Contents

List of Tables

List of Figures

Chapter 1

Introduction

The topology of wireless sensor network (WSN) is similar to ad-hoc network. Each sensor node in the network should communicate and cooperate with its neighbors to achieve the goal of task. For implementation stage, sensor nodes were carried in flight, sent to air, and downward scattered over desirably observing area. In such sensor network system, these nodes have memory, microprocessor, and power constraints. Each node collects, stores, and processes the sensed data, such as temperature, brightness, sound, and so on [1], and communicates with neighboring nodes to provide the environmental observation.

Because sensor nodes have small size, cheap, disposable, and failure-tolerant advantages, they are widely applied to several domains. For example, we can monitor certain places nobody garrison or difficult to reach like two poles of the earth, a marsh, or a jungle. We can also use them as emergent announcements when gas leaking, forest firing, or machine breakdown. Actually these nodes are often applied to military demand, e.g. detecting enemy and tracking unclear objects. In addition, in other aspects such as meteorology, agriculture, medicine, ecology, and so on, have had developing technology for recent years.

In order to perform these applications well, sensor nodes not only sense data but also realize where it is. The sensed data will be useful if it contains the location (or position) information. In Location Aided Routing [2], geocast [3], nodes use position information to determine routing direction for the destination zone. Smart Kindergarten [4] is used to observe children's physical and mental development by tracking their interactions with others and toys. Additionally, in hospitals, these nodes can be used to record the healthy status of the patient and examining manner of the doctor and nurse.

In recent years, Global Positioning System (GPS) [5] has been proven to be an integrated part of modern navigation. However, it is not suitable for all sensor nodes. For sensor networks, the defects of GPS are shown in the following:

1. GPS signals are degraded by the environment or jamming. Under tree canopies, inside cities between skyscrapers, and in indoor environments, GPS signals would be blocked. Especially in indoor, GPS signal is almost unavailable.

2. The power consumption of GPS is high. For a sensor node with limited energy, the energy will rapidly be run out.

3. The production cost of GPS is expensive if the sensed area needs a large number of sensor nodes.

4. The size of GPS is larger than sensor node. It is not suitable for sensor nodes to integrate with GPS receiver.

Although the defects of GPS are shown in previous descriptions, it remains a popular, accurate, and convenient solution for location system. Considering the advantages and disadvantages of GPS, few sensor nodes known their positions by combining GPS is a feasible solution. Sensor nodes without GPS also can obtain their positions by location algorithm. According to the ranging technology and the limited location information, the location algorithm can determine the positions of sensor nodes. The operation flow of location algorithm is shown in Figure 1.1.

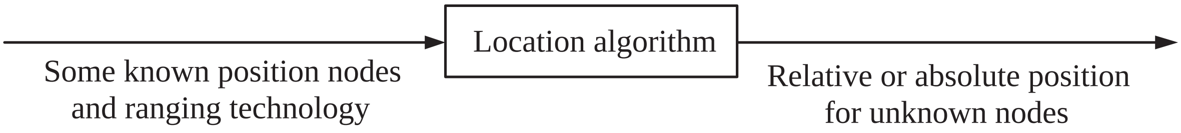
Some known position nodes and ranging technology
Location algorithm
Relative or absolute position for unknown nodes

Figure 1.1: Design engine for location algorithm

Based on the concept of location algorithm, there are some methods to be created. These location algorithms can be divided into two classes, one is centralized system and the other is distributed system. The centralized system has a central location server receiving the location query, calculating the location information, and replying it back to the query node. In contrast, the distributed system without location server utilize the location algorithm to calculate its position by itself. The distributed system is more feasible than the centralized system because of the following reasons:

1. The centralized system has time-synchronization requirement. When network topology changes, the positions of nodes can not be updated immediately.

2. The bottleneck of traffic load happens on the location server. Nodes near

the location server consume relatively large energy to forward position information.

3. System stability depends on the communication links of location server. If these links are broken, sensor nodes fail to determine their positions.

Summarized above mentions, there are some challenges for locating sensor nodes needing to be solved. The first challenge is the energy consumption problem. It is worthy to consider how to utilize the limited energy of sensor nodes to achieve position determination. The consumption origin contains activating RF Module (RFM) when they communicate with neighbors, and activating microprocessor when they compute estimated positions. The second challenge is the ranging error problem. This problem comes from anisotropic environment and multipath interference, and these factors lead to distance or angle measurement errors. The third challenge is sparse known (position) nodes. It causes inaccurate position estimations for unknown (position) nodes. The fourth challenge is the movement of sensor nodes. If unknown nodes move, they can estimate their position by computing repeatedly or routing update. If known nodes move, they can bring much more position information, because when they move to new positions, they can be considered as new known nodes. The feature of sensor network is low mobility, so this problem is not serious.

In this thesis, we call the known node equipping with GPS as "beacon". Our location algorithm includes two phases. In the first phase each node estimates its initial position by a modified DV-hop method [6]. In the second phase each node uses neighbors' positions and distances to neighbors as information, but neighbors may be known or unknown and distances to neighbors may be near or far. So we operate the multilateration with different weight values. The weight is defined as an error function of positions and distances to show the effect of data accuracy. These weighted error functions operate Minimum Square Estimation (MSE) to estimate unknown node position. Usually a node requires three neighbors to locate, but in the second phase we try to use only two neighbors to locate it, however this has to get help from the initial position of the first phase. From simulation results, we can observe when the average number of neighbors of nodes is 4, the average position error has only 15% radio range after refining in 20% beacon ratio.

The thesis is organized as follows. In chapter 2, we introduce the background and related work for location systems. In chapter 3, we give the weighted multilateration algorithm in detail. Simulation and analyses are shown in chapter 4. Finally, the conclusion and future work are given in chapter 5.

Chapter 2

Background and Related Work

In this chapter, we will introduce the existing popular location systems. According to the characteristic of location systems, signal propagation medium and ranging technology will be discussed respectively. After that, the basic location principle will also be discussed later. Finally, we will briefly introduce previously related work and present the consideration of proposed location algorithm.

2.1 Background

Before understanding how sensor network location operates, we first introduce the signal propagation medium for nodes' communication. In addition, we will talk about ranging technology for estimating distances or angles to neighbors, and basic location principle.

2.1.1 Signal Propagation Medium

In the sensor network, sensor nodes have to collect sensed data and exchange information with their neighbors. One certain medium for signal propagation between transmitter and receiver will be used. Some possible signal propagation media are shown as follows:

1. Radio Frequency (RF): In RADAR [7], it can receive signal strength from three fixed beacons, and then locate user position in the building. Its error comes from multipath fading, shadowing and antenna height.

2. Ultrasound: In Cricket [8], it uses radio and ultrasound signals to estimate distances to beacons, and then perform trilateration to locate node positions. The disadvantage of this medium is requiring additional hardware cost.

3. Infrared (IR): In Active Badge [9], everyone can wear a badge sending globally unique identifier, and data center collects information from fixed infrared sensors to locate. This accuracy of this medium is not good because it is easily influenced by the exterior visible spectrum.

2.1.2 Ranging Technology

Ranging means a node can use certain instruments to measure the distances or angles to neighbors. If one sensor node knows neighbors' positions, it may utilize some location techniques (introduced in 2.1.3) to estimate its position.

1. Received Signal Strength Indicator (RSSI): If a node knows transmitting power, receiving power, and one path loss model, it can estimate the distances to all neighbors. But these distance estimations are easily influenced by multipath fading, shadowing, and non-line of sight (NLOS). Though the accuracy of this technology is rather low, it is simple and used widely. RSSI

mainly uses RF as signal propagation medium. [10] adopts RSSI as ranging technology.

2. Time Of Arrival (TOA) or Time Difference Of Arrival (TDOA): Because propagation speed of RF is faster, it needs to collocate slower speed ultrasound. By measuring the relationship of sending time and receiving time between RF and ultrasound, and known ultrasound propagation speed, a node can estimate the distances to neighbors. Above is called TOA, AHLoS [11] adopts TOA as ranging technology. But it is called TDOA that a node measures Round-Trip Time (RTT) of RF to estimate the distances to neighbors, [12] adopts TDOA as ranging technology. It has the better accuracy that using these technologies to estimate distances, but the cost is requiring a microprocessor for powerful computation ability to reach high synchronization. TOA and TDOA may use RF, infrared, or ultrasound as signal propagation medium.

3. Angle Of Arrival (AOA): It is used to estimate the angles (or directions) to neighbors, this technology is also influenced by multipath reflection, and requires expensive antenna array. AOA mainly uses RF and ultrasound as signal propagation media. [13] adopts AOA as ranging technology.

2.1.3 Basic Location Principle

In the previous subsection, nodes can obtain some location information such as distances or angles to their neighbors. In this subsection, we will illustrate some simple and general basic location principles to perform location estimation via location information.

1. Trilateration: If a node knows the distances to three beacons and certainly knows these beacons' positions, that node can estimate its position, and this method is called trilateration. Much research about location system adopt trilateration. The basic concept is shown in Figure 2.1. In addition, if a node receives more than three beacons, it also can apply this principle called as multilateration.

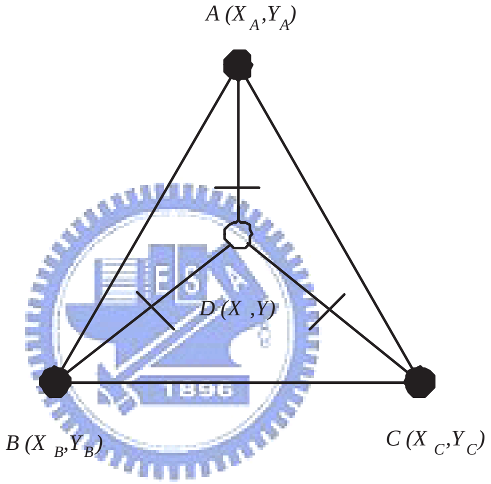

Figure 2.1: A,B,C are beacons, $\overline{DA}$,$\overline{DB}$,$\overline{DC}$ are known, thus D can estimate its position

2. Triangulation: If a node knows the angles to three beacons and certainly knows these beacons' positions, that node can estimate its position, and this method is called triangulation. The paper in [13] adopts this method and the basic concept is shown in Figure 2.2.

3. Maximum Likelihood (ML): A node use RSSI or others to measure the distances to neighbors, these distances are called "measured distances". By

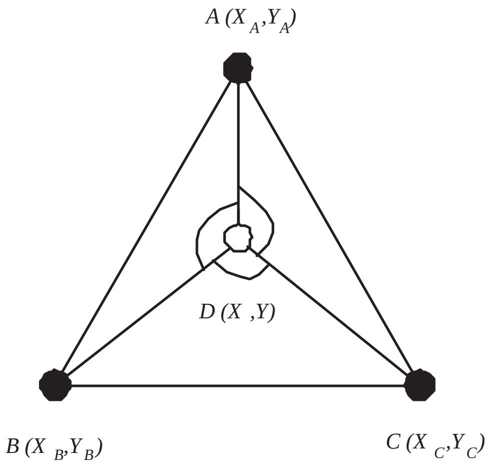

Figure 2.2: A,B,C are beacons, ∠BDA,∠BDC,∠ADC are known, thus D can estimate its position

means of unknown node position and neighbors' positions to get Euclidean distances to neighbors, these distances are called "estimated distances". By squaring and minimizing errors between measured distances and estimated distances to estimate node position, this method is called Minimum Square Estimation (MSE). The papers in [11][14] adopt this method and the basic concept is shown in Figure 2.3.

2.2 Related Work

For WSN applications, location is one of the important issues. In [15], a survey paper notes that early classical location systems may not be suitable for sensor networks. Some papers just talked about how to get global relative coordinate system. In [16], it randomly selects three neighboring nodes to assign one coordinate system, and then uses known distances between nodes to infer other

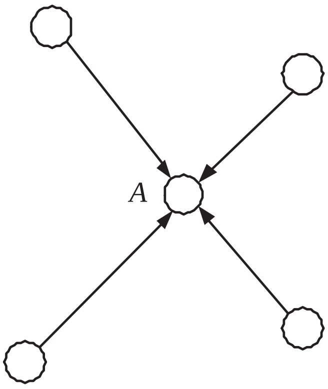

Figure 2.3: Node A uses MSE to estimate its position

nodes' coordinates by triangle relationship. In [10] and [17], they use Multidimensional Scaling (MDS) to get global relative coordinate system, and then map this system into absolute coordinate system by self-designed location algorithm.

Some papers use refinement after getting absolute coordinate system. In [10], it moves the anchor's (anchor is also called beacon in this thesis) estimated position to its real position, and then operates simultaneous movement to all unknown nodes, and the movement is the same as anchor's. In [18] unknown nodes estimate initial positions by Hop-TERRAIN, and then update their positions by interacting with neighbors and computing repeatedly.

According to the limitation of location system, it can be divided into two classes: indoor and outdoor. RADAR [7], Cricket [8], and Active Badge [9] are early classical indoor location system. They usually install some beacons in a building to locate user's position in the room. In contrast, Cellular [19] lets the base station transmit a signal, and then the mobile phone receives and reflects. The signal's TDOA generates the distance estimation, so operates multilateration to locate the mobile phone.

According to the operation model of location system, it can be divided into two classes: anchor-based and anchor-free. In anchor-based location system, it utilizes some anchor nodes to achieve location determination. In contrast, the location system without anchors perform location determination is called as anchor-free location system. In [16][20] nodes exchange local distance information with others to generate their relative coordinates. The transformation from relative coordinates to absolute coordinates was handed over to certain post-process methods.

Some incremental algorithms [11][13][21] also use beacons to perform the location estimation. The execution process radiates from the beacons to its neighbors and then these estimated neighbor nodes will become beacons to repeat this process. However, the location estimation error in incremental algorithm will be accumulated from one node to another. In addition, some concurrent algorithms [10][20][22] estimate and refine nodes coordinates simultaneously. The estimation error will be reduced by iterative and optimal computation.

2.3 The Consideration for Proposed Algorithm

According to previous mention, we propose a location algorithm that utilizes the RF as our signal propagation medium and the RSSI as our ranging technology. This is because that RF is an available signal for sensor network without additional hardware components. The consideration for ranging technology is similar to signal propagation medium. The RSSI can be measured from RF without additional antenna. It is the most common method and easy to obtain the distance between nodes and its neighbors. In our proposed location algorithm, it contains two phases to perform the location determination. In the first phase, the processing step is modified from [6] that uses the classical multilateration to estimate the initial absolute positions. In the second phase, estimated positions that obtain in the first phase will be refined by modified weighted multilateration. After finishing the two phases, nodes' locations can be obtained with reasonable and accurate results. Note that our proposed method belongs to anchor-based category because our method utilizes few beacons to supply absolute location information for their neighbors. In addition, unknown nodes concurrently and iteratively estimate their positions to reach global optimization.

Chapter 3

Weighted Multilateration Algorithm

In this chapter, we will introduce the proposed algorithm that extends the method in [18] with two phases. In the first phase, the DV-hop method in APS [6] is modified. Beacons broadcast their position information, and then those unknown nodes can get their initial positions. In the second phase, unknown nodes only exchange information with their neighbors and then estimate their location by multilateration with weighted coefficient to refine initial node positions. This is because neighbors' positions and distances to neighbors cause the effect of data accuracy. In the following, we will present the weighted multilateration algorithm in details.

3.1 The First Phase

The goal of this phase is to give the initial positions to unknown nodes. These location information will be exchanged between nodes in the second phase. In this phase, we adopt a modified DV-hop method and the processing steps are shown in the follows:

1. Beacons broadcast their positions in the network.
2. Beacons maintain the shortest paths to other beacons, and unknown nodes maintain the shortest paths to beacons.
3. Beacons receive information from other beacons to compute their one-hop distances, and then broadcast these values to unknown nodes.
4. Nodes receive information from at least three beacons, and use average one-hop distance multiplied by shortest path hop count to beacons to estimate distances to beacons. Finally they operate classical multilateration to estimate their positions.

Take Figure 3.1 for example, node 1, 2, and 3 are beacons and others are unknown nodes.

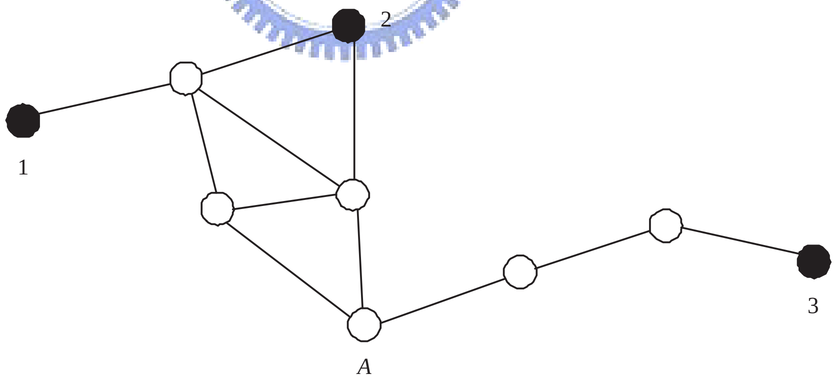

Figure 3.1: The example for the first phase

- From step 1 and step 2, beacons and unknown node A can know following information:

  - Real Euclidean distance between 1 and 2, $d_{12} = d_{21} = 40m$;
  - Real Euclidean distance between 2 and 3, $d_{23} = d_{32} = 75m$;
  - Real Euclidean distance between 1 and 3, $d_{13} = d_{31} = 100m$;
  - Hop count between 1 and 2, $h_{12} = d_{21} = 2$;
  - Hop count between 2 and 3, $h_{23} = d_{32} = 5$;
  - Hop count from between 1 and 3, $h_{13} = d_{31} = 6$;
  - Hop count from A to 1, $h_{A1} = 3$;
  - Hop count from A to 2, $h_{A2} = 2$;
  - Hop count from A to 3, $h_{A3} = 3$;

- From step 3, beacons can compute their one-hop distances:
  - Beacon1's one-hop distance= $\frac{d_{12}+d_{13}}{h_{12}+h_{13}} = \frac{40+100}{2+6} = 17.5$,
  - Beacon2's one-hop distance= $\frac{d_{21}+d_{23}}{h_{21}+h_{23}} = \frac{40+75}{2+5} = 16.43$,
  - Beacon3's one-hop distance= $\frac{d_{31}+d_{32}}{h_{31}+h_{32}} = \frac{100+75}{6+5} = 15.91$,

- From step 4, node A computes average one-hop distance and then estimates distances to beacons:
  - Average one-hop distance= $\frac{17.5+16.43+15.91}{3} = 16.61$;
  - Measured distance from A to 1, $d_{A1} = 16.61 \times 3 = 49.83$;
  - Measured distance from A to 2, $d_{A2} = 16.61 \times 2 = 33.22$;
  - Measured distance from A to 3, $d_{A3} = 16.61 \times 3 = 49.83$;

- Finally, node A operates classical multilateration to estimate its initial position as follows:

- Error functions between estimated distances to beacons and measured distances to beacons:

$$\begin{cases} f_{A1} = \sqrt{(x_A - x_1)^2 + (y_A - y_1)^2} - d_{A1} \\ f_{A2} = \sqrt{(x_A - x_2)^2 + (y_A - y_2)^2} - d_{A2} \\ f_{A3} = \sqrt{(x_A - x_3)^2 + (y_A - y_3)^2} - d_{A3} \end{cases} \tag{3.1}$$

- To estimate (x_A, y_A), we apply Minimum Square Estimation (MSE) to Equations 3.1:

$$minimizing \quad F = f_{A1}^2 + f_{A2}^2 + f_{A3}^2$$

- Now, we put the detailed process of getting estimation (x_A, y_A) into Appendix A.

3.2 The Second Phase

Based on the first phase, unknown nodes have got their initial positions. In this phase, nodes exchange information with their neighbors to refine their positions. When nodes receive information from at least three neighbors, they can apply certain ranging technology (e.g. RSSI) to measure distances to neighbors, and then use classical multilateration to estimate positions. However, positions and distances to neighbors have different accuracy. It is obviously that the position information from beacons is more accurate than that from other nodes. Similarly, according to the signal degression properties for ranging measurement, the position information from near node is more accurate than that from far node. Hence, we define a parameter, "weight", that is the production of neighbor's position weight and distance weight. Position weight reflects the accuracy of neighbor's position, and distance weight reflects the accuracy of ranging measurement. The classical multilateration can be transformed into ”weighted multilateration” with given separate weight for each neighbor. The steps of the second phase are shown in the follows:

1. A node, say node v, which can receive the position data from at least three neighbors, calculates the weights for its neighbors as follows.
   (a) Find the position weight w_p for each neighbor, where $w_p = 0.1$ for unknown node and $w_p = 1$ for beacon.
   (b) Use ranging technology (e.g. RSSI) to measure distances to each neighbor, and assign the distance weight w_d according to received power, where $0.1 \leq w_d \leq 1$.
   (c) Set the weight w by $w = w_p \times w_d$.
2. Node v estimates its position by the weighted multilateration.
3. Node v updates its position weight by $w_p = \frac{\sum w}{N}$ (where N is the number of neighbors), and then floods its w_p and position to neighbors.
4. Repeat steps $1 \sim 3$ until position error converges. In our simulation, the number of iterations needed to converge is less than ten.

Take Figure 3.2 for example, node 3 is beacon and others are unknown nodes.

- From step 2, node 1 uses ranging technology to get the distances to its neighbors:
  - distance between 1 and 2 is d_{12},

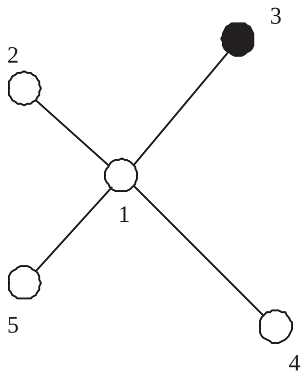

Figure 3.2: The example for the second phase

- distance between 1 and 3 is d_{13},
- distance between 1 and 4 is d_{14},
- distance between 1 and 5 is d_{15},

- Next node 1 gets neighbors' positions and then operates classical multilateration like the first phase:
  - Error functions between estimated distances to neighbors and measured distances to neighbors:

$$\begin{cases} f_{12} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} - d_{12} \\ f_{13} = \sqrt{(x_1 - x_3)^2 + (y_1 - y_3)^2} - d_{13} \\ f_{14} = \sqrt{(x_1 - x_4)^2 + (y_1 - y_4)^2} - d_{14} \\ f_{15} = \sqrt{(x_1 - x_5)^2 + (y_1 - y_5)^2} - d_{15} \end{cases} \tag{3.2}$$

  - To estimate (x_1, y_1), we apply Minimum Square Estimation (MSE) to Equations 3.2:

$$minimizing \quad F = f_{12}^2 + f_{13}^2 + f_{14}^2 + f_{15}^2$$

- But these error functions will have the same influence on minimizing F, this is unreasonable, so we will add weight information by steps $1 \sim 3$:

- Each neighbor's position weight is $w_{2p}, w_{3p}, w_{4p}, w_{5p}$ respectively. Position weight of beacon is always 1, and initial position weight of unknown node is 0.1.
- Each neighbor's distance weight is $w_{2d}, w_{3d}, w_{4d}, w_{5d}$ respectively.
- Weight representing the reliability of neighbors' positions and distances to neighbors is $w_2 = w_{2d} \times w_{2p}, w_3 = w_{3d} \times w_{3p}, w_4 = w_{4d} \times w_{4p}, w_5 = w_{5d} \times w_{5p}$

- From step 4, the classical multilateration is modified to "weighted multilateration":
  - "Weighted error functions" between estimated distances to neighbors and measured distances to neighbors:

$$\begin{cases} w_2 \cdot f_{12} = w_2 \cdot (\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} - d_{12}) \\ w_3 \cdot f_{13} = w_3 \cdot (\sqrt{(x_1 - x_3)^2 + (y_1 - y_3)^2} - d_{13}) \\ w_4 \cdot f_{14} = w_4 \cdot (\sqrt{(x_1 - x_4)^2 + (y_1 - y_4)^2} - d_{14}) \\ w_5 \cdot f_{15} = w_5 \cdot (\sqrt{(x_1 - x_5)^2 + (y_1 - y_5)^2} - d_{15}) \end{cases} \tag{3.3}$$

  - To estimate (x_1, y_1), we apply Minimum Square Estimation (MSE) to Equations 3.3:

$$minimizing \quad F = (w_2 \cdot f_{12})^2 + (w_3 \cdot f_{13})^2 + (w_4 \cdot f_{14})^2 + (w_5 \cdot f_{15})^2$$

  - Now, we put the detailed process of getting estimation (x_1, y_1) into Appendix B.

- From step 5, node 1 updates its position weight to be average neighbors' weight:

$$w_{1p} = \frac{w_2 + w_3 + w_4 + w_5}{4}$$

3.2.1 Location from Only Two Neighbors

According to previous descriptions, a node requires at least three neighbors to locate by multilateration in the second phase. In other words, if it receives less than two neighbors, it can not locate its position. In order to solve this problem, we propose a method to obtain nodes’ position if it only receives the signals of two neighbors. Take Figure 3.3 for example, we will show that how to obtain node’s position with only two neighbor’s information.

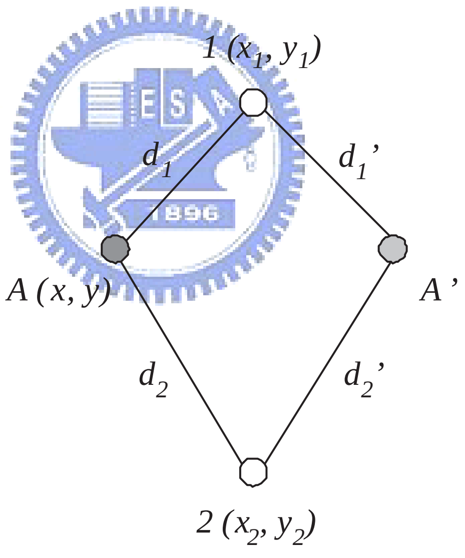

Figure 3.3: Node A has only two neighbors

- Assume that node A has only two neighbors, node 1 and node 2, it may receive their position information $(x_1, y_1), (x_2, y_2)$. Using one ranging technology, the distance d_1 and d_2 can be estimated by receiving power strength.

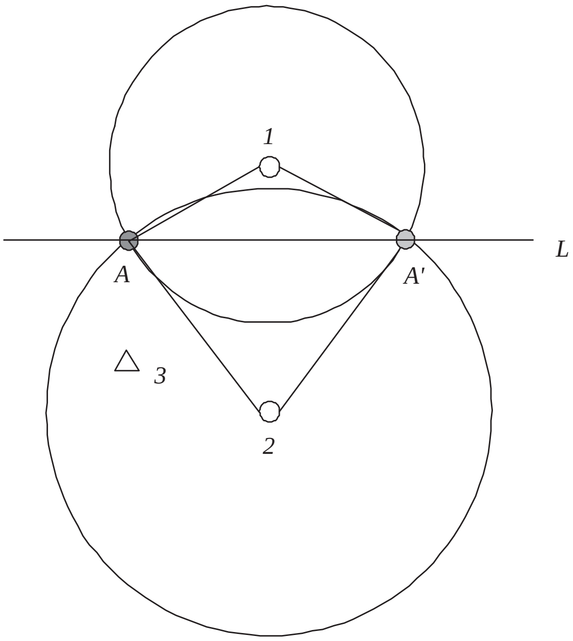

Figure 3.4: Node A has two possible positions

- Node A might also be located to another mirror position A' if d_1' equals d_2'. The geometric relationship of these two possible position solutions A and A' is shown in Figure 3.4.
  - According to the known information $(x_1, y_1), (x_2, y_2), d_1$, and d_2, we can get two circle equations:

$$(x - x_1)^2 + (y - y_1)^2 = d_1^2 \tag{3.4}$$

$$(x - x_2)^2 + (y - y_2)^2 = d_2^2 \tag{3.5}$$

    where (x, y) is the estimated node's position.

  - Subtracting equations 3.4 and 3.5, a straight line equation L can be obtained:

$$(-2x_1 + 2x_2)x + (-2y_1 + 2y_2)y = d_1^2 - d_2^2 - (x_1^2 - x_2^2) - (y_1^2 - y_2^2) \tag{3.6}$$

  - Based on equations 3.4 $\sim$ 3.6, the positions of node A or A' can be obtained.

- After getting two possible positions, we still have to select the better one as an estimation. The selection method is performed by the rule that finding the minimum distance between the estimated position and the initial position to be the better solution. For instance, Figure 3.4 shows that node 3 is the initial position and $\overline{A3}$ is less than $\overline{A'3}$. So node A is better than A'.

Chapter 4

Simulation and Analyses

In this chapter, we use the simulation environment with ns-2, and create two modules in ns-2. The main object of the two modules is to operate the two-phase location processes respectively. These modules also can send their format packets at the top of network layer. In addition, nodes use a mathematical model to compute the distances to their neighbors according to receiving power, and then give distance weight to these distances. Finally, we show some simulation results on different environments.

4.1 Protocol Stack

In order to perform the simulation with ns-2, we follow its system architecture to create two modules for the proposed method. The two modules belong to agent components, and then these modules generate and consume packets, deal with information, and operate location algorithm. One agent module is called dvhop agent, that implements the first phase. The goal of this agent is to generate the initial position for each node. The other agent module is called refine agent, that implements the second phase. The goal of it is to generate refined position for each node.

Figure 4.1 is a conceptual diagram for implemented protocol stack. The figure generally expresses relationship between separate layers in the network. The detailed network component architecture in ns-2 can reference to [23]. In the physical and data link layer, we adopt the RF and 802.11 MAC respectively. In the network layer I use DSDV routing agent. The dvhop and refine agent modules are at the top of network layer. So the two agent modules can get information of routing table because the first phase requires hop counts to beacons.

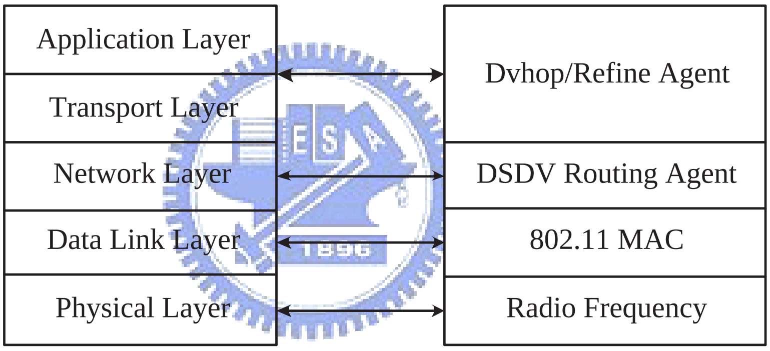

Figure 4.1: Protocol stack

4.2 Packet Header Format

In the first phase, each node is attached to a dvhop agent. Only beacons will broadcast packets to the network, and this packet is called dvhop packet. Dvhop packet header contains whether the node is beacon or not, beacon's x and y coordinate, and one-hop distance from beacon computing. Table 4.1 shows dvhop packet header.

beacon or not	x coordinate	y coordinate	one-hop distance

Table 4.1: Dvhop packet header

In the second phase, each node is attached to a refine agent. All nodes (all unknown nodes and beacons) flood packets to their neighbors, and this packet is called refine packet. Refine packet header also contains whether the node is beacon or not, estimated or real x and y coordinate, and position weight w_p. Table 4.2 shows refine packet header.

beacon or not	x coordinate	y coordinate	position weight w_p

Table 4.2: Refine packet header

4.3 Path Loss Model

In the second phase, nodes exchange information with neighbors. After nodes receive packets from neighbors, they can use the ranging technology (e.g. RSSI) to estimate distances to their neighbors by measuring receiving power. Because of the property of signal propagation, we use a certain path loss model for implementation.

Friis [24] presented free space propagation model, that described the relationship between receiving power and distance from transmitter to receiver. The relationship is shown in the Equation 4.1.

$$P_r = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2 L} \tag{4.1}$$

where P_r is receiving power, P_t is the transmitting power, G_t is the transmitter antenna gain, G_r is the receiver antenna gain, L is the system loss, λ is the wavelength, and d is the distance between transmitter and receiver.

Equation 4.1 may be transformed into Equation 4.2, I would use Equation 4.2 to compute distances between nodes according to receiving power.

$$d = \sqrt{\frac{P_t G_t G_r \lambda^2}{(4\pi)^2 P_r L}} \tag{4.2}$$

4.4 Distance Weight Determination

From section 4.3, we know nodes can estimate distances to neighbors by receiving power. In the second phase, nodes not only know the distances but also know the accuracy of the distance measurements, i.e. determining distance weight. Table 4.3 is our implementation example that describes how to give distance weight according to receiving power. This method is transmitting power is divided by 10 successively, and then receiving power levels are divided into 10 equivalent intervals. These 10 intervals correspond to distance weight 0.1, 0.2, ..., 0.9, 1.

4.5 Simulation Results

In this section, we verify the proposed weighted multilateration algorithm. From the previous sections, we know the simulation processes happen in ns2 envi-

transmitting power =0.28 (watt)	
P_r: Receiving power	w_d: distance weight
$0.28 \times 10^{-1} \leq P_r < 0.28$	1
$0.28 \times 10^{-2} \leq P_r < 0.28 \times 10^{-1}$	0.9
$0.28 \times 10^{-3} \leq P_r < 0.28 \times 10^{-2}$	0.8
$0.28 \times 10^{-4} \leq P_r < 0.28 \times 10^{-3}$	0.7
$0.28 \times 10^{-5} \leq P_r < 0.28 \times 10^{-4}$	0.6
$0.28 \times 10^{-6} \leq P_r < 0.28 \times 10^{-5}$	0.5
$0.28 \times 10^{-7} \leq P_r < 0.28 \times 10^{-6}$	0.4
$0.28 \times 10^{-8} \leq P_r < 0.28 \times 10^{-7}$	0.3
$0.28 \times 10^{-9} \leq P_r < 0.28 \times 10^{-8}$	0.2
$P_r < 0.28 \times 10^{-9}$	0.1

Table 4.3: Distance weight determination

ronment. We analyze the simulation results on different conditions or perspective. In subsection 4.5.1, we try to vary connectivity, i.e. varying the average number of neighbors of nodes, and observe both of the first phase and the second phase about the average position error and the number of located nodes. In addition, we also consider the effect of beacon ratios. In subsection 4.5.2, we try to vary the number of nodes, and observe similar results in subsection 4.5.1 for different conditions.

4.5.1 Connectivity

In this subsection, we try to change connectivity which represents the average number of neighbors of nodes. Connectivity depends on the variance of the radio range, and we will observe the average position error and the number of nodes which can be located. The working area in my simulation is $1000 \times 1000\ m^2$. There are 100 nodes in this working area with random placement.

In Figure 4.2, it shows connectivity versus the position error. We note that beacon ratio means the ratio of the number of beacons and total nodes. When connectivity is greater than 12, the position error decreases under 50%R in the first phase. 50%R means half of the radio range, and it decreases under 10%R in the second phase. The position error decreases about one-fifth from the first phase to the second phase when connectivity is 12.

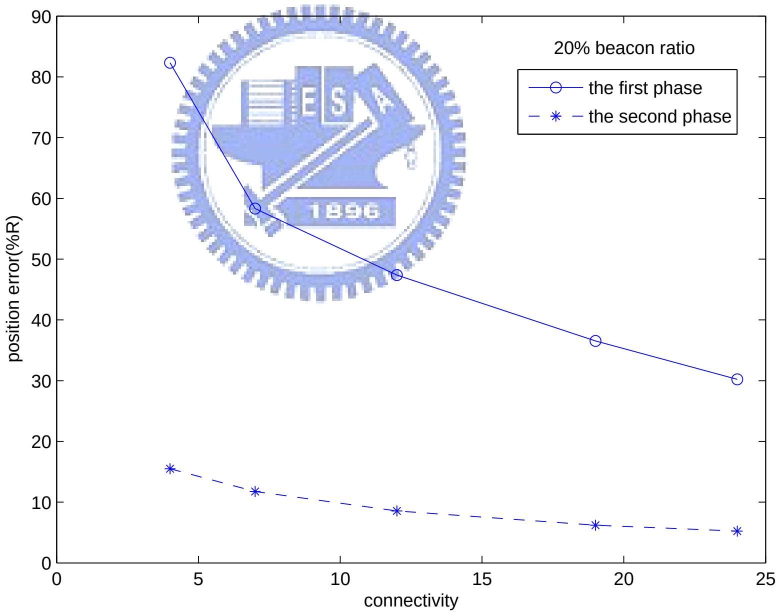

Figure 4.2: Connectivity versus the position error in two phases

Figure 4.3 shows connectivity versus the number of located nodes. The same as Figure 4.2, beacon ratio is 20%. When connectivity is only 4, the number of located nodes is greater than 70% whenever in the first phase or the second phase. When connectivity is over 24, the number of located nodes is close to 100% in two phases. We can see the number of located nodes in the first phase is greater than that in the second phase. This is because a node can not be refined in the second phase when it has no at least two neighbors.

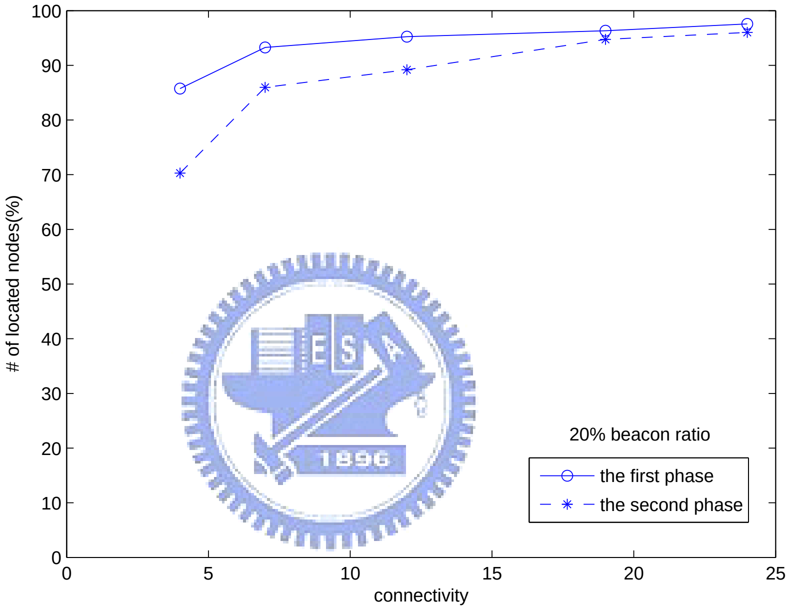

Figure 4.3: Connectivity versus the number of located nodes in two phases

Then Figure 4.4 shows connectivity versus the position error in the first phase. We can see when connectivity is only 4, the position error is greater than 200%R in 5% beacon ratio and less than 100%R in 20% beacon ratio. But when connectivity exceeds 24, the position error decreases under 30%R in 20% beacon ratio.

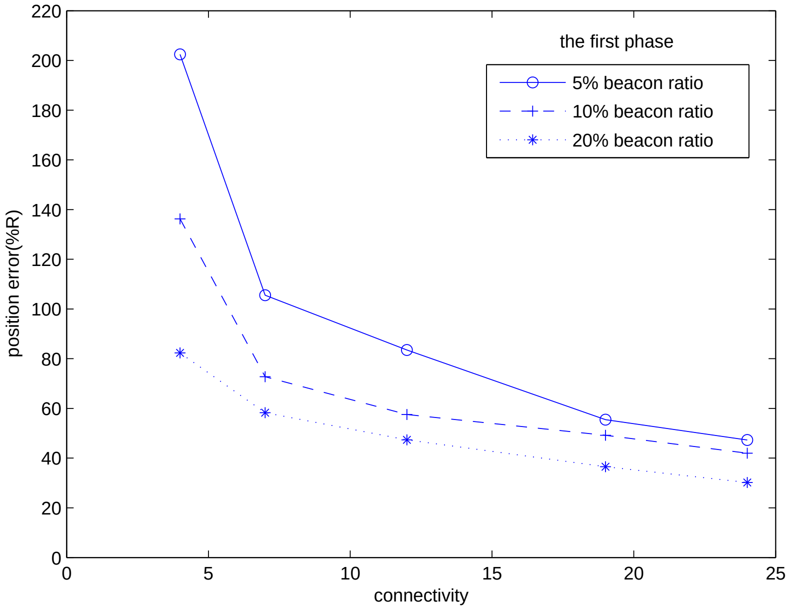

Figure 4.4: Connectivity versus the position error in the first phase

Similarly in Figure 4.4, Figure 4.5 shows connectivity versus the position error in the second phase. We can observe the performance is obviously better than the first phase. When connectivity is greater than 12, the position error decreases under 20%R whatever beacon ratio is. And when connectivity exceeds 24, the position error is down to 5%R in 20% beacon ratio.

Comparing proposed algorithm with referenced algorithm [18], Figure 4.6 shows that our algorithm has better performance when connectivity is low. The position error in the second phase is even decreases 10%R in 20% beacon ratio. But when connectivity approaches 25, the position error converges to its lower bound, so the difference is not very apparent. The number of located nodes in proposed algorithm is greater than referenced algorithm, especially in lower connectivity. Of

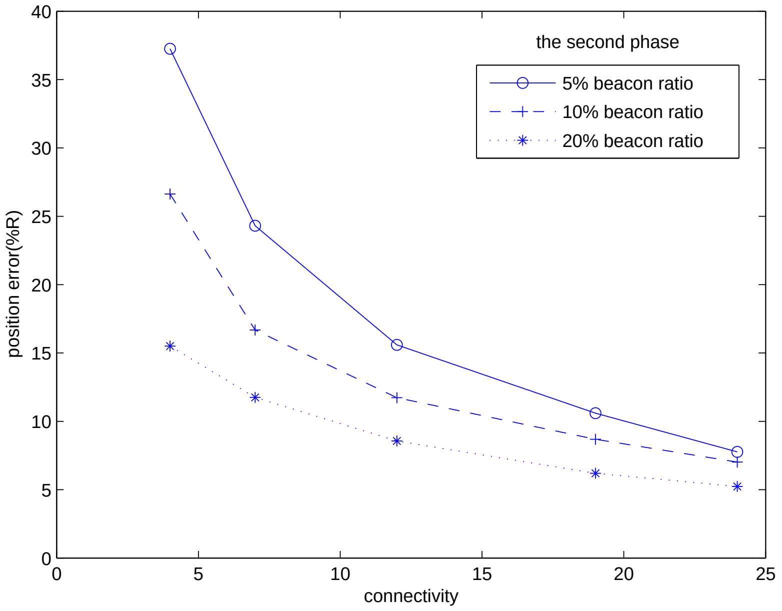

Figure 4.5: Connectivity versus the position error in the second phase

course it is hardly no difference (approaches 100%) when connectivity is high.

4.5.2 The Number of Nodes

In this subsection, we try to change the number of nodes, and observe the average position error and the number of nodes which can be located. The working area in our simulation is $1000 \times 1000\ m^2$. The radio range is 100 m and nodes are put in random placement.

Figure 4.7 shows the number of nodes versus the position error. In 20% beacon ratio, when the number of nodes is 70, the position error in the first phase approximates 100%R, and the position error in the second phase decreases down to half of the first phase. If the number of nodes reaches 100, the position error in

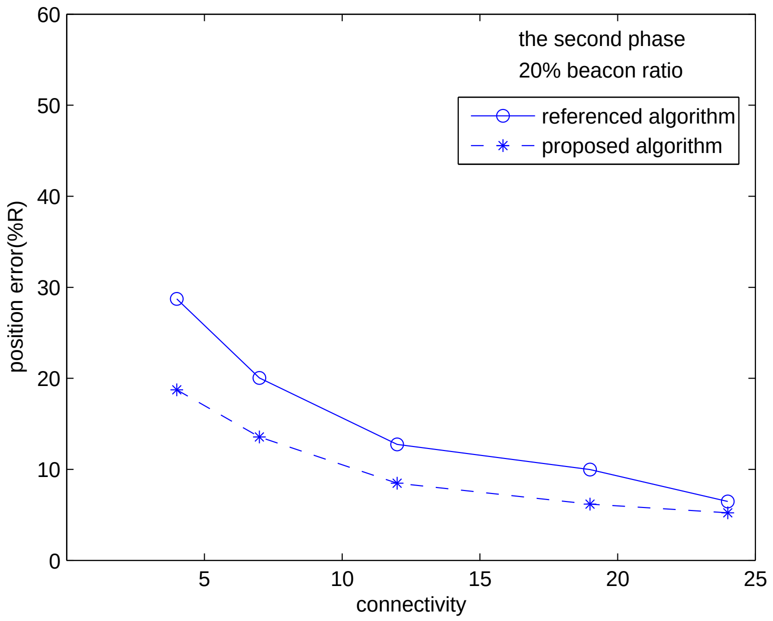

Figure 4.6: Comparison of referenced algorithm and proposed algorithm

the second phase is even less than 20%R.

Figure 4.8 shows the number of nodes versus the number of located nodes. The same as Figure 4.7, beacon ratio is 20%. When the number of nodes is only 30, the number of located nodes exceeds 70% whenever in the first or second phase. When the number of nodes is greater than 80, the number of located nodes is up to 90% in two phases and even reaches 100%. Similarly, we understand the number of located nodes in the first phase is greater than in the second phase, the reason is the same as Figure 4.3.

Then Figure 4.9 shows the number of nodes versus the position error in the first phase. We can see when the number of nodes is very low, the position error

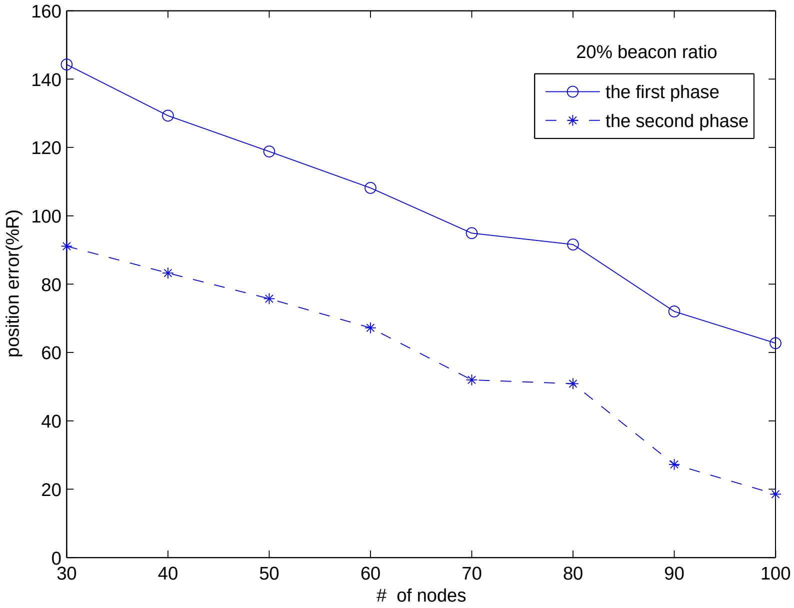

Figure 4.7: The number of nodes versus the position error in two phases

is over 130%R whatever beacon ratio is. When the number of nodes is 70 in 20% beacon ratio, the position error is about 100%R. And when the number of nodes is up to 100 in 30% beacon ratio, the position error finally decreases approximately 50%R.

Similarly in Figure 4.9, Figure 4.10 shows the number of nodes versus the position error in the second phase. We discover the performance is obviously better than the first phase. When the number of nodes is greater than 40, the position error decreases down to 100%R whatever beacon ratio is. When the number of nodes is up to 90, the position error is down to about 50% whatever beacon ratio is. And when the number of nodes reaches 100 in 30% beacon ratio, the position error excellently descends under 20%R.

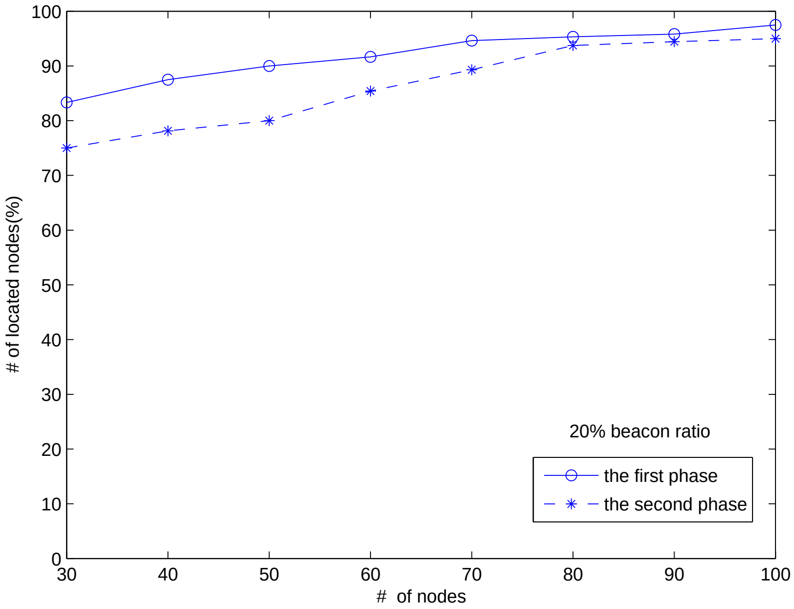

Figure 4.8: The number of nodes versus the number of located nodes in two phases

4.5.3 Discussions

In this thesis, nodes use one-hop distance to estimate distances to beacons in the first phase. Intuitively, one-hop distance is not very accurate. Instead we can try to accumulate the distances by ranging measurement. The simulation environment is the same as the section 4.5.1, and the results for the first phase and the second phase are shown in Figure 4.11 and Figure 4.12 respectively. We can see that this method does not improve the performance by comparing with Figure 4.4 and Figure 4.5. The reasons for causing more error are as follows.

Take Figure 4.13 for example, the real distance between unknown node A and

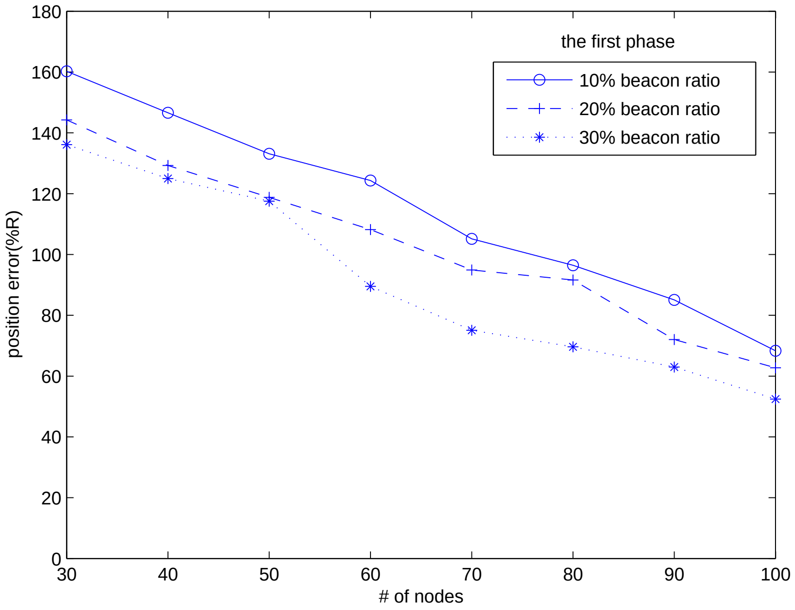

Figure 4.9: The number of nodes versus the position error in the first phase

beacon B is r_{AB}. Node A exchanges the ranging information with its neighbors, and gets the estimated distance to B. In Figure 4.13, if the path from A to B is a little curved, the distance error on A is $|d_1 + d_2 + d_3 - r_{AB}|$. If the path is more curved, the distance error on A is $|d'_1 + d'_2 + d'_3 - r_{AB}|$. So the node distribution causes the large variance. But using one-hop distance in this thesis, the distance error on A is always $|(one - hop\ distance) \times 3 - r_{AB}|$ however curved the path is. So the variance is smaller.

In addition, if nodes accumulate the ranging distances, they could consume larger energy to communicate and compute. But nodes use one-hop distance to estimate, they does not need exchanging the ranging information with neighbors. So it saves more energy in the first phase.

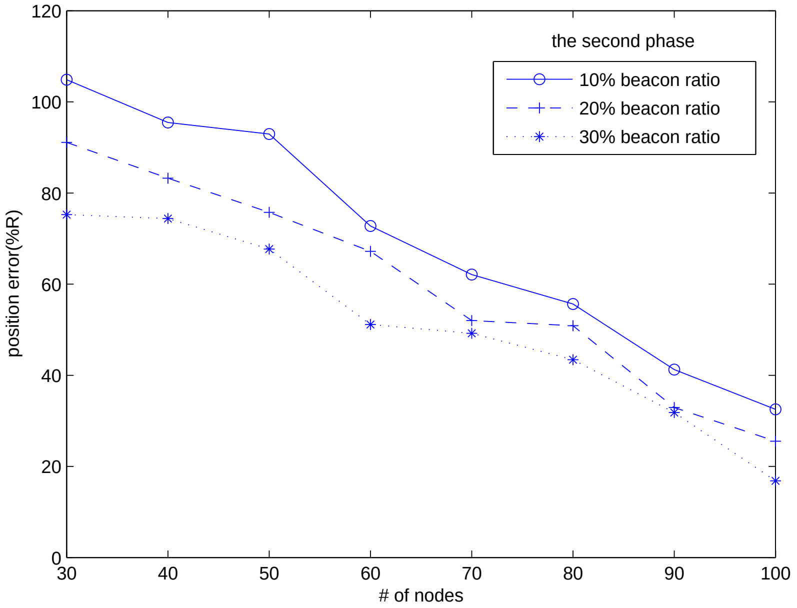

Figure 4.10: The number of nodes versus the position error in the second phase

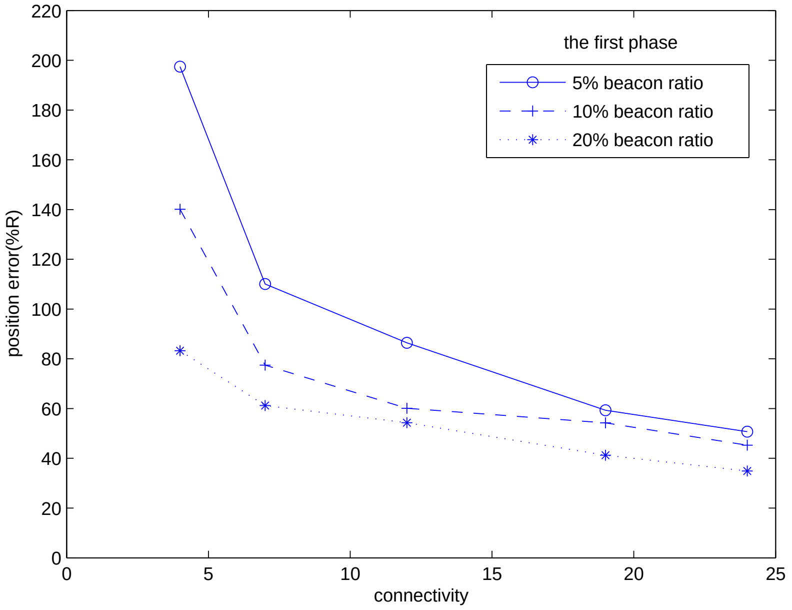

Figure 4.11: Connectivity versus the position error in the first phase

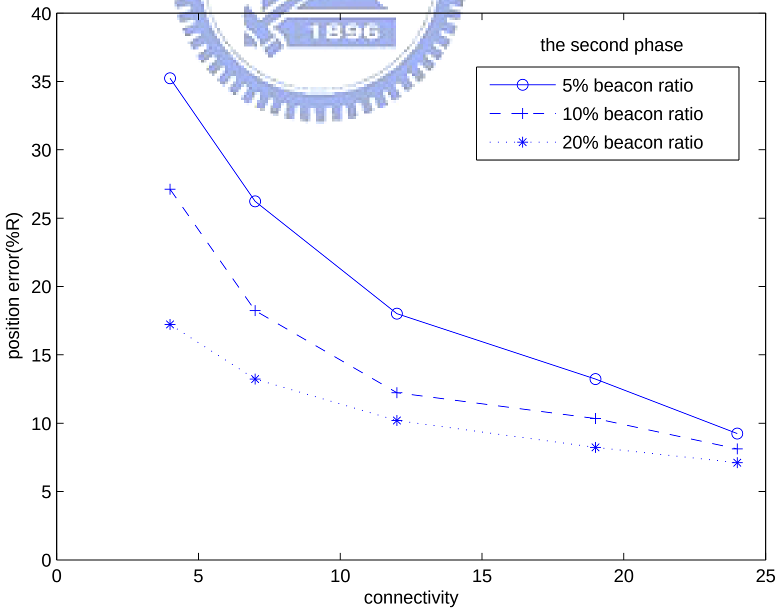

Figure 4.12: Connectivity versus the position error in the second phase

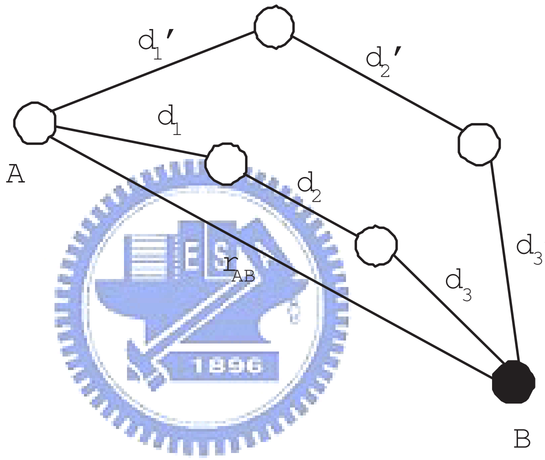

Figure 4.13: Accumulating the ranging distances in the first phase

Chapter 5

Conclusion and Future Work

In this thesis, we proposed a weighted multilateration positioning method for wireless sensor networks. Our method needs fewer beacons (known nodes) without calculating relative positions. Our location algorithm includes two phases. The first phase adopts the method modified from DV-hop. Beacons broadcast their positions and one-hop distances, and then unknown nodes use information from many beacons to get distances to beacons. Unknown nodes finally operates classical multilateration to estimate their initial positions. In the second phase, unknown nodes exchange information with their neighbors. According to whether neighbor is beacon and unstable ranging measurement because of near or far, giving position weight and distance weight to represent their data accuracy. So they can operate modified weighted multilateration to refine their positions repeatedly. Additionally, we solve the location problem that a node only has two neighbors in the second phase.

According to the simulation analyses, the proposed algorithm can decrease the average position error down to 10%R and increase the average number of located nodes up to 80% when connectivity is greater than 12 in 20% beacon ratio.

On the side, when the number of nodes reaches 100 in 20% beacon ratio, the average position error is under 20%R and the number of located nodes approximates 100%.

The proposed algorithm still has some problems that should be solved in future work. In the first phase, it uses the method modified from DV-hop. This intermediate result has poor accuracy for node positions. This is a challenge to design an accurate method for initial estimations. In the second phase, some unknown nodes that have no at least two neighbors can not be located. Actually they might use only one neighbor and certain information to roughly estimate their positions.

Appendix A

Classical Multilateration

We can minimize $F = f_{A1}^2 + f_{A2}^2 + f_{A3}^2$ so that we get the estimated approximate (x_A, y_A).

- Let $f_{A1} = f_{A2} = f_{A3} = 0$, then get the following non-linear system:

$$(x_A - x_1)^2 + (y_A - y_1)^2 = d_{A1}^2 \tag{A.1}$$

$$(x_A - x_2)^2 + (y_A - y_2)^2 = d_{A2}^2 \tag{A.2}$$

$$(x_A - x_3)^2 + (y_A - y_3)^2 = d_{A3}^2 \tag{A.3}$$

- Reduce to the linear system:

$$A.2 - A.1:$$
$$2(x_1 - x_2)x_A + 2(y_1 - y_2)y_A = d_{A2}^2 - d_{A1}^2 - (x_2^2 - x_1^2) - (y_2^2 - y_1^2)$$

$$A.3 - A.1:$$
$$2(x_1 - x_3)x_A + 2(y_1 - y_3)y_A = d_{A3}^2 - d_{A1}^2 - (x_3^2 - x_1^2) - (y_3^2 - y_1^2)$$

- Transform to the matrix form:

$$\begin{pmatrix} 2(x_1 - x_2) & 2(y_1 - y_2) \\ 2(x_1 - x_3) & 2(y_1 - y_3) \end{pmatrix} \begin{pmatrix} x_A \\ y_A \end{pmatrix} = \begin{pmatrix} d_{A2}^2 - d_{A1}^2 - (x_2^2 - x_1^2) - (y_2^2 - y_1^2) \\ d_{A3}^2 - d_{A1}^2 - (x_3^2 - x_1^2) - (y_3^2 - y_1^2) \end{pmatrix}$$

$\Rightarrow Ax = b$

where $A = \begin{pmatrix} 2(x_1 - x_2) & 2(y_1 - y_2) \\ 2(x_1 - x_3) & 2(y_1 - y_3) \end{pmatrix}$, $x = \begin{pmatrix} x_A \\ y_A \end{pmatrix}$,

$b = \begin{pmatrix} d_{A2}^2 - d_{A1}^2 - (x_2^2 - x_1^2) - (y_2^2 - y_1^2) \\ d_{A3}^2 - d_{A1}^2 - (x_3^2 - x_1^2) - (y_3^2 - y_1^2) \end{pmatrix}$

$\Rightarrow$ If the number of the equations is larger than 2, x has no solution but approximation, so this problem is equivalent to solve normal equation $A^T Ax = A^T b$

$\Rightarrow x = (A^T A)^{-1} A^T b$

Appendix B

Weighted Multilateration

We can minimize $F = (w_2 \cdot f_{12})^2 + (w_3 \cdot f_{13})^2 + (w_4 \cdot f_{14})^2 + (w_5 \cdot f_{15})^2$ so that we get the estimated approximate (x_1, y_1).

- Let $(w_2 \cdot f_{12}) = (w_3 \cdot f_{13}) = (w_4 \cdot f_{14}) = (w_5 \cdot f_{15}) = 0$, then get the following non-linear system:

$$w_2^2 \cdot [(x_1 - x_2)^2 + (y_1 - y_2)^2] = w_2^2 \cdot d_{12}^2 \tag{B.1}$$

$$w_3^2 \cdot [(x_1 - x_3)^2 + (y_1 - y_3)^2] = w_3^2 \cdot d_{13}^2 \tag{B.2}$$

$$w_4^2 \cdot [(x_1 - x_4)^2 + (y_1 - y_4)^2] = w_4^2 \cdot d_{14}^2 \tag{B.3}$$

$$w_5^2 \cdot [(x_1 - x_5)^2 + (y_1 - y_5)^2] = w_5^2 \cdot d_{15}^2 \tag{B.4}$$

- Reduce to the linear system:

  $B.2 \times w_2^2 - B.1 \times w_3^2 :$
  $2w_2^2w_3^2(x_2 - x_3)x_1 + 2w_2^2w_3^2(y_2 - y_3)y_1 = w_2^2w_3^2[d_{13}^2 - d_{12}^2 - (x_3^2 - x_2^2) - (y_3^2 - y_2^2)]$

  $B.3 \times w_2^2 - B.1 \times w_4^2 :$
  $2w_2^2w_4^2(x_2 - x_4)x_1 + 2w_2^2w_4^2(y_2 - y_4)y_1 = w_2^2w_4^2[d_{14}^2 - d_{12}^2 - (x_4^2 - x_2^2) - (y_4^2 - y_2^2)]$

  $B.4 \times w_2^2 - B.1 \times w_5^2 :$
  $2w_2^2w_5^2(x_2 - x_5)x_1 + 2w_2^2w_5^2(y_2 - y_5)y_1 = w_2^2w_5^2[d_{15}^2 - d_{12}^2 - (x_5^2 - x_2^2) - (y_5^2 - y_2^2)]$

- Transform to the matrix form:

$$\begin{pmatrix} 2w_2^2w_3^2(x_2-x_3) & 2w_2^2w_3^2(y_2-y_3) \\ 2w_2^2w_4^2(x_2-x_4) & 2w_2^2w_4^2(y_2-y_4) \\ 2w_2^2w_5^2(x_2-x_5) & 2w_2^2w_5^2(y_2-y_5) \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} =$$

$$\begin{pmatrix} w_2^2w_3^2[d_{13}^2-d_{12}^2-(x_3^2-x_2^2)-(y_3^2-y_2^2)] \\ w_2^2w_4^2[d_{14}^2-d_{12}^2-(x_4^2-x_2^2)-(y_4^2-y_2^2)] \\ w_2^2w_5^2[d_{15}^2-d_{12}^2-(x_5^2-x_2^2)-(y_5^2-y_2^2)] \end{pmatrix}$$

$\Rightarrow Ax = b$

where $A = \begin{pmatrix} 2w_2^2w_3^2(x_2-x_3) & 2w_2^2w_3^2(y_2-y_3) \\ 2w_2^2w_4^2(x_2-x_4) & 2w_2^2w_4^2(y_2-y_4) \\ 2w_2^2w_5^2(x_2-x_5) & 2w_2^2w_5^2(y_2-y_5) \end{pmatrix}$, $x = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}$,

$b = \begin{pmatrix} w_2^2w_3^2[d_{13}^2-d_{12}^2-(x_3^2-x_2^2)-(y_3^2-y_2^2)] \\ w_2^2w_4^2[d_{14}^2-d_{12}^2-(x_4^2-x_2^2)-(y_4^2-y_2^2)] \\ w_2^2w_5^2[d_{15}^2-d_{12}^2-(x_5^2-x_2^2)-(y_5^2-y_2^2)] \end{pmatrix}$

$\Rightarrow$ If the number of the equations is larger than 2, x has no solution but approximation, so this problem is equivalent to solve normal equation $A^TAx = A^Tb$

$\Rightarrow x = (A^TA)^{-1}A^Tb$

Bibliography

[1] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K.S.J. Pister. ”System architecture directions for networked sensors”. *ASPLOS*, pp.93-104, 2000

[2] Y.B. Ko and N.H. Vaidya. ”Location-Aided Routing(LAR) in mobile adhoc networks”. *In Proceedings of Mobile Computing and Networking*, October 1998.

[3] J.C. Navas and T. Imielinski. ”Geographic addressing and routing”. *In Proceedings of Mobile Computing and Networking*, pp. 26-30, September 1997.

[4] M.B. Srivastava, R. Muntz, and M. Potkonjak. ”Smart Kindergarten: sensorbased wireless networks for smart developmental problem-solving environments”. *In Proceedings of Mobile Computing and Networking*, pp. 132-139, July 2001.

[5] B. Parkinson and J. Spilker. ”Global Positioning System(GPS): theory and application”. *American Institute of Astronautics and Aeronautics*, vol. 1 and 2, 1996.

[6] D. Niculescu and B. Nath. ”Ad-hoc positioning system(APS)”. *In IEEE GlobeCom*, vol. 5, pp. 2926-2931, November 2001.

[7] P. Bahl and V. Padmanabhan. ”RADAR: an in-building RF-based user location and tracking system”. *In Proceedings of IEEE Computer and Communications Societies*, pp. 775-84, vol. 2, March 2000.

[8] N. Priyantha, A. Chakraborthy, and H. Balakrishnan. ”The cricket location support system”. *In ACM/IEEE International Conference on Mobile Computing and Networking*, pp. 32-43, August 2000.

[9] R. Want, A. Hopper, V. Falcao, and J. Gibbons. ”The active badge location system”. *ACM Transactions on Information Systems 10*, pp. 91-102, Janauary 1992.

[10] X. Ji and H. Zha. ”Robust sensor localization algorithm in wireless ad-hoc sensor networks”. *In IEEE International Conference on Computer Communications and Networks*, pp. 527-532, 2003.

[11] A. Savvides, C.C. Han, and M. Srivastava. ”Dynamic fine-grained localization in ad-hoc networks of sensors”. *In Proceedings of the 7th ACM International Conference on Mobile Computing and Networking*, pp. 166-179, July 2001.

[12] A. Nasipuri and K. Li. ”A directionality based location discovery scheme for wireless sensor networks”. *In 1st ACM Int'l Workshop on Wireless Sensor Networks and Applications*, pp. 105-111, September 2002.

[13] D. Niculescu and B. Nath. ”Ad-hoc positioning system(APS) using AOA”. *In Proceedings of the 20st Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 3, pp. 1734-1743, April 2003.

[14] A. Savvides, H. Park, and M. Strvastava. "The bits and flops of the n-hop multilateration primitive for node localization problems". *In 1st ACM Int'l Workshop on Wireless Sensor Networks and Application*, pp. 112-121, September 2002.

[15] J. Hightower and G. Boriello. "Location systems for ubiquitous computing". IEEE Computer, vol. 8, pp. 57-66, August 2001.

[16] S. Capkun, M.Hamdi, J.P. Hubaux. "GPS-free positioning in mobile ad-hoc networks". *In Proceedings of the 34th Annual Hawaii International Conference on System Sciences*, pp. 3481-3490, January 2001.

[17] Y. Shang, W. Ruml, Y. Zhang, and P.J Fromherz. "Localization from mere connectivity". *In Proceedings of the 4th ACM international symposium on Mobile Ad-hoc Networking and Computing*, pp. 201-212, June 2003.

[18] C. Savarese, J. Rabaey, and K. Langendoen. "Robust positioning algorithm for distributed ad-hoc wireless sensor networks". *In USENIX Technical Annual Conference*, pp. 317-327, June 2002.

[19] J. Caffery and G. Stuber. "Overview of radiolocation in CDMA cellular system". *IEEE Communications Magazine*, vol. 36, pp. 38-45, April 1999.

[20] N.B. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller. "Anchor-free distributed localization in sensor networks". *Technical Report MIT-LCS-TR-892, MIT Lab for Computer Science*, April 2003.

[21] C. Savarese, J. Rabaey, and J. Beutel. ”Locationing in distributed ad-hoc wireless sensor networks”. *In Proceedings of ICASSP*, pp. 2037-2040, May 2001.

[22] N. Bulusu, J. Heidemann, and D. Estrin. ”GPS-less low cost outdoor localization for very small devices”. *Technique Report 00-729, Computer Science Department*, April 2000.

[23] UCB/LBNL/VINT, Official ns manual [Online], Available:
http://www.isi.edu/nsnam/ns/doc/index.html

[24] H.T. Friss. ”A note on a simple transmission formula”. Proc. IRE, vol. 34, pp. 254-256, May 1946.