achievable rates are calculated using the per-link scheduling scheme described in the last section. We use the simulation results to validate our theoretical bounds.

Here are our simulation settings. The communication range R_c we use is $10m$. We deploy the nodes in an $120m \times 120m$ rectangular area according to a uniform distribution. We specify the Cartesian coordinates of the lower left corner and the upper right corner as $(0,0)$ and $(120,120)$, correspondingly.

In subsection 5.2 we define $d_m(d, k, pr)$, as the $1 - pr$ percentile of the k -hop distances when the node density is d . Values for $d_m(d, k, pr)$ are needed when Theorem 5.3 and Theorem 5.6 are applied to randomly-deployed networks. Here we run simulations to get $d_m(d, k, 90\%)$. We specify a stream with its source located at $(20, 20)$ and its sink located at $(100, 100)$, and randomly deploy the nodes according to various densities d from $0.05/m^2$ to $0.10/m^2$. For each simulation round we run the simulation for 100 times. Each time we use a different random deployment of the nodes with the same density and find the minimum k -hop distance. From the 100 distances we get the 10% percentile. For each density we run 70 rounds of simulation and calculate the mean values and confidence intervals of the 10% percentiles. The simulation show that all 95% confidence intervals are smaller than 5% of the mean values. The results are shown in Figure 5.9 and the mean values showed here can be used as estimates for $d_m(d, k, 90\%)$. The simulations shows that the 10% percentile of the minimum k -hop distances increases very slowly when the density increases, and the values of the percentiles are almost proportional to the numbers of hops with the same density.

Figure 5.10: Curve Fitting for 5% Percentile of k -hop distances

$d_m(d, 2, 95\%)$	$-2/dR_c^2 - 1.5432/(R_c\sqrt{d}) + 2$
$d_m(d, 3, 95\%)$	$-3/dR_c^2 - 2.0614/(R_c\sqrt{d}) + 3$
$d_m(d, 4, 95\%)$	$-4/dR_c^2 - 2.3405/(R_c\sqrt{d}) + 4$
$d_m(d, 5, 95\%)$	$-5/dR_c^2 - 2.5979/(R_c\sqrt{d}) + 5$
$d_m(d, 6, 95\%)$	$-6/dR_c^2 - 1.7818/(R_c\sqrt{d}) + 6$

Figure 5.11: Least-Square Fitting Functions for $d_m(d, k, 95\%)$

In addition to estimation of the value for $d_m(d, k, pr)$ through simulation, we also apply curve fitting approaches to get the formulas for $d_m(d, k, pr)$ in close forms. We run simulations with the same setting as the aforementioned set, expect that this time we collect data for $d_m(d, k, 95\%)$ for density from $0.05/m^2$, $0.06/m^2$ to $0.25/m^2$ with a step of $0.01/m^2$, in which k is selected from 2 to 6. We then apply the “polytool” utility in MATLAB to find the most appropriate least-square fitting function for $d_m(d, k, 95\%)$ with varying densities. It turns out that the best fitting function for $d_m(d, k, 95\%)$ is a quadratic function of $1/R_c\sqrt{d}$ in the form of $-k/dR_c^2 + a(k)/(R_c\sqrt{d}) + k$, in which $a(k)$ needs to be determined by a first-order least-square fitting function. Figure 5.10 shows that the fitting functions match the simulation data very well. The $d_m(d, k, 95\%)$ fitting functions are listed in Figure 5.11. $d_m(d, k, pr)$ for other pr and k values can be obtained in a similar manner.

We then validate the theoretical analysis based on the $d_m(d, k, pr)$ values we obtain from the

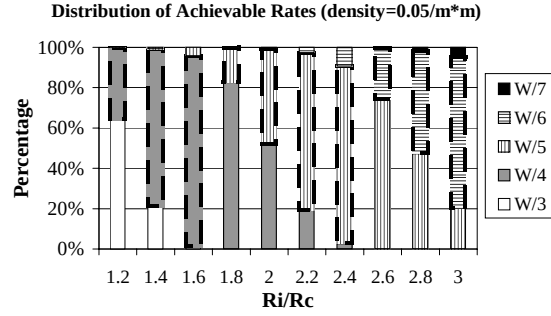


Figure 5.12: Distribution of achievable rates

above simulations and evaluate the impact of various interference ranges on achievable rates. The stream location is the same as specified in the last paragraph. We use various R_i values from $12m$ to $30m$ in the simulations. The density is chosen as $0.05/m^2$. For each value of R_i , we run 126 rounds of simulations. In each round, we obtain maximum achievable rates for 100 different random deployments. Then we obtain the percentage of each achievable rate for each round. Then for the 126 rounds we calculate the mean values and confidence intervals of the percentages. The 95% confidence intervals are all smaller than 2%. Figure 5.12 shows the distribution of the achievable rates in our simulation results. The boxes with thick dashed boundaries show the theoretical achievable rates with 90% probability we get using Theorem 5.3 with $d_m(0.05, k, 90\%)$, obtained from the results shown in Figure 5.9. This verifies that by using $d_m(d, k, 90\%)$ obtained from Figure 5.9, we can compute the maximum data rates that are achievable with a probability of at least 90%. The simulation results closely match our theoretical achievable rates with 90% probability for all tested interference ranges. This result shows that our theoretical bounds can be used to predict achievable rates without accurate knowledge about the network topology. From the simulation results we see the tendency that when the R_i/R_c ratio increases from 1.2 to 3, the achievable rate decreases significantly from $W/4$ to $W/6$.

Next we simulate a network with two parallel streams: $(60, 20)$ to $(60, 100)$ and $(60 + dist, 20)$ to $(60 + dist, 100)$. Define the *source-sink line* of a stream as the straight line connecting its source and its link. The distance *dist* between the two source-link lines are chosen from $10m$ to $20m$. Note that due to the network topology the distance between relays from different streams may be shorter

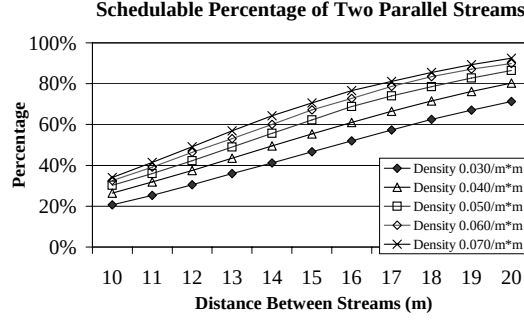


Figure 5.13: Schedulable percentage for two parallel streams

than $dist$. The interference range is $10m$. The data rate for each stream is chosen as $W/3$, so that if there is any interference between the streams, they are not schedulable. Due to lack of complete knowledge of the network topology and therefore the use of relay/interference ranges in stead of exact relay locations, our theoretical analysis gives a non-schedulable result. However, we will see that because of the randomness of relay locations, there is still a chance that the two streams are schedulable for some deployments. For each distance and density (selected from $0.03/m^2$ to $0.07/m^2$) pair, we run the simulation for 50 rounds. For each round, we choose 100 random node deployments, and for each deployment we run the simulation to see if the two streams are actually schedulable based on the per-link scheduling scheme. Then we get the schedulable percentage for each round and calculate the mean value and confidence interval of the percentages for all 50 rounds. Figure 5.13 shows the simulation results. The 95% confidence intervals are smaller than the mean values plus or minus 2%. This result shows that when the distance between two source-link lines is as close as the interference range, the probability of inter-stream interference is as high as 80% in this simulation setting. When the distance increases, the probability of inter-stream interference decreases, but there is still a nonnegligible chance that these two streams interfere with each other. This result shows that our theoretical analysis is safe, but not overly pessimistic for randomly deployed networks.

Another scenario we simulate is a network with three streams. The source-link lines of these streams cross each other and the intersection points are close to each other. The angles between two source-link lines are all 60° . The distances from the center of the system area $(60, 60)$ to the

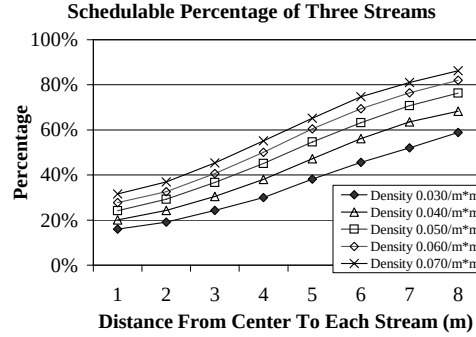


Figure 5.14: Schedulable percentage for three streams

three source-sink lines are always uniform, chosen from $1m$ to $8m$. The data rate for each stream is $W/6$. Other assumptions are the same as the above simulation setup. Due to the short distances and lack of complete knowledge of the network topology, our theoretical analysis gives a non-schedulable result. We also use 100 deployments for each round, and run 48 rounds of simulation for each distance/density pair and get the mean values and confidence intervals of the schedulable percentages. The results are shown in Figure 5.14. The 95% confidence intervals are smaller than the mean values plus or minus 2%. When the distance from the center to each source-link line is $1m$, the non-schedulable probability is as high as 80%. When the distance increases, the probability decreases significantly. However, there is still a nonnegligible chance that the streams are not schedulable, which again validates our theoretical analysis and the necessity of using safe relay/interfere regions.

From the two-stream and three-stream simulations it can be seen that the feasible rates do not increase significantly with the increase of density if the density is above $5/R_c^2$. Therefore, simply increasing density does not help improve achievable rates significantly.

5.6 Implications for Designers

We may roughly divide the spatial relations between two streams that interfere with each other into two categories: intersecting pairs and parallel pairs. In order to increase the data rates beyond the threshold shown in the theorems, the only thing a routing scheme can do is to bypass one stream

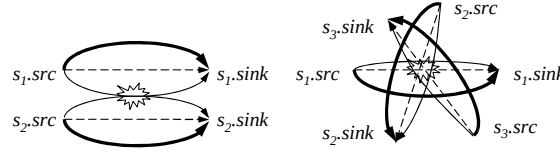


Figure 5.15: Impact of routes on feasibility

around another one for intersecting pairs. Without global knowledge it is difficult. For parallel pairs, if the distance between source-sink lines of the two streams is greater than the interference range, there exists a gray region. It is because of the randomness of routing. If by chance the two routes are further away, there may be no interference between them. If the two routes are closer than the interference range, there may be. This shows that a routing scheme that pushes two parallel routes away from each other may help increase feasible data rates.

Another interesting scenario happens when three or more streams intersect. For simplicity we only consider a three stream case. If the streams intersect at the same point, the point becomes a bottleneck if the stream rates are high. If the intersection points are further away from each other, higher data rates are possible. Therefore, a routing scheme that pushes the intersection points apart may also help increasing feasible data rates. One possible approach is to use an analogy of electric charges. The heavily loaded nodes can be thought as electric charges, when other routes are being established, these nodes repel them so that the other intersection points move further away from the already heavily loaded intersection point. A node calculates the sum of its time slots in a period by transmission or interference. If this sum is close to the period p , it broadcasts its duty cycle to its neighborhood. If another stream intends to route through its neighborhood, the stream is aware that the node is heavily loaded and will try to avoid this area.

These two scenarios are shown in Figure 5.15. In the figure, the dashed lines are the source-sink lines. The thin lines are routes that may require higher bandwidth. The thick lines show better routes that make the streams schedulable with lower bandwidth.

But the improvement due to better routing schemes has a limit. Based on an analysis technique from [27], we can get an optimistic bound for multi-hop periodic streams. One of the key implications in [27] is that “each transmission consumes valuable area.” A transmission at least

consumes a circular area with radius $(R_i - R_c)/2$ and the center is the location of the receiver of the transmission [27]. At any time the circular area of any active transmission cannot overlap with that of another active transmission, otherwise a collision occurs. Therefore, we obtain an optimistic bound:

$$\sum_{s \in S} \pi(R_i - R_c)^2 \cdot s.N \cdot s.r/4 \leq AW, \quad (5.11)$$

where S is the set of all streams in the system, $s.N$ is the hop number for stream s , and A is the whole system area. Note that this bound is optimistic, but no matter what routing scheme is taken, the feasible rates must satisfy this inequality.

5.7 Conclusions

Our main contribution of this work is providing a theoretical analysis for capacity of periodic streams in multi-hop wireless sensor networks. This analysis is important for designers to make decisions on the selection of protocols and system parameters according to a set of stream specifications. This knowledge of the relationship between system parameters such as communication and interference ranges and achievable stream rates also help correctly tune the system parameters when requirements change after the system is deployed and put into use. These benefits are not achievable from best-effort or trial-and-error approaches.

Chapter 6

Integration with AADL

In this chapter, the analytical results we have obtained in the previous chapters are integrated as plug-ins into OSATE, a popular analysis and design tool, based on AADL. Section 6.1 gives a brief introduction to AADL and OSATE. Section 6.2 describes the integration of the analysis approaches into OSATE. Section 6.3 presents the benefits and issues of using AADL and OSATE as an integration approach.

6.1 Introducing AADL and OSATE

Architecture Analysis and Design Language (AADL) is a language used to analyze and design architectures for real-time embedded systems. The development and standardization of AADL are conducted by International Society for Automotive Engineers (SAE) in its Avionics Systems Division (ASD), with the support of quite a few industrial and academic partners. Its main focus is on prediction and validation of system architecture and implementations, especially the runtime characteristics, for automotive, avionic and other industrial systems.

Several categories of components can be specified to represent various software and hardware entities in a real-time embedded system. As an example, the designer can specify memory, bus, processors, subprograms and processes in a system. A larger component can be composed of smaller subcomponents. Components are connected with connections. One of the most common connections is a port connection, in which data and control are exchanged between components. Logic

flows of information are represented by flows, which are used to support modeling and analysis such as end-to-end latency. Properties can be assigned to each component, and the name and type for each property are defined in a property set. Operational modes are also supported to design and analyze runtime behaviors so that various states of the systems can be modeled as predefined operational modes. Extensions of the core language can be defined as annexes.

An annex of the AADL language defines the meta-model and XML-based interchange formats for AADL models. The meta-model defines the object representation of AADL models, which can be manipulated through an Application Programming Interface (API). The meta-model also enables the information specified in an AADL model to be translated to an XML file based on a standardized interchange format, so that other commercial off-the-shelf tools that deal with XML formats can easily interoperate with AADL models.

The Software Engineering Institute (SEI) at Carnegie Mellon University developed an Open Source AADL Tool Environment (OSATE) based upon Eclipse, a popular open source Java-based development platform. OSATE is provided as a set of plug-ins in Eclipse and developers can easily develop tools and extensions based upon OSATE, also as plug-ins in Eclipse.

The following section describes the efforts of integrating the analytical approaches presented in the previous three chapters into OSATE as plug-ins.

6.2 Integrating the Analysis Approaches into OSATE

In this section we discuss the integration of the analytical approaches into OSATE. In particular, we describe the implementation of the duty-cycle based tracking analysis presented in Chapter 4 in details as an example.

In OSATE, when an AADL project is created, users can add AADL models into the project. Once an AADL model is saved or imported from out of the project, its corresponding XML model is created automatically with a “.aaxl” extension, and the AADL plug-ins can be applied to the .aaxl file. Figure 6.1 is an example.

In Figure 6.1, the three icons circled in the toolbar correspond to the tools for low duty-cycle

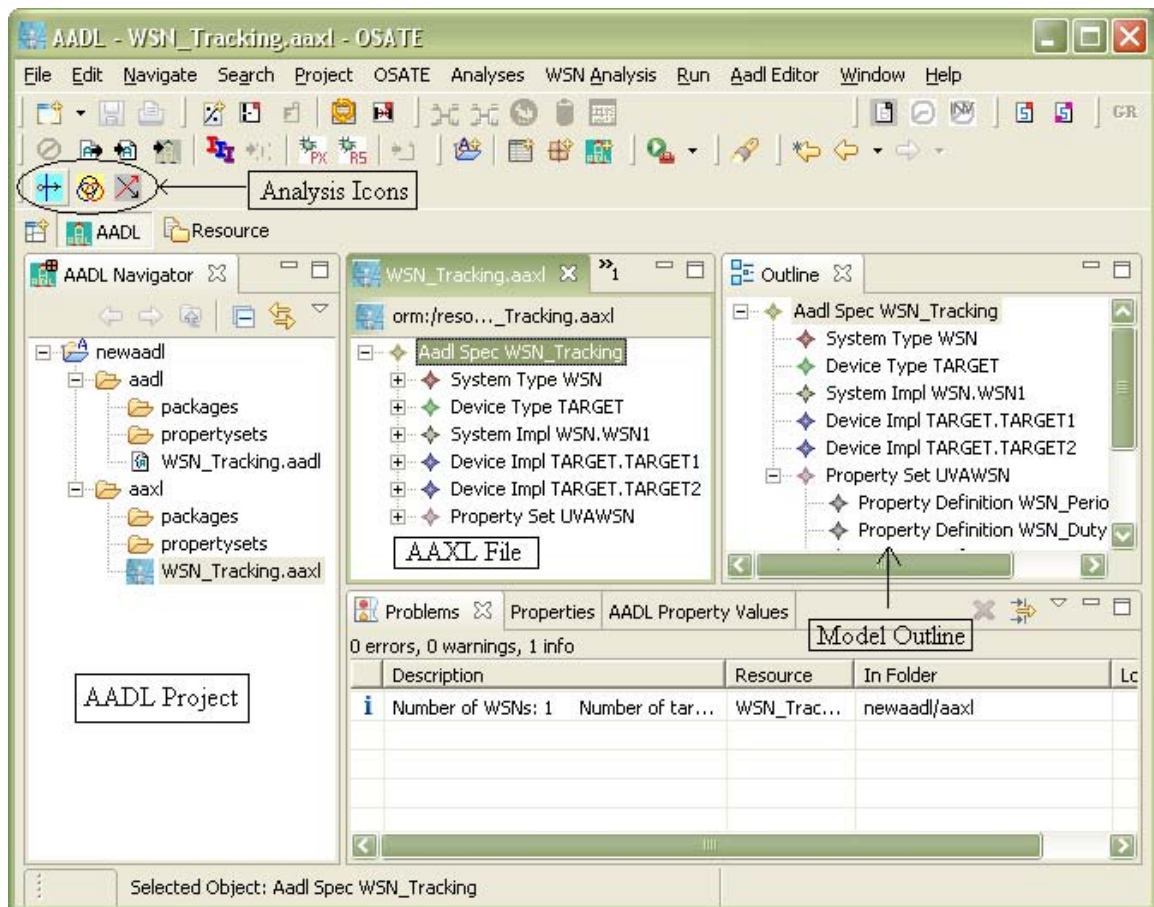


Figure 6.1: AADL Plug-In Screenshot

```

property set UVAWSN is
    WSN_Period: aadlreal applies to (system);
    WSN_DutyCycle: aadlreal applies to (system);
    WSN_SensingRange: aadlreal applies to (system);
    WSN_Density: aadlreal applies to (system);
    WSN_Width: aadlreal applies to (system);
    TARGET_Speed: aadlreal applies to (device);
    TARGET_FromOutside: aadlboolean applies to (device);
end UVAWSN;

```

Figure 6.2: Specifying Property Set

surveillance analysis, sensing coverage analysis and stream analysis, from left to right. In order to apply the surveillance analysis, the user needs to click on the “WSN_Tracking.aaxl” file, the XML representation of the AADL file, and then click on the icon for the low duty-cycle surveillance analysis.

The analysis tool for low duty-cycle surveillance, written in Java, does the following to implement the analysis functionality. First, it searches for all the necessary properties definitions in the AADL model required for the analysis. Then it traverses the AADL model to collect the property values of all components. Finally, it applies formulas derived in Chapter 4 to calculate the results, in this case, the detection probability and the expected delay.

In order to apply this analysis tool, the user has to define the properties that are necessary for the analysis in a “Property Set”. For the surveillance analysis, the deployment density, the sensing range, the target speed, the duty cycle and the period are necessary. The property set is defined as in Figure 6.2.

The “Target_FromOutside” property is used to specify whether the target comes from outside the network. If this boolean property is true, the target is a type I target or a type III target. If the property is false, it is a type II target or a type IV target. Other properties are self-explanatory. The wireless sensor network is specified as a system, and the target to be analyzed is specified as

```

system WSN
  properties
    UVAWSN::WSN_DutyCycle => 0.3;
    UVAWSN::WSN_SensingRange => 8.0;
    UVAWSN::WSN_Density => 0.1;
end WSN;

device TARGET
  properties
    UVAWSN::TARGET_Speed=> 20.0;
    UVAWSN::TARGET_FromOutside => true;
end TARGET;

```

Figure 6.3: Specifying Component Types

a device. Before an instance of a system or a device and its property values are specified, its type must be defined as in Figure 6.3.

Note that each property is given a default value. If a property value is not specified in a component instant, the default value is used. After the component types (here a system type “WSN” and a device type “TARGET”) are defined, the users can specify the wireless sensor networks and the targets as in Figure 6.4.

Figure 6.4 specifies one wireless sensor network and two targets. The users can specify more than one wireless sensor network instance, and the analysis tool calculates the detection probability and expected delay for each combination of WSN and TARGET instances. Figures 6.2, 6.3, and 6.4 compose a complete model for surveillance analysis. After applying the analysis tool by clicking on the left icon, a window appears with the analysis results as shown in figure 6.5.

6.3 Benefits and Issues

One benefit of integrating the analysis approaches into an AADL based tool is that it makes it very easy for the users to specify the system and protocol properties and use the tools. AADL and OSATE are gaining popularity very fast and more tools will be built upon this platform. There-

```

system implementation WSN.WSN1
  properties
    UVAWSN::WSN_Period => 1.0;
    UVAWSN::WSN_DutyCycle => 0.1;
    UVAWSN::WSN_SensingRange => 4.0;
    UVAWSN::WSN_Density => 0.016;
    UVAWSN::WSN_Width => 15.0;
end WSN.WSN1;

device implementation TARGET.TARGET1
  properties
    UVAWSN::TARGET_Speed=> 20.0;
    UVAWSN::TARGET_FromOutside => true;
end TARGET.TARGET1;

device implementation TARGET.TARGET2
  properties
    UVAWSN::TARGET_Speed=> 5.0;
    UVAWSN::TARGET_FromOutside => false;
end TARGET.TARGET2;

```

Figure 6.4: Specifying Component Instances

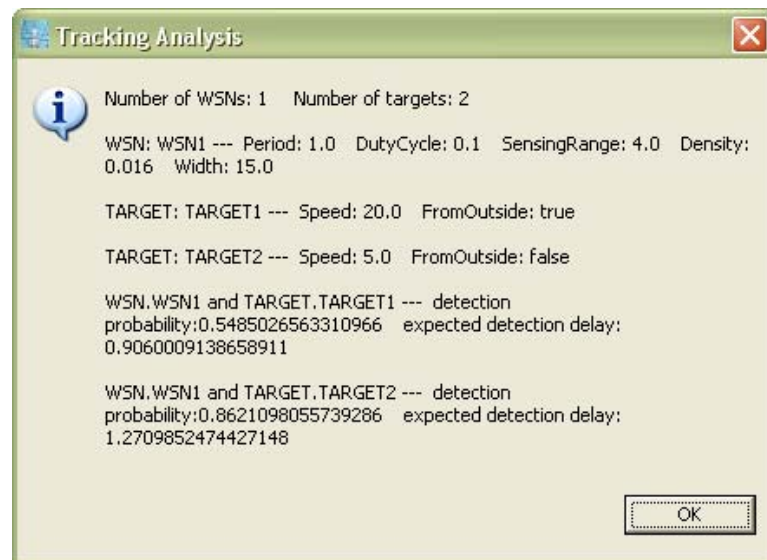


Figure 6.5: Surveillance Analysis Results

fore, choosing AADL and OSATE as an integration platform is also beneficial for interoperability purposes.

On the other hand, there are certain shortcomings of the AADL core language that makes it hard to incorporate some modeling concepts in wireless sensor networks. One of them is lack of convenient means of specifying the network topology. Currently in the core language there is not a way to automatically generate node locations for random or grid topologies. The user has to input the locations by hand, which is very inconvenient. Another shortcoming is the lack of capability of specifying logic rules. For example, it is rather common to define two nodes to be neighbors if the distance between them is shorter than the communication range, while due to the shortcoming, there is no straightforward way to define this condition in AADL. Although most of our analysis approaches only deal with system-wide parameters, some of them can be easily extended to handle individual node information. For example, the capacity analysis for periodic streams can be easily extended to the case when the node locations are known. In this case, lack of convenient means of specifying the node locations is an important issue. Therefore, if more complicated analysis approaches that involve properties of individual nodes or individual links need to be incorporated into an AADL based tool, it will be beneficial to define an extension, or annex, to the AADL core language to handle multiplicity and other issues.

Chapter 7

Conclusions and Future Work

To conclude this dissertation, in Section 7.1 this chapter revisits the theoretical and practical contributions of the presented work, and in Section 7.2 proposes potential extensions to be explored in the future.

7.1 Contributions Revisited

As proven by various successful application prototypes, wireless sensor networks are a powerful information gathering paradigm that has evident advantages in terms of both functionality and performance over conventional sensor systems. On the other hand, the achievable performance of wireless sensor networks is limited by several resource constraints. Some resource constraints, such as the processing power of microcontrollers, can be projected to be relaxed drastically in the near future. Other resource constraints are less likely to be relaxed significantly in the foreseeable future due to technological limitations, laws of physics and government regulations. Among these constraints, battery life, sensing range and communication bandwidth are deemed very important because their close relationship to system performance. It is therefore very important to understand the quantitative impacts of these constraints on performance metrics for both theoretical and practical purposes.

This dissertation identifies three major aspects of performance of wireless sensor networks and study the impacts of the above mentioned constraints on these aspects. The three aspects are sensing

coverage, tracking and periodic communications. Here we summarize the theoretical and practical contributions of the work on these three areas presented in this dissertation.

7.1.1 Theoretical Contributions

The theoretical contributions are summarized as follows:

1) The work on sensing coverage is the first to present a energy efficient sensing coverage protocol that provide the flexibility to select various degrees of sensing coverage over the target area with provable sensing coverage guarantees. The protocol outperforms a well-known peer protocol by as much as 130%.

2) The work on low duty-cycle surveillance provides the first thorough theoretical analysis on the impact of duty-cycle and other factors on the performance of low duty-cycle surveillance systems, which gives quantitative relationship between the system lifetime and tracking performance.

3) The work on capacity and feasibility analysis of periodic streams gives quantitative relationship between the achievable rates of a specified set of periodic communication streams and the system bandwidth, which is a significant improvement over asymptotic capacity bounds.

7.1.2 Practical Contributions

On the practical side, the analytical approaches presented in the dissertation provide criteria for choosing system and protocol parameters for wireless sensor network designers. Compared with conventional approaches that involve best-effort designs and trial and error testings, the analytical approaches are much less costly, easier to use, and more adaptive to environmental variations. The following lists the detailed practical contributions in the three application areas.

1) The work in sensing coverage provides the designers the quantitative relationship among node density, average duty-cycle and guaranteed degree of sensing coverage. The designers are able to select a set of parameters according to the requirements on system lifetime and desired degree of sensing coverage based on the results of our work.

2) The work in tracking performance provides the designers the quantitative relationship among sensor duty-cycles, node density and tracking performance metrics. The designers are able to select

a set of parameters according to the tracking performance requirements based on our analysis.

3) The work in capacity of periodic streams provides the designers the quantitative relationship between system bandwidth and achievable rates of a set of specified streams. The designers are able to select the system bandwidth if the set of streams are fixed, or to adjust the stream set if the system bandwidth is fixed based on our analysis.

Furthermore, the dissertation also describes how to integrate the analytical approaches into OSATE plug-ins based on AADL, a standardized analysis and design language.

In addition to these specific contributions, the approaches on the individual analysis problems also provide methodological insights on how to generate analytical approaches for predictable performance for wireless sensor networks. Although all three pieces of work reported in this dissertation have an analytical flavor, the manners that individual problems are tackled are quite different. For sensing coverage, the problem is solved constructively in the sense that we propose a brand new protocol and the analytical guarantee of degree of sensing coverage is proved during the design of the protocol. For low duty-cycle tracking, we take a commonly used scheme and conduct mathematical analysis based on probabilistic and geometric calculations. For capacity analysis of periodic streams, the approach is a combination of geometric analysis and scheduling analysis due to the spatiotemporal characteristics of the problem. The efforts of integration of these analytical approaches also set examples for how to incorporate similar analysis into a well known design tool.

7.2 Potential Extensions

Due to the complexity of wireless sensor network systems and their broad applications, there are ample opportunities to extend the results presented here based on similar methodologies used in this dissertation. Several potential extensions are listed as follows.

1) Mobile sensor networks. Although this dissertation does analyze the performance of tracking mobile targets, mobile sensor networks are not considered here. It will be very interesting work to explore analytical approaches that help predict performance of sensing, tracking and communications for mobile sensor networks.

2) Probabilistic sensing model. This dissertation assumes that a target is detected for certain by a sensor node when it appears in its sensing area. In reality, some types of sensors detect targets with a probability, so a probabilistic sensing model is more appropriate for these sensors. It will be interesting to extend the work on sensing coverage and tracking performance presented in this dissertation to a probabilistic sensing model.

3) 3-dimensional sensing model. This dissertation only consider sensing in a 2-dimensional plane, while some applications do require 3-dimensional sensing. Therefore, it is necessary to analyze the sensing coverage and tracking performance when the 3-dimensional sensing model is applied.

4) Data aggregation. In the analysis work on capacity analysis of periodic streams presented in this dissertation, data aggregation is not considered. It will be valuable to analyze streams with data aggregation, since data aggregation is a widely utilized technique in real data collection applications.

5) Quality of service. In many applications, due to bandwidth and energy consumption constraints it is almost impossible to send all the information gathered at the sensors back to the data sinks. Processing at the sensor node that gathers the information or at the relays are unavoidable. Due to the fact that most sensor networks employ low end microcontrollers for energy, form factor and cost constraints, their computational capability is extremely limited. The designers have to balance between processing and communications, to deliver data with reasonable quality within the capacities of both communication and computation.

Evidently there are many more potential areas that extensions of the work presented in the dissertation can help predict performance in various aspects, and the fields listed above are far from complete. Either new protocols with analytical guarantees or analysis over existing protocols can add significant values for both theory and practice of wireless sensor networks.

It is also interesting to transform some of the off-line analysis approaches into on-line approaches that handle more dynamic scenarios. For example, we can transform the off-line analysis approach for periodic streams into an on-line scheduling scheme. After calculating the schedules, a base station sends them to the nodes for all streams. Each time a stream enters or exits the system,

the base station may need to recalculate the schedules and send them to the sensor nodes.

Another interesting potential extension is to improve our approaches for preventing attacks from adversaries. For example, if an adversary can detect the work/sleep schedules of a low duty-cycle wireless sensor network based surveillance system, he or she may maneuver a target to avoid being detected. A potential solution is to switch to new schedules frequently to prevent the detection of the schedules by adversaries.

Bibliography

- [1] In <http://www.cast.cse.ohio-state.edu/exscal/>.
- [2] T. F. Abdelzaher, S. Prabh, and R. Kiran. On real-time capacity limits of multihop wireless sensor networks. In *IEEE RTSS 2004*.
- [3] A. Agarwal and P. R. Kumar. Improved capacity bounds for wireless networks. *Wireless Communications and Mobile Computing*, 4:251–261, 2004.
- [4] G.-S. Ahn, A. T. Campbell, A. Veres, and L.-H. Sun. Swan: Service differentiation in stateless wireless ad hoc networks. In *IEEE INFOCOM*, 2002.
- [5] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless Sensor Networks: A Survey. 38(4):393–422, March 2002.
- [6] H. Alt, D. Hsu, and J. Snoeyink. Computing the largest inscribed isothetic rectangle. In . *Proceeding of 7th Canadian Conf. Comput. Geom.*, pages 67–72, 1995.
- [7] O. Arpacioglu and Z. J. Haas. On the scalability and capacity of wireless networks with omnidirectional antennas. In *IPSN'04*.
- [8] J. Aslam, Z. Butler, F. Constantin, V. Crespi, G. Cybenko, and D. Rus. Tracking a Moving Object with a Binary Sensor Network. In *First ACM Conference on Embedded Networked Sensor Systems (SenSys 2003)*, November 2003.
- [9] L. Bao and J. J. Garcia-Luna-Aceves. A new approach to channel access scheduling for ad hoc networks. In *ACM SIGMOBILE'01*.

- [10] G. Barrenechea, B. Beferull-Lozano, and M. Vetterli. Lattice sensor networks: Capacity limits, optimal routing and robustness to failures. In *IPSN'04*.
- [11] S. Bhatnagar, B. R. Deb, and B. Nath. Service Differentiation in Sensor Networks. In *Intl. Symposium on Wireless Personal Multimedia Communications*, September 2001.
- [12] S. Bhattacharya, H. Kim, S. Prabh, and T. Abdelzaher. Energy-Conserving Data Placement and Asynchronous Multicast in Wireless Sensor Networks. In *The First International Conference on Mobile Systems, Applications, and Services (MobiSys)*, May 2003.
- [13] R. R. Brooks, P. Ramanathan, and A. Sayeed. Distributed target tracking and classification in sensor networks. *Proceedings of the IEEE*, 2002.
- [14] B. M. Blum C. Lu, T. F. Abdelzaher, J. A. Stankovic, and T. He. Rap: A real-time communication architecture for large-scale wireless sensor networks. In *IEEE RTAS*, 2002.
- [15] Q. Cao, T. Abdelzaher, T. He, and J. A. Stankovic. Towards optimal sleep scheduling in sensor networks for rare-event detection. In *the Fourth International Conference on Information Processing in Sensor Networks (IPSN'05)*, 2005.
- [16] A. Cerpa and D. Estrin. ASCENT: Adaptive Self-Configuring Sensor Networks Topologies. In *Proc. of the IEEE Computer and Communications Societies (INFOCOM)*, June 2002.
- [17] A. Chakarabarti, A. Sabharwal, and B. Aazhang. Multi-hop communication is order-optimal for homogeneous sensor networks. In *IPSN'04*.
- [18] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. In *6th ACM MOBICOM Conference*, 2001.
- [19] CrossBow. In <http://www.xbow.com>.
- [20] CrossBow. *MicaZ Data Sheet*. Available at http://www.xbow.com/products/Product_pdf_files/Wireless_pdf/MICAz_Datasheet.pdf.

- [21] CrossBow Technology, Inc. *Mica2 Data Sheet*. Available at http://www.xbow.com/Products/Product_pdf_files/wireless_pdf/6020-0042-01_A_MICA2.pdf.
- [22] K. Daniels, V. Milenkovic, and D. Roth. Finding the largest area axis-parallel rectangle in a polygon. pages 125–148, 1997.
- [23] J. Elson, L. Girod, and D. Estrin. Fine-grained network time synchronization using reference broadcasts. In *Symposium on Operating Systems Design and Implementation*, 2002.
- [24] L. Gu and J. A. Stankovic. Radio-Triggered Wake-Up Capability for Sensor Networks. In *Proceedings of RTAS*, 2004.
- [25] C. Gui and P. Mohapatra. Power conservation and quality of surveillance in target tracking sensor networks. In *ACM Mobicom*, 2004.
- [26] C. Guo, L. C. Zhong, and J. M. Rabaey. Low power distributed mac for ad hoc sensor radio networks. In *IEEE GlobeCom*, 2001.
- [27] P. Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE Transaction on Information Theory*, 46(2), March 2000.
- [28] T. He, B. M. Blum, J. A. Stankovic, and T. F. Abdelzaher. AIDA: Adaptive Application Independent Data Aggregation in Wireless Sensor Networks. *ACM Transactions on Embedded Computing System, Special issue on Dynamically Adaptable Embedded Systems*, 2004.
- [29] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher. Range-Free Localization Schemes in Large-Scale Sensor Networks. In *Proc. of the Intl. Conference on Mobile Computing and Networking (MOBICOM)*, September 2003.
- [30] T. He, S. Krishnamurthy, J. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, J. Hui, and B. Krogh. An energy-efficient surveillance system using wireless sensor networks. In *ACM Mobisys*, 2004.

- [31] T. He, J. Stankovic, C. Lu, and T. Abdelzaher. SPEED: A Stateless Protocol for Real-Time Communication in Ad Hoc Sensor Networks. In *Proc. of International Conference on Distributed Computing Systems (ICDCS)*, May 2003.
- [32] T. He, P. Vicaire, T. Yan, Q. Cao, G. Zhou, L. Gu, L. Luo, R. Stoleru, J. A. Stankovic, , and T. Abdelzaher. Achieving Long-Term Surveillance in VigilNet. In *IEEE Infocom*, 2006.
- [33] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-Efficient Communication Protocol for Wireless Microsensor Networks. In *Proc. of the Intl. Conference on System Sciences*, January 2000.
- [34] C. Hsin and M. Y. Liu. Network coverage using low duty-cycled sensors: Random and coordinated sleep algorithms. In *Proceedings of IPSN*, 2004.
- [35] C.-F. Hsin and M. Liu. Network Coverage Using Low Duty-Cycle Sensors: Random & Coordinated Sleep Algorithms. In *The 3rd International Symposium on Information Processing in Sensor Networks (IPSN'04)*, Berkeley, California, April 2004.
- [36] H. Takagi and L. Kleinrock. Optimal transmission ranges for randomly distributed packet radio terminals. *IEEE Trans. on Communication*, 32(3), 1984.
- [37] C. Intanagonwiwat, D. Estrin, R. Govindan, and J. Heidemann. Impact of network density on data aggregation in wireless sensor networks. In *International Conference on Distributed Computing Systems*, 2002.
- [38] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu. Impact of interference on multi-hop wireless network performance. In *ACM MobiCom'03*.
- [39] D. Kirkpatrick and J. Snoeyink. Tentative prune-and-search for computing fixed-points with applications to geometric computation. pages 353–370, 1995.
- [40] M. Kodialam and T. Nandagopal. Characterizing achievable rates in multihop wireless networks: The joint routing and scheduling problem. In *ACM MobiCom'03*.

- [41] U. C. Kozat and L. Tassiulas. Throughput capacity of random ad hoc networks with infrastructure support. In *ACM MobiCom'03*.
- [42] B. Krishnamachari, D. Estrin, and S. Wicker. Impact of Data Aggregation in Wireless Sensor Networks. In *Proc. of Intl. Workshop on Distributed Event-Based Systems*, July 2002.
- [43] J. Li, C. Blake, D. D. Couto, H. I. Lee, and R. Morris. Capacity of ad hoc wireless networks. In *ACM MobiCom'01*.
- [44] J. J. Liu, J. Liu, J. Reich, P. Cheung, and F. Zhao. Distributed group management for track initiation and maintenance in target localization applications. In *Proceedings of IPSN*, 2003.
- [45] R. Min, M. Bhardwaj, S.-H. Cho, A. Sinha, E. Shih, A. Wang, and A. Chandrakasan. An architecture for a power-aware distributed microsensor node. In *IEEE Workshop on Signal Processing Systems*, 2000.
- [46] S. Port. *Theoretical Probability for Applications*. John Wiley and Sons, Inc, 1994.
- [47] M. Rahimi, R. Baer, O. I. Iroez, J. C. Garcia, J. Warrior, D. Estrin, and M. Srivastava. Cyclops: In situ image sensing and interpretation in wireless sensor networks. In *ACM SenSys'05*.
- [48] R. Ramanathan and R. Rosales-Hain. Topology control of multihop wireless networks using transmit power adjustment. In *IEEE INFOCOM*, 2000.
- [49] S. Ren, Q. Li, H. N. Wang, X. Chen, and X. D. Zhang. Analyzing object tracking quality under probabilistic coverage in sensor networks. In *ACM Mobile Computing and Communications Review*, 2005.
- [50] D. Son, B. Krishnamachari, and J. Heidemann. Experimental study of the effects of transmission power control and blacklisting in wireless sensor networks. In *IEEE SECON'04*.
- [51] D. Tian and N.D. Georganas. A Node Scheduling Scheme for Energy Conservation in Large Wireless Sensor Networks. *Wireless Communications and Mobile Computing Journal*, May 2003.

- [52] E. W. Weisstein. Kolmogorov-smirnov test. In *MathWorld* at <http://mathworld.wolfram.com/Kolmogorov-SmirnovTest.html>.
- [53] R. Williams. *Geometrical Foundation of natural Structure:A Source Book of Design*. Dover Publications Inc, New York, 1979.
- [54] G. Xing, C. Lu, R. Pless, and Q. Huang. Impact of Sensing Coverage on Greedy Geographic Routing Algorithms. *IEEE Transactions on Parallel and Distributed Systems, Special Issue on Localized Communication and Topology Protocols for Ad Hoc Networks*, 2006.
- [55] G. Xing, C. Lu, R. Pless, and J. A. O'Sullivan. Co-Grid: an Efficient Coverage Maintenance Protocol for Distributed Sensor Networks. In *The 3rd International Symposium on Information Processing in Sensor Networks (IPSN'04)*, Berkeley, California, April 2004.
- [56] G. Xing, X. Wang, Y. Zhang, C. Lu, R. Pless, and C. Gill. Integrated Coverage and Connectivity Configuration for Energy Conservation in Sensor Networks. *ACM Transactions on Sensor Networks*, 1(1), August 2005.
- [57] Y. Xu, J. Heidemann, and D. Estrin. Geography-informed energy conservation for ad hoc routing. In *MobiCom*, 2001.
- [58] Y. Xue and B. Li. A location-aided power-aware routing protocol in mobile ad hoc networks. In *IEEE GlobeCom*, 2001.
- [59] F. Ye, G. Zhong, S. Lu, and L. Zhang. Energy Efficient Robust Sensing Coverage in Large Sensor Networks. Technical report, UCLA, 2002.
- [60] F. Ye, G. Zhong, S. Lu, and L. Zhang. PEAS: A Robust Energy Conserving Protocol for Long-lived Sensor Networks. In *Proc. of International Conference on Distributed Computing Systems (ICDCS)*, May 2003.
- [61] G. Zhou, T. He, S. Krishnamurthy, and J. A. Stankovic. Impact of Radio Irregularity on Wireless Sensor Networks. In *ACM MobySys*, 2004.

- [62] G. Zhou, T. He, J. A. Stankovic, and T. F. Abdelzaher. RID: Radio Interference Detection in Wireless Sensor Networks. In *IEEE InfoCom*, 2005.