

NODE PLACEMENT, ROUTING AND LOCALIZATION ALGORITHMS FOR HETEROGENEOUS  
WIRELESS SENSOR NETWORKS

Except where reference is made to the work of others, the work described in this thesis is my own or was done in collaboration with my advisory committee. This thesis does not include proprietary or classified information.

---

Shaoqiang Dong

Certificate of Approval:

---

Stanley Reeves  
Professor  
Electrical and Computer Engineering

---

Prathima Agrawal, Chair  
Samuel Ginn Distinguished Professor  
Electrical and Computer Engineering

---

Shiwen Mao  
Assistant Professor  
Electrical and Computer Engineering

---

Joe F. Pittman  
Interim Dean  
Graduate School

NODE PLACEMENT, ROUTING AND LOCALIZATION ALGORITHMS FOR HETEROGENEOUS  
WIRELESS SENSOR NETWORKS

Shaoqiang Dong

A Thesis

Submitted to

the Graduate Faculty of

Auburn University

in Partial Fulfillment of the

Requirements for the

Degree of

Master of Science

Auburn, Alabama  
May 10, 2008

NODE PLACEMENT, ROUTING AND LOCALIZATION ALGORITHMS FOR HETEROGENEOUS  
WIRELESS SENSOR NETWORKS

Shaoqiang Dong

Permission is granted to Auburn University to make copies of this thesis at its discretion, upon the request of individuals or institutions and at their expense. The author reserves all publication rights.

---

Signature of Author

---

Date of Graduation

## VITA

Shaoqiang Dong was born in Shandong province of China on November 14th, 1971. He received B.S. and M.S. degrees in 1994 and 1997 respectively, both in Mechanical Engineering, from Xi'an Jiaotong University of China. He also received his Ph.D degree in mechanical engineering in 2004 from University of Massachusetts at Amherst. From August 2004 to December 2005, he worked as a research engineer at Migma Systems Inc. He began his graduate studies in Electrical and Computer Engineering at Auburn University in January 2006.

THESIS ABSTRACT

NODE PLACEMENT, ROUTING AND LOCALIZATION ALGORITHMS FOR HETEROGENEOUS  
WIRELESS SENSOR NETWORKS

Shaoqiang Dong

Master of Science, May 10, 2008  
(Ph.D, University of Massachusetts, Amherst, MA, 2004)

73 Typed Pages

Directed by Prathima Agrawal

Wireless sensor network is a very promising technology for many military and civilian applications. The network usually consists large number of nodes and many existing network technologies can not be used directly in such networks. To deal with such problems, heterogeneous and hierarchical network architecture have been proposed in recent years. This thesis studies node placement, routing and localization problems in such network architecture.

For node placement, we consider a two-tiered wireless sensor network where Lite Nodes (LN, sensor nodes) are placed randomly to sense the environment and Sophisticated Nodes (SN) are placed to aggregate data from LNs and send them to the sink. We intend to minimize the number of SNs and make the SNs form a connected network. We formulate this as an optimization problem and solve it with integer linear programming.

Routing algorithm for sensor networks must be scalable and energy-efficient due to large-scale and resource-constraint characteristics of sensor networks. We proposes a reinforcement learning based geographic routing algorithm. The algorithm addresses several

issues in traditional geographic routing and considers energy-efficiency. Simulations demonstrate that the routing algorithm can significantly improve the network lifetime.

For node localization, we propose a hierarchical localization method for sensor node localization. In addition to LNs and SNs, a small number of anchor nodes with known locations are placed in the network. By using a hierarchical network architecture and UWB technology, only a small number of anchor nodes is needed. Our localization method works in two steps. First, SNs estimate their locations by using the built-in high-precision distance measurement capabilities and locations of anchor nodes. Next, lite nodes use SNs as references to estimate their locations. If a lite node has fewer than three neighboring SNs, it uses SNs and other localized lite sensor nodes as references. Both iterative and non-iterative methods are studied. Extensive MATLAB simulations are performed to study the resulting error performance. It is observed that iterative method produces far lower localization error. The effects of UWB node placement and other parameters such as measurement noise, number of anchor nodes, and communication ranges on the resulting localization error are also presented.

## ACKNOWLEDGMENTS

First and foremost, I would express my sincere gratitude to my advisor Professor Prathima Agrawal for her guidance and help during my study and research at Auburn. Thanks for leading me into the research area of wireless networks. Without her patience and support, this thesis would not be possible.

I also thank Professors Stanley Reeves and Shiwen Mao for serving on my advisory committee. My thanks also go out to my colleagues in the Wireless Research Laboratory, Santosh Pandey, Pratap Simha, Priyanka Sinha, Srivathsan Soundararajan and Yogesh Reddy. In particular, I thank Priyanka for helping me with many computer and programming problems. I would like to thank Santosh and Pratap for the discussions and valuable suggestions on our research.

I also am grateful to my wife, Juxiang Cao, for her love and support. Many thanks to my parents for their consistent support during my study.

Part of the research was supported by a grant from Air Force Office of Scientific Research (AFOSR) grant No. FA9550-06-1-0103.

Style manual or journal used Journal of Approximation Theory (together with the style known as “aums”). Bibliography follows van Leunen’s *A Handbook for Scholars*.

---

Computer software used The document preparation package T<sub>E</sub>X (specifically L<sup>A</sup>T<sub>E</sub>X) together with the departmental style-file `aums.sty`.

---



## TABLE OF CONTENTS

LIST OF FIGURES	x
1 INTRODUCTION	1
1.1 Wireless Sensor Network and Its Applications . . . . .	1
1.2 Sensor Network Architecture . . . . .	2
1.3 Node Placement . . . . .	3
1.4 Reinforcement Learning and Position Based Routing Algorithm . . . . .	3
1.5 Hierarchical Localization for Heterogeneous Sensor Networks . . . . .	4
1.6 Thesis Outline . . . . .	6
2 LITERATURE REVIEW	7
2.1 Node Placement . . . . .	7
2.2 Routing Algorithms in Sensor Network . . . . .	9
2.3 Node Localization in Sensor Network . . . . .	11
3 HETEROGENEOUS SENSOR NETWORK NODE PLACEMENT	15
3.1 Problem Definition . . . . .	15
3.2 Binary Integer Linear Programming Solution . . . . .	19
3.3 Simulation Results . . . . .	21
4 REINFORCEMENT LEARNING BASED GEOGRAPHIC ROUTING ALGORITHM	26
4.1 Proposed Algorithm . . . . .	26
4.2 Simulation . . . . .	33
4.2.1 Simulation with grid network . . . . .	33
4.2.2 Simulation of Randomly Distributed Network . . . . .	36
5 LOCALIZATION ERROR EVALUATION IN HETEROGENEOUS SENSOR NETWORKS	40
5.1 System Architecture . . . . .	40
5.2 Algorithm Description . . . . .	41
5.2.1 Non-Iterative Method . . . . .	43
5.2.2 Iterative Approach . . . . .	44
5.3 Simulation Results . . . . .	45
5.3.1 UWB Node Localization . . . . .	46
5.3.2 Sensor Node Localization . . . . .	50
6 CONCLUSION AND FUTURE WORK	56
BIBLIOGRAPHY	58

## LIST OF FIGURES

3.1	BILP results. $N(SN)=42$ nodes . . . . .	22
3.2	Greedy results. $N(SN)=40$ nodes . . . . .	22
3.3	GA results. $N(SN)= 32$ nodes . . . . .	23
3.4	Performance of different algorithms. LN placed over a grid . . . . .	24
3.5	Performance of different Algorithms. LN (uniform) randomly distributed . . . . .	24
4.1	Illustration of "hole" problem . . . . .	30
4.2	Energy distribution of GRLR after network partition . . . . .	35
4.3	Energy distribution of GPSR after network partition . . . . .	35
4.4	Energy distribution of GRLR after network partition with hole . . . . .	37
4.5	Residual energy distribution of GRLR after network partition . . . . .	38
4.6	Residual energy distribution of GPSR after network partition . . . . .	39
5.1	Illustration of system architecture . . . . .	41
5.2	Illustration of triangulation . . . . .	42
5.3	Localization of UWB nodes with anchor nodes . . . . .	47
5.4	Histogram of initial adjacent anchor nodes . . . . .	48
5.5	Histogram of number of anchor nodes before localization . . . . .	49
5.6	Average localization error as a function of distance measurement error . . . . .	50
5.7	Node distribution with more anchor nodes . . . . .	51
5.8	Average localization error for different number of anchor nodes . . . . .	52

5.9	Error analysis using non-iterative and iterative methods . . . . .	53
5.10	Localization of sensor nodes . . . . .	54
5.11	Distribution of three types of nodes . . . . .	54
5.12	Coverage of sensor nodes by UWB nodes . . . . .	55
5.13	Variation of location error of sensor nodes for different communication ranges	55

## CHAPTER 1

### INTRODUCTION

#### 1.1 Wireless Sensor Network and Its Applications

Wireless sensor network consist of large number of densely deployed sensor nodes to monitor and control the environment. Sensor nodes are small, low-cost, low power devices with sensing, computing and communication capabilities. A sensor node can only cover a small area, but cooperation of many sensors can provide more reliable and better monitoring of large areas. The sensor nodes can be deployed randomly and they can establish ad hoc networkss to collect and send data to the remote base station.

Sensor networks have a variety of military and civilian applications [1]. The rapid deployment, self-organization and fault tolerance characteristics of sensor networks make them a very useful sensing technique for military applications such as battlefield surveillance, damage assessment, biological and nuclear attack detection, equipment monitoring and reconnaissance of opposing forces and terrain. For homeland security, sensor network can be deployed for border surveillance.

For civilian applications, sensor networks can be used for environment monitoring, including habitat monitoring (determining the plant and animal species population and behavior), seismic detection, forest fire detection and flood detection. Sensor networks can provide better precision and more information at lower cost than using satellite surveillance for these purposes. Sensor network can also be used for city traffic and pollution monitoring. For instance, Harvard University and BBN Technologies are working on a project to cover Cambridge, MA, with wireless sensor nodes mounted to telephone poles to monitor

pollutions and weather [2]. The network, called CitySense, will be an open testbed on which anyone can run experiments.

Other applications include industrial process control and automation, machine health monitoring and diagnosis, vehicle tracking and detection.

## 1.2 Sensor Network Architecture

In sensor networks, sensor nodes collect local data and send them to the base station (sink). It is usually not practical for sensor nodes to transmit data directly to the sink, because the sink is usually far away from the sensors and sensor nodes have limited transmission range. Two network architectures are used in recent research: multi-hop based architecture and cluster-based architecture.

In multi-hop architecture, each node forwards data to the nodes closer to the sink. The disadvantage is that data forwarding consumes large volume of the limited energy.

In the cluster-based network architecture, the nodes close to each other form clusters. One of the nodes acts as clusterhead. Other nodes send data directly to the clusterhead. The clusterhead is responsible for forwarding the data to the sink. Various schemes have been proposed to select the clusterheads.

It is well known that the cluster-based architecture is more scalable. In earlier research, the clusterhead nodes are selected from some of the sensor nodes [3]. The sensor nodes rotate as clusterheads to prolong network lifetime. Recently, some papers proposed to place some more powerful nodes in the network to act as clusterheads. These nodes are called relay nodes, gateway nodes or application nodes. The architecture can dramatically extend the network lifetime and reduce network traffic. In this thesis, we use this network architecture.

In the following sections, we provide an overview of the issues and proposed solutions in this thesis.

### 1.3 Node Placement

As discussed in the previous section, we consider a two-tiered hierarchical network architecture. The sensor nodes (called Lite Nodes) are placed randomly in an area. Some Sophisticated Nodes (SNs) which have more computation, energy and communication capabilities, are placed as clusterheads. Assuming that the location of sensor nodes is known, we intend to place the minimum number of SNs to handle the traffic generated by LNs (*traffic constraint*) and ensure that the SNs form a connected network (*connectivity constraint*). The problem is formulated into an optimization problem and Binary Integer Linear Programming (BILP) is proposed as one solution to the problem.

### 1.4 Reinforcement Learning and Position Based Routing Algorithm

Many routing protocols have been proposed for mobile and ad hoc networks. However, they cannot be directly used in sensor networks due to high-energy cost and overhead. In this thesis, we propose a reinforcement learning and position based routing algorithm. In the network architecture, we assume that the sophisticated nodes are equipped with UWB transceivers.

Ultra-Wide Band [4] has attracted research interests in recent years after FCC granted unlicensed frequency bandwidth for UWB communication. UWB has several advantages over narrow band systems, such as high data rate, low power cost, and high immunity to multipath fading. The high-precision ranging capability of UWB can be used to estimate

the location of UWB nodes [5, 6]. The location information is very useful for geographic routing [7].

Geographic routing is very attractive for sensor networks because the nodes need only local information and therefore improve algorithm scalability. Because the nodes in sensor networks have limited energy, many routing protocols [6, 8, 9] attempt to improve energy efficiency to increase network lifetime. Although UWB is more energy efficient than narrow band radio, the routing protocol still needs to consider energy cost to maximize network lifetime. This is because in the heterogeneous architecture, a UWB node is responsible for collecting data from many sensor nodes in its range and forward data to the sink. The failure of one UWB node can thus affect many sensor nodes.

We proposed a reinforcement learning [10] based routing algorithm. A new reward function is proposed to consider energy efficiency, delay, routing failure and the local maximum problem in geographic routing. It is very suitable for sensor networks because the algorithm considers node energy and network lifetime. It has learning capability to adapt to the network topology changes. The proposed routing algorithm is evaluated in simulation and compared with GPSR.

## **1.5 Hierarchical Localization for Heterogeneous Sensor Networks**

Localization is the process of estimating locations of the nodes. In wireless sensor networks (WSN), locations of the nodes are essential to many network functionalities and applications. Node location is necessary for the geographic routing protocols discussed in the previous section. For event detection and target tracking, location information is indispensable.

GPS is the mostly used localization method. However, GPS has several limitations that affects its application in sensor networks. First, GPS can only be used in outdoor environment. Second, nodes in WSN need to be low cost and small in size. Installing GPS on sensor nodes increases node cost and size. The power consumption of GPS also reduces battery life of the nodes. For these reasons, many localization algorithms have been proposed specially for wireless sensor networks [11] without assuming using GPS on every sensor node.

Several measurable parameters can be used for localization. The most popular methods are Received Signal Strength (RSS), Time of Arrival (ToA), Angle of Arrival (AoA) and Time Difference of Arrival (TDoA) [12]. ToA method uses the signal propagation time to measure inter-node distance. It requires the nodes to be synchronized. TDoA method measure the difference between propagation times and does not need node synchronization. AoA method measures the angle of arrival of signals by using antenna array, which is not very practical for WSN.

In many proposed localization algorithms, the sensor nodes use a small number of anchor nodes with known locations to estimate their locations. These sensor nodes then become anchor nodes for other sensor nodes. However, WSN usually consists large number of nodes. One important problem with this method is error propagation. If RSS is used to estimate distance between nodes, the accumulation of distance estimation error and location error can cause significant localization error. In this thesis, we propose a hierarchical node localization method. The clusterhead nodes are placed randomly. They are equipped with UWB and RF radios. Their location can be estimated very accurately. The sensor nodes



use the UWB nodes as anchor nodes to estimate their location. The performance of the proposed localization scheme is evaluated through simulations.

## **1.6 Thesis Outline**

In summary, we study three important issues in sensor network: sophisticated node placement, energy-efficient routing and node localization. The remainder of the thesis is organized as follows. Chapter 2 provides review of some related previous work in the three areas. Chapter 3 discusses sophisticated node placement problem. Chapter 4 presents the reinforcement based geographic routing algorithm and simulation results. Chapter 5 deals with node localization. Finally, Chapter 6 provides conclusions of the thesis and suggests future work.

## CHAPTER 2

### LITERATURE REVIEW

#### 2.1 Node Placement

In wireless sensor networks, the sensor nodes have limited energy and communication range. Several algorithms have been proposed to place relay nodes or clusterhead nodes to extend network lifetime or maintain network connectivity. In some proposed deployment schemes, the nodes are assumed to be homogeneous. For example, in [13], an incremental self-deployment method was proposed to determine the locations of sensor nodes that are used as relay nodes in a homogeneous sensor network. One relay node is placed at each step in the most energy-efficient way to solve the nonlinear non-convex optimization problem. However, the method can possibly result in a sub-optimal deployment. In [14], an integer linear programming method was proposed for sensor deployment. The sensor nodes belong to two different types with different costs. The objective is to find the number and locations of each type of sensors to cover a grid area with minimum cost. In [15], sensor nodes are deployed as relay nodes to maintain network connectivity. The minimum node placement problem is modeled as a *minimum Steiner tree with bounded edge length problem*. Several approximation algorithms were proposed to solve the optimization problem. These algorithms do not consider a hierarchical architecture and the LN traffic constraints.

Several algorithms have been proposed for node placement in the two-tiered sensor network architecture [16]. For example, in [17], the problem of finding minimum number and locations of backbone nodes in a heterogeneous ad hoc network was formulated as a *constrained Steiner network problem*. Three approximation methods were proposed to solve

the problem. In [18], two algorithms were proposed for relay node placement in a two-tier sensor network. The problem is formulated as a *Minimum Geometry Disk Cover problem*. Different from the previous methods, this algorithm considers 2-connectivity to provide fault tolerance. Each sensor node is connected to two relay nodes to form a 2-connected network. The objective is minimize the number of relay nodes. Similarly, [19] studied fault-tolerant minimum relay node placement problem in a two-tiered wireless sensor network. The first step of the method solved a *minimum disk cover problem* so that the relay node can connect the sensor nodes. The second step solves the minimum Steiner tree problem to add Steiner nodes so that the relay nodes are connected. In these papers, the sensor nodes (LNs) are assumed to reach the relay nodes (SNs) in single hop.

In addition to the connectivity constraint, some research work is concerned about network life time. In [20], the joint problem of energy provisioning and relay node placement was studied to increase network lifetime. The locations of clusterhead nodes are assumed known. Additional relay nodes are deployed and their necessary energy provision are optimized. An efficient heuristic algorithm was proposed to solve the formulated mixed integer nonlinear programming problem. In [21], the problem of optimal placement of relay nodes is modeled as a *minimum set cover problem* and a recursive algorithm is proposed to solve the same. The objective is to deploy heterogeneous nodes so that the total network cost is minimized and the constraints of lifetime, coverage and connectivity are satisfied. The research in [22] and [23] extends the work in [21] and proposes a two-phase relay node placement algorithm. The relay nodes placed in the first phase are directly connected to sensor nodes. In the second phase, two strategies, Far-Near strategy and Max-Min strategy, were proposed to place more relay nodes to satisfy the lifetime and connectivity constraints.

However, the above papers consider a single hop from a LNs to SNs. This may not be cost effective, especially when there is a greater disparity between the capabilities of LNs and SNs.

## 2.2 Routing Algorithms in Sensor Network

Many routing protocols have been proposed for wireless sensor networks. Energy efficiency is an important consideration when designing routing algorithm for sensor networks. For example, in Geographical and Energy Aware Routing (GEAR) [8], a node uses both location and energy information of its neighboring nodes to select the next hop to forward a packet towards the target region. When the packet is in the destination region, restricted flooding is used to disseminate the data to guarantee geographic forwarding can terminate when node density is low. The cost function considers both distance and consumed energy at a node. However, GEAR is based only on local information and cannot consider the cost along the path. An energy and latency aware routing protocol is proposed in [9] for sensor networks. The algorithm tries to balance energy consumption from a global view. However, it needs the base station to send sensing tasks to all the nodes through flooding. Each node adds remaining energy and number of hops to the received flooding message. Such piggybacked information is used by nodes to find a path with optimal energy and latency. The sink node and the sensor nodes refresh the global information through periodic flooding. Similarly, a probabilistic energy-efficient routing protocol is proposed in [24]. The sink floods the sensing tasks through the network. The probability of choosing a neighboring node is proportional to its residual energy and inversely proportional to the routing cost.

However, flooding costs energy to all nodes in the network, even though they do not participate in routing. The nodes also need memory to keep the path and costs. The scalability of these routing protocols is a problem. Sasanka et al [25] proposed an energy band based routing protocol for wireless sensor networks. The algorithm tries to balance loads across the network to increase network lifetime. It divides the network into energy bands which are ordered based on energies. Each node uses the energy band information of its neighbors to forward packets. However, this protocol also uses flooding to calculate energy bands.

Geographical routing is very attractive for sensor networks and has been used in several papers. Geographical routing only needs local information and does not need to use flooding for path searching or collecting global information. GPSR [26] is one of the most successful geographic routing protocols for ad hoc networks. It tries to solve the local maximal problem, in which a node cannot find a neighboring node to the destination. However, GPSR is not designed for sensor networks and did not consider energy efficiency. The probabilistic geographic routing algorithm in [27] assigns probability to a node's neighbor based on the node's residual energy and link reliability estimated based on the sent and received "Hello" packets. It can improve network lifetime compared to GPSR. However, it only searches the region closer to the destination and never sends a packet backwards. This can avoid loops but the algorithm fails if a node has no neighbor closer to the sink. Seungjoon et al [28] proposed several link cost metrics including proximity, packet error rate, delay and power consumption. However, the cost metrics were used individually in simulations. The GEAR algorithm discussed above is also a geographic routing algorithm designed specially for sensor networks.

Reinforcement learning [10] is an unsupervised learning algorithm that can learn an optimal policy by interacting with the environment. Reinforcement learning was first proposed in [29] for the static network routing problem to minimize delay. The simulation results show that reinforcement learning-based routing algorithm can adapt to network topology and traffic changes. In [30], reinforcement learning is used to build an adaptive spanning tree with the sink as the root. The performance was evaluated in response to node failure and sink mobility. Reference [31] used the reinforcement learning algorithm proposed in [32] to speed up the learning process. However, these algorithms did not consider other performance metrics such as energy, network lifetime and delay, which are important for sensor networks. Dowling et al [33] proposed a model-based collaborative reinforcement learning routing algorithm for Mobile Adhoc Network (MANET). However, the process to build the state transition model can be very time and energy consuming. It mainly focused on network dynamics to improve throughput for ad hoc network and did not consider the network lifetime problem of sensor networks.

### **2.3 Node Localization in Sensor Network**

In recent years, there is tremendous research in node localization for sensor networks. A comparison of several localization algorithms can be found in [11]. The existing algorithms can be grouped according to several criteria: infrastructure-based and infrastructure-free methods; centralized and distributed methods; range-based and range-free methods; one-hop and multi-hop methods. There is some overlap among these categories and one algorithm can possibly belong to several categories.

In centralized algorithms, the entire network topology need to be collected at a certain powerful server node to estimate locations of all nodes. In [34], the localization problem is formulated a convex optimization problem. In this algorithm, known peer-to-peer communication in the network is modeled as a set of geometric constraints on the node positions. The global solution of a feasibility problem for these constraints yields estimates for the unknown positions of the nodes. One advantage of the method is that it can give optimal solution. MDS-MAP proposed in [35] is a centralized localization method based on multi-dimensional scaling (MDS). Similar to [34], it estimates node location based on connectivity. The method can generate relative maps that represent the relative positions of nodes when there are no anchor nodes. If there are 3 anchor nodes for 2-D and 4 anchor nodes for 3-D localization, MDS-MAP can determines the absolute coordinates of all nodes in the network. However, MDS-MAP does not work well on irregularly-shaped networks. The disadvantage of centralized algorithm is the need of a powerful central server node and the large communication and energy cost to collect the network topology information. It also has problem to deal with node mobility.

Distributed localization algorithms do not need the central processing node. Most localization algorithm for WSN belong to this category. In [36], a distributed algorithm called AHLoS (Ad Hoc Localization System) is proposed for sensor nodes to discover their locations. It require only a small fraction of nodes (beacons) to know their exact locations (either through GPS or manual configuration). The nodes use the ranging information and known beacon nodes to iteratively estimate their locations. Several multilateration algorithm are described in the paper: atomic, iterative and collaborative multilateration. The algorithm is tested with *Medusa* sensor nodes and a PC. The proposed cooperative

multilateration can improve the percentage of nodes that can determine their locations. Reference [37] proposed a robust distributed sensor node localization algorithm based on noisy distance measurements. The localization problem is formulated as a graph realization problem: given a planar graph with edges of known length, recover the position of each vertex. One drawback of the algorithm is that when node connectivity is low or measurement noise is high, the algorithm may be unable to localize a useful number of nodes. The algorithm is implemented on a physical network to assess its performance. The scalability of the algorithm is demonstrated through simulation. The ad hoc positioning system (APS) proposed in [38, 39] is a hybrid of two concepts: distance vector (DV) routing and beacon based positioning (GPS). The distance or other measurement information is forwarded in a hop by hop fashion independently with respect to each landmark. Each node eventually estimates its position based on landmark readings it gets. The method has been shown to work using simple connectivity, range measurement, angle measurement and a combination of them. The disadvantage of APS is that it requires a somewhat uniform distribution of landmarks. The DV-based method fails for highly irregular network topologies.

Most of the above methods need to anchor nodes with known location. Reference [40] proposed a distributed localization algorithm for ad hoc network without the support of GPS. Each node first estimate their position in a local coordinate system. These local coordinate systems are then combined to form a relative global coordinate system. However, this algorithm is designed for mobile ad hoc network and the required communication overhead and power consumption are not suitable for sensor network. A decentralized anchor-free algorithm (AFL) is proposed in [41]. In AFL, nodes start from a random



initial coordinate assignment and converge to a consistent solution using only local node interactions.

## CHAPTER 3

### HETEROGENEOUS SENSOR NETWORK NODE PLACEMENT

In the two-tiered sensor network architecture, the sophisticated nodes (SNs) placement needs to satisfy several constraint. In most previous research, the LNs are assumed to reach the SN via one hop. We consider a more practical case where the LNs can reach SN via multiple hops. The number of hops is used as a design parameter. The SN placement problem is first formulated as an optimization problem. We propose to use integer linear programming to solve the optimization problem and compared the results with those of greedy algorithm and genetic algorithm.

#### 3.1 Problem Definition

In the heterogeneous sensor network, nodes have different capabilities such as communication ranges and data traffic. Considering the notations given in Table 3.1, we make the following assumptions:

- Different LNs are used to measure different parameters such as temperature, humidity and video feed and generate different amounts of traffic.
- Two LNs can communicate with each other if  $d(LN_i, LN_j) \leq R_{LN}$ .
- An SN ( $SN_j$ ) have two interfaces: one to communicate with neighboring  $LN_i$ 's (if  $d(LN_i, SN_j) \leq R_{LN}$ ) and the other to communicate with neighboring  $SN_i$ 's (if  $d(SN_i, SN_j) \leq R_{SN}$ ).

Parameter	Description
$N(LN), N(SN)$	Number of lite nodes (LN) and sophisticated nodes (SN) respectively
$R_{LN}, R_{SN}$	Communication range of LN nodes and SN nodes respectively
$LN_i, SN_j$	The $i^{th}$ LN and $j^{th}$ SN respectively
$d(X_i, X_j)$	Distance between node $X_i$ and $X_j$
$h(X_i, X_j)$	The minimum number of hops required for the traffic from node $X_j$ to reach node $X_i$
$h_{max}$	The maximum hops of LN traffic that an SN will handle (design parameter)
$w_{hi}$	Weighing factor for $i^{th}$ hop LN traffic considered at an SN (design parameter)
$N(LN_{hi}, SN_j)$	The number of LN that are $i$ hops away from $SN_j$
$T(LN_i)$	The traffic generated (data rate) by node $LN_i$
$O_f$	The overprovisioning factor for traffic generated by LN
$C(SN_j)$	The capacity (data rate) of node ( $SN_j$ )
$C(LN_k, SN_j)$	The capacity of $SN_j$ offered to a single $LN_k$

Table 3.1: Table of parameters and notations.

- All the SNs need to be connected to the sink directly or via other SNs to send data to sink.
- The LNs can forward their data via multiple LN hops to the most feasible SN.
- The sensor network is static where the locations of LNs are known and do not change after deployment.

The objective is to determine the locations of SNs to satisfy the *traffic constraint* and *connectivity constraint*. The traffic constraint guarantees that all LN traffic can be handled by SNs. This can be satisfied if an LN can reach at least one SN directly or via other LNs. To avoid long paths from a LN to its closest SN, we limit the maximum number of hops ( $h_{max}$ ) from an LN to a SN. Mathematically, the set of LNs that a  $SN_j$  can serve is given

by  $Serve_{SN_j}$  in equation (3.1)

$$Serve_{SN_j} = \{LN_i | h(SN_j, LN_i) < h_{max}\} \quad (3.1)$$

where  $h_{max}$  the maximum number of LNs that can be connected to a SN. The greater the  $h_{max}$ , the more LNs can be served by a single SN and fewer SNs will be required.

In a sensor network, due to packet collision and intermediate node failure, the probability for a packet to reach a destination decreases with the increase of number of hops [42]. To reflect this in our constraints, we assume that the reliability of an SN to handle traffic generated by an LN,  $T(LN)$ , decreases with the number of hops to reach the SN, i.e.,  $w_{hi} > w_{h(i+1)}$ , where  $w_{hi}$  are weight indicating reliability of LN traffic reaching a SN in  $i$  hops. The weights can cause LNs further from SNs to have access to multiple SNs within  $h_{max}$  hops and hence increasing the robustness of the deployment.

The capacity of SNs is shared amongst LNs in  $Serve_{SN_j}$  based on weights assigned to LNs. Let  $C(SN_j)$  denote the capacity of  $SN_j$  in terms of data rate. This capacity is allocated to different LNs. LNs with higher weights will be allocated greater proportion of  $C(SN_j)$ . Let  $N(LN_{hi}, SN_j)$  denote the number of LNs that are  $i$  hops away from  $SN_j$ . We assume that traffic generated by each LN is equal to  $T(LN)$ . These nodes will all have weights  $w_{hi}$  assigned to them. The cumulative traffic generated by nodes  $i$  hops away will be:  $N(LN_{hi}, SN_j) \times w_{hi} \times T(LN)$ . If a  $LN$  cannot reach a  $SN$  directly, its traffic is forwarded by  $LNs$  closer to  $SN_j$ . Thus the effective traffic ( $E_i$ ) from all  $i^{th}$  hop LNs includes the traffic generated by  $LN_k$ s in  $Serve_{SN_j}$  and  $h(LN_k, SN_j) > i$ . This is given by

equation 3.2.

$$E_i = \sum_{k=i}^{h_{max}} (N(LN_{hk}, SN_j) \times w_{hk} \times T(LN)) \quad (3.2)$$

The proportion of  $C(SN_j)$  allocated to  $LN$  nodes which are  $i$  hops away is denoted by  $P_{hi}$  in equation 3.3.

$$P_{hi} = \frac{E_i}{\sum_{i=1}^{h_{max}} E_i} = \frac{\sum_{i=h_i}^{h_{max}} N(LN_{hi}, SN_i) \times w_{hi}}{\sum_{i=1}^{h_{max}} i \times N(LN_{hi}, SN_i) \times w_{hi}} \quad (3.3)$$

where  $P_{hi} = 0$  for  $hi > h_{hmax}$  indicating that no portion of  $C(SN_j)$  is allocated to LNs that are further than  $h_{max}$  hops away. Also,  $\sum_{i=1}^{h_{max}} (P_{hi}) = 1$ , indicating that the entire  $C(SN_j)$  is divided amongst the LNs in  $Serve_{SN_j}$ .

Based on the above equations, the capacity ( $C(LN_k, SN_j)$ ) of  $SN_j$  allocated to a single  $LN_k$  of  $i$  hops away is expressed in equation 3.4:

$$C(LN_k, SN_j) = \frac{1}{N(LN_{hi}, SN_j)} \times P_{hi} \times C(SN_j) \quad (3.4)$$

Considering the above constraints, the placement problem can be formulated as the following optimization problem. Find the minimum number and locations of all  $SNs$  denoted by

$$\min\{N(SN)\} \quad (3.5)$$

subject to,

$$\sum_{j=1}^{N(SN)} C(LN_k, SN_j) \geq O_f \times T(LN_k), \forall k = \{1, 2, \dots, N(LN)\} \quad (3.6)$$

and

$$\text{All } SN_j \text{ are connected} \quad (3.7)$$

The *traffic constraint* in equation (3.6) specifies that the capacity allocated to  $LN_k$  via different  $SN$ s, which is represented as  $C(LN_k) = \sum_{j=1}^{N(SN)} C(LN_k, SN_j)$ , is sufficient to handle the over provisioned traffic ( $O_f \times T(LN_k)$ ) of  $LN_k$ . The overprovisioning factor ( $O_f$ ) gives additional control over design by accommodating the possibility of higher LN traffic loads. The *connectivity constraint* in equation 3.7 specifies that the  $SN$ s have to form a connected network to forward data to the sink via multi-hops.

### 3.2 Binary Integer Linear Programming Solution

We formulate the node placement problem as a binary integer linear programming (BILP) problem. The LNs are randomly distributed throughout the RoI. The SNs are to be placed at grid intersection of an assumed grid over RoI. These placements are obtained by solving a BILP problem. Let  $N(G)$  be the total number of grid points. The binary variable  $X_i$  represents whether an SN is placed at the  $i^{th}$  point of this grid.

Consider the binary variable  $X_i$  where  $i = \{1, 2, \dots, N(LN)\}$ . Here  $X_i = 1$  indicates that a  $SN$  is placed at the location of  $LN_i$  and vice versa. Thus the problem in (3.5) and (3.6) can be re-written as

$$\min \left\{ \sum_{i=1}^{N(LN)} (X_i) \right\} \quad (3.8)$$

subject to,

$$C \times X \geq O_f \times T \quad (3.9)$$

where

$$C = \begin{bmatrix} C(LN_1, SN_1) & \cdots & C(LN_1, SN_{N(LN)}) \\ C(LN_2, SN_1) & \cdots & C(LN_2, SN_{N(LN)}) \\ \vdots & \ddots & \vdots \\ C(LN_{N(LN)}, SN_1) & \cdots & C(LN_{N(LN)}, SN_{N(LN)}) \end{bmatrix} \quad (3.10)$$

Here,  $C(LN_i, SN_j)$  indicates the capacity provisioned to node  $LN_i$  by  $SN_j$ . Although this would result in optimal solution (for the grid deployment) with respect to the traffic constraint, it does not consider the connectivity constraint of equation (3.7).

In order to find the locations of additional SNs to connect the SN nodes obtained by BILP method, we use a greedy implementation of Steiner tree problem as proposed in [43]. Unlike the implementation of [43], we add a single edge at a time and re-evaluate the various distances after adding additional Steiner nodes along the edge. This modification considers shorter links that may terminate on Steiner nodes. In the worst case, when the added edge terminates on original nodes, the performance of our greedy algorithm will be equivalent to the algorithm in [43]. Our algorithm is given as follows:

#### Greedy Steiner Tree Algorithm

1. Find connected components of SN graph.
2. Find the shortest edge required to connect two disconnected components.
3. Add Steiner nodes (additional SNs) along the edge to connect the components.
4. If resultant graph is disconnected
  - (a) Consider all SNs (from BILP and Steiner nodes) in the SN graph and GOTO Step 1.

(b) Else END.

The solution steps of using BILP can be summarized as:

#### BILP Algorithm

1. Find  $SN$  locations using BILP on the LN location. The resultant solution is called pure BILP solution.
2. Using greedy steiner tree solution place additional nodes to obtain connectivity amongst  $SN$  from pure BILP solution.

### **3.3 Simulation Results**

First, we show the SN placement solution by the algorithms for a single test case. The test case consist of 80 LNs distributed in an  $20 \times 20$  unit RoI. The placement results of BILP are compared with results of greedy algorithm and genetic algorithm proposed in [44].

The results of BILP, greedy algorithm and genetic algorithm are shown in Figure 3.1, 3.2 and 3.3 respectively. All the algorithms provide solutions that satisfy the traffic and connectivity constraints. Also, the GA obtains a solution with the smallest number of  $N(SN)$  with 32 nodes. But it is noticed in simulation that GA method needs a large number of iterations and a long time. In some simulations, genetic algorithm does not converge. The result shown here is the best result obtained. In contrast, BILP method can solve the problem very quickly.

To compare the performance of the three algorithms for different LN placement, the number of LNs is varied and the number of SNs are computed with the three algorithms. We first assume a grid network layout where the LNs are placed on the grid intersections.



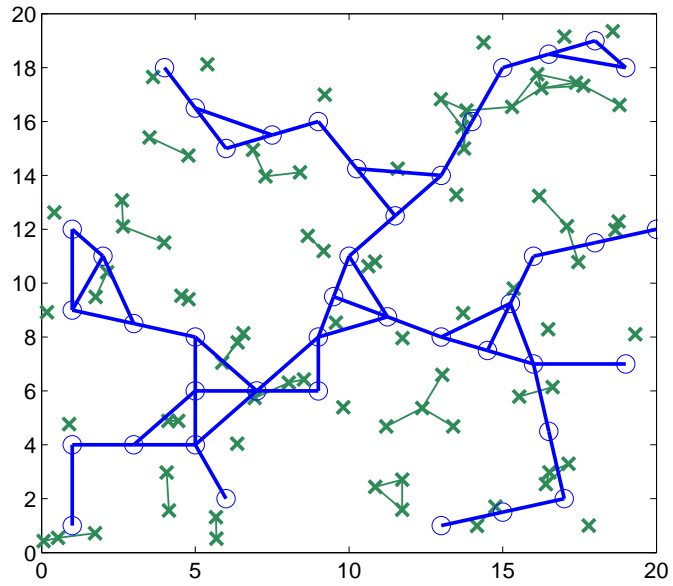


Figure 3.1: BILP results.  $N(\text{SN})=42$  nodes

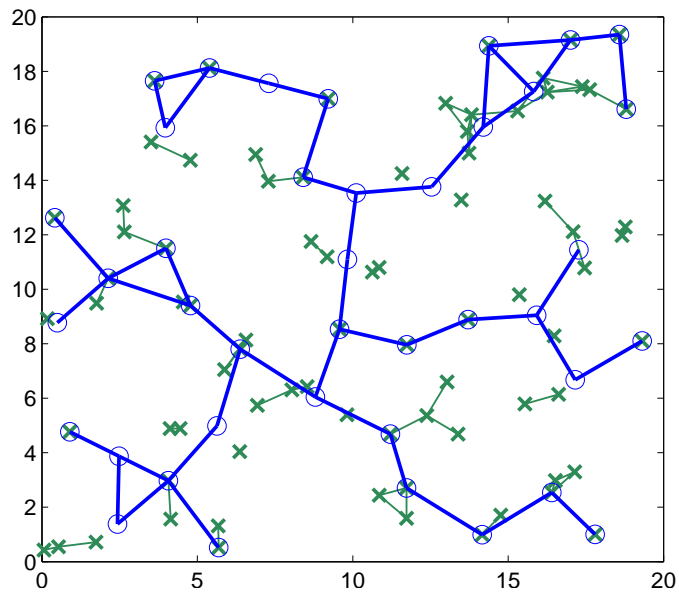


Figure 3.2: Greedy results.  $N(\text{SN})=40$  nodes

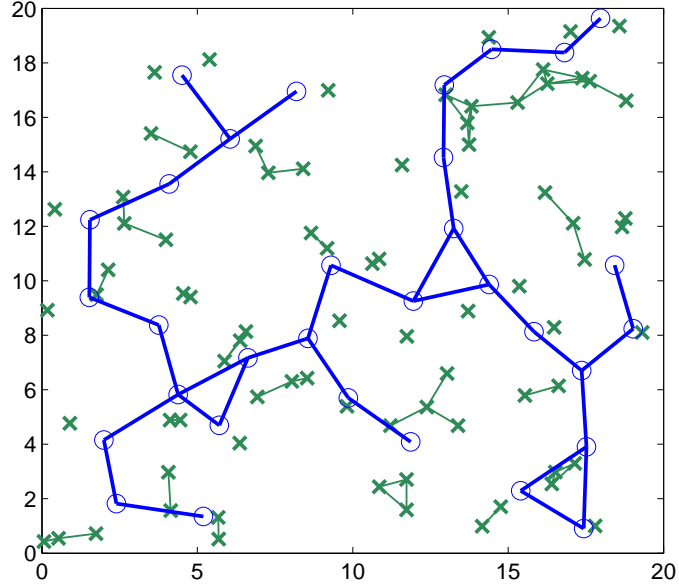


Figure 3.3: GA results.  $N(\text{SN})= 32$  nodes

This is not very practical in practice and we only use this for comparison.  $N(LN)$  is varied but takes only perfect square values due to the grid based arrangement of LNs. We use the BILP formulation and  $h_{max}$  value. It should be noted that as derived in [44], the SN placements from the BILP solution satisfied the connectivity constraint. The outcome of BILP thus represents the optimal solution considering SN placements over the grid. The GREEDY and GA described in [44] are also used to obtain SN placements. As seen from Figure 3.4, the BILP solution results in lowest number of SNs for the grids. The solutions obtained by GREEDY and GA algorithms also satisfy both the constraints but result in higher number of SNs for the same LN grid. The GA solution is closer to the BILP solution as compared to the GREEDY solution.

For more realistic application, a uniform random distribution of LNs is considered next. The number of LNs was varied from 40 to 200 nodes in steps of 40. Since LNs are randomly distributed, a single distribution cannot be used to characterize the performance

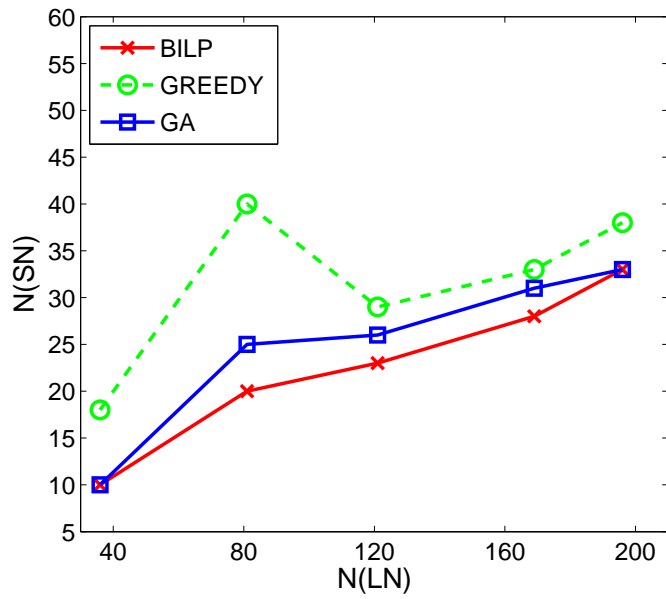


Figure 3.4: Performance of different algorithms. LN placed over a grid

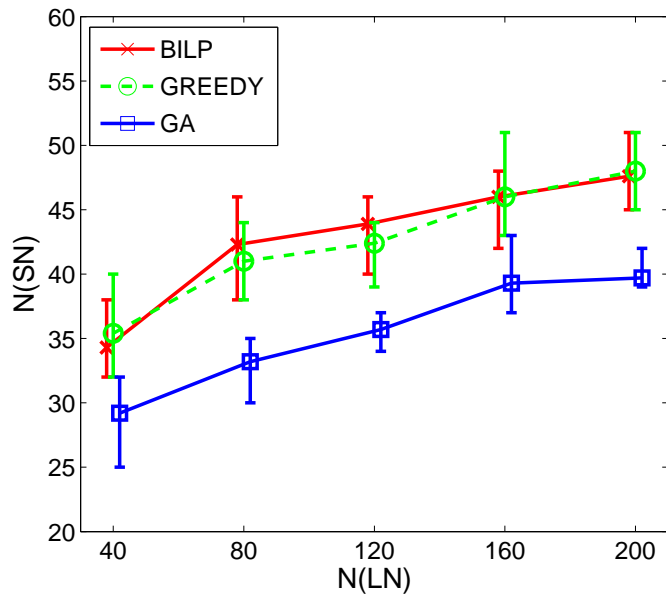


Figure 3.5: Performance of different Algorithms. LN (uniform) randomly distributed

of an algorithm. Hence, we consider 10 distinct cases of node distribution for every value of  $N(LN)$ . Thus, in all  $5 \times 10 = 50$  cases are considered. The three algorithms are validated over these 50 cases. The  $N(SN)$  along with the design parameters are shown in the Figure 3.5. Note that for a given algorithm and  $N(LN)$ , 10 cases are plotted as error bars. In the figure, each algorithm's performance represents the average number of SNs required for different values of  $N(LN)$ s. As seen from the figure, the GA outperforms BILP and GREEDY in this case.

In summary, we can see that BILP gives the best result if the LNs are placed on the grid points. However, for randomly distributed LNs, BILP solution will require Steiner nodes to ensure connectivity thus making the solution sub-optimal. The heuristic approaches do not guarantee optimal solution for both type of LN distributions. GA performs better than BILP for random LN distribution. However, it takes much longer to run as compared to the BILP implementation.

## CHAPTER 4

### REINFORCEMENT LEARNING BASED GEOGRAPHIC ROUTING ALGORITHM

Location-based geographic routing algorithm is very suitable for sensor networks because it needs only local information and is scalable. We propose a reinforcement learning based geographic routing algorithm for UWB sensor networks. The algorithm performance is evaluated with NS2 and compared with Geographic Perimeter Stateless Routing (GPSR). Simulation results demonstrate that the proposed algorithm can improve network robustness and network lifetime to be 75% to 213% better than GPSR. Although the algorithm is simulated only for UWB nodes, it can also be used for other sensor nodes if their locations can be estimated. The results of this chapter are presented at Globecom 2007 [45].

#### 4.1 Proposed Algorithm

In this section, we describe the proposed reinforcement learning based geographic routing protocol. The popular  $Q$ -learning algorithm [10] is used as the basic learning algorithm. To deal with several problems that may occur in geographic routing, we propose a reward function to consider network lifetime, routing failure and the local maximum problem.

In our heterogeneous network architecture, some sophisticated nodes are equipped with UWB transceivers to be used as clusterheads. These nodes aggregate data from sensor nodes and send them to the sink. In the proposed algorithm, we consider the routing problem among the UWB nodes. However, other sensor nodes can use the location of UWB nodes

to determine their locations and use the proposed routing protocol to send data to the associated UWB nodes.

We assume that each node can obtain its location and locations of the sink and its neighbors by using accurate ranging capability of UWB. Some localization techniques are discussed in the next chapter. Each node also knows its residual energy and its neighbors'. This can be achieved by exchanging short "hello" messages when the energy of a node is changed. A node saves the location and energy information of its neighbors in a table. We also assume that UWB nodes are stationary. If node mobility needs to be considered, the location information can be included in the "hello" message.

When a node needs to send data to the sink and cannot reach the sink directly, it uses the location and energy information of its neighbors to select the next hop. The objectives are to reduce delay and maximize network lifetime. The routing problem can be considered as a Markov decision process, which can be solved by using reinforcement learning. In reinforcement learning, the "agent" decides at each state which action to take based on its experiences. After taking an action, the agent gets a reward or cost from the environment. The agent uses the reward to update its policy. The advantage of reinforcement learning based routing is that each node does not need global network information, but can still approximate global optimality. Each node runs the reinforcement learning algorithm to select the next hop. Therefore, it is a distributed reinforcement learning algorithm.

Reinforcement learning algorithms can be classified into model-based and model-free methods. Model-based methods try to build state transition probability model of the environment. Model-free methods do not need a model and only uses experiences. One popular model-free learning algorithm is  $Q$ -learning. A  $Q$ -matrix is used to store the learned reward

for each state and action pair.  $Q(s, a)$  is the estimated reward for taking action  $a$  at state  $s$  and follow an optimal policy thereafter. The  $Q(s, a)$  value is updated as follows:

$$Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma \max_{a'} Q(s', a') - Q(s, a)] \quad (4.1)$$

where  $s$  is a state and  $a$  represents an action.  $r$  denotes reward and  $\gamma$  is a discount factor.  $\alpha$  is the learning rate. In our routing algorithm, when a node receives or generates a data packet, the state  $s$  of the packet is the current node. The action of the node is to choose one neighboring node  $a$  as the next hop based on the learned  $Q(s, a)$  values.  $Q(s, a)$  is the estimated total reward from node  $s$  to the sink through neighboring node  $a$ . The  $Q(s, a)$  value is then updated using equation (4.1) based on the reward. The state  $s$  is deterministic at each node. Each node therefore only needs to store a vector of the  $Q$  values of its neighbors.

The most important consideration to successfully achieve the desired routing performance is to define the reward function. In sensor networks, we are interested in improving network lifetime by uniformly distributing energy consumption and reducing packet delivery delay. We propose a reward function as in equation (4.2). It considers several routing scenarios to improve network lifetime and packet delivery ratio. The purpose of each item of the function are explained.

$$r = \begin{cases} \beta * adv/adv_{avg} + (1 - \beta)E_r/E_l & \text{if next hop is not sink} \\ R_C & \text{if next hop is sink} \\ -R_D & \text{if next hop is not found} \\ -R_E & \text{if nexthop is found but remaining energy is low} \end{cases} \quad (4.2)$$

The first item in equation (4.2) combines normalized advance to the sink through the next hop and normalized energy of the next hop.  $adv = d(i, k) - d(n, k)$  is the advance of the packet at node  $i$  to the sink  $k$  by choosing neighboring node  $n$  as the next hop.  $adv_{avg} = \sum_{j=1}^m |adv_j| / m$  is the averaged absolute advance through all neighboring nodes. The absolute value is used because it is observed through simulations that symmetric neighbors of a node can result in zero average advance and cause division by zero problem.  $E_r$  is residual energy of the selected neighboring node and  $E_I$  is its initial energy. As a result, more reward is assigned when choosing a neighboring node with large relative advance and more residual energy. The second term in equation (4.2) is the reward if a node can reach the sink in one hop. We assume the sink has unlimited energy. If a node can reach the sink directly, the reward is a constant  $R_C$ .

The third item in the above reward function is to solve the "hole" problem in geographic routing [28]. Most geographic routing protocols use greedy forwarding algorithm. Greedy algorithm sends the packet to the next hop which can lead to the most positive advancement towards the destination. But greedy routing algorithm fails if a node cannot find a neighbor closer the sink. This problem can be caused by node deployment or lack of energy in some nodes. Many solutions based on graph theory or heuristics have been proposed [28, 46]. The planarization algorithms used in GPSR [28] assumes a unit-graph model, which can be violated in real networks and makes the algorithms fail. These algorithms are usually complicated and only consider distance as a metric. We propose a reinforcement learning based solution. When a node receives a data packet but cannot find a neighbor to forward the packet, it drops the packets and sends a negative reward to the sending node to indicate



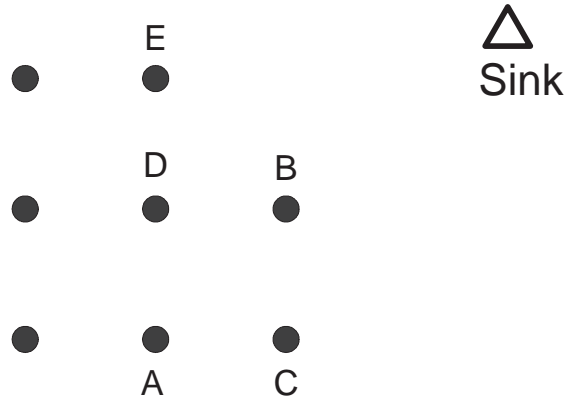


Figure 4.1: Illustration of "hole" problem

forwarding failure. The sending node will try to select other neighbors to re-send the packet. This process is illustrated in Figure 4.1.

When node  $A$  needs to send a packet to the sink, it compares the  $Q$  values of its neighbor nodes. For instance, it selects  $B$  as the next hop and sends the packet. Node  $B$  compares the distance of its neighbors to the sink and finds no neighbor is closer to the sink. It drops the packet and sends a negative reward ( $-R_D$ ) to node  $A$ . Node  $A$  will select another neighbor (for example  $D$ ) based on the  $Q$  values as the next hop and re-send the packet. Node  $A$  learns that node  $B$  has no neighbor closer to the sink and will not send following packets to node  $B$ .

This scheme solves the problem when a node (such as  $B$ ) needs to forward packets for other nodes (node  $A$ ) but cannot find a neighboring node closer to the sink. However if a node near the hole (node  $B$ ) needs to send packets to the sink, it has to send packets to a neighboring node whose distance is longer to the sink. For example, in Figure 4.1, if node  $B$  itself has packet to send to the sink, it has to send the packets to one of its neighbors

$(E, D, A, C)$ . In our algorithm, the node first searches for a neighboring node with shorter distance than itself to the sink as the next hop. If it cannot find such a neighbor, it uses its neighbors'  $Q$  values to select a neighbor farther to the sink than itself. It then updates the  $Q$ -value using equation (4.1). In order to prevent a loop, a node includes its location in the packet header. If a node receives a packet and has to search its neighbors farther to the sink, it compares the location of its neighbors and ignores the node with the same location as in the header. By using this scheme, a node will not forward packets for other nodes if it does not have neighboring nodes closer to the sink.

The fourth item in the reward function is used as a low-energy warning signal. In order to improve network lifetime, each UWB node has a low energy threshold. When its residual energy reaches the threshold and the node receives a packet to forward, it forwards the packets if it has sufficient energy, but also sends a negative  $(-R_E)$  reward to its neighboring nodes. The neighboring nodes will update their  $Q$  vectors and will avoid sending packets to the low-energy neighbor in future.

In reinforcement learning, there is a tradeoff between exploitation and exploration. Most of the time, the agent selects the next action with the largest  $Q$  value, however, it also needs to try other nodes to explore potential better path. Although we assume the nodes have fixed location, the energy of the nodes is always changing. We use  $\epsilon$ -greedy algorithm [10] in which a node selects a neighboring node with non-maximum  $Q$  value with small probability  $\epsilon$ . A node thus can estimate the cost of different path to the destination.

Based on the above description, the routing algorithm can be summarized in the following steps:

1. Initialization: Each node used the received "Hello" message from neighboring nodes and localization results to build a neighboring node table. The  $Q$  vector is initialized based on the initial energy and the relative distance to the sink.
2. If a node  $i$  has a packet to send, it compares each of its neighbors  $N_i$ .
  - (a) If the residual energy of a neighboring node is below the threshold, it is skipped.
  - (b) If  $N_i$  is farther to the sink than  $k$ , it is placed in a list  $L_1$  and temporarily skipped.
  - (c) If  $N_i$  is closer to the sink and its residual energy is higher than the threshold, its  $Q$  value  $Q_{N_i}$  is compared with the current largest  $Q$  value  $Q_{max}$ . If  $Q_{N_i}$  value is larger than  $Q_{max}$ ,  $N_i$  is a candidate next hop and placed in list  $L_2$ .
  - (d) Select the next neighbor and repeat 2a to 2c.
3. If  $L_2$  is not empty, generate a random number  $rnd$  between 0 and  $\epsilon$ . If  $rnd < \epsilon$ , the next hop is the node with maximum  $Q$  value. If  $rnd > \epsilon$ , a node is randomly selected from  $L_2$  as the next hop. This implements the  $\epsilon - greedy$  algorithm.
4. If  $L_2$  is empty, search list  $L_1$  to select the neighbor with the largest  $Q$  value as the next hop. The node also broadcasts a message to its neighbors that it does not have neighboring nodes closer to the sink.
5. Forward the packet to the next hop found in the above steps.
6. If next hop is not found, the node drops the packet.
7. The node calculates reward based on the reward function in equation (4.2) and updates the  $Q(s, a)$  value using equation (4.1).

## 4.2 Simulation

The proposed routing algorithm is implemented in NS2 to evaluate its performance in several scenarios. We first evaluate the performance with a network where nodes are placed on a grid. This simple configuration is used because the performance of the routing algorithm is predictable. We then simulated a network with randomly distributed nodes. For comparison, the performance of GRLR is compared with GPSR, which does not consider node energy. The implementation of GPSR by Ke Liu [47] is modified to calculate performance metrics and compare with those of GRLR.

### 4.2.1 Simulation with grid network

In this simulation, 100 UWB nodes are placed on a 10x10 grid shown as in Figure 4.2. Both GRLR and GPSR are simulated with the same network configuration and initial parameters. The distance between two adjacent nodes in the horizontal and vertical directions is 60m. The transmission range of UWB nodes is assumed to be 100m. In this configuration, a node can reach adjacent nodes in one-hop range. The sink marked with a triangle in Figure 4.2 is located at the upper right corner. The sink has unlimited energy and all other nodes have initial energy of 5J. A node updates its residual energy after sending or receiving a packet. Four nodes marked with circles at the lower left corner send Constant Bit Rate (CBR) data packets to the sink at a period of 0.5 second. In equation (4.1),  $\alpha = 0.7$  and  $\beta = 0.8$ , and the exploration probability is  $\epsilon = 0.2$ .

Several performance metrics, including the delivery ratio, network lifetime and average delay are used for evaluation. The lifetime of the network is defined as the time when the

Table 4.1: Performance metric for grid network

Protocol	Number of Packets Sent	Number of Packets Received	Delivery Ratio (%)	Average Delay (sec)	Network Lifetime (sec)	Average Path Length (# Hops)
GRLR	3406	3399	99.79	0.028	593.75	11.16
GPSR	1892	1891	99.94	0.022	339.34	9.10

energy of one node falls below a threshold. The simulation results are compared in Table 4.1.

By comparison, we can see that the network lifetime of GRLR is 74.97 % longer than GPSR. This causes GRLR sends more packet than GPSR to the sink. The delivery ratios of the two protocols are very close. The delay of each packet is the time difference between the receiving and sending time of the packet, and the average delay is calculated based on delay of all received packets. The average path length and delay of GPSR are shorter than GRLR. GPSR is an geographic routing algorithm originally proposed to solve the "hole" problem when greedy forwarding fails. For the above network distribution where there is initially no hole, GPSR tends to work as greedy forwarding and results in shorter delay and path length. But this caused the nodes along the path to exhaust their energy and lead to network partition sooner than GRLR. The energy distribution of GRLR and GPSR at the end of the simulation are shown in Figure 4.2 and 4.3, respectively. Lighter color indicates more residual energy. We can see that the residual energy distribution is more uniform in GRLR than GPSR. In Figure 4.3, the node energy along the diagonal paths became very small at the end.

In order to evaluate performance of GRLR when there is "hole" in network distribution, several nodes are removed from the grid and the node distribution is in Figure 4.4. In Figure

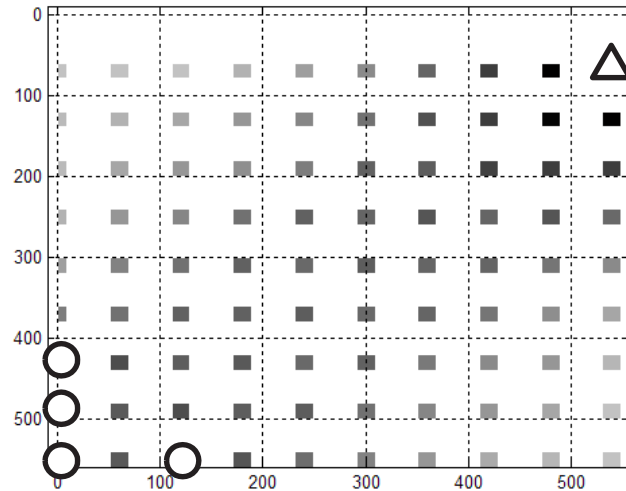


Figure 4.2: Energy distribution of GRLR after network partition

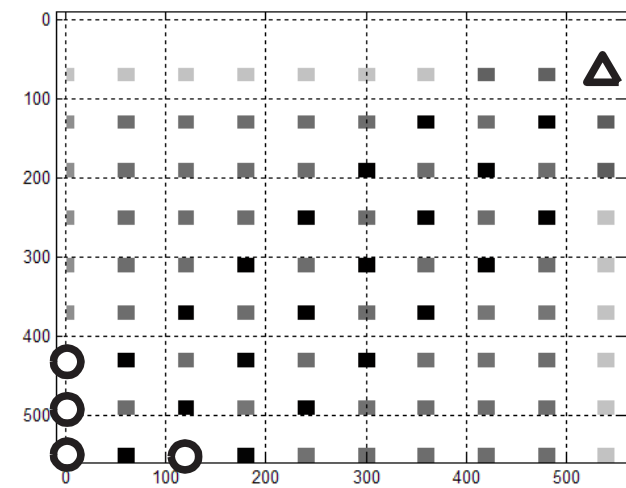


Figure 4.3: Energy distribution of GPSR after network partition

Table 4.2: Performance metric for grid network with hole

Protocol	Number of Packets Sent	Number of Packets Received	Delivery Ratio (%)	Average Delay (sec)	Network Lifetime (sec)	Average Path Length (# Hops)
GRLR	3541	3400	97.15	0.030	468.71	12.11
GPSR	Not available due to algorithm failure					

4.4, node  $A$  has no neighbor closer to the destination than itself. If greedy forwarding is used, the hole causes problem for packets from the source nodes, especially the node  $A$  and nodes that select  $A$  as the next hop. Table 4.2 lists the values of performance metrics. GPSR is also applied to the node distribution in Figure 4.4, but the simulation terminated after nodes start sending packets, so the result is not available. By examining the GPSR implementation code, it is found that the algorithm failed when it tried to perform planarization, which is a critical step for GPSR. But the results in Table 4.2 show that GRLR can still achieve high delivery ratio. Compared with the results of GRLR in Table 4.1 without hole, the average delay and path length are both longer. This is because the reward function in equation (4.2) caused the learning agents to learn to bypass the hole. The longer routing path also caused the network lifetime shorter than without hole. The energy distribution of GRLR at the end of simulation in Figure 4.4 shows many nodes, especially nodes near the hole and the sink, contributed to packet forwarding and the hole is bypassed.

#### 4.2.2 Simulation of Randomly Distributed Network

In real application, the UWB nodes are randomly distributed based on the distribution of sensor nodes. We evaluate the algorithm performance for a UWB network in which 100 UWB nodes are randomly distributed in a 600x600 area shown in Figure 4.5. Four nodes

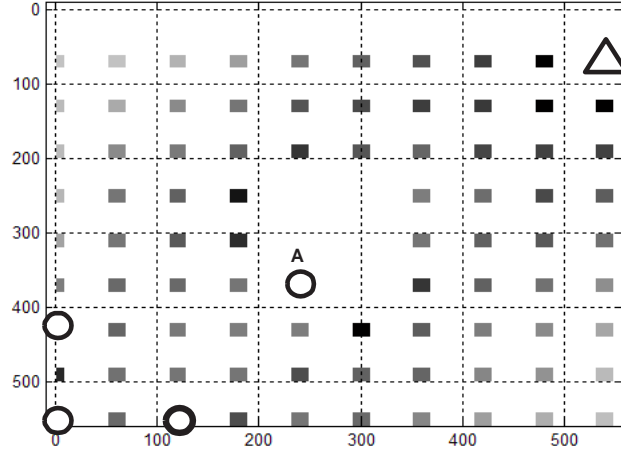


Figure 4.4: Energy distribution of GRLR after network partition with hole

marked with circle send packets to the sink marked with triangle at the upper right corner with CBR packet with interval of 0.5 sec. The performance metrics of GRLR and GPSR are compared in Table 4.3. The results indicate that the network lifetime of GRLR is 213% longer than GPSR. However, the delivery ratio of GRLR (95.16%) is much lower than GPSR (99.88%). Many packets are dropped when GRLR is used as routing protocol. To further examine the performance of GRLR during simulation, the performance metrics at several time steps are shown in Table 4.4.

From Table 4.4, we can see that before 400sec, the delivery ratio of GRLR is very close to that of GPSR. The average delay of GRLR increases with the progress of simulation. This is caused by the variations of node energy. The routing agent selects the next hop with more residual energy and causes longer path to the sink. Near the end of the network partition in GRLR, more packets are dropped due to insufficient energy of some nodes, but



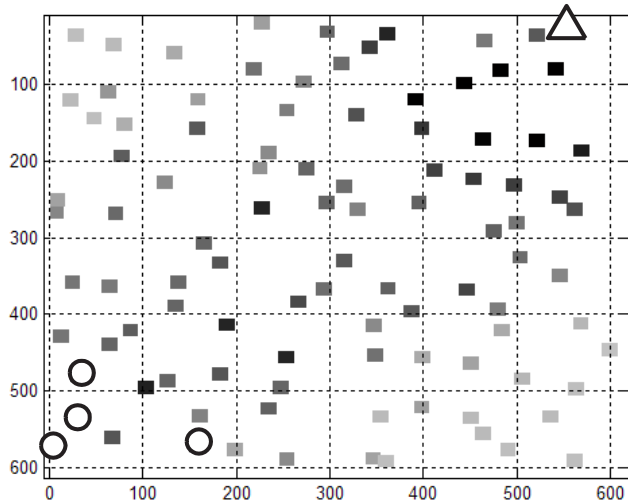


Figure 4.5: Residual energy distribution of GRLR after network partition

a large percentage (95.16%) of the packets are still forwarded to the sink before network partition. Even if the network is considered as partitioned at 400 sec, the network lifetime of GRLR is still 194.2% longer than GPSR.

The residual energy at the end of simulations is shown in Figure 4.5 and Figure 4.6 respectively. Lighter color indicates more residual energy. We can see in Figure 4.5 that a larger number of nodes have low residual energy at the end. In Figure 4.6, only a small number of nodes between the source nodes and sink have low residual energy. This comparison indicates that GRLR algorithm can adapt to energy variations and select the neighbor nodes to increase the network lifetime. GPSR does not consider energy and only select the shortest path, which cause the energy between the source nodes and sink to be exhausted and causes earlier network partition.

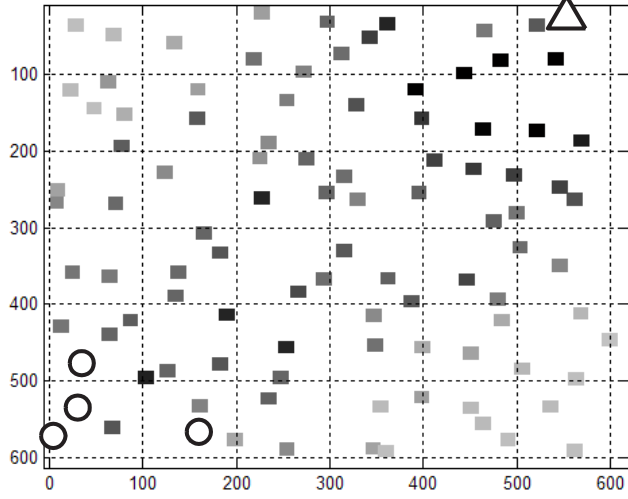


Figure 4.6: Residual energy distribution of GPSR after network partition

Table 4.3: Performance metric for random network

Protocol	Number of Packets Sent	Number of Packets Received	Delivery Ratio (%)	Average Delay (sec)	Network Lifetime (sec)	Average Path Length (# Hops)
GRLR	3205	3050	95.16	0.033	425.27	14.22
GPSR	896	895	99.88	0.024	136.24	9.72

Table 4.4: Performance metrics for GRLR at different steps

Time (sec)	Number of Packets Sent	Number of Packets Received	Delivery Ratio (%)	Average Delay (sec)	Average Path (Hops)
150	995	992	99.70	0.031	11.71
200	1395	1392	99.78	0.032	11.92
300	2210	2207	88.86	0.032	12.26
400	2999	2995	99.86	0.033	12.65
425	3205	3050	95.16	0.034	14.22

## 5.1 System Architecture

This chapter discusses the proposed localization methods for sensor networks. We assume a heterogeneous hierarchical network architecture as shown in Figure 5.1. The network consists of three types of nodes: anchor nodes (also called beacon nodes), clusterhead nodes and sensor nodes. An anchor node is a node with known accurate location, either through GPS or manual programming during deployment [48]. These anchor nodes are essential for the localization algorithm. The clusterhead nodes are more capable than ordinary sensor nodes. In this network architecture, clusterhead nodes are equipped with UWB and RF radios. For convenience, we call them UWB nodes. UWB nodes collect data from sensor nodes, do some processing such as feature extraction, data compression and send results to the sink through multiple hops. UWB nodes communicate with each other through UWB radios and with sensor nodes through RF radio. This hierarchical architecture is more scalable than flat network architecture [49]. It is also different from the clustering architecture in which clusterhead nodes are elected and rotated among sensor nodes [3].

UWB technology allows centimeter level accuracy in ranging as well as low-power and low-cost implementation of communication system [50]. It is suitable for wireless sensor networks [51]. However, due to size and cost considerations, it is not practical to equip every sensor node with UWB technology. In our network, the anchor nodes are equipped with UWB so that other UWB nodes can communicate with anchor nodes to estimate their

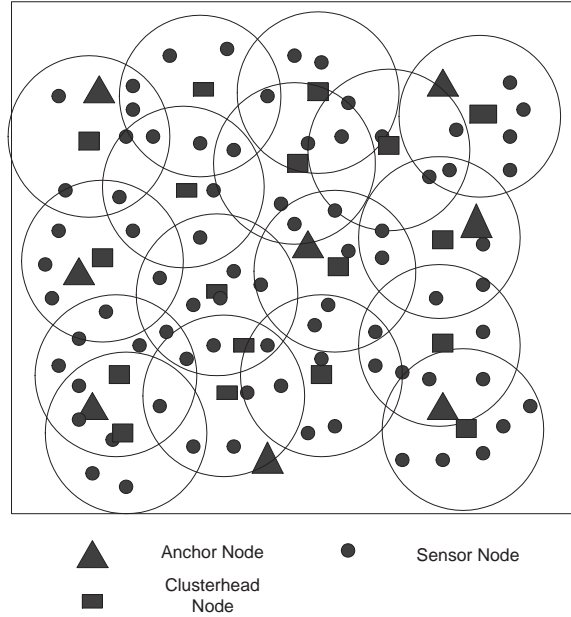


Figure 5.1: Illustration of system architecture

location. The distance between UWB nodes can be estimated from Time of Arrival (TOA) or Time Difference of Arrival (TDOA) measurements [50].

## 5.2 Algorithm Description

In our network, localization is performed in two steps. Locations of UWB nodes are first estimated by using the anchor nodes as references. If a UWB node has three or more neighboring anchor nodes which are not located along one line, its location can be estimated through multilateration. Once the location of the a UWB node is known, it becomes an anchor node. After locations of the UWB nodes are estimated, the sensor nodes use the UWB nodes as anchor nodes to estimate their positions. We assume that a UWB node can estimate its distance to anchor nodes and other UWB nodes with TOA. A sensor node can measure its distance to UWB node or other sensor nodes with RSS.

The location of a node covered by three or more anchor nodes can be estimated through multilateration. If the distance can be measured accurately, the location of the unknown node is the intersection of three circles as shown in Figure 5.2. However, distance measurement is often corrupted by noise and the three circles will not intersect at one point. The goal of the location estimation algorithm is to minimize the localization error. Assume the location of the unknown node is  $(x, y)$  and the  $i$ th anchor node is at  $(x_i, y_i)$ . If the estimated distance between the unknown node and the anchor node is  $\hat{d}_i$ , the error of distance estimation can be written as

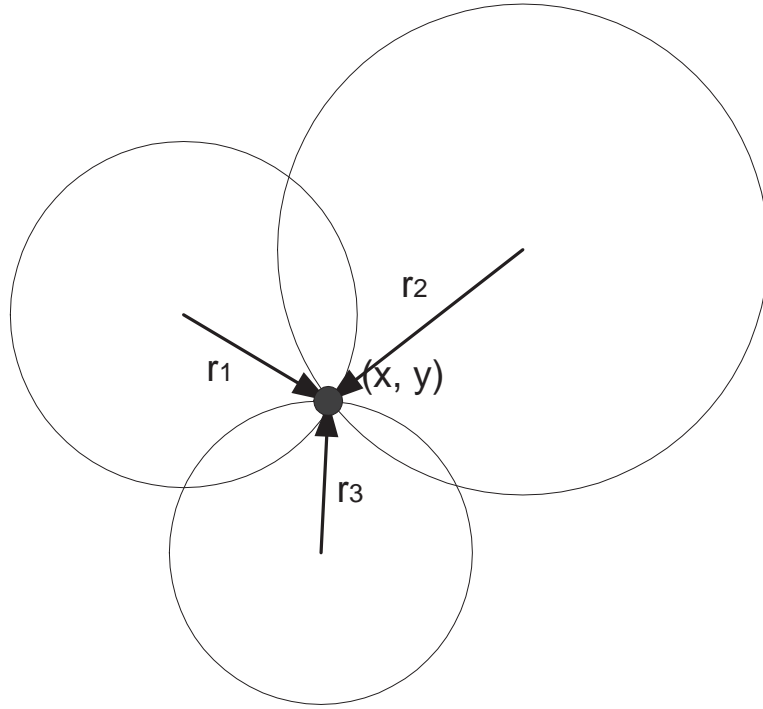


Figure 5.2: Illustration of triangulation

$$e_i(x, y) = \hat{d}_i - \sqrt{(x - x_i)^2 + (y - y_i)^2} \quad (5.1)$$

For convenience we call a node with unknown location an unknown node. If three or more anchor nodes are available to an unknown node, its location can be estimated by minimizing the sum of squared distance measurement error expressed in equation (5.2)

$$J(x, y) = \sum_i^n e_i^2(x, y) \quad (5.2)$$

The methods used to solve the above optimization problem can be classified as iterative or non-iterative methods [50]. In the following, we present derivation of the two methods.

### 5.2.1 Non-Iterative Method

The non-iterative method is simple and easy to implement. In equation (5.1), let  $e_i(x, y) = 0$ , we get:

$$\sqrt{(x - x_i)^2 + (y - y_i)^2} = \hat{d}_i \quad (5.3)$$

Squaring both sides gives:

$$x_i^2 + y_i^2 + x^2 + y^2 - 2x_i x - 2y_i y = \hat{d}_i^2 \quad (5.4)$$

Subtracting equation (5.4) for  $i = 1$  from (5.4) for  $(i = 2, \dots, n)$ , we can eliminate the non-linear term  $x^2 + y^2$  and obtain a set of linear equations.

$$\begin{aligned} (x_2^2 - x_1^2) + (y_2^2 - y_1^2) + 2(x_1 - x_2)x + 2(y_1 - y_2)y &= \hat{d}_2^2 - \hat{d}_1^2 \\ &\vdots \\ (x_n^2 - x_1^2) + (y_n^2 - y_1^2) + 2(x_1 - x_n)x + 2(y_1 - y_n)y &= \hat{d}_n^2 - \hat{d}_1^2 \end{aligned} \quad (5.5)$$

Reordering the terms, we can obtain a set of linear equation of  $x$  and  $y$  in the following format

$$AX = b \quad (5.6)$$

where

$$A = 2 \begin{bmatrix} x_1 - x_2 & y_1 - y_2 \\ \vdots & \vdots \\ x_1 - x_n & y_1 - y_n \end{bmatrix}$$

$$X = [x \ y]^T$$

$$b = \begin{bmatrix} (\hat{d}_2^2 - \hat{d}_1^2) + (x_1^2 - x_2^2) + (y_1^2 - y_2^2) \\ \vdots \\ (\hat{d}_n^2 - \hat{d}_1^2) + (x_1^2 - x_n^2) + (y_1^2 - y_n^2) \end{bmatrix}$$

For  $n \geq 3$ , this can be solved using a least square approach:

$$X = (A^T A)^{-1} A^T b \quad (5.7)$$

### 5.2.2 Iterative Approach

Iterative method starts with an initial solution and the variables are estimated iteratively. The iterations stop when the required accuracy or predefined number of iterations is reached. We derive the iterative method using Taylor series linearization to solve the optimization in (5.2).

Expanding equation (5.1) with Taylor series around an initial estimation and retain the first two terms gives:

$$e_i(x_0, y_0) \approx \hat{d}_i - (d_0 + \frac{x_0 - x_i}{d_0} \Delta x + \frac{y_0 - y_i}{d_0} \Delta y) \quad (5.8)$$

where  $d_0 = \sqrt{(x_0 - x_i)^2 + (y_0 - y_i)^2}$ ,  $i = 1 \dots n$ .

The above minimization problem can be solved with least square method to obtain  $\Delta x_0$  and  $\Delta y_0$ . The estimates of  $(x_0, y_0)$  can then be updated according to

$$x_0 = x_0 + \Delta x, y_0 = y_0 + \Delta y \quad (5.9)$$

Repeat the above process until  $\Delta x$  and  $\Delta y$  are sufficiently small.

This method need to start with an initial estimation  $(x_0, y_0)$ . One intuitive estimate is the average coordinates of the adjacent anchor nodes.

$$x_0 = \sum_{i=1}^n x_i/n; y_0 = \sum_{i=1}^n y_i/n \quad (5.10)$$

### 5.3 Simulation Results

In this section, we evaluate the performance of the proposed localization scheme through simulations. The algorithm is implemented in Matlab. As described in the previous sections, localization is performed in two steps, UWB node localization and sensor node localization. The simulation results of both steps are presented. Effects of anchor node placement and several parameters such as number of anchor nodes are also demonstrated through simulations.



### 5.3.1 UWB Node Localization

The simulation setup is shown in Figure 5.3. We assume the network is in a 100 x 100 square area. The location of anchor nodes is important for performance of the localization algorithm. Multilateration performs best when unknown nodes are surrounded by anchor nodes [52]. For this reason, the 12 anchor nodes are placed uniformly along the border of the area. The distance between adjacent anchor nodes is 33.33 units. 75 UWB nodes are placed randomly with a uniform distribution in the area. The communication range of UWB is set to 30. Figure 5.4 shows the initial histogram of number of the anchor nodes in the communication range of the UWB nodes. We can see that only 8 ( $\approx 10\%$ ) UWB nodes are covered by 3 anchor nodes. These nodes are near the corners of the area. Some UWB nodes are covered by 1 or 2 anchor nodes. Some nodes near the center are not covered by any anchor nodes. This setup is used to test the idea of using some localized UWB nodes as anchor nodes for other UWB nodes. If a node can find 3 or more anchor nodes within its range, it estimates its location using multilateration. Otherwise, it waits until it has sufficient neighbors with known location. Another reason is that in sensor networks, it is preferred to place only a few anchor node. To simulate measurement error, zero mean Gaussian noise is added to the actual distance between UWB nodes and anchor nodes. The noise is assumed to be proportional to the actual distance and the measured distance is expressed in equation (5.11).

$$dm(u, a) = d(u, a) + d(u, a) * randn * \sigma \quad (5.11)$$

where  $\sigma$  is the percentage of noise.

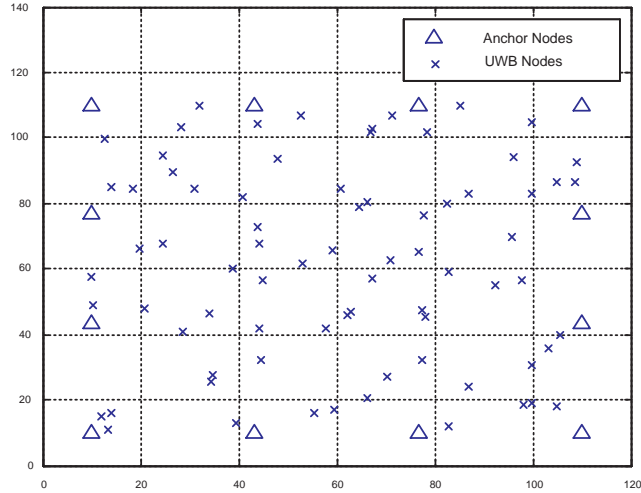


Figure 5.3: Localization of UWB nodes with anchor nodes

With some UWB nodes localized and become anchor nodes, the unknown nodes will have more adjacent anchor nodes. Figure 5.5 shows the histogram of the number of anchor nodes that a UWB can use to estimate its location. We can see that the number varies from three to twenty. This is much more than the number of anchor nodes at the beginning. However, relatively long-distance anchor nodes cause larger measurement error. It is also very computationally expensive to use large number of anchor nodes. To reduce localization error and computation, we set a limit to the number of anchor nodes actually used. If an unknown node has more than  $A_{max}$  adjacent anchor nodes, we sort the anchor nodes by distance and use only the  $A_{max}$  anchor nodes with shortest distance. In the following simulations, we use  $A_{max} = 5$ .

Firstly, we investigate the localization accuracy for different level of noise measurement. The measurement error in equation (5.11) is varied from 1% to 5%. Since the distance between UWB nodes can be estimated very accurately, larger measurement error is not

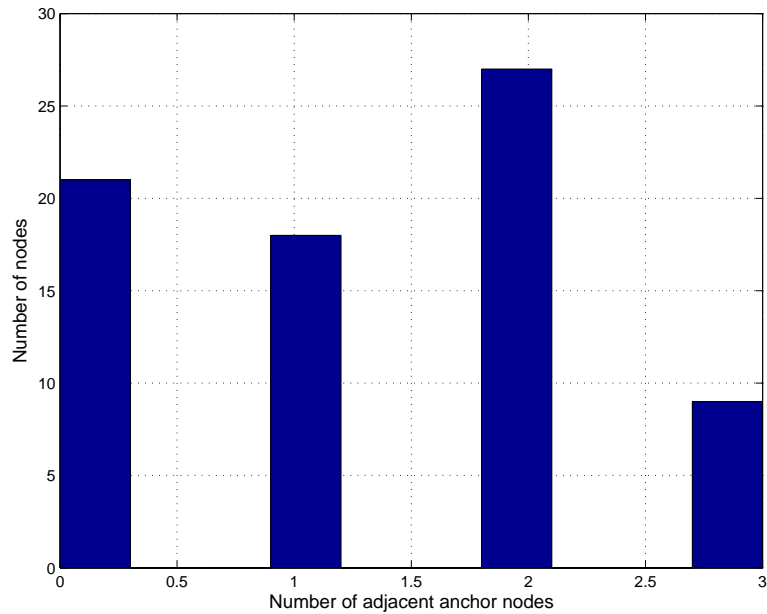


Figure 5.4: Histogram of initial adjacent anchor nodes

used. For each measurement error, the simulation is repeated 500 times and the average localization error of all nodes is calculated. The non-iterative algorithm is first used and the result is shown in Figure 5.6. The result indicates that the location error rises almost linearly with the increase of distance measurement error.

To study the effect of variance of the anchor nodes placement on localization error, four more anchor nodes are added inside the area as shown in Figure 5.7. The simulation is also repeated for different distance measurement errors. Figure 5.8 shows the comparison of the average location error of the using 12 and 16 initial anchor nodes. The result shows that location error can be decreased by using more anchor nodes. The difference is significant for larger distance measurement error. The introduction of more anchor nodes reduces the error caused by location error propagation and distance measurement error. More nodes

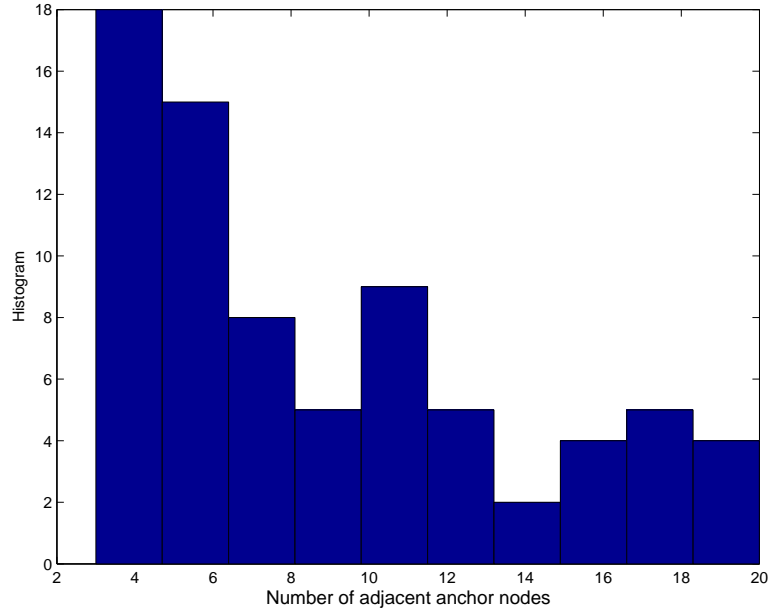


Figure 5.5: Histogram of number of anchor nodes before localization

can estimate their location by using the original anchor nodes instead of other UWB nodes as anchor nodes.

In section 5.2, we presented two methods to solve the nonlinear optimization multilateration problem. To compare their performance, the iterative method is used in simulation of the 16 anchor node configuration. The average location error of the two methods is shown in Figure 5.9. We can see that iterative method yields much smaller localization error. More importantly, the error does not increase significantly with the increase of distance measurement error. However, it should be noted that iteration method is more time-consuming to achieve the small error. In practice, the selection of the method depends on localization accuracy requirement and the computational speed of hardware used.

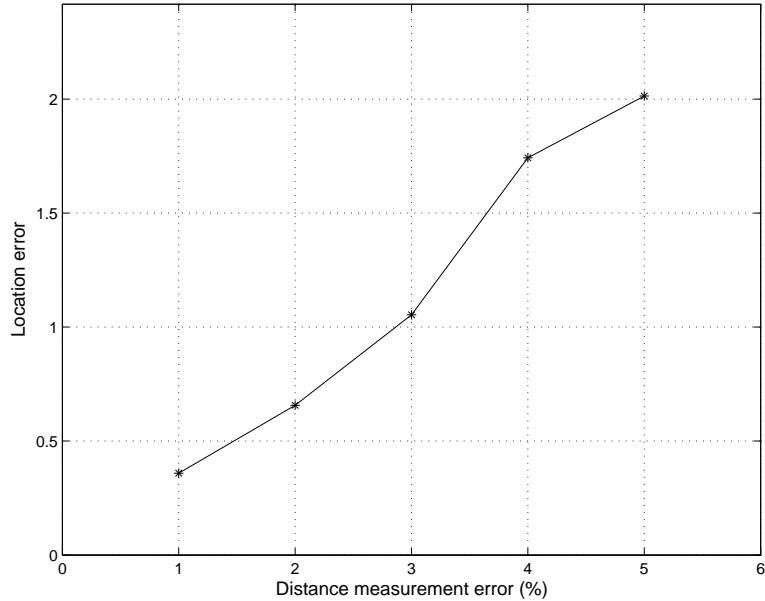


Figure 5.6: Average localization error as a function of distance measurement error

### 5.3.2 Sensor Node Localization

Once the UWB nodes are localized, sensor nodes can use the UWB nodes as anchor nodes to estimate their locations. The distance between a sensor node and UWB nodes can be estimated based on received signal strength (RSS). If a sensor node is covered by at least three UWB nodes, it can use the non-iterative or iterative method to estimate its location. However, it is possible that some sensor nodes are covered by fewer than three anchor nodes due to the randomness of node distribution. For these nodes, we employ the same method in the previous section. A sensor can use other localized sensor nodes as beacon nodes. This is illustrated in Figure 5.10. Sensor node  $A$  is covered by only two UWB nodes  $U_1$  and  $U_2$ , so its location cannot be uniquely determined. However, if sensor node  $B$ , which is covered by three UWB nodes is localized, node  $A$  can use its distances to the two UWB nodes and  $B$  to estimate its location. However, the distance estimated from RSS measurements is less

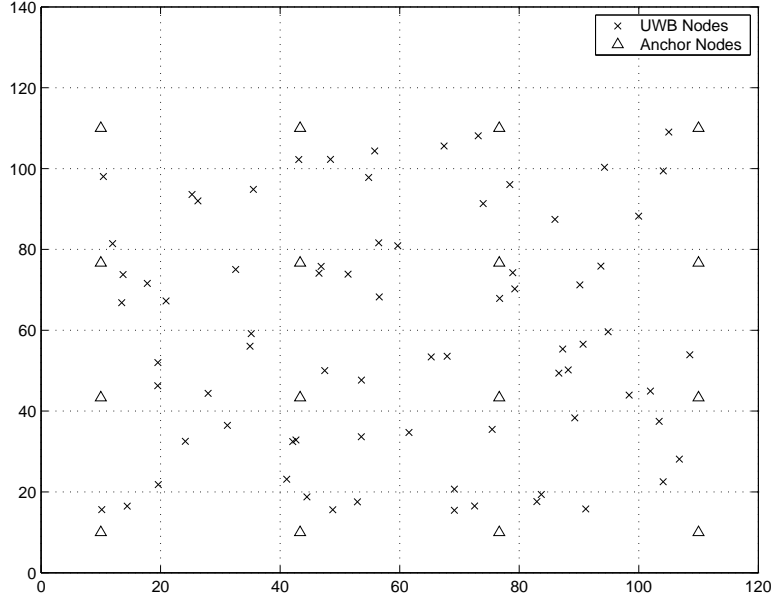


Figure 5.7: Node distribution with more anchor nodes

accurate than the distance from UWB estimation. The location estimation of sensor node B is also less accurate than UWB nodes. To express the confidence in location estimation, we use weighted least square method to estimate the location of this type of sensor node and add more weights to the UWB nodes. For non-iterative method, equation (5.6) becomes

$$WAX = b \quad (5.12)$$

where  $W$  is a weight matrix  $W = \text{diag}w_i$ . The solution becomes:

$$X = (A^TWA)^{-1}AW \quad (5.13)$$

The iterative method can be modified similarly.

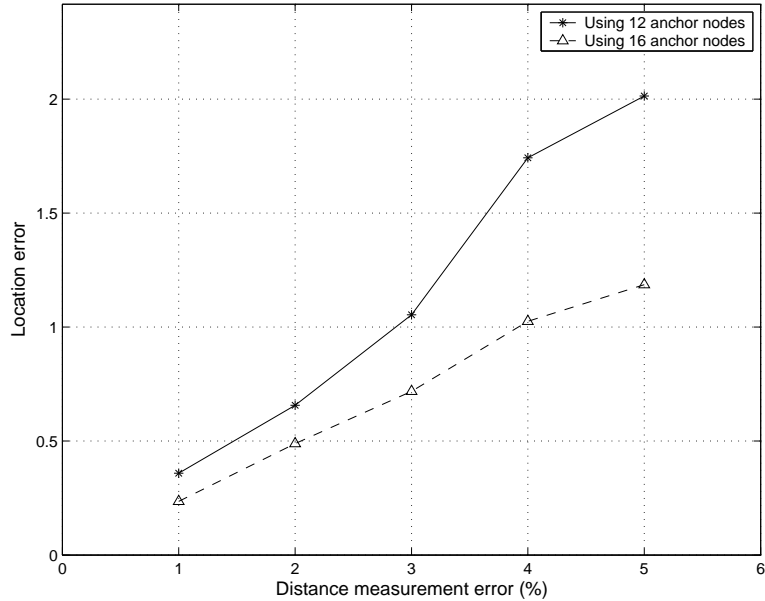


Figure 5.8: Average localization error for different number of anchor nodes

To evaluate the performance of the sensor localization method, 500 sensor nodes are placed randomly with uniform distribution in the 100x100 area together with 16 anchor nodes and 75 randomly distributed UWB nodes. The node distribution is illustrated in Figure 5.11. After the nodes are placed, the coverage of sensor nodes by UWB nodes depends on its communication range. To evaluate the algorithm performance, communication range of sensor nodes is varied from 11 to 18 in the simulation. The distance estimation error of UWB nodes and sensor nodes are fixed at 5% of the actual distance. All results are averaged over 200 trials. Figure 5.12 shows the average percentage of sensor nodes covered by at least 3 UWB nodes for different communication ranges. The result shows that when the communication range increases from 11 to 18, the percentage of sensor nodes covered by at least 3 UWB nodes increases from less than 60% to almost 100%. Figure 5.13 shows the corresponding localization error of sensor nodes. We can see that the error decreases as more

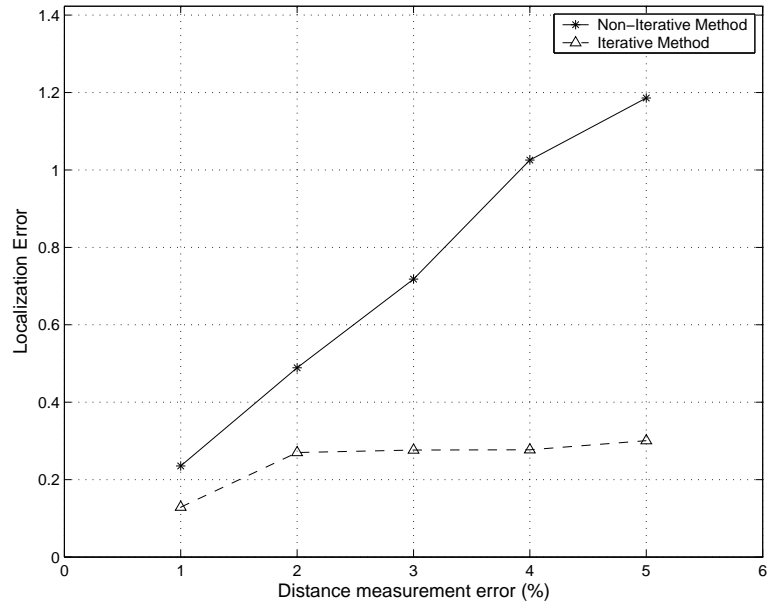


Figure 5.9: Error analysis using non-iterative and iterative methods

sensor nodes are covered by UWB nodes. If a sensor node can use adjacent UWB nodes to estimate its location, the accuracy is affected by its distance measurement error and the location accuracy of UWB nodes. However, if a sensor node is not covered by sufficient UWB nodes, it has to wait until its neighboring sensor node is localized to estimate its own location. This is why increasing the communication range results in smaller location estimation error. The error can be reduced by placing more UWB nodes and anchor nodes, although this will certainly increase the cost. The results of this chapter will be submitted to [53].



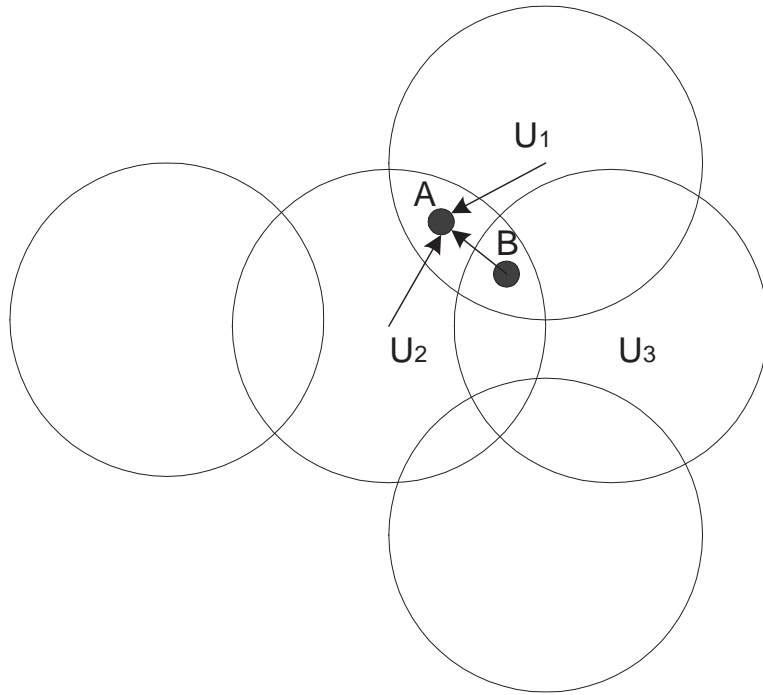


Figure 5.10: Localization of sensor nodes

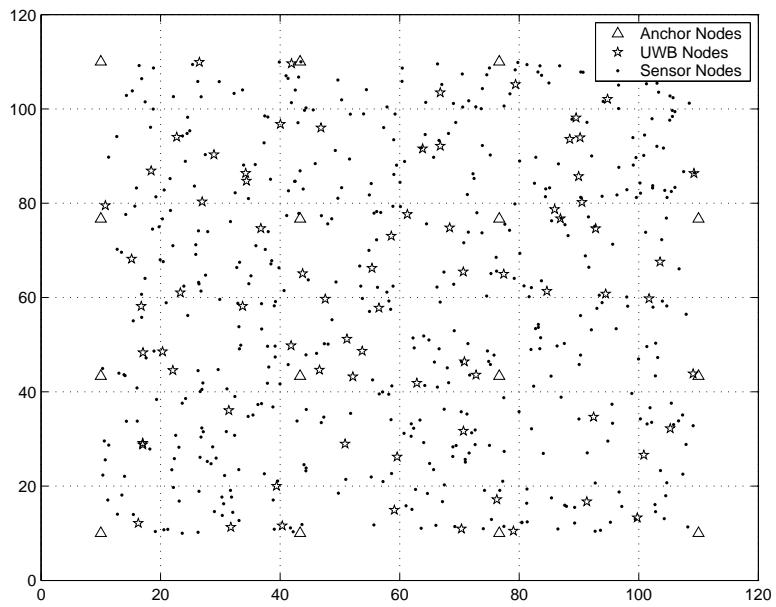


Figure 5.11: Distribution of three types of nodes

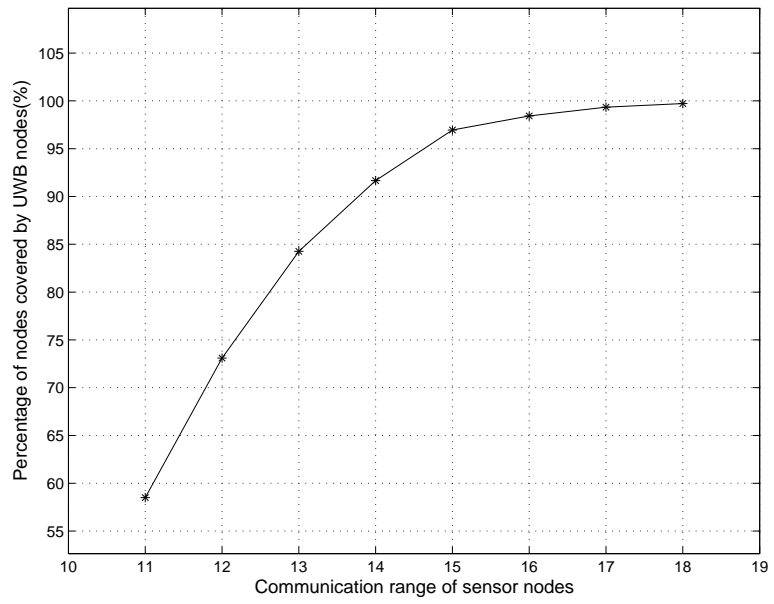


Figure 5.12: Coverage of sensor nodes by UWB nodes

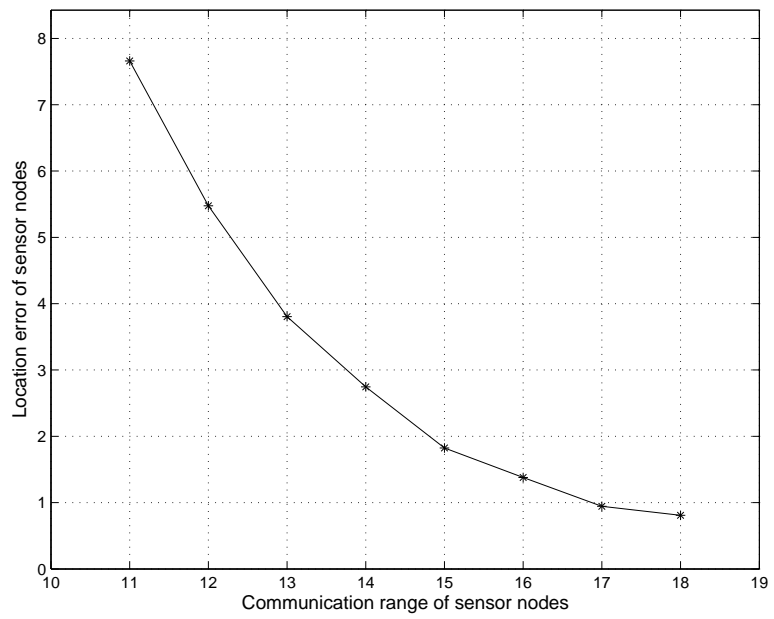


Figure 5.13: Variation of location error of sensor nodes for different communication ranges

## CHAPTER 6

### CONCLUSION AND FUTURE WORK

In this thesis, we investigate node placement, routing and localization problems in wireless sensor networks. These problems are very important for application of sensor networks.

The sophisticated node placement problem formulation considers lite node traffic and network connectivity constraint. The optimization problem is solved with integer linear programming. In the algorithm, we only require the sophisticated nodes to be connected in order to minimize the node number. As discussed in the literature review, some papers have considered 2-connected or k-connected network to provide fault tolerance. However, this has been proven to be a NP-hard problem. Some approximation algorithms have been proposed in the literature. This can be investigated in the future work.

A routing protocol based on geographic information and reinforcement learning is proposed. A new reward function is proposed to solve several routing problems. The performance is evaluated in NS2 and compared with GPSR. The algorithm shows superior performance to GPSR. It can still provide satisfactory performance when GPSR fails. In the hierarchical sensor network, the sensor nodes can also use the same routing algorithm to send data to the clusterhead if the locations of the sensor nodes can be determined. However, the current version of NS2 does not support hierarchical routing for wireless networks. This will be developed in the future. We can also consider using other network simulators such as OPNET to perform the simulations.

For node localization, we propose a two-step node localization method for two-tiered hierarchical heterogeneous sensor networks. The proposed method uses the high-accurate ranging capability of UWB nodes to estimate location of some nodes from some anchor nodes. The sensor nodes then estimate their locations by using UWB nodes as anchor nodes. The performance of algorithm is evaluated through MATLAB simulations. The effects of anchor node placement and node parameters are also investigated. The two-tiered network architecture is more scalable and practical for large scale sensor network. Only a small fraction of anchor nodes are needed in the network. The proposed localization scheme is very useful for this network architectures.

## BIBLIOGRAPHY

- [1] T. Arampatzis, J. Lygeros, and S. Manesis, "Survey of applications of wireless sensors and wireless sensor networks," in *Proceedings of the 13th Mediterranean Conference on Control and Automation*, Limassol Cyprus, June 2005.
- [2] K. Greene, "A wireless sensor city, a wireless-sensor network to report pollution and traffic comes to Cambridge, MA," April 2007. [Online]. Available: <http://www.technologyreview.com/Infotech/18533/>
- [3] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocols for wireless microsensor networks," in *Proceedings of the Hawaii International Conference on Systems Sciences*, January 2000.
- [4] X. Sheng, M. Guizani, R. Qiu, and T. Le-Ngoc, Eds., *Ultra-Wideband Wireless Communications and Networks*. West Sussex, England: John Wiley Sons Ltd, 2006.
- [5] K. Yu and I. Oppermann, "Performance of uwb position estimation based on time-of-arrival measurements," in *Proc. Joint UWBST IWUWBS, Kyoto, Japan, 2004*, pp. 400–404.
- [6] K. Cheng, W. H.C. So, and Y. Chan, "Least squares algorithms for time-of-arrival-based mobile location," *IEEE Trans. Signal Processing*, vol. 52, pp. 1121–1128, 2004.
- [7] I. Oppermann, L. Stoica, A. Rabbachin, Z. Shelby, and J. Haapola, "Uwb wireless sensor networks: Uwen - a practical example," *IEEE Communications Magazine*, vol. 42, no. 12, pp. 27–32, 2004.
- [8] Y. Yu, R. Govindan, , and D. Estrin, "Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks," UCLA Computer Science Department, Technical Report UCLA/CSD-TR-01-0023, May 2001.
- [9] P. Zeng, H. Yu, and W. Liang, "A new data gathering paradigm for wireless sensor networks," in *IEEE/Sarnoff Symposium on Advances in Wired and Wireless Communication*, 2005, pp. 85 – 89.
- [10] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.
- [11] D. Niculescu, "Positioning in ad hoc sensor networks," *IEEE Network*, vol. 18, no. 4, pp. 24–29, 2004.

- [12] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero, R. L. Moses, , and N. S. Correal, "Locating the nodes, cooperative localization in wireless sensor networks," *IEEE Signal Processing*, pp. 54–69, 2005.
- [13] W. Li and C. G. Cassandras, "A minimum-power wireless sensor network self-deployment scheme," *IEEE Wireless Communications and Networking Conference*, vol. 3, pp. 1897–1902, March 2005.
- [14] K. Chakrabarty, S. S. Iyengar, H. Qi, and E. Cho, "Grid coverage for surveillance and target location in distributed sensor networks," *IEEE Transactions on Computers*, vol. 51, no. 12, pp. 1448–1453, 2002.
- [15] X. Cheng, D. Du, L. Wang, and B. Xu, "Relay sensor placement in wireless sensor networks," 2001. [Online]. Available: [citeseer.ist.psu.edu/cheng01relay.html](http://citeseer.ist.psu.edu/cheng01relay.html)
- [16] G. Gupta and M. Younis, "Fault-tolerant clustering of wireless sensor networks," in *Proceeding of IEEE WCNC*, 2003, pp. 1579–1584.
- [17] Y. Xin, T. Guven, and M. A. Shayman, "Topology design for wireless ad-hoc networks with backbone support," in *The IEEE Conference on Information Sciences and Systems*, March 2005.
- [18] J. Tang, B. Hao, and A. Sen, "Relay node placement in large scale wireless sensor networks," *Journal of Computer Communications, Special Issue on Sensor Networks*, vol. 29, no. 4, pp. 490–501, 2006.
- [19] H. Liu, P. Wan, and X. Jia, "On optimal placement of relay nodes for reliable connectivity in wireless sensor networks," *Journal of Comb. Optim.*, vol. 11, pp. 249–260, 2006.
- [20] Y. Hou, Y. Shi, H. Sherali, and S.F.Midkiff, "On energy provisioning and relay node placement for wireless sensor network," *IEEE Transactions on Wireless Communications*, vol. 4, no. 5, pp. 2579–2590, 2005.
- [21] K. Xu, Q. Wang, H. Hassanein, and G. Takahara, "Optimal wireless sensor networks (wsns) deployment: Minimum cost with lifetime constraint," in *IEEE International Conference on Wireless And Mobile Computing, Networking And Communications*, 2005, pp. 454–461.
- [22] Q. Wang, K. Xu, H. Hassanein, , and G. Takahara, "Minimum cost guaranteed lifetime design for heterogeneous wireless sensor networks," in *Proc. IEEE IPCCC*, April 2005, pp. 599–604.
- [23] Q. Wang, K. Xu, G. Takahara, and H. Hassanein, "Locally optimal relay node placement in heterogeneous wireless sensor networks," in *IEEE Global Telecommunications Conference*, vol. 6, December 2005, pp. 3549–3553.

- [24] L. Gan, J. Liu, and X. Jin, "Agent-based, energy efficient routing in sensor networks," in *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, vol. 1, 2004, pp. 472 – 479.
- [25] S. Madiraju, R. K. C. Mallanda, A. Durresi, and S. Iyengar, "Ebrp: Energy band based routing protocol for wireless sensor networks," in *Proceedings of the Intelligent Sensors, Sensor Networks and Information Processing Conference*, 2004, pp. 67–71.
- [26] B. Karp and H. Kung, "Gpsr: Greedy perimeter stateless routing for wireless networks," in *In Proc. ACM/IEEE MobiCom, Boston*, 2000, pp. 243–254.
- [27] T. Roosta, M. Menzo, and S. Sastry, "Probabilistic geographic routing in ad hoc and sensor networks," in *Proc. of International Workshop on Wireless Ad-hoc Networks (IWVAN), London, UK*, 2005.
- [28] S. Lee, B. Bhattacharjee, and S. Banerjee, "Efficient geographic routing in multihop wireless networks," in *International Symposium on Mobile Ad Hoc Networking and Computing*, Urbana-Champaign, IL, 2005, pp. 230 – 241.
- [29] J. Boyan and M. Littman, "Packet routing in dynamically changing networks: A reinforcement learning approach," in *Proc. of NIPS*, 1999.
- [30] Y. Zhang and Q. Huang, "Adaptive tree: a learning-based meta-routing strategy for sensor networks," in *Consumer Communications and Networking Conference*, vol. 1, 2006, pp. 122–126.
- [31] P. Wang and T. Wang, "Adaptive routing for sensor networks using reinforcement learning," in *Proceeding of the sixth IEEE International Conference on Computer and Information Technology*, 2006.
- [32] M. Lagoudakis and R. Parr, "Model-free least square policy iteration," in *Proc of NIPS*, 2001.
- [33] J. Dowling, E. Curran, R. Cunningham, and V. Cahill, "Using feedback in collaborative reinforcement learning to adaptively optimize manet routing," *IEEE Transactions on Systems, Man and Cybernetics, Part A*, vol. 35, no. 3, pp. 360–372, 2005.
- [34] L. Doherty, K. S. Pister, and L. E. Ghaoui, "Convex position estimation in wireless sensor networks," in *IEEE INFOCOM*, 2001, pp. 1655–1663.
- [35] Y. Shang and W. Ruml, "Localization from mere connectivity," in *ACM Mobihoc*, June 2003.
- [36] A. Savvides, C.-C. Han, and M. B. Strivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*, 2001, pp. 166–179.

- [37] D. Moore, J. Leonard, D. Rus, and S. Teller, “Robust distributed network localization with noisy range measurements,” in *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, 2004, pp. 50–61.
- [38] D. Niculescu and B. Nath, “Ad-hoc positioning system,” in *Proceedings of IEEE Global Communications Conference, (GLOBECOM2001)*, 2001.
- [39] —, “Dv based positioning in ad hoc networks,” *Telecommunication Systems*, vol. 22, pp. 267–280, 2003.
- [40] S. Capkun, M. Hamdi, and J. P. Hubaux, “Gps-free positioning in mobile ad hoc networks,” in *34th Hawaii International Conference on System Science*, January 2001.
- [41] N. B. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller, “Anchor-free distributed localization in sensor networks,” MIT LCS, Tech. Rep. TR-892, April 2003.
- [42] S. Shakkottai, R. Srikant, and N. Shroff, “Unreliable sensor grids: Coverage, connectivity and diameter,” in *Infocom*, March 2003.
- [43] J. L. Bredin, E. D. Demaine, M. Hajiaghayi, and D. Rus, “Deploying sensor networks with guaranteed capacity and fault tolerance,” in *MobiHoc '05: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, 2005, pp. 309–319.
- [44] S. Pandey, S. Dong, P. Agrawal, and K. Sivalingam, “A hybrid approach to optimize node placements in hierarchical heterogeneous networks,” in *Wireless Communications and Networking Conference, IEEE WCNC*, March 2007.
- [45] S. Dong, P. Agrawal, and K. Sivalingam, “Reinforcement learning based geographic routing protocol for uwb wireless sensor network,” in *IEEE Global Communication Conference*, November 2007.
- [46] Y. Kim, R. Govindan, B. Karp, and S. Shenker, “Geographic routing made practical,” in *Proceedings of the 2nd Symposium on Networked Systems Design and Implementation*, Boston, MA, 2005.
- [47] K. Liu, “Gpsr: Greedy perimeter stateless routing code for version 2.26 or later,” 2007. [Online]. Available: <http://www.cs.binghamton.edu/~kliu/research/ns2code/index.html>
- [48] H. Chan, M. Luk, and A. Perrig, “Using clustering information for sensor network localization,” in *IEEE/ACM International Conference on Distributed Computing in Sensor Systems*, June 2005.
- [49] J. Pan, Y. T. Hou, L. Cai, Y. Shi, and S. X. Shen, “Topology control for wireless sensor networks,” in *MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking*, 2003, pp. 286–299.



- [50] S. Gezici, Z. Tian, G. B. Giannakis, H. Kobayashi, A. F. Molisch, H. V. Poor, and Z. Sahinoglu, "Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks," *IEEE Signal Processing Magazine*, vol. 70, July 2005.
- [51] K. Yu, J. philippe Montillet, A. Rabbachin, P. Cheong, and I. Oppermann, "Uwb location and tracking for wireless embedded networks," *Signal Process.*, vol. 86, no. 9, pp. 2153–2171, 2006.
- [52] K. Langendoen and N. Reijers, "Distributed localization in wireless sensor networks: a quantitative comparison," *Computer Networks*, vol. 43, no. 4, 2003.
- [53] S. Dong, P. Agrawal, and K. Sivalingam, "Hierarchical localization for heterogeneous sensor networks," *Under Preparation*.