

# Self Organisation in Random Geometric Graph models of Wireless Sensor Networks

A Project Report

Submitted in partial fulfilment of the

requirements for the Degree of

**Master of Engineering**

in

**Telecommunication**

by

**Swaprava Nath**

under the guidance of

**Prof. Anurag Kumar**



Electrical Communication Engineering

Indian Institute of Science

Bangalore – 560 012 (INDIA)

June 2008

# Abstract

We consider the problem of self-organisation in dense wireless sensor networks. Wireless sensor networks can be viewed in terms of deployment of a large number of nodes in an Euclidean space. After deployment, the nodes are required to build a topology to communicate among themselves and also to a *base station*. In this process they should meet some performance criteria, e.g. coverage of the area to be monitored, connectivity of all the nodes in the network, the capacity of the wireless network, etc. Also, once an event is detected in the network, we need to localise the occurrence of the event with the information reaching the base station in an energy efficient way with minimum delay. These performance objectives are the issues addressed in self-organisation of wireless sensor networks (WSN).

In this report, we first introduce the problem of self-organisation in general and then present a survey of the existing literature in this area. Later we formalise a very commonly used approximation of proportionality between the *hop-distance* (the minimum number of hops) and Euclidean distance for three different scenarios in dense networks. Our proofs bank on a certain geometric construction and union bound, and provide a sufficient condition. We provide simulation results that illustrate the theoretical result, and serve to show how large the number of nodes needs to be before the asymptotics are useful. We propose a localisation algorithm that uses this theory for a fixed anchor and a random node. We also introduce another algorithm for localisation that uses the empirical distribution of Euclidean distance given the hop-distance, which performs better than the previous one. Finally, we discuss few more issues related to the non-idealities in real sensor networks that require more understanding of the stochastic geometry of these networks and theoretical formalisation.

# Acknowledgements

I am grateful to my research supervisor *Prof. Anurag Kumar* for introducing me to an interesting and challenging research topic. It is under his supervision, I learnt to be methodical and tried to learn how to *write in mathematics* rather than *writing in English!*

I am also grateful to my thesis examiners, *Dr. Utpal Mukherji* and *Dr. Rajesh Sundaresan*, and all other faculty members of ECE department and some members of the CSA department, who offered the excellent courses. In particular, I had very useful discussions with Dr. Rajesh Sundaresan, who helped me in improving my work.

I must mention the constant encouragement and support of my parents from a place 2000 kms away, which had been my inspiration all along the tenure of my study here.

I feel myself lucky to be part of the IISc, ECE student community, where I came across plenty of great co-students and I thank them all. In particular, I should mention *Achintya*, who showed me how to handle the pressure of IISc coursework, *Avhishek*, my lab-mate, from whom I learnt many things, both academic and non-academic, and *Venkatesan*, with whom I loved working on topics related to our project. Members of our lab, *Onkar*, *Arjun*, *Prem*, *Venkat*, *Chandramani*, *Manoj*, *Vineeth*, *Naveen* and former member *Srivathsa* made this a great place for working.

The IISc B-Mess used to be a fresh breeze during the heavy coursework. Later too, with the arrival of some juniors of my undergrad university, it had become much more enjoyable.

The people in the Network Labs and in the ECE office were excellent, including *Mr. Srinivasa Murthy*, who kept us updated about all official matters and *Mahesh*, who kept the lab computers up and running, and also gave me a chance to learn Linux.

Last but not the least, I would like to thank all the people associated with Gymkhana for believing in the philosophy that *a healthy mind lives in a healthy body*.

# Contents

<b>Abstract</b>	<b>i</b>
<b>Acknowledgements</b>	<b>ii</b>
<b>1 Introduction to Self-Organisation in Wireless Sensor Networks</b>	<b>1</b>
1.1 Overview . . . . .	1
1.2 System Model . . . . .	2
1.3 Organisation of the Report . . . . .	4
<b>2 Previous Work</b>	<b>6</b>
2.1 Connectivity . . . . .	6
2.1.1 Power Control Mechanism . . . . .	6
2.1.2 Power Management Mechanism . . . . .	8
2.2 Capacity . . . . .	9
2.3 Routing . . . . .	10
2.3.1 Routing protocols . . . . .	11
2.4 Localisation . . . . .	20
2.4.1 Localisation in isotropic atmosphere . . . . .	22

2.4.2	Localisation in anisotropic atmosphere . . . . .	23
<b>3</b>	<b>Distance Discretisation between Two Fixed Points</b>	<b>25</b>
3.1	Distance Discretisation: A Brief Introduction . . . . .	25
3.2	Motivation for random node placement . . . . .	26
3.3	HD-ED Relationship in Arbitrary Geometric Graphs . . . . .	26
3.4	ED-HD proportionality in Random Geometric Graph (RGG) . . . . .	30
3.5	Proportionality for Two Fixed Points Separated by a Distance $d$ . . . . .	31
3.6	Generalisation for all Pairs of Points at a Distance $d$ apart on a Unit Square . . . . .	39
3.7	Generalisation for all Pairs of Points on a Unit Square . . . . .	44
3.8	Illustration of Theorem 1 through Simulations . . . . .	45
<b>4</b>	<b>Distance Discretisation between Two Random Nodes</b>	<b>51</b>
<b>5</b>	<b>Distance Discretisation between A Fixed Point and A Random Node</b>	<b>57</b>
5.1	HD-ED Relationship in Random Geometric Graphs . . . . .	58
5.1.1	Distance distribution from a fixed anchor $b_l$ : Uniform i.i.d. deployment . . . . .	60
5.1.2	Distance distribution from fixed anchors $b_l, l = 1, \dots, L$ : Uniform i.i.d. deployment . . . . .	65
5.2	HD-ED Relationship in Random Geometric Graphs: Fixed Radius . . . . .	66
5.3	Extension to Randomised Lattice Deployment . . . . .	68
5.4	ED bound on a Single Blade with Poisson Deployment . . . . .	70
5.5	Simulation Results . . . . .	71
5.5.1	Illustration of Theorem 5 with increasing $n$ for a fixed $\epsilon$ and HD . . . . .	71

5.5.2	Illustration of Theorem 5 with decreasing HD for a fixed $n$ and a fixed lower bound on probability . . . . .	72
5.5.3	Illustration of convergence in probability for the geometric graph with fixed radius . . . . .	74
<b>6</b>	<b>Application to Node Localisation</b>	<b>75</b>
6.1	Algorithm: Hop Count-derived Distance-based Localisation (HCDL) . . . . .	76
6.1.1	The Algorithm . . . . .	76
6.1.2	Estimating the value of $r(n)$ . . . . .	77
6.2	Simulations implementing HCDL on $\mathcal{G}_{crit}(\mathbf{V})$ . . . . .	80
6.3	A Heuristic for Localisation . . . . .	82
<b>7</b>	<b>Real Scenarios</b>	<b>83</b>
7.1	Non-homogeneous node deployment . . . . .	84
7.1.1	Node deployment density is positive on all points . . . . .	84
7.1.2	Node deployment density hits zero at some points . . . . .	84
7.2	Anisotropic radiation . . . . .	86
7.3	Anisotropic propagation . . . . .	90
<b>8</b>	<b>Conclusions and Work Ahead</b>	<b>91</b>
<b>A</b>	<b>Proof of Lemma 1</b>	<b>93</b>
<b>B</b>	<b>Localisation using the Empirical Distribution of <math>\frac{d}{hr_{crit}}</math></b>	<b>96</b>
	<b>Bibliography</b>	<b>103</b>

# List of Tables

3.1	Simulation Results . . . . .	47
5.1	$(1 - \epsilon)(h_1 - 1)r(n)$ and $h_1 r(n)$ are found from Theorem 5. $\underline{D}_1$ and $\overline{D}_1$ are the maximum and minimum EDs from anchor 1 given the hop-distance $h_1 = 5$ . The theoretical Probability Lower Bound (PLB) = $1 - (h_1 - 1) \left\lceil \frac{\pi h_1}{2\sqrt{1-p^2(\epsilon)}} \right\rceil e^{-ng(\epsilon)r^2(n)}$ , and the Empirical Probability (EP) is found from this experiment. $r(n) = \frac{4}{\sqrt{\pi}} \sqrt{\frac{\ln n}{n}}$ , $\epsilon = 0.4$ . . . . .	72
5.2	Radius $r = 0.1$ , $h_1 = 5$ , $\epsilon = 0.36$ . The theoretical PLB = $1 - (h_1 - 1) \left\lceil \frac{\pi h_1}{2\sqrt{1-p^2(\epsilon)}} \right\rceil e^{-ng(\epsilon)r^2}$ . Abbreviations are as defined in Table 5.1. . . . .	74

# List of Figures

1.1	Various stages in the formation of a sensor network . . . . .	3
3.1	An example deployment with the 4 anchors. . . . .	27
3.2	Graphical illustration of $\overline{D}_l(\mathbf{v}, h_l)$ and $\underline{D}_l(\mathbf{v}, h_l)$ . . . . .	27
3.3	Condition for sequential neighbours in arbitrary geometric graph. . . . .	28
3.4	Construction to find the lower bound on Euclidean distance. . . . .	29
3.5	Achievability of the lower bound. . . . .	29
3.6	Illustration of <i>hop distance</i> . Starting with each point $b_i$ , we first seek a node within the radius $r(n)$ of $b_i$ . . . . .	31
3.7	The construction of a path taking the line joining $b_1$ and $b_2$ as the axis. We are looking at the distance traveled along this path in $N(n)$ hops. . . . .	32
3.8	All the points in the arc shown are also reachable from $b_1$ in $N(n)$ hops. . . . .	37
3.9	Corollary for all pairs of points $z_1$ and $z_2 \in \mathcal{B}(b_1, b_2)$ . . . . .	38
3.10	Construction for all pairs $b_1$ and $b_2$ . There is a circle of radius $d$ centred at the centre of each squarelet and each such circle is covered by blades. . . . .	40
3.11	Detailed view of a squarelet and the first strip. . . . .	41
3.12	Let the angle subtended by the arc at $b_1$ be $\alpha(n)$ . . . . .	41



3.13	Histogram of $\frac{d}{h}$ ratio for 1400 nodes (100 runs) . . . . .	46
3.14	Comparing the histograms (100 runs) of $\frac{d}{h}$ ratio on $\mathcal{G}(\mathbf{V}, r(n))$ and $\mathcal{G}_{crit}(\mathbf{V})$ for 1000 nodes . . . . .	48
3.15	Histogram of $\frac{d}{h}$ ratio on $\mathcal{G}_{crit}(\mathbf{V})$ for 1000 nodes (100 runs): points $b_1$ and $b_2$ . . . .	49
3.16	Histogram of $\frac{d}{h}$ ratio on $\mathcal{G}_{crit}(\mathbf{V})$ for 1000 nodes (100 runs): points $a_1$ and $a_2$ . . . .	50
4.1	Construction using the blades cutting the circumference of the circle of radius $hr(n) + \frac{r(n)}{2}$ and the squarelet splitting the region $\mathcal{A}$ . . . . .	53
4.2	Construction to find $J(n)$ . . . . .	54
5.1	Construction using the blades cutting the circumference of the circle of radius $h_l r(n)$ . . . . .	61
5.2	The construction with $h_l$ hops. . . . .	62
5.3	Construction to find $J(n)$ . . . . .	63
5.4	$g(\epsilon)$ vs $\epsilon$ plot. . . . .	64
5.5	Graphical illustration of how Theorem 6 yields a location region for a node that is at a HD $h_l$ from anchor $b_l$ , $1 \leq l \leq 4$ . . . . .	66
5.6	1000 nodes, 5 hops, $\epsilon = 0.4$ , $\mathbb{P}^n(E_1(n)) \geq 0.37$ . $r(n) = 0.1876$ . . . . .	68
5.7	5000 nodes, 5 hops, $\epsilon = 0.4$ , $\mathbb{P}^n(E_1(n)) \geq 0.79$ . $r(n) = 0.0931$ . . . . .	68
5.8	5000 nodes, 12 hops. . . . .	73
5.9	5000 nodes, 10 hops. . . . .	73
5.10	5000 nodes, 8 hops. . . . .	73
5.11	5000 nodes, 5 hops. . . . .	73
6.1	Construction to show how $r(n)$ is estimated in $\mathcal{G}(\mathbf{V}, r(n))$ using the ED and HD between the anchors only and the node-point theory. . . . .	78

6.2	$r(n)$ estimation error for $\epsilon = 0.02$ . . . . .	80
6.3	$r(n)$ estimation error for $\epsilon = 0.20$ . . . . .	80
6.4	HCDL: 1000 nodes. . . . .	81
6.5	PDM: 1000 nodes. . . . .	81
6.6	HCRL: 1000 nodes. . . . .	81
6.7	Comparison of error CDFs. . . . .	81
7.1	We see that the shortest path from the anchor to the tagged node takes a path along the boundary of the hole on the geometric graph. Hence, ED is no longer proportional to HD in this setup. . . . .	85
7.2	The setup for the Anisotropic Network. . . . .	87
7.3	GG for 1000 nodes: Isotropic. . . . .	88
7.4	GG for 1000 nodes: Anisotropic. . . . .	88
7.5	GG for 5000 nodes: Isotropic. . . . .	88
7.6	GG for 5000 nodes: Anisotropic. . . . .	88
7.7	Error for 1000 nodes: Isotropic. . . . .	89
7.8	Error for 1000 nodes: Anisotropic. . . . .	89
7.9	Error for 5000 nodes: Isotropic. . . . .	89
7.10	Error for 5000 nodes: Anisotropic. . . . .	89
7.11	Error CDF comparison for 1000 nodes. . . . .	90
7.12	Error CDF comparison for 5000 nodes. . . . .	90
A.1	$\phi = \frac{(h_l-1)\pi}{h_l+1} + \delta$ and $\phi' = \frac{(h_l-1)\pi}{h_l+1} - \frac{(h_l-1)}{2}\delta$ . . . . .	94
A.2	$c$ is the point where the angle bisectors meet. . . . .	94

B.1	$\frac{d}{hr_{crit}}$ Histogram. . . . .	97
B.2	Localisation error CDFs of MMSE, HCDL, PDM & HCRL. . . . .	100
B.3	Error for MMSE. . . . .	101
B.4	Error for HCDL. . . . .	101
B.5	Error for PDM. . . . .	101
B.6	Error for HCRL. . . . .	101

# Chapter 1

## Introduction to Self-Organisation in Wireless Sensor Networks

### 1.1 Overview

An ad-hoc wireless sensor network is composed of a large number of nodes deployed densely in close proximity to some phenomenon to be monitored. Each of these nodes makes observations via one or more sensors, the overall purpose being to deduce an inference about the phenomenon and to route this information back to a base station<sup>1</sup>. The network must possess self-organising capabilities since the positions of individual nodes are not predetermined. Cooperation among nodes is the dominant feature of this type of network, where groups of nodes cooperate to process and disseminate the information gathered in their vicinity to the user.

Sensor networks are made up of smart sensor nodes called *motes*. The core of a mote is a small, low-cost, low-power microprocessor. The microprocessor monitors one or more sensors and connects to the outside world with a radio link. The digital radio transceiver allows a mote to transmit reliably to a distance of a few meters. The typical power consumption is about 10 milliamps when the mote is running, and about 10 microamps in sleep mode. Each sensor node is driven by one

---

<sup>1</sup>A base station is a node responsible for the fusion of sensor data. A base station is also known as fusion centre, collector node, or sink.

or two 1.5 V cells. The microprocessor, sensors, antenna and batteries are all packaged in small containers, typically a few millimetres thick [13].

Sensor networks may consist of many different types of sensors such as seismic, thermal, electrical, visual, acoustic, radar and so on. Sensor networks can find a wide variety of applications in a number of domains. Some common applications of sensor networks are:

- Military applications such as battlefield surveillance, nuclear, biological and chemical (NBC) attack detection, and reconnaissance.
- Environmental applications such as wild animal tracking, air and water pollution level monitoring, forest fire detection and precision agriculture.
- Health applications such as heart rate monitoring, telemedicine and drug administration.
- Commercial applications such as highway traffic analysis, building security, structural fault detection, and power consumption measurement.

## 1.2 System Model

We will focus on sensor networks for *inferencing* where the sensor network is deployed for the purpose of deducing the occurrence and the location of an *event*. An event results in a change in the level of some form of ambient energy, in the region monitored, that needs to be sensed by the nodes and used for inferencing, e.g., in border security applications, an event could be an entry or presence of an enemy or intruder into the region.

Figure 1.1 shows the flow of operations in a Wireless Sensor Network (WSN). The stages are briefly described below.

1. **Node deployment:** The first step in the formation of a wireless sensor network is node deployment which addresses the problems of how and where the nodes should be placed, given an operational region, the number of sensor nodes to be deployed, and the application.

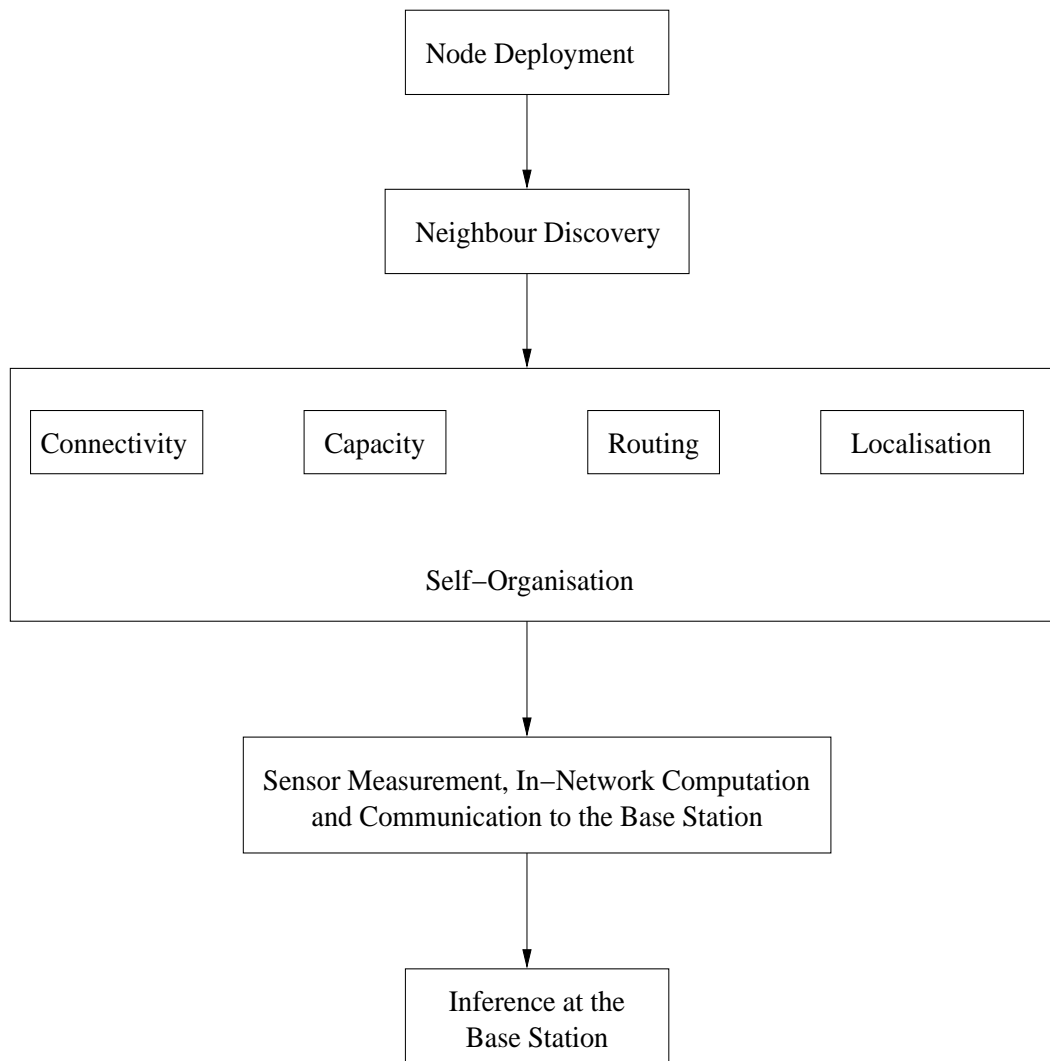


Figure 1.1: Various stages in the formation of a sensor network

The node deployment can be deterministic, or random, or some combination of the two depending on the accessibility of the region.

2. **Neighbour discovery:** Once node deployment is done, every node has to identify its neighbours, i.e., nodes with whom it can directly communicate in the absence of interference. The output of the neighbour discovery process can be represented by the graph  $G_0$  (over the nodes), which has a link between each node and each of the discovered neighbours of the node.
3. **Self-organisation:** After the deployment of nodes, the network self-organises, which means that the connectivity between nodes are ensured, nodes find their routes to the base station

and the locations of the nodes are estimated. So, self-organisation comprises of the following aspects.

- **Connectivity:** This addresses the problem connecting each node with every other node via single or multi-hop path. Connectivity ensures at least one path by which information can be transmitted from one node to any other node in the network.
- **Capacity:** Capacity addresses the question of how much information can be propagated through the network, and gives an upper limit on this maximum possible amount of information flow.
- **Routing:** When an event is detected in the network, the information needs to be sent to the Base Station in an energy efficient manner with minimum delay. Routing addresses these problems in sensor networks.
- **Localisation:** To estimate the location of a detected event, we need to know the location of the nodes in the network. The node location in a dense sensor network is often estimated using the heuristic of the minimum number of hops between two nodes (we will call this as the hop-distance) to be proportional to the Euclidean distance. This report focuses on this issue and provides a theoretical formalisation for this approximation.

□ Although several algorithms for routing and localisation uses a topology as the starting point, a topology is not absolutely necessary. We will discuss about a few routing algorithms that are topology-free. However, topology-free localisation is possible if each node has independent way of locating itself, e.g., if each of them are equipped with GPS transceivers. But for GPS-free localisation, we need to have a topology. Topology-free approaches are interesting because it requires less power, as it doesn't need to maintain the topology and is robust against node failures.

## 1.3 Organisation of the Report

The rest of the report is organised as follows. In Chapter 2 we review some literature on the various aspects of self-organisation in wireless sensor networks. In Chapter 3, we provide a theory of

*distance discretisation*, which is the use of integral hop-distance as a measure of Euclidean distance (a real number), for two fixed points on an area. We model the sensor node locations as a random deployment of points on a region of Euclidean space, e.g. a unit square, and the communication topology as a *Geometric Graph* (GG)<sup>2</sup>. We show that the approximation of distance discretisation is valid asymptotically almost surely in this topology if certain conditions are satisfied for the radius ( $r$ ) of the GG. In Chapter 4, we show the results on distance discretisation for two random nodes in the RGG, and in Chapter 5, we prove that given the hop-distance  $h$  from a fixed anchor to a random node, the Euclidean distance lies within  $[(1 - \epsilon)(h - 1)r, hr]$ , for arbitrarily small  $\epsilon > 0$ . We illustrate the theoretical results with simulations in each chapter. The emphasis of this report is in *localisation*. In Chapter 6, we illustrate a new algorithm called Hop Count-yielded Distance-based Localisation (HCDL), that uses the node-point theory of Chapter 5. Also, we show a heuristic algorithm in Appendix B, that banks on the empirical distribution of  $\frac{d}{hr_{crit}}$  ( $d$  and  $h$  are the Euclidean and hop distances respectively and  $r_{crit}$  is the critical radius), which performs better than the previous in the sense of location error. Finally in Chapter 8, we draw the conclusions from this work and look at some future works.

---

<sup>2</sup>A geometric graph  $\mathcal{G}(\mathbf{V}, r)$  on  $\mathcal{A} \subseteq \mathbb{R}^2$ , and with  $n$  vertices, is a graph with vertex location vector  $\mathbf{V} \in \mathcal{A}^n$ , and an undirected edge between all the nodes that are at a distance  $\leq r$ . If  $\mathbf{V}$  is random, the GG is called Random Geometric Graph (RGG) [21].



# Chapter 2

## Previous Work

In this chapter, we survey some of the existing literature on self-organisation and classify them according to the flowchart of Figure 1.1.

### 2.1 Connectivity

One of the most important issues related to any Wireless Ad-hoc network and particularly WSN (Wireless Sensor Network) is that of connectivity. The topology control algorithm must always ensure that the resulting network or the sub-networks of the original network are strongly connected, so that the information can flow from any node to the other within the entire network or within the sub-networks. Any node should have at least a (single or multiple hop) path to any other node or a certain base station within the network or the sub-network, and that is why we need a connected communication topology. This connectivity problem can be classified as,

#### 2.1.1 Power Control Mechanism

The goal of power control mechanisms is to dynamically change the nodes' transmitting range in order to maintain some property, e.g., connectivity, of the communication graph, while reducing the energy consumed by node transceivers because they are one of the primary sources of energy

consumption in WSNs. Power control mechanisms are fundamental to achieving a good network energy efficiency. Power control is studied in homogeneous (nodes have the same transmitting range) and non-homogeneous (nodes have different transmitting ranges) scenarios.

**Homogeneous Power Control** For homogeneous networks, the connectivity problem has been addressed in detail by Gupta and Kumar [9], where it has been shown that if  $n$  nodes are placed in a disc of unit area in  $\mathbb{R}^2$  and each node transmits at a power so as to cover an area of  $\pi r^2 = \frac{\log n + c(n)}{n}$ , then the resulting network is asymptotically connected w. p. 1 iff  $c(n) \rightarrow +\infty$  and asymptotically disconnected w. p. 1 iff  $c(n) \rightarrow c < \infty$

In homogeneous networks, the CTR (Critical Transmitting Range) problem has been investigated in theoretical ways as well as practical viewpoints. Narayanaswamy et al. [19] present a distributed protocol, called COMPOW that attempts to determine the minimum common transmitting range needed to ensure network connectivity. They show that setting the transmitting range to this value has the beneficial effects of maximising network capacity, reducing the contention to access the wireless channel, and minimising energy consumption.

Connectivity issue has been discussed also by Xue and Kumar in [26], where the authors prove that in a network with  $n$  randomly placed nodes, each should be connected to  $\Theta(\log n)$  nearest neighbours. If each node is connected to less than  $0.074 \log n$  nearest neighbours, the network is asymptotically disconnected w. p. 1. While if each node is connected to more than  $5.1774 \log n$  nearest neighbours, the network is asymptotically connected w. p. 1. It appears that the critical constant may be close to one, but remains an open problem.

In [22], Ramaiyan et al. have demonstrated joint control of power and hop length for a single cell scenario. The objective was to maximise the transport capacity of the network.

**Non-homogeneous Power Control** Non-homogeneous networks are more challenging because nodes are allowed to have different transmitting ranges. The problem of assigning a transmitting range to nodes in such a way that the resulting communication graph is strongly connected and the energy cost is minimum is called the Range Assignment (RA) problem, and it was first studied by Kirousis et al. [15]. The computational complexity of RA has also been analysed in [15]. It

is shown to be NP-hard in the case of 2D and 3D networks. However the optimal solution can be approximated within a factor of 2 using the range assignment generated in [15]. An important variant of RA has been recently studied is based on the concept of symmetry of the communication graph. Due to the high overhead needed to handle unidirectional links in routing protocols or MAC protocols which are naturally designed to work under the symmetric assumption, Symmetric Range Assignment (SRA) shows more practical significance.

In practical wireless networks the links are neither homogeneous in all directions nor time invariant. This issue has been addressed in [3], where the authors use ideas from percolation theory and compare networks of geometric discs to other simple shapes, including probabilistic connections, and they find that when transmission range and node density are normalised across experiments so as to preserve the *expected number of connections* (ENC) enjoyed by each node, the discs are the *hardest* shape to connect together. In other words, anisotropic radiation patterns and spotty coverage allow an unbounded connected component to appear at lower ENC levels than perfect circular coverage allows. This indicates that connectivity claims made in the literature using the geometric disc abstraction will in general hold also for the more irregular shapes found in practice.

### 2.1.2 Power Management Mechanism

Power management is concerned of which set of nodes should be turned on/off and when, for the purpose of constructing energy saving connected topology to prolong the network lifetime.

There are several algorithms that discuss about the sleep-wake cycling of the sensor nodes so that the network lifetime is extended. Also care needs to be taken so that the power of the network falls uniformly over the nodes of the network. Here we survey a particular example. In [6] Chen et al. propose SPAN, a power saving topology maintenance algorithm for multi-hop ad hoc wireless networks which adaptively elects coordinators from all nodes to form a routing backbone and turn off other nodes' radio receivers most of the time to conserve power.

## 2.2 Capacity

Capacity of Wireless Networks has been discussed by Gupta and Kumar in [10], where they have used *Throughput Capacity* as the number of successful bits transmitted in the network in unit time. But in networks, a packet is routed to its destination via a multi-hop path. If the throughput capacity is considered, the same packet which is getting replicated in several nodes are counted multiple times. We should focus on the packet that is transmitted from source to destination and count it as a successful transmission from source to destination, i.e. the throughput per node should be divided by the mean number of hops to get a quantity called the *Transport Capacity*. It implies that in a wireless network the throughput is not the only parameter to be concerned with, the distance to which a bit is transmitted is also important. So, mean hop length  $\times$  throughput capacity = transport capacity is the quantity to maximise.

In this paper, the authors have taken two types of networks, viz. the *Arbitrary* and *Random* Networks and models considered were *Protocol* and *Physical* models. For arbitrary networks the node locations are unknown, whereas in random networks they are random.

- **Protocol Model:** The transmission from node  $i$  to  $j$  in the  $m^{th}$  sub channel is successful if  $|X_k - X_j| \geq (1 + \Delta)|X_i - X_j|$ , for any simultaneous transmitter  $k \neq i$  transmitting in the same sub channel.  $X_l$  denotes the location of the node  $l$ ,  $\Delta$ : a constant.
- **Physical Model:** Let,  $\{X_k, k \in \tau\}$  be the set of simultaneously transmitting nodes at some time slot over a certain sub channel (We are using notation for nodes and their location interchangeably). The transmission from node  $i$  to  $j$  in that sub channel is successful if

$$\frac{\frac{P_i}{|X_i - X_j|^\alpha}}{N_0 + \sum_{k \in \tau, k \neq i} \frac{P_k}{|X_k - X_j|^\alpha}} \geq \beta$$

Where,  $N_0$ : Noise variance,  $\alpha$ : Path loss factor,  $\beta$ : Threshold

It is assumed that whenever these conditions are satisfied the nodes put across a fixed  $W$  bits over the wireless channel. For a setting like this, the results presented in the paper are as follows:

- Under a Protocol Model of noninterference, the capacity of wireless networks with  $n$  randomly located nodes each capable of transmitting at  $W$  bits per second and employing a common range, and each with randomly chosen and therefore likely far away destination, is  $\Theta\left(\frac{W}{\sqrt{n \log n}}\right)$ . This is true whether the nodes are located on the surface of a three-dimensional sphere or on a planar disc. Even when the nodes are optimally placed in a disc of unit area, and the range of each transmission is optimally selected, a wireless network cannot provide a throughput of more than  $\Theta\left(\frac{W}{\sqrt{n}}\right)$  bits per second to each node for a distance of the order of 1 m away. In fact, summing over all the bits transported, a wireless network on a disc of unit area in the plane cannot transport a total of more than  $\Theta(W\sqrt{n})$  bit-meters per second, irrespective of how the load is distributed. Under a Physical Model of noninterference, the lower bounds are the same as those above for the Protocol Model, while the upper bounds on throughput are  $\Theta\left(\frac{W}{\sqrt{n}}\right)$  for Random Networks and  $\Theta\left(\frac{W}{n^\alpha}\right)$  for Arbitrary Networks.
- Splitting the channel into several sub channels does not change any of these results.

## 2.3 Routing

Routing can be defined as the algorithm to send a packet from source to destination. Routing may use a topology or can be topology-free. Typically there are one or more sink nodes or base stations which serve as collection points and connect the wireless nodes to a wired infrastructural network, for example, the Internet. Since the radio range of sensor nodes is of the order of a few meters, the farthest nodes may not be able to reach the sink node in a single hop transmission. Moreover, the nodes may be deployed over uneven terrain in a nonuniform manner (as would be the case for example when several sensor nodes are airdropped over a mountainous region). These factors combined with the resource limitations of sensor nodes make the problem of routing highly nontrivial. The obvious solution to this problem is to resort to *multi-hop routing*, wherein sensor nodes communicate with the sink node via multiple hops through other intermediate nodes. Each sensor node serves as a router in addition to sensing its environment. Conventional *link state* routing algorithms consume a lot of expensive memory space for maintaining their tables and are hence unsuitable for the sensor network scenario.

The lifetime of a fully active sensor node is of the order of a few days. The most energy-intensive operations for a node are those of radio transmission and reception. To maximise the network lifetime, therefore, the amount of network traffic should be minimised. One way of accomplishing this is for certain network nodes to collect raw sensor readings from a number of sensor nodes and combine them into a single composite signal which is then forwarded towards the sink node. This process is called *data aggregation*. Data aggregation can greatly reduce the number of packets transmitted, which can result in large energy savings.

The routing protocols that have been proposed for sensor networks can be broadly classified as *flat* and *hierarchical* protocols. Hierarchical protocols organise the network nodes into several logical levels. This is typically implemented by a process called *cluster formation*. A cluster consists of a set of geographically proximal sensor nodes; one of the nodes serves as a cluster head. The cluster heads can be organised into further hierarchical levels. The key advantage of hierarchical routing protocols is that the cluster heads can perform efficient in-network data aggregation. Routing proceeds by forwarding packets up the hierarchy until the sink node is reached. Flat routing protocols, on the other hand, attempt to find good-quality routes from source nodes to sink nodes by some form of *flooding*. Since flooding is a very costly operation in resource starved networks, smart routing algorithms restrict the flooding to localised regions. Some algorithms use probabilistic techniques based on certain heuristics to establish routing paths.

### 2.3.1 Routing protocols

#### Flat Routing Protocols

Flat routing protocols are similar to the conventional multi-hop ad-hoc routing protocols. Each sensor node determines its next hop neighbour node(s) to forward data packets. The nodes are not organised into hierarchical clusters as is done in the hierarchical protocols. The advantage of this approach is that all the nodes can reach the base station irrespective of their position.

Most of the flat routing protocols that have been proposed for sensor networks incorporate distance vector routing algorithms. In distance vector routing, nodes maintain estimates of their distances

from the destination nodes. Each node transmits its distance estimates to its neighbours. Each node updates its distance vector so as to minimise the distance to each destination by examining the cost to that destination reported by each of its neighbours and then adding its distance to that neighbour. The problem with the straightforward distance vector algorithm is that it takes a long time to converge after a topological change. The following subsections describe the routing protocols that can be classified as flat routing protocols:

**TinyOS Beaconing** This is a topology-free routing technique. The TinyOS embedded sensor network platform [11] employs a very simple ad-hoc routing protocol. The base station periodically broadcasts a route update beacon message to the network. The beacon message is received by a few nodes that are in the vicinity of the base station. These nodes mark the base station as their parent and rebroadcast the beacon to their neighbours. The algorithm proceeds recursively with nodes progressively propagating the beacon to their neighbours; each node marks the first node that it hears from as its parent. The beacon is thus flooded throughout the network, setting up a breadth-first spanning tree rooted at the base station. This process is repeated at periodic intervals known as epochs.

Each network node periodically reads its sensor data and transmits the data packet to its parent in the spanning tree. The parent node in turn forwards the packet to its parent and so on. This process is repeated until the data finally reaches the base station.

The attractive feature of TinyOS Beaconing is its simplicity, nodes do not have to maintain large routing tables or other complicated data structures. Each node needs to remember only its parent node in the path to the base station. By combining the beaconing with a MAC layer scheduling scheme such as TDMA, the nodes can conserve power by keeping their radio off most of the time. In spite of its attractive features, the beaconing protocol suffers from one main disadvantage: it is not resilient to node failures. If a parent node fails, then its entire sub-tree is cut off from the base station during the current epoch. Moreover, the protocol results in uneven power consumption across network nodes. The nodes nearer to the base station consume a lot of power in forwarding packets from all the nodes in their sub-tree, whereas the leaf nodes in the spanning tree do not have to perform any forwarding at all and consume the least power.

**Directed Diffusion** A data-centric communication protocol for sensor networks has been proposed in [12]. All sensor data are characterised by attribute-value pairs. A node that requires data sends out *interests* for named data; interests are diffused through the network towards the nodes that are capable of responding. Data are in turn drawn towards the requesting node via *gradients* established along the reverse path of interest propagation. This style of data-centric communication is fundamentally different from the node-centric end-to-end communication mechanism of traditional IP networks. An interest for data may contain several fields such as type, interval, duration, time stamp and the coordinates of the target region. The duration refers to the time period for which data is desired, and the interval refers to the data rate. The sink broadcasts interests to its neighbours; due to the unreliable nature of broadcast networks, interests are refreshed periodically with updated time-stamp values. The initial interest specifies a large interval value; when the path to the event source is established, a higher data rate is requested. Each node maintains an *interest cache* that contains several fields. One of the fields is called a *gradient* that specifies the node's downstream neighbour. The gradients in each node are used to set up the reverse path for information flow from the source to the sink. A gradient also specifies the data rate requested by the neighbouring node.

Whenever an interest is received, the node looks up its interest cache. If there is no matching entry in the cache, a new interest entry is created. If a matching entry exists already, its time-stamp is updated. The node further broadcasts the interest to its neighbours, and thus the interest is flooded throughout the network, ultimately reaching the source. When the source node detects an event, it searches its interest cache for matching event entries. If a matching interest entry is found, the node starts relaying its readings at the highest requested data rate among all its outgoing gradients. Intermediate nodes that receive a data message from their neighbours also check their interest caches for matching entries. If no matching entry is found, the data packet is silently discarded. Otherwise, the node searches its *data cache* associated with the matching interest entry. If there is no recently seen data item corresponding to the interest, a new entry is created and the data is forwarded to the neighbouring nodes; if the data is already present in the cache, the data packet is silently dropped. This mechanism helps in preventing the formation of loops in data dissemination.



The sink may finally receive low-rate event data from several paths. It *reinforces* one of its neighbours to draw high-rate events. Reinforcement is done by sending out an interest with a higher data rate (smaller interval). The same procedure is adopted by all the upstream nodes to reinforce one or more paths that deliver high-quality event data. This finally results in an empirically low-delay path between the source and the sink. In case multiple paths are created and some paths are found to perform consistently better, an option is available to *negatively reinforce* the other paths. The reinforcement rules can also be applied by intermediate nodes along previously reinforced paths to enable local repair of failed or degraded paths.

**Rumor Routing Algorithm** Braginsky and Estrin [4] propose an algorithm to route user queries to nodes that have observed certain events. Events are assumed to be localised phenomena, occurring in a fixed region of space. Queries can be requests for information or commands to initiate collection of more data. If the number of observed interesting events is high and the number of queries for the events is low, it is better to flood the *queries* through the network. On the other hand, if the number of user queries is very high compared to the number of interesting events, it makes sense to flood *event information*. The rumor routing algorithm tries to fit in between query flooding and event flooding.

Rumor routing aims to create paths leading to events; whenever a query is issued for an event, it is sent on a random walk through the network until it intersects one of the event paths. If the random query path fails to intersect any event path, the application resubmits the query, or in the worst case, floods the query throughout the network.

Each node maintains a *neighbour table* and an *event table*. The event table contains a list of events that the node has observed. The neighbour table can be maintained by actively initiating hello messages or passively eavesdropping on network broadcasts.

The algorithm employs a set of long-lived packets called *agents* that traverse the network, record interesting events that they observe and disseminate this event information to network nodes. Agents are generated by nodes randomly with a tunable probability based on whether the nodes have observed an event in the recent past. Agents also contain event tables similar to nodes, which include the number of hops to each event. When an agent crosses a node that has information about some

event that the agent has not yet seen, it updates its event table to include the event. Agents travel for a specified number of hops and then die. Nodes can update their routing tables when they encounter agents that have cheaper paths to certain events.

**Highly resilient energy-efficient multipath routing** Ganesan et al. [8] present a multipath routing technique to improve the resilience of a sensor network to node failure. Constructing  $k$  disjoint paths from the source to the sink ensures that the network does not get disconnected even if  $k$  nodes fail. However, finding completely disjoint paths tends to be very energy-inefficient. To overcome this problem, an algorithm to construct partially disjoint paths (called *braided paths*) with some common nodes between paths is presented.

The algorithm aims to extend the concept of directed diffusion to eliminate the energy-intensive flooding used to discover alternate paths. The basic idea of the algorithm is to set up multiple paths along which data is disseminated at low data rates at the same time when the primary path is established. If there is any node or link failure on the primary path, nodes can quickly reinforce one of the alternate paths without resorting to expensive flooding. Nodes have to fall back on flooding only if all the multiple paths fail simultaneously.

A mechanism to discover strictly disjoint multipaths is presented first. Following the directed diffusion algorithm, the sink reinforces the link with its most preferred neighbour. At the same time, it sends an alternate reinforcement message to its next most preferred neighbour, say A. A propagates this reinforcement to its most preferred neighbour, say B, in the direction of the source. If B already happens to lie on the primary path between the source and the sink, it sends a *negative reinforcement* message back to A; A then tries its next most preferred neighbour and so on. Otherwise B continues to propagate the alternate reinforcement to its neighbours. This procedure can be extended to discover  $k$  disjoint multipaths between the source and the sink.

The problem of finding braided multipaths can be defined as finding the best path from the source to the sink that does not contain one of the nodes on the primary path. This results in finding an alternate path that is expected to be physically close to the primary path, and hence dissipates energy proportional to the primary path. Braided paths are constructed using a procedure similar to that of the disjoint multipaths. Each node on the primary path sends reinforcement messages to

its first and second most preferred neighbours, thus trying to route around its immediate neighbour on the primary path. A node not on the primary path that receives a reinforcement message from a primary node propagates the message to its most preferred neighbour. If this neighbour happens to lie on the primary path, then the reinforcement is *not* propagated any further (since a braided path has already been found).

Constructing strictly disjoint multipaths ensures that any number of failures on the primary path does not affect any of the alternate paths. In contrast, in the case of the braided multipaths, failure of a certain set of nodes on the primary path can disrupt all the multipaths. However, the advantage of braided multipaths stems from the fact that the total number of *distinct* alternate paths through a braid is much higher than the number of nodes along the primary path, thus greatly increasing the resilience of the braid. It will be interesting to study the extension of the braided multipath algorithm to multiple sources and sinks with respect to complexity, resilience to isolated and patterned failures, and maintenance overhead.

**ASCENT** The ASCENT (Adaptive Self-Configuring sEnsor Networks Topologies) system [5] builds on the observation that only a subset of the nodes is actually required to establish connectivity in a dense sensor network. Each node determines its connectivity and decides whether or not to participate in the routing mechanism.

Nodes in ASCENT can be in one of four states: *active*, *passive*, *test* or *sleep*. Active and test nodes are involved in the forwarding of data and routing control messages. Sleeping nodes keep their radios turned off to save power, whereas passive nodes only listen to the network traffic in promiscuous mode. Initially nodes start in the test state which is used to determine if the addition of a new node is likely to improve the connectivity of the network. When a node enters the test state, it initialises a timer  $T_t$  and transmits *neighbour announcement* messages to other nodes. The node upgrades itself to the active state as soon as the timer expires. Before the timer goes off, however, if the number of active neighbours is above a certain threshold or the data loss rate is higher than before, the node falls back to the passive state. A passive timer  $T_p$  is started when the node enters the passive state. As soon as this timer expires, the node turns its radio off and enters the sleep state. Before this timer goes off, however, if the number of active neighbours is less

than a threshold, or the node hears a help message (described below) from an active neighbour, it transitions back to the test state. Nodes in the sleep state wake up and move to the passive state after a timeout interval,  $T_s$ .

The source transmits packets towards the sink via the active nodes. Since there are only a few active nodes to start with, many losses are encountered. This prompts the active nodes to broadcast *help* messages, asking for more active nodes to join the network. Some of the nodes in the passive state react to the help messages and become active nodes. This process continues until a sufficient number of active nodes is available for reliable data transmission.

**GRAdient Broadcast (GRAB) Routing** Ye et al. [28] propose a routing algorithm named GRAdient Broadcast (GRAB), which is an example of a topology-free routing technique. GRAB addresses the problem of robust data forwarding to a data collecting unit (called the sink) using unreliable sensor nodes with error-prone wireless channels. The model for the sensor network consists of large number of small, stationary sensor nodes deployed over a field. The user collects sensing data via a stationary sink that communicates with the network. Each event is detected by multiple nearby sensor nodes and one of them generates reports as a source. Due to the limited radio range, reports are forwarded over many hops before reaching the sink. Nodes can tune their transmitting powers to control how far the transmission may reach. Such power adjustments save energy and reduce collisions whenever possible. This forwarding technique is called ‘mesh forwarding’ and it is claimed that GRAB exploits the large scale property of sensor networks and achieves robust data delivery through controlled mesh forwarding. The algorithm is topology-free, hence requires no energy cost to maintain the topology. Also, since the route is discovered after the occurrence of the event, it is robust against node failures.

**Energy Efficient Routing** Schurgers et al. [23] propose a set of techniques to improve the routing in sensor networks. They argue that uniform utilisation of resources such as power can be obtained by shaping the traffic flow. For instance, the routing paths for several data streams are likely to share many common nodes. These common nodes burn out faster owing to the heavy load and thereby limit the system lifetime. A traffic flow that spreads the energy utilisation over all the

nodes uniformly is highly desirable to maximise the lifetime of the network.

All previously discussed protocols do not diminish the total network energy uniformly. This is the difference of this protocol with the rest.

Three techniques for spreading the network traffic uniformly are proposed. In the first scheme, a *stochastic model* is used by any node to select the next hop. The randomly chosen links tend to distribute the traffic load across the network. In the *energy-based scheme*, a node that has depleted its energy reserves below a certain threshold discourages neighbouring nodes from forwarding packets to it. This is done by appropriate advertisements to neighbours. In the *stream-based scheme*, a node on the path of one data stream tries to divert traffic from other streams away from it.

**Summary of Flat Routing Protocols** The preceding sections have described several flat routing protocols for sensor networks. Since the routing algorithms have to operate based only on local knowledge, some form of distance vector routing is adopted. However, as discussed previously, distance vector routing protocols converge very slowly.

### **Hierarchical and cluster-based Routing Protocols**

Hierarchical routing protocols organise the network into groups called *clusters*. Each cluster selects a node that serves as the *cluster-head*. The cluster-head is responsible for collecting the sensor data from all the cluster members, aggregating them and transmitting a summary to the base station. This results in eliminating a large number of redundant messages from the nodes, thereby reducing the overall power consumption in the network. It also avoids many MAC layer collisions that waste the available bandwidth. This enables the sensor network to scale to a large number of nodes.

The disadvantage of cluster-based algorithms is that the base station should be reachable from all the cluster-heads. This drains the power reserves of the cluster-heads quickly, thereby disconnecting the corresponding clusters from the network. It is possible to avoid this problem by periodically rotating the cluster heads among the nodes to ensure uniform energy consumption.

**LEACH** Heinzelman et al. [2] describe LEACH (Low Energy Adaptive Clustering Hierarchy), a cluster-based routing protocol. LEACH aims to uniformly distribute the energy consumed by sensor nodes across the network to extend system lifetime. This is accomplished by periodically rotating the cluster head nodes. The cluster heads collect the sensor readings from the other nodes in the cluster, perform local compression or aggregation on the data to reduce global communication and transmit a summary of the readings back to a central base station. Thus the cluster heads are the most critical nodes in the network since the entire cluster would be disconnected if the corresponding cluster-head were to run out of energy. A fundamental assumption of the LEACH algorithm is that nodes can adjust their transmission power to transmit signals to varying distances.

The LEACH algorithm runs in rounds, with each round beginning with a *setup phase* in which the cluster-heads are selected and the clusters are formed, and the *steady-state phase*, in which the sensor data transfer takes place. Each node determines by itself whether to serve as a cluster head or not during the current round, based on its remaining energy level and a predetermined desired percentage of cluster-heads in the network. The algorithm guarantees that each node will become a cluster-head eventually, after some fixed number of rounds. This contributes towards uniform energy dissipation of the nodes.

Once a node decides to act as a cluster-head for the current round, it broadcasts an *advertisement message* to the rest of the nodes. Each of the non-cluster-head nodes affiliate themselves with the cluster-head from which they receive the advertisement message with the highest signal strength, with ties being broken randomly. The cluster-head is informed about this affiliation by a message from each of the affiliating nodes. This process organises the entire network into clusters, with a single cluster-head for each cluster.

After a cluster-head receives affiliation messages from all the nodes in its cluster, it creates a TDMA schedule and broadcasts it to the nodes. The TDMA schedule divides time into a set of slots, the number of slots being equal to the number of nodes in the cluster. Each node is assigned a unique time slot during which it can transmit its readings to the cluster-head. The advantage of this approach is that a node can turn off its radio transceiver during all of the other time slots, leading to large energy savings. When the cluster-head receives the sensor readings from all of its cluster nodes, it compresses and aggregates them into a composite signal and transmits it to

the base station. This transmission potentially requires high energy since the cluster-head may be very distant from the base station. An additional disadvantage of this scheme is that if there is any physical obstruction (such as a tree, a hill or a building) between the cluster-head and the base station, the entire cluster is cut off from the base station.

**Summary of Hierarchical and cluster-based Routing Protocols** Hierarchical routing protocols greatly increase the scalability of a sensor network. The overall energy consumption of the nodes is reduced, leading to prolonged network lifetime. The organisation of the network into clusters lends itself to efficient data aggregation which in turn results in better utilisation of the channel bandwidth. Cluster-based routing holds great promise for many-to-one and one-to-many communication paradigms that are prevalent in sensor networks.

## 2.4 Localisation

Given an event has occurred in the region where sensor nodes are deployed, to know the exact location of the event is the goal of *Localisation*. In situations where the entire or part of the terrain is inaccessible, e.g. in battlefields, the nodes are deployed in a random manner over the area and the nodes have no information about its location. To estimate the location of the sensor nodes, we need localisation algorithms.

One approach for localisation can be to equip all the nodes with Global Positioning System (GPS) receivers. GPS gives near accurate estimate of location. This approach requires the GPS satellites and GPS enabled receivers. A typical GPS receiver calculates its position using the signals from four or more GPS satellites. Four satellites are needed since the process needs a very accurate local time, more accurate than any normal clock can provide, so the receiver internally solves for time as well as position. In other words, the receiver uses four measurements to solve for 4 variables, namely, x, y, z and t (for these 3 co-ordinates we need 4 satellites). These values are then turned into more user-friendly forms, such as latitude/longitude or location on a map, then displayed to the user.

Each GPS satellite has an atomic clock, and continually transmits messages containing the current time at the start of the message, parameters to calculate the location of the satellite (the ephemeris), and the general system health (the almanac). The signals travel at a known speed, the speed of light, through outer space, and slightly slower through the atmosphere. The receiver uses the arrival time to compute the distance to each satellite, from which it determines the position of the receiver using geometry and trigonometry.

Although four satellites are required for normal operation, fewer may be needed in some special cases. For example, if one variable is already known (for example, a sea-going ship knows its altitude is 0), a receiver can determine its position using only three satellites. Also, in practice, receivers use additional clues (doppler shift of satellite signals, last known position, dead reckoning, inertial navigation, and so on) to give degraded answers when fewer than four satellites are visible.

GPS receivers are composed of an antenna, tuned to the frequencies transmitted by the satellites, receiver-processors, and a highly-stable clock (often a crystal oscillator). They may also include a display for providing location and speed information to the user. A receiver is often described by its number of channels: this signifies how many satellites it can monitor simultaneously. Originally limited to four or five, this has progressively increased over the years so that, as of 2006, receivers typically have between twelve and twenty channels.

The advantage of using GPS is its accuracy. Autonomous civilian GPS receivers are typically accurate to about 15 meters. But the complexity of computing the location of the receiver makes it energy intensive and each of the nodes more expensive.

Since the GPS-based approach is expensive, we will look at some algorithms that do not use GPS in all the nodes. However in these methods, the beacon nodes should have their location information beforehand, and thus they may be equipped with GPS receiver. Here we are going to discuss two such methods of localisation, namely *DV hop based localisation* and *Hop Count Ratio based Localisation* (HCRL).

Depending upon the sensor nodes' radiation pattern, which can be either isotropic or anisotropic, the localisation algorithms can be classified as the following.



### 2.4.1 Localisation in isotropic atmosphere

**DV Hop based localisation** In [20], Niculescu and Nath have proposed an algorithm named DV hop where each node exchanges information containing the location of, and the hop-counts to, the anchor nodes. After the information exchanges between anchor nodes are complete, an average distance per hop is calculated. Now, to locate a certain node, the number of hops from at least 3 beacons are computed and multiplied with the average distance per hop to get an approximate distance from each of the beacons. Using *triangulation* method, the location of the node can be determined.

**HCRL** In [27], Yang et al. propose an algorithm named *Hop Count Ratio based Localisation* (HCRL), where unlike the above algorithm, the average distance per hop is not evaluated. Rather, the ratio of the *hop distances* (hop distance is defined as the minimum number of hops between any two nodes) from a node to a pair of beacons are computed. It doesn't require the knowledge of the actual average distance per hop, only assumes that the hop distance and the Euclidean distances are proportional. However, the minimum number of beacons in this case is 4, since we are not using the actual distance but the ratio of the distances.

**Observation:** In both the approaches, the assumption made is that the Euclidean distance between two nodes in a dense network is proportional to the hop distance. This is because, in a dense enough network, there are plenty of nodes and the minimum hop path between any two nodes are likely to be on a straight line connecting the nodes. In the following three chapters, we are going to provide a theoretical formalisation of this approximation.

**Proximity Distance Map (PDM)** This approach was proposed by Lim and Hou [17]. The relation between the Euclidean distance and hop-distance is characterised as a linear transformation. If  $D$  is the matrix of the pairwise distances between the beacons and  $H$ , the hop-distance matrix, then the following relation is assumed  $D = TH$ , where  $T$  is the linear transformation from  $H$  to  $D$ . Given  $D$  and  $H$ ,  $T$  is then obtained as  $T = DH^T(HH^T)^{-1}$ . Now, for any node with a hop-tuple  $\mathbf{h}$  the Euclidean distance vector is estimated as  $\mathbf{d} = T\mathbf{h}$  and this ED vector is used to localise each node via triangulation. This method has been tested for anisotropic networks and is found to work

well.

The relation between the number of hops and the Euclidean distance traversed has been studied analytically in previous literature. Vural and Ekici [24] have derived the distribution of the maximum Euclidean distance traveled along a RGG with radius  $r$ , in one dimension. They have studied the distribution of the maximum Euclidean distance traveled along the line by a path of a given number of hops, and have obtained approximations to the mean and variance of these distributions. A similar analysis has been performed in Dulman *et al.* [7], where two dimensional node deployments have been considered in some detail.

The work presented in this report is different from the type above. Here our focus is on finding the bound on Euclidean distance (ED) given hop-distance (HD) and vice-versa. In Chapter 3, we took any two points on a two-dimensional plane and found a bound on the HD. In Chapter 4, we took any two nodes at a certain HD away and found the bound on the ED between them. In Chapter 5, we took a fixed point and a node placed randomly at a certain HD away and found the bound on the ED between them. Hence, the results presented in this report are different in addressing the question of proportionality between the ED and HD, and gives a design rule for HD-based localisation algorithm presented in Chapter 6.

### 2.4.2 Localisation in anisotropic atmosphere

In real networks, the radiation pattern of antennas and the radio propagation are anisotropic. Also the deployment of the nodes may not be homogeneous. These non-idealities forces the above-mentioned algorithms to perform worse. Following are two papers that address some of these issues.

**Concave Environment** The concave environment is defined as a certain kind of node deployment where the nodes have a positive probability of falling only on an area, whose shape is non-convex. Using a setup of this kind, in [25], the authors have proposed a new algorithm named *i-Multihop*, that tries to estimate the location of a node using the distance information got from a graph with nodes deployed in a non-convex region. The distance information can be imprecise due

to the concavity and so they formulate an optimisation problem to solve for the location of a node given the distance estimates. Through simulation they have shown the algorithm to perform better than the DV-Hop based approach in a non-convex setting.

**Deployment with *holes*** Since sensor networks are used for monitoring some process on an area, it can have *holes*. The holes are defined as regions where the node density is zero. In a network with holes, it is difficult to use the hop-distance as a measure of Euclidean distance directly. This issue has been addressed in the paper by Li and Liu [16], where they propose a new algorithm called REndered Path (REP). This algorithm tries to figure out the true ED between two nodes from the path information and the shape of the holes. The assumption there is that the number of holes and their boundary information is known. Also they assume that outside the holes the node density is homogeneous and the ED-HD proportionality is valid there. When a hole comes in between the node and an anchor, the shortest path takes a route over the edge of the hole, and using several geometric propositions, they prove that the true ED can be figured out in such a setup.

## Chapter 3

# Distance Discretisation between Two Fixed Points

### 3.1 Distance Discretisation: A Brief Introduction

We consider a dense wireless sensor network comprising a large number of nodes,  $n$ , distributed uniformly over a region in Euclidean space, e.g., the unit square. If the communication range of every node is  $r$ , then the communication topology becomes a geometric graph, i.e., each node is connected to every other node that is at a distance  $\leq r$ . If the node deployment is random in some sense, e.g., uniform i.i.d. deployment, then the network topology becomes a random geometric graph (RGG) (see, e.g., [21]). Given a dense deployment of nodes, and a topology over them, a frequently used approximation is to take the minimum number of hops between nodes (i.e., the *hop distance*) as a measure of the Euclidean distance between them. By *distance discretisation*, we mean the use of the integral valued hop-distances as a measure of Euclidean distances, which are real numbers. Niculescu and Nath [20], Nagpal *et al.* [18] and Yang *et al.* [27] have used this approximation to develop techniques for GPS-free localisation in dense wireless sensor networks. Yang *et al.*, in particular, make a key assumption that the ratio of the Euclidean distance between a node and two anchor nodes is well approximated by the ratio of the corresponding hop distances.

In the following section, we give a motivation for studying the Geometric Graph with random node

placements and study the ED-HD relationships in a Random Geometric Graph in the sections to follow.

## 3.2 Motivation for random node placement

A natural question that arises when we use random node placement for the sensor network, is the part played by randomness in bringing out the ED from the HD information. The answer is that, if we use an arbitrary placement of nodes, the HD does not give any useful information about the ED. This has been shown in the following section, where we show that we can always come up with an arbitrary node placement for which if the  $HD \geq 2$ ,  $r < ED \leq HD \times r$ , where  $r$  is the radius of the geometric graph on which we measure the hops. So, this fact motivates us to use the random placement of nodes, and we will prove that, in such a setup, HD gives useful information about the ED, thus making the Random Geometric Graphs interesting to study.

## 3.3 HD-ED Relationship in Arbitrary Geometric Graphs

In this section, we evaluate the performance of distance-hop proportionality in an *arbitrary* geometric graph (by arbitrary we mean the node locations are arbitrary) with radius  $r$ , and show that this approximation is coarse. The setting and few notations are as follows.

### Setting:

- $n$  nodes are deployed on a unit 2-dimensional area  $\mathcal{A}$  in an *arbitrary* fashion. The node locations are denoted by the vector  $\mathbf{v} = [v_1, v_2, \dots, v_n] \in \mathcal{A}^n$ , where  $v_i$  is the location of the  $i^{th}$  node.
- We form the geometric graph  $\mathcal{G}(\mathbf{v}, r)$  by connecting the nodes that are within the radius  $r$  of each other, where  $r$  is the radius for the arbitrary geometric graph.

We define *anchors* as nodes whose locations are known apriori, e.g., in Figure 3.1, we have shown 4 anchors  $b_1, b_2, b_3$  and  $b_4$ , with their position fixed at the 4 corners of the unit square  $\mathcal{A}$ .

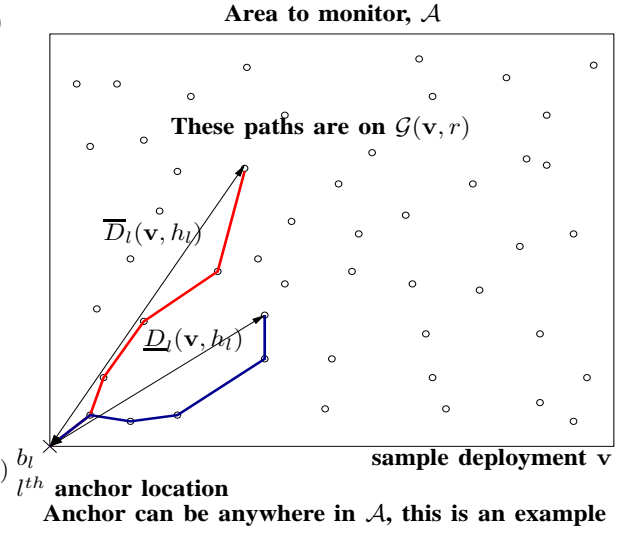
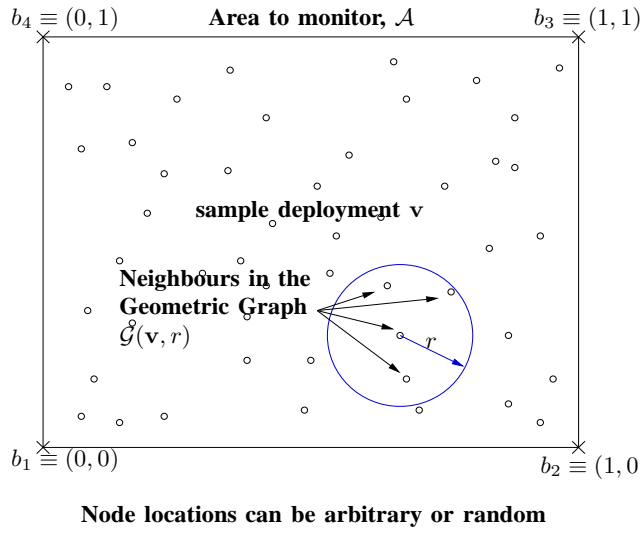


Figure 3.1: An example deployment with the 4 anchors.

Figure 3.2: Graphical illustration of  $\overline{D}_l(\mathbf{v}, h_l)$  and  $\underline{D}_l(\mathbf{v}, h_l)$ .

#### Notation:

- $\mathcal{N} = [n] = \{1, 2, \dots, n\}$ , the index set of the nodes, i.e., node  $i \in \mathcal{N}$  has a location  $v_i$  on  $\mathcal{A}$ .
- $b_l$  = Location of the  $l^{th}$  anchor node,  $l = 1, \dots, L$ , e.g., in Figure 3.1,  $L = 4$ .
- $H_{l,i}(\mathbf{v})$  = Minimum number of hops of node  $i$  from anchor  $b_l$  on the graph  $\mathcal{G}(\mathbf{v}, r)$  for the deployment  $\mathbf{v}$ .
- $D_{l,i}(\mathbf{v})$  = Euclidean distance of node  $i$  from anchor  $b_l$  for the deployment  $\mathbf{v}$ .

$$\overline{D}_l(\mathbf{v}, h_l) = \max_{\{i \in \mathcal{N} : H_{l,i}(\mathbf{v}) = h_l\}} D_{l,i}(\mathbf{v})$$

$$\underline{D}_l(\mathbf{v}, h_l) = \min_{\{i \in \mathcal{N} : H_{l,i}(\mathbf{v}) = h_l\}} D_{l,i}(\mathbf{v})$$

A graphical illustration of the above two quantities is given in Figure 3.2.

- $\mathcal{N}_j = \{k \in \mathcal{N} : \|v_j - v_k\| \leq r, k \neq j\}$ ,  $j \in \mathcal{N}$ . This is the neighbour set of node  $j$  in  $\mathcal{G}(\mathbf{v}, r)$ .

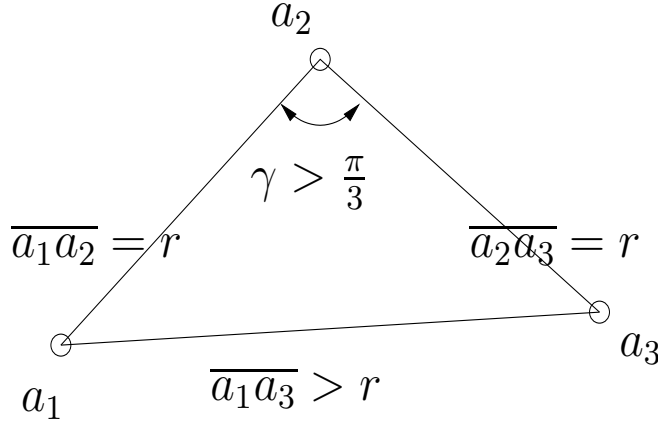


Figure 3.3: Condition for sequential neighbours in arbitrary geometric graph.

With this setting, given the hop distance  $h_l$  on  $\mathcal{G}(\mathbf{v}, r)$  between a node and an anchor, we wish to obtain constraints on the Euclidean distance of the node from anchor  $b_l$ . We define a sequence of nodes  $\{a_1, a_2, \dots, a_K\}$ , where all  $a_i \in \mathcal{N}, i = 1, \dots, K$ , as *sequential neighbours* iff

$$\mathcal{N}_{a_i} = \begin{cases} \{a_{i-1}, a_{i+1}\} & \text{for } i = 2, \dots, K-1 \\ \{a_{i+1}\} & \text{for } i = 1 \\ \{a_{i-1}\} & \text{for } i = K \end{cases}$$

We can observe if  $\{a_1, a_2, a_3\}$  have the following properties,  $\overline{a_1 a_2} = r$ ,  $\overline{a_2 a_3} = r$  and  $\overline{a_1 a_3} > r$ ,<sup>1</sup> they will be sequential neighbours on  $\mathcal{G}(\mathbf{v}, r)$  (See Figure 3.3). Then, by using cosine law in  $\triangle a_1 a_2 a_3$ , we get,  $\overline{a_1 a_3} = \sqrt{r^2 + r^2 - 2r \cdot r \cos \gamma} = r\sqrt{2(1 - \cos \gamma)} > r$ , for  $\gamma > \frac{\pi}{3}$ . Now, to find a bound on ED for HD  $h_l$ , we construct a regular polygon with  $h_l + 1$  sides, all with length  $r$ , as shown in Figure 3.4. We know that the total interior angle is  $(h_l + 1 - 2)\pi = (h_l - 1)\pi$ . Hence each angle is  $\frac{(h_l - 1)\pi}{h_l + 1}$ . We see that,  $\frac{(h_l - 1)\pi}{h_l + 1} > \frac{\pi}{3}$ , iff  $h_l > 2$ . So, for all hop distances  $h_l \geq 3$ , each of the internal angles will be  $> \frac{\pi}{3}$ . Now we pick two adjacent nodes  $s$  and  $d$  as shown in Figure 3.4. We want the hop distance between them to be  $h_l$ , we delete the edge  $sd$  and increase all the other angles by a very small amount  $\delta$ , hence we get the node sequence as shown in Figure 3.5. This sequence of nodes will be sequential neighbours iff the ED between  $s$  and  $d$ , i.e.  $r_1$ , becomes  $> r$ .

---

<sup>1</sup>Define,  $\overline{a_j a_k} = \|v_{a_j} - v_{a_k}\|$

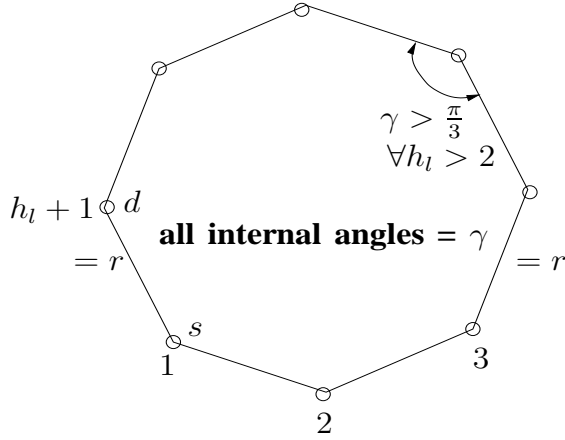
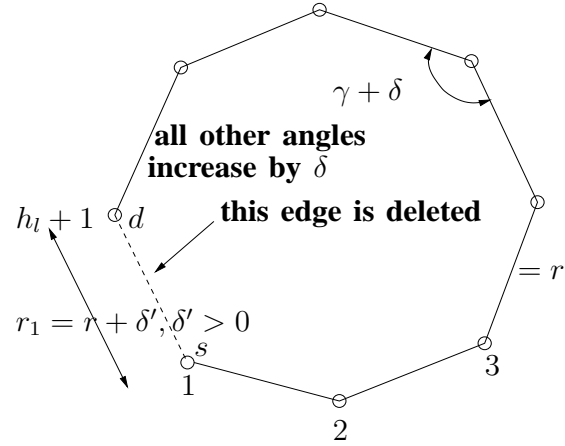
A regular  $h_l + 1$  sided polygonhop distance between  $s$  and  $d = h_l$ 

Figure 3.4: Construction to find the lower bound on Euclidean distance.

Figure 3.5: Achievability of the lower bound.

Now the following lemma says that for a certain choice of  $\delta$ ,  $r_1 > r$  with the nodes on the path from  $s$  to  $d$  being sequential neighbours.

**Lemma 1** For  $h_l > 2$  and  $0 < \delta < \frac{4\pi}{h_l+1}$ ,  $r_1 > r$

**Proof:** In Appendix A.

Hence  $s$  and  $d$  cease to be neighbours and the nodes in the path from  $s$  to  $d$  still follow the properties of being sequential neighbours. Assuming other nodes to be more than  $r$  from all the nodes of this set of  $h_l + 1$  nodes, the hop distance between  $s$  and  $d$  becomes  $h_l$ , however the Euclidean distance between them is just more than  $r$ , for  $h_l \geq 3$ . For  $h_l = 2$ , a construction similar to Figure 3.3 can be done to show that the distance between  $s = a_1$  and  $d = a_3$  is just a little more than  $r$ . Hence for any arbitrary geometric graph in 2-dimensions, given the hop distance between a node and an anchor being  $h_l \geq 2$ , the Euclidean distance can be arbitrarily close but more than  $r$ , which is a trivial lower bound. The upper bound on ED remains  $h_l r$  as usual, which can be achieved by placing the nodes on a straight line  $r$  distance from each other to form a set of  $h_l + 1$  sequential neighbours. Hence, we have proved the following lemma.

**Lemma 2** For arbitrary  $\mathbf{v}$  and  $h_l \geq 2$ ,  $r < \underline{D}_l(\mathbf{v}, h_l) \leq \overline{D}_l(\mathbf{v}, h_l) \leq h_l r$  and both bounds are sharp.



Hence the hop distance in an arbitrary geometric graph on a plane does not provide useful information about the Euclidean distance between the nodes.

However, the situation changes when the distribution of nodes has positive density over all points on  $\mathcal{A}$ , e.g., the node distribution is Uniform i.i.d. or Randomised Lattice. As we will find out in the following sections, in a random geometric graph with a sufficient number of nodes, the hop-distance serves as a good measure of the Euclidean distance.

### 3.4 ED-HD proportionality in Random Geometric Graph (RGG)

In this chapter, we formalise the notion of proportionality between the hop distance (HD) between any two points (not necessarily nodes) on a unit area  $\mathcal{A}$  and the Euclidean distance (ED) between them. The nodes are distributed in a uniform i.i.d. fashion over  $\mathcal{A}$ , i.e., the location of each node is uniformly distributed over  $\mathcal{A}$ , independent of the locations of the other nodes (a formal definition is given later). On such a deployment of nodes, we consider the RGG with radius  $r(n) = c\sqrt{\frac{\ln n}{n}}$ ,  $c > \frac{1}{\sqrt{\pi}}$ , which ensures connectedness of the RGG with probability approaching 1, as  $n \rightarrow \infty$  (Gupta and Kumar [9]). The notion of hop distance between a pair of arbitrary points  $b_1$  and  $b_2$  separated by Euclidean distance  $d$  is illustrated in Figure 3.6. A node that falls within a radius of  $r(n)$  from point  $b_i$ ,  $i = 1, 2$ , is connected to  $b_i$ , and this is counted as one hop. For a connected RGG, there is at least one path between these two nodes, and hence we get a path between the points  $b_1$  and  $b_2$ . Connectivity of the RGG is ensured by the choice of  $r(n)$  and existence of at least one node within a radius  $r(n)$  of  $b_i$  is ensured by Lemma 3. The minimum number of hops for all such possible paths is called the hop-distance between  $b_1$  and  $b_2$ . In this setting, we show that the hop distance is nearly proportional to the Euclidean distance in the following sense. For each  $\epsilon$ ,  $0 < \epsilon < 1$ , if  $r(n) = c(\epsilon)\sqrt{\frac{\ln n}{n}}$ , for an appropriate choice of  $c(\epsilon)$ , the probability that the hop distance lies in the interval  $\left[\frac{d}{r(n)}, \frac{d}{(1-\epsilon)r(n)}\right)$  goes to 1 as  $n \rightarrow \infty$ .

In Section 3.5, we establish the result for two *fixed* points  $b_1$  and  $b_2$  separated by a distance  $d$  on  $\mathcal{A}$ . A construction ensures a path of  $\frac{d}{(1-\epsilon)r(n)}$  hops w.h.p.<sup>2</sup>; hence the hop distance is no more

---

<sup>2</sup>w.h.p. (with high probability) means that the probability of the said event  $\rightarrow 1$  as  $n \rightarrow \infty$

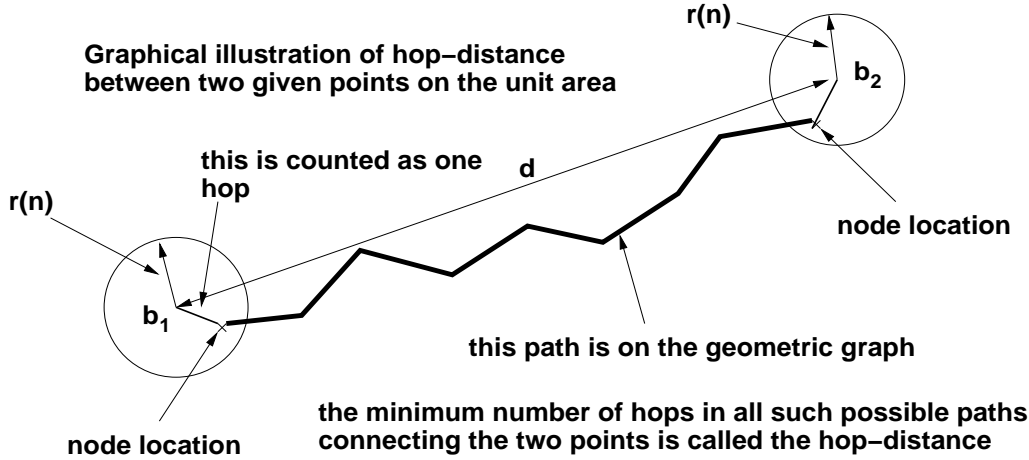


Figure 3.6: Illustration of *hop distance*. Starting with each point  $b_i$ , we first seek a node within the radius  $r(n)$  of  $b_i$ .

than this quantity w.h.p.. The lower bound,  $\frac{d}{r(n)}$ , follows trivially by the triangle inequality. This gives us Theorem 1. In Section 3.6, we generalise the result to hold simultaneously for *all pairs* of points  $(b_1, b_2)$  separated by the distance  $d$  (Theorem 2). Finally in Section 3.7 we show that, the constructions made to prove Theorem 2 extend to yield Theorem 3, which is the generalisation of this result for any pair of points on  $\mathcal{A}$ . We note here that Khude *et al.* [14] have shown that the hop distance lies w.h.p. in the interval  $\left[\frac{d}{r(n)}, a\frac{d}{r(n)}\right)$  for a fixed number  $a > 1$ , and hence their result does not provide a means to control the accuracy of the approximation between hop distance and Euclidean distance. Here we show that the constant  $a$  can be made arbitrarily close to 1 by appropriate choice of  $\epsilon$ .

## 3.5 Proportionality for Two Fixed Points Separated by a Distance $d$

### Setting:

- $n$  nodes are deployed on a unit area  $\mathcal{A}$  in the uniform i.i.d.<sup>3</sup> fashion. The random node

---

<sup>3</sup> $\Pr(i^{th} \text{ node falls in } \mathcal{A}_1 \subseteq \mathcal{A}) = \frac{|\mathcal{A}_1|}{|\mathcal{A}|}, \forall i$ , independent of all nodes  $j \neq i$ , where  $|\mathcal{A}|$  is the area of the region  $\mathcal{A}$ . Deployment of this kind is called uniform i.i.d. deployment.

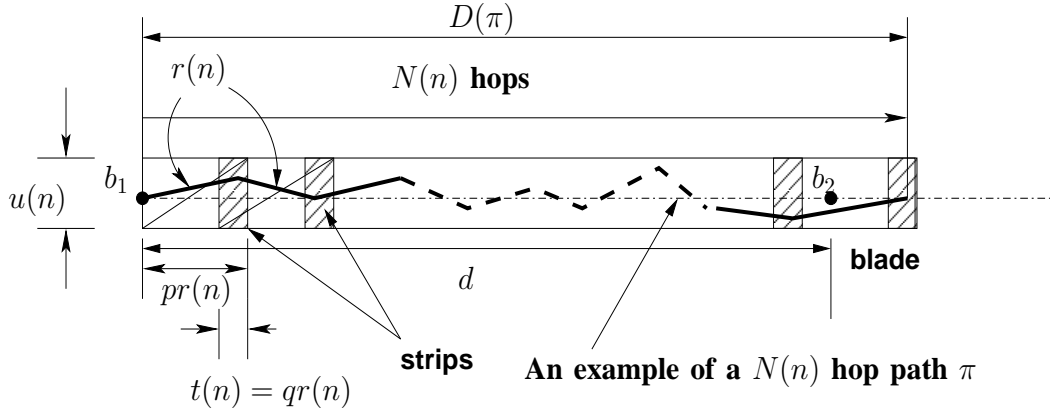


Figure 3.7: The construction of a path taking the line joining  $b_1$  and  $b_2$  as the axis. We are looking at the distance traveled along this path in  $N(n)$  hops.

locations are denoted by the random vector  $\mathbf{V} \in \mathcal{A}^n$ , with a particular realisation being denoted by  $\mathbf{v}$ . We denote by  $\mathbb{P}^n(\cdot)$  the probability measure on  $\mathcal{A}^n$  so obtained.

- We form the RGG  $\mathcal{G}(\mathbf{v}, r(n))$  by connecting the nodes that are within the radius  $r(n)$  of each other, where  $r(n)$ , the radius of the geometric graph is chosen so that the network remains asymptotically connected. We take  $r(n) = c\sqrt{\frac{\ln n}{n}}$ ,  $c > \frac{1}{\sqrt{\pi}}$ , a constant; this ensures asymptotic connectivity (see [9]).

#### Notation:

- Fix two points  $b_1$  and  $b_2$  in  $\mathcal{A}$  such that  $\overline{b_1 b_2} = d$ , where  $\overline{b_1 b_2}$  denotes the Euclidean distance between the points  $b_1$  and  $b_2$ .
- If a node exists within a radius of  $r(n)$  from each of the points  $b_i$ , then we obtain a path between these nodes in the RGG  $\mathcal{G}(\mathbf{v}, r(n))$  (see Figure 3.6). The hop count of such a path is the number of hops traversed on the RGG, plus 2. Let  $\mathcal{P}(\mathbf{v}, r(n))$  denote the set of all such paths between  $b_1$  and  $b_2$  for the deployment  $\mathbf{v}$ . Then we define the hop distance between  $b_1$  and  $b_2$ , for the node deployment  $\mathbf{v}$ , by

$$H_{b_1 b_2}(\mathbf{v}) = \begin{cases} \min\{\text{hop count for paths in } \mathcal{P}(\mathbf{v}, r(n))\} \\ n + 2, \text{ if } \mathcal{P}(\mathbf{v}, r(n)) \text{ is empty} \end{cases}$$

The following lemma assures that there exists at least one node within a radius of  $r(n)$  for any given points  $b_1$  and  $b_2$ , w.h.p.. Define,  $B_i = \{\mathbf{v} : \exists \text{ at least one node within a radius of } r(n) \text{ from } b_i\}$ ,  $i = 1, 2$ .

**Lemma 3**  $\lim_{n \rightarrow \infty} \mathbb{P}^n(B_1 \cap B_2) = 1$

**Proof:**  $\mathbb{P}^n(B_1 \cap B_2) = 1 - \mathbb{P}^n(B_1^c \cup B_2^c)$ , and  $\mathbb{P}^n(B_1^c \cup B_2^c) \leq \mathbb{P}^n(B_1^c) + \mathbb{P}^n(B_2^c) = (1 - \pi r^2(n))^n + (1 - \pi r^2(n))^n \leq 2e^{-n\pi r^2(n)} \xrightarrow{n \rightarrow \infty} 0$  since  $r(n) = c\sqrt{\frac{\ln n}{n}}$ . ■

This Lemma tells us that the set  $\mathcal{P}(\mathbf{v}, r(n))$  will be non-empty w.h.p.. Given  $\epsilon$ ,  $0 < \epsilon < 1$ , we will write  $c(\epsilon)$  to denote the dependence of  $c$  on  $\epsilon$ , and then we will write  $r(n, \epsilon) = c(\epsilon)\sqrt{\frac{\ln n}{n}}$ . Now define the event  $E_1 = \left\{ \mathbf{v} : \frac{d}{r(n, \epsilon)} \leq H_{b_1 b_2}(\mathbf{v}) < \frac{d}{(1-\epsilon)r(n, \epsilon)} \right\}$ . The following theorem states that if  $c(\epsilon)$  is chosen appropriately then  $\mathbb{P}^n(E_1) \rightarrow 1$  as  $n \rightarrow \infty$ , i.e., w.h.p.,  $H_{b_1 b_2}(\mathbf{v})$  is proportional to  $d (= \overline{b_1 b_2})$ , where the proportionality constant is  $\frac{1}{r(n, \epsilon)}$ , but for an error that can be made arbitrarily small by choosing  $\epsilon$  to be small.

**Theorem 1** For fixed  $b_1$  and  $b_2$  s.t.  $\overline{b_1 b_2} = d$ , for all  $\epsilon$ ,  $1 > \epsilon > 0$ , if  $c^2(\epsilon) \geq \frac{1}{2q\sqrt{1-p^2}}$ , where  $p$  and  $q$  are any two constants satisfying  $1-\epsilon < p < 1$  and  $0 < q < p-(1-\epsilon)$ , then  $\lim_{n \rightarrow \infty} \mathbb{P}^n(E_1) = 1$ .

**Discussion:** We see that as  $\epsilon$  is made small (in order to achieve greater accuracy in the proportionality between hop distance and  $d$ )  $p$  becomes closer to 1, and  $q$  closer to 0, thus increasing  $c(\epsilon)$ , and hence making  $r(n, \epsilon)$  larger than the critical radius for connectedness, i.e.,  $\sqrt{\frac{\ln n}{\pi n}}$ . Now, if it is desired that  $r(n, \epsilon) \leq r_{\max}$ , for some given  $r_{\max}$ , then  $n$  can be appropriately chosen to achieve this. If  $n$  is so chosen (and is large enough for the asymptotics to “kick in”) then we can expect to get proportionality of hop distance and  $d$  in the sense of the theorem, and also a distance discretisation accuracy of  $r_{\max}$ . We see that, due to the  $c(\epsilon)$  factor, we may need  $n$  to be larger than if the only objective was that of connectedness of the RGG.

**Proof:** The proof proceeds via a few lemmas. First, we define some quantities that will be needed in the lemmas and then state and prove the lemmas. The theorem will follow thereafter.

We are given an  $\epsilon > 0$ . We take an  $\epsilon' < \epsilon$  and define,  $N(n) = \left\lceil \frac{d}{(1-\epsilon')r(n)} \right\rceil$ . So,

$$\frac{d}{(1-\epsilon')r(n)} \leq N(n) < \frac{d}{(1-\epsilon')r(n)} + 1 \quad (3.1)$$

The construction in Figure 3.7 is done taking the straight line joining points  $b_1$  and  $b_2$  as the axis.  $b_1$  is at the centre of the left side of the rectangle and  $b_2$  is at a distance  $d$  along the horizontal axis of the rectangle. The rectangular box up to the  $N(n)^{th}$  strip, hereafter referred to as a *blade* and denoted by  $\mathcal{B}(b_1, b_2)$ , has been drawn along that axis and has shaded strips of width  $qr(n)$ . The maximum distance between any two points in consecutive shaded strips is  $r(n)$  as shown in the figure. We will show that  $b_2$  is reachable (by reachable we mean that there exists a path as shown in Figure 3.6 by which we can reach  $b_2$  from  $b_1$ ) from  $b_1$  in  $N(n)$  hops w.h.p.. If there exists at least one node in all the  $N(n)$  consecutive shaded strips starting from  $b_1$ , we can connect these consecutive nodes each with one hop and consequently get a  $N(n)$  hop path as shown in Figure 3.7. Define,

- $\pi(b_1, \mathcal{B}(b_1, b_2), N(n), \mathbf{v})$  : a path obtained by hopping over the nodes of the  $N(n)$  consecutive shaded strips in the blade  $\mathcal{B}(b_1, b_2)$  starting from point  $b_1$  for a sample deployment  $\mathbf{v}$ , see Figure 3.7. For brevity of notation, we will use  $\pi$  to denote such a path.
- $D(\pi(b_1, \mathcal{B}(b_1, b_2), N(n), \mathbf{v}))$  : Euclidean distance traveled along the axis of the blade  $\mathcal{B}(b_1, b_2)$  by the path  $\pi(b_1, \mathcal{B}(b_1, b_2), N(n), \mathbf{v})$ , as shown in Figure 3.7. To make the notation simple, we will use  $D(\pi)$  to denote this distance.
- $A_i = \{\mathbf{v} : \exists \text{ at least one node in the } i^{th} \text{ strip of the blade } \mathcal{B}(b_1, b_2)\}$

In each hop at least a distance of  $(p - q)r(n)$  is traveled along the blade. If there exists at least one node in each of the  $N(n)$  consecutive shaded strips starting from  $b_1$ , a distance of at least  $N(n)(p - q)r(n)$  is traveled along  $\mathcal{B}(b_1, b_2)$  in  $N(n)$  hops. Hence,

$$\begin{aligned} & \bigcap_{i=1}^{N(n)} \{A_i\} \\ & \subseteq \{ \mathbf{v} : \exists \text{ a path } \pi(b_1, \mathcal{B}(b_1, b_2), N(n), \mathbf{v}) \text{ s.t. } N(n)(p - q)r(n) \leq D(\pi(b_1, \mathcal{B}(b_1, b_2), N(n), \mathbf{v})) \} \end{aligned} \quad (3.2)$$

Which implies,

$$\begin{aligned}
& \mathbb{P}^n \{ \mathbf{v} : \exists \text{ a path } \pi(b_1, \mathcal{B}(b_1, b_2), N(n), \mathbf{v}) \text{ s.t.} \\
& N(n)(p - q)r(n) \leq D(\pi(b_1, \mathcal{B}(b_1, b_2), N(n), \mathbf{v})) \} \\
& \geq \mathbb{P}^n \left( \bigcap_{i=1}^{N(n)} \{A_i\} \right) \\
& = 1 - \mathbb{P}^n \left( \bigcup_{i=1}^{N(n)} \{A_i^c\} \right) \\
& \geq 1 - \sum_{i=1}^{N(n)} \mathbb{P}^n (A_i^c)
\end{aligned} \tag{3.3}$$

Where the first inequality follows from Equation 3.2, the equality is due to De-Morgan's Theorem and the second inequality follows from the Union Bound. We will now prove the following lemma:

**Lemma 4**  $\lim_{n \rightarrow \infty} \sum_{i=1}^{N(n)} \mathbb{P}^n (A_i^c) = 0$ , if and only if  $c^2 \geq \frac{1}{2q\sqrt{1-p^2}}$ .

**Proof:** Using the fact that node deployment is uniform i.i.d. over the unit area of  $\mathcal{A}$ ,

$$\begin{aligned}
& \sum_{i=1}^{N(n)} \mathbb{P}^n (A_i^c) \\
& = N(n)(1 - u(n)t(n))^n \\
& < \left( \frac{d}{(1 - \epsilon')r(n)} + 1 \right) (1 - u(n)t(n))^n \\
& \leq \left( \frac{d}{(1 - \epsilon')r(n)} + 1 \right) e^{-nu(n)t(n)} \\
& = \frac{d}{(1 - \epsilon')c} \frac{n^{\frac{1}{2} - c^2 q \sqrt{1-p^2}}}{\sqrt{\ln n}} + e^{-nc^2 q \sqrt{1-p^2} \frac{\ln n}{n}} \\
& = \frac{d}{(1 - \epsilon')c} \sqrt{\frac{n^{1-2c^2 q \sqrt{1-p^2}}}{\ln n}} + e^{-c^2 q \sqrt{1-p^2} \ln n}
\end{aligned} \tag{3.4}$$

$r(n) = c\sqrt{\frac{\ln n}{n}}$ ,  $c$  a constant, and by construction,  $u(n) = \sqrt{1 - p^2}r(n)$ ,  $t(n) = qr(n)$ . Here the first inequality follows from (3.1). In the second inequality, we have used the fact  $1 - x \leq e^{-x}$ . The second term in the above equation goes to zero as  $n \rightarrow \infty$ . To drive the first term to zero, it is necessary and sufficient to take  $1 - 2c^2 q \sqrt{1 - p^2} \leq 0$ , i.e.  $c^2 \geq \frac{1}{2q\sqrt{1-p^2}}$ , which proves the lemma. ■

Thus, under the hypothesis of Lemma 4, R.H.S. of Equation 3.3 goes to 1. Now we take  $p - q = 1 - \epsilon' > 1 - \epsilon$ , as  $\epsilon' < \epsilon$  for the given  $\epsilon > 0$ , this is possible since  $1 > p > q > 0$ , by the construction. Which is same as  $1 - \epsilon < p < 1$  and  $0 < q < p - (1 - \epsilon)$ . Then we have, using (3.1),

$$N(n)(p - q)r(n) = N(n)(1 - \epsilon')r(n) \geq d$$

So, when  $\mathbf{v} \in \cap_{i=1}^{N(n)} \{A_i\}$ , by such a choice of  $p, q$ , we travel at least a distance of  $d$  from  $b_1$  along the blade in  $N(n)$  hops. Since all the points within the blade up to the  $N(n)^{th}$  strip are reachable,  $b_2$  is also reachable from  $b_1$  in  $N(n)$  hops, see Figure 3.8. Hence the hop distance between  $b_1$  and  $b_2$  can be no more than  $N(n)$ .

$$\begin{aligned} & \cap_{i=1}^{N(n)} \{A_i\} \\ & \subseteq \{ \mathbf{v} : \exists \text{ a path } \pi \text{ s.t. } D(\pi) \geq N(n)(p - q)r(n) \} \\ & = \{ \mathbf{v} : \exists \text{ a path } \pi \text{ s.t. } D(\pi) \geq N(n)(p - q)r(n) \geq d \} \\ & = \{ \mathbf{v} : b_2 \text{ is reachable from } b_1 \text{ in } N(n) \text{ hops} \} \\ & \subseteq \{ \mathbf{v} : H_{b_1 b_2}(\mathbf{v}) \leq N(n) \} \end{aligned} \tag{3.5}$$

$$\Rightarrow \mathbb{P}^n \left( \cap_{i=1}^{N(n)} \{A_i\} \right) \leq \mathbb{P}^n \{ \mathbf{v} : H_{b_1 b_2}(\mathbf{v}) \leq N(n) \} \tag{3.6}$$

The second equality holds from the fact that existence of such a path  $\pi$  will guarantee reachability of  $b_2$  from  $b_1$  in  $N(n)$  hops, see Figure 3.8. So, from this result together with Lemma 4, we get the following lemma:

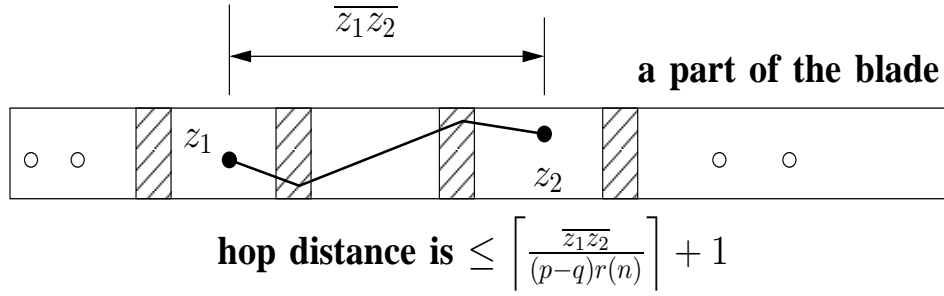
**Lemma 5** *If  $c^2 \geq \frac{1}{2q\sqrt{1-p^2}}$ , then  $\lim_{n \rightarrow \infty} \mathbb{P}^n \{ \mathbf{v} : H_{b_1 b_2}(\mathbf{v}) \leq N(n) \} = 1$ .*

■

In Lemma 4 we got a bound on  $c$  that depends on the given  $\epsilon$ . So,  $c$  and consequently  $r(n)$  becomes function of  $\epsilon$ . We invoke that dependence by changing  $c$  to  $c(\epsilon)$  and  $r(n)$  to  $r(n, \epsilon)$ . Where,  $r(n, \epsilon) = c(\epsilon)\sqrt{\frac{\ln n}{n}}$ . So, from this point onwards, we will use  $r(n, \epsilon)$  instead of  $r(n)$  to denote that all the subsequent results are true for  $r(n, \epsilon)$ , where  $c(\epsilon)$  satisfies  $c^2(\epsilon) \geq \frac{1}{2q\sqrt{1-p^2}}$ , where  $p$  and  $q$  are any two constants satisfying  $1 - \epsilon < p < 1$  and  $0 < q < p - (1 - \epsilon)$ .





Figure 3.9: Corollary for all pairs of points  $z_1$  and  $z_2 \in \mathcal{B}(b_1, b_2)$ 

Hence the theorem. ■

**Extension:** We see from Equation 3.5, that by ensuring the existence of at least one node in each of the  $N(n)$  strips, not only we have a path of  $N(n)$  hops between  $b_1$  and  $b_2$ , but also have for all pairs of points  $z_1$  and  $z_2 \in \mathcal{B}(b_1, b_2)$  (see Figure 3.9), with the Euclidean distance between them being denoted by  $\overline{z_1 z_2}$ , the hop-distance will be  $\leq \left\lceil \frac{\overline{z_1 z_2}}{(p-q)r(n, \epsilon)} \right\rceil + 1 = \left\lceil \frac{\overline{z_1 z_2}}{(1-\epsilon')r(n, \epsilon)} \right\rceil + 1$ , i.e.

$$\{\cap_{i=1}^{N(n)} A_i\} \subseteq \left\{ \mathbf{v} : \forall z_1, z_2 \in \mathcal{B}(b_1, b_2), H_{z_1 z_2}(\mathbf{v}) \leq \left\lceil \frac{\overline{z_1 z_2}}{(1-\epsilon')r(n, \epsilon)} \right\rceil + 1 \right\} \quad (3.8)$$

Now as in Equation 3.7, we can show the following,

$$\begin{aligned} \left\lceil \frac{\overline{z_1 z_2}}{(1-\epsilon')r(n, \epsilon)} \right\rceil + 1 &< \left( \frac{\overline{z_1 z_2}}{(1-\epsilon')r(n, \epsilon)} + 1 \right) + 1 \\ &= \frac{\overline{z_1 z_2}}{(1-\epsilon)r(n, \epsilon)} \left( \frac{1-\epsilon}{1-\epsilon'} + \frac{2(1-\epsilon)r(n, \epsilon)}{\overline{z_1 z_2}} \right) \\ &\leq \frac{\overline{z_1 z_2}}{(1-\epsilon)r(n, \epsilon)} \quad \forall n \geq N_\epsilon \end{aligned} \quad (3.9)$$

The triangle inequality holds for all deployment  $\mathbf{v}$ , hence  $\frac{\overline{z_1 z_2}}{r(n, \epsilon)} \leq H_{z_1 z_2}(\mathbf{v})$ ,  $\forall \mathbf{v} \in \mathcal{A}^n$ . So, we can state the above result in the following corollary,

**Corollary 1** For all  $\epsilon$ ,  $1 > \epsilon > 0$ , if  $c^2(\epsilon) \geq \frac{1}{2q\sqrt{1-p^2}}$ , where  $p$  and  $q$  are any two constants satisfying  $1 - \epsilon < p < 1$  and  $0 < q < p - (1 - \epsilon)$ , then

$$\lim_{n \rightarrow \infty} \mathbb{P}^n \left\{ \mathbf{v} : \forall z_1, z_2 \in \mathcal{B}(b_1, b_2), \frac{\overline{z_1 z_2}}{r(n, \epsilon)} \leq H_{z_1 z_2}(\mathbf{v}) < \frac{\overline{z_1 z_2}}{(1-\epsilon)r(n, \epsilon)} \right\} = 1$$

■

### 3.6 Generalisation for all Pairs of Points at a Distance $d$ apart on a Unit Square

Now we generalise the result so that it holds simultaneously for all pairs of points  $b_1, b_2$  in a unit square  $\mathcal{A}$ , with  $\overline{b_1 b_2} = d$ . For that we split the area  $\mathcal{A}$  into small squarelets of side  $\frac{r(n)}{\sqrt{2}}$  as shown in Figure 3.10, so that the diagonal of each squarelet is  $r(n)$ . From the centre of each squarelet, we construct a circle of radius  $d + \frac{r(n)}{2}$  and segment the entire circle by blades, as shown in Figure 3.10. We start with one blade. It will cover some portion of the circumference of the circle of radius  $d + \frac{r(n)}{2}$ . We construct the next blade so that it covers the adjacent portion of the circumference that has not been covered by the previous blade. We go on constructing these blades until the entire the circle is covered. In each blade we construct strips in the same way we did previously. The only difference of these blades with that in Figure 3.7 is that here we have a strip right in the beginning of the blade, which was not there in the previous construction. So, there are  $N(n) + 1$  strips now in each blade.

Now we choose in  $\mathcal{A}$ , any two points  $b_1$  and  $b_2$  at a distance  $d$ . Point  $b_1$  will fall in some squarelet. If we ensure that each shaded strip of each blade drawn from the centre of each squarelet is nonempty, we can travel in one hop from  $b_1$  to any node in the first strip (since it is within  $r(n)$  from all points in the squarelet, under the condition as explained in Figure 3.11) of the blade that contains  $b_2$  and travel along that blade to reach  $b_2$ , as shown in Figure 3.10. The radius of the circle drawn from the centre of each squarelet is  $d + \frac{r(n)}{2}$  since no  $b_2$  that is at a distance  $d$  from any  $b_1$  in the squarelet can be beyond this range, and by ensuring the existence of at least a node in each strip of each blade, all points inside and on this circle is reachable in  $N(n)$  hops (As the radius of the circle has increased to  $d + \frac{r(n)}{2}$ , we need to redefine  $N(n) = \left\lceil \frac{d + \frac{r(n)}{2}}{(1-\epsilon')r(n)} \right\rceil$ ). Hence  $b_2$  is reachable from  $b_1$  in  $N(n) + 1$  hops and the hop distance between  $b_1$  and  $b_2$  can be no more than  $N(n) + 1$ .

Some notations are in order,



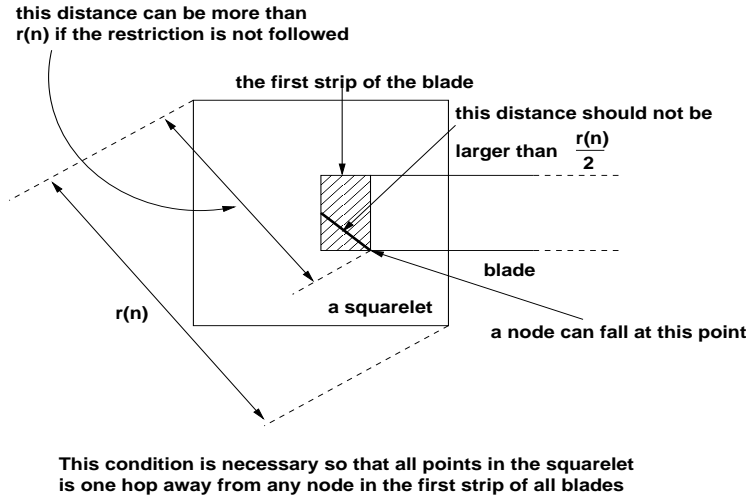
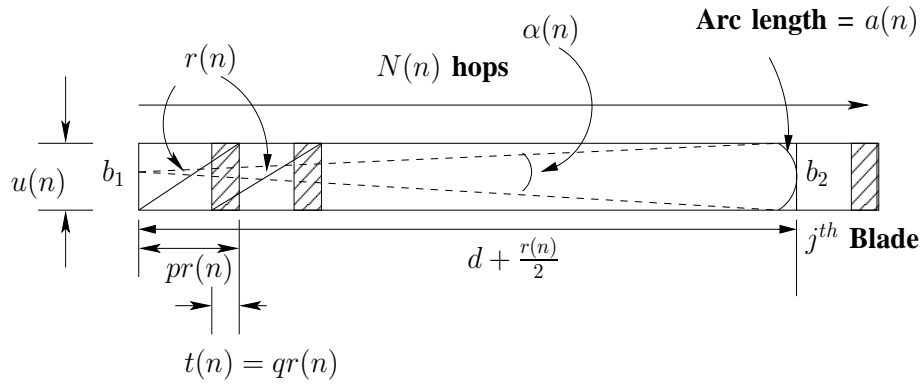


Figure 3.11: Detailed view of a squarelet and the first strip.

Figure 3.12: Let the angle subtended by the arc at  $b_1$  be  $\alpha(n)$ 

- $\mathcal{B}_{k,j} : j^{th}$  blade drawn from the centre of the  $k^{th}$  squarelet of the unit square.
- $J(n) : \text{Number of blades that cover the circle of radius } d + \frac{r(n)}{2} \text{ drawn from the centre of each squarelet.}$
- $N(n) = \left\lceil \frac{d + \frac{r(n)}{2}}{(1-\epsilon')r(n)} \right\rceil, \epsilon' < \epsilon$ , the number of strips in each blade is now  $N(n) + 1$ .
- $K(n) : \text{Number of squarelets in the unit square.}$
- $a(n)$  is the length of the arc of radius  $d$  that lies within a blade, drawn taking  $b_1$  as centre, as shown in Figure 3.12.
- $\alpha(n) : \text{angle subtended by } a(n) \text{ at } b_1$ , see Figure 3.12.

- $\pi_{k,j}(b_1, \mathcal{B}_{k,j}, N(n), \mathbf{v})$  : a path obtained by hopping over the nodes of the  $N(n)$  consecutive shaded strips in the blade  $\mathcal{B}_{k,j}$  starting from the node in the first strip that connects point  $b_1$  for a sample deployment  $\mathbf{v}$ . For brevity of notation, we will use  $\pi_{k,j}$  to denote the same.
- $D(\pi_{k,j}(b_1, \mathcal{B}_{k,j}, N(n), \mathbf{v}))$  : Euclidean distance traveled along the axis of the blade  $\mathcal{B}_{k,j}$  by the path  $\pi_{k,j}(b_1, \mathcal{B}_{k,j}, N(n), \mathbf{v})$ . To make the notation simple, we will use  $D(\pi_{k,j})$  to denote the same.
- $A_{k,j,i} = \{\mathbf{v} : \exists \text{ at least one node in the } i^{th} \text{ strip of } \mathcal{B}_{k,j}\}$
- $\overline{H}(d, \mathbf{v}) = \sup_{\{(b_1, b_2) \in \mathcal{A}^2 : \overline{b_1 b_2} = d\}} H_{b_1 b_2}(\mathbf{v})$
- $\underline{H}(d, \mathbf{v}) = \inf_{\{(b_1, b_2) \in \mathcal{A}^2 : \overline{b_1 b_2} = d\}} H_{b_1 b_2}(\mathbf{v})$

Now, we choose  $p - q = 1 - \epsilon'$  as before.

$$\begin{aligned}
 & \{\cap_{k,j,i} A_{k,j,i}\} \\
 & \subseteq \{\mathbf{v} : \exists \text{ a path } \pi_{k,j}, b_2 \in \mathcal{B}_{k,j} \text{ s.t. } D(\pi_{k,j}) \geq N(n)(p - q)r(n)\} \\
 & = \{\mathbf{v} : \exists \text{ a path } \pi_{k,j}, b_2 \in \mathcal{B}_{k,j} \text{ s.t. } D(\pi_{k,j}) \geq N(n)(p - q)r(n) \geq d\} \\
 & \subseteq \{\mathbf{v} : H_{b_1 b_2}(\mathbf{v}) \leq N(n) + 1, \forall (b_1, b_2) \text{ s.t. } \overline{b_1 b_2} = d\} \\
 & = \left\{ \mathbf{v} : \overline{H}(d, \mathbf{v}) < \left( \frac{d + \frac{r(n)}{2}}{(1 - \epsilon')r(n)} + 1 \right) + 1 \right\} \tag{3.10}
 \end{aligned}$$

To ensure that all nodes falling in the first strips of the blades pertaining to a squarelet is within a distance of  $r(n)$  from any point on the squarelet, we need to ensure the diagonal line from the centre of the squarelet to the edge of the strip as shown in Figure 3.11 is  $\leq \frac{r(n)}{2}$ , which gives  $p \geq 2q$ , i.e. no two strips in a blade should overlap. If it is not satisfied, there can be one event for which the distance between the node falling in the first strip and some point on the squarelet can be more than  $r(n)$ , as shown in Figure 3.11.

By similar arguments as in Equation 3.7, it can be shown that  $\exists$  a  $N_\epsilon$  s.t. the upper bound of

$\overline{H}(d, \mathbf{v})$  in Equation 3.10 can be made  $\leq \frac{d}{(1-\epsilon)r(n)}$ ,  $\forall n \geq N_\epsilon$ . We have,

$$\mathbb{P}^n(\cap_{k,j,i} A_{k,j,i}) \geq 1 - \sum_{k=1}^{K(n)} \sum_{j=1}^{J(n)} \sum_{i=1}^{N(n)+1} \mathbb{P}^n(A_{k,j,i}^c) \quad (3.11)$$

The upper limit of  $i$  is now  $N(n) + 1$ , because of the extra strip in the beginning of the blade. Now using  $\alpha(n)$  as shown in Figure 3.12, we get,  $J(n) = \left\lceil \frac{2\pi}{\alpha(n)} \right\rceil$ . Further, recalling the definition of  $a(n)$ ,

$$\begin{aligned} \alpha(n) \left( d + \frac{r(n)}{2} \right) &= a(n) \geq u(n) \Rightarrow \alpha(n) \geq \frac{u(n)}{\left( d + \frac{r(n)}{2} \right)} \\ \Rightarrow J(n) = \left\lceil \frac{2\pi}{\alpha(n)} \right\rceil &\leq \left\lceil \frac{2\pi \left( d + \frac{r(n)}{2} \right)}{\sqrt{1-p^2}r(n)} \right\rceil \end{aligned} \quad (3.12)$$

Also,  $K(n) = \left\lceil \frac{\sqrt{2}}{r(n)} \right\rceil^2$ . We carry out the calculation to find the condition on  $c$  to drive the rightmost quantity in Equation 3.11 to zero without the ceiling on  $J(n)$  and  $K(n)$ , since the condition on  $c$  is the same even with the ceiling. Let us focus on the following term,

$$\begin{aligned} &\sum_{k=1}^{K(n)} \sum_{j=1}^{J(n)} \sum_{i=1}^{N(n)+1} \mathbb{P}^n(A_{k,j,i}^c) \\ &= K(n)J(n)(N(n)+1)(1-u(n)t(n))^n \\ &< \left( \frac{\sqrt{2}}{r(n)} \right)^2 \frac{2\pi \left( d + \frac{r(n)}{2} \right)}{\sqrt{1-p^2}r(n)} \left[ \frac{d + \frac{r(n)}{2}}{(1-\epsilon')r(n)} + 2 \right] (1-u(n)t(n))^n \\ &\leq \left( \frac{\sqrt{2}}{r(n)} \right)^2 \frac{2\pi \left( d + \frac{r(n)}{2} \right)}{\sqrt{1-p^2}r(n)} \left[ \frac{d + \frac{r(n)}{2}}{(1-\epsilon')r(n)} + 2 \right] e^{-q\sqrt{1-p^2}c^2 \ln n} \\ &= \frac{4\pi}{\sqrt{1-p^2}} \left[ \frac{d}{r^3(n)} + \frac{1}{2r^2(n)} \right] \left[ \frac{d}{(1-\epsilon')r(n)} + \frac{1}{2(1-\epsilon')} + 2 \right] e^{-q\sqrt{1-p^2}c^2 \ln n} \\ &= \frac{4\pi}{\sqrt{1-p^2}} \left[ \frac{d^2}{(1-\epsilon')c^4} \frac{n^{2-q\sqrt{1-p^2}c^2}}{(\ln n)^2} + \left( \frac{d}{2(1-\epsilon')} + d \left( \frac{1}{2(1-\epsilon')} + 2 \right) \right) \frac{n^{\frac{3}{2}-q\sqrt{1-p^2}c^2}}{c^3(\ln n)^{\frac{3}{2}}} + \right. \\ &\quad \left. \frac{1}{2} \left( \frac{1}{2(1-\epsilon')} + 2 \right) \frac{n^{1-q\sqrt{1-p^2}c^2}}{c^2 \ln n} \right] \end{aligned} \quad (3.13)$$

To drive this quantity to zero as  $n \rightarrow \infty$ , it is necessary and sufficient to take  $c^2 \geq \frac{2}{q\sqrt{1-p^2}}$ . We

know, for all  $(b_1, b_2) \in \mathcal{A}^2$  s.t.  $\overline{b_1 b_2} = d$  and  $\forall \mathbf{v} \in \mathcal{A}^n$ ,  $H_{b_1 b_2}(\mathbf{v}) \geq \frac{d}{r(n)}$ . Hence,  $\underline{H}(d, \mathbf{v}) \geq \frac{d}{r(n)}$ ,  $\forall \mathbf{v} \in \mathcal{A}^n$ . We also have,  $\forall n \geq N_\epsilon$ ,  $\left( \left\lceil \frac{d + \frac{r(n)}{2}}{(1-\epsilon')r(n)} \right\rceil + 1 \right) + 1 \leq \frac{d}{(1-\epsilon)r(n)}$ . So, defining  $E_3 = \left\{ \mathbf{v} : \frac{d}{r(n, \epsilon)} \leq \underline{H}(d, \mathbf{v}) \leq \overline{H}(d, \mathbf{v}) < \frac{d}{(1-\epsilon)r(n, \epsilon)} \right\}$ , the theorem can be stated as,

**Theorem 2** For all  $\epsilon, 1 > \epsilon > 0$ , if  $c^2(\epsilon) \geq \frac{2}{q\sqrt{1-p^2}}$ , where  $p$  and  $q$  are any two constants satisfying  $1 - \epsilon < p < 1$  and  $0 < q < p - (1 - \epsilon)$  and  $p \geq 2q$ , on a unit square, then  $\lim_{n \rightarrow \infty} \mathbb{P}^n(E_3) = 1$ .

■

### 3.7 Generalisation for all Pairs of Points on a Unit Square

Theorem 2 takes account of any two points on  $\mathcal{A}$  at a fixed distance  $d$ . We can generalise this result for all distances  $\leq d$  using the same line of arguments as in Corollary 1. If we pick any pair of points  $z_1$  and  $z_2$  in  $\mathcal{A}$ , such that  $\overline{z_1 z_2} \leq d$ , we will encounter a situation similar as depicted in Figure 3.10, the points  $b_1$  and  $b_2$  being replaced by  $z_1$  and  $z_2$ ,  $d$  replaced by  $\overline{z_1 z_2}$  and the radius of the circle by  $\overline{z_1 z_2} + \frac{r(n)}{2}$ . Ensuring the existence of at least a node in each of the strips of the blades in the original circle of radius  $d + \frac{r(n)}{2}$ , also ensures a path from  $z_1$  to  $z_2$  with hop-distance being  $\leq \left( \left\lceil \frac{\overline{z_1 z_2} + \frac{r(n)}{2}}{(p-q)r(n)} \right\rceil + 1 \right) + 1 = \left( \left\lceil \frac{\overline{z_1 z_2} + \frac{r(n)}{2}}{(1-\epsilon')r(n)} \right\rceil + 1 \right) + 1$ . The last 1 comes from the first hop from  $z_1$  to the node in the first strip of the blade that contains  $z_2$ . Following the algebra similar to Equation 3.7, it can be shown that  $\left( \left\lceil \frac{\overline{z_1 z_2}}{(1-\epsilon')r(n)} \right\rceil + 1 \right) + 1 < \frac{\overline{z_1 z_2}}{(1-\epsilon)r(n)}$ , for all  $n \geq N_\epsilon$ . So we have,

$$\begin{aligned} & \{\cap_{k,j,i} A_{k,j,i}\} \\ & \subseteq \left\{ \mathbf{v} : \forall z_1, z_2 \in \mathcal{A}, \overline{z_1 z_2} \leq d, H_{z_1 z_2}(\mathbf{v}) < \left( \left\lceil \frac{\overline{z_1 z_2} + \frac{r(n)}{2}}{(1-\epsilon')r(n)} \right\rceil + 1 \right) + 1 \right\} \end{aligned} \quad (3.14)$$

Now we take  $d$  to be equal to the diameter of the area  $\mathcal{A}$  (we define diameter as the maximum of the minimum distance between any two points on  $\mathcal{A}$ , here it turns out to be the diagonal of the square  $\mathcal{A}$ ). Then any pair of points  $z_1$  and  $z_2$  on  $\mathcal{A}$  will have the property  $\overline{z_1 z_2} \leq d$ . So, we can state Theorem 2 in more general setting as,

**Theorem 3** For all  $\epsilon, 1 > \epsilon > 0$ , if  $c^2(\epsilon) \geq \frac{2}{q\sqrt{1-p^2}}$ , where  $p$  and  $q$  are any two constants satisfying  $1 - \epsilon < p < 1$  and  $0 < q < p - (1 - \epsilon)$  and  $p \geq 2q$ , on a unit square,

$$\lim_{n \rightarrow \infty} \mathbb{P}^n \left\{ \mathbf{v} : \forall z_1, z_2 \in \mathcal{A}, \frac{\overline{z_1 z_2}}{r(n, \epsilon)} \leq H_{z_1 z_2}(\mathbf{v}) < \frac{\overline{z_1 z_2}}{(1 - \epsilon)r(n, \epsilon)} \right\} = 1$$

■

### 3.8 Illustration of Theorem 1 through Simulations

In this section we illustrate the result in Theorem 1 via simulation. This also provides some insight into how large  $n$  might need to be for the asymptotics to be useful. We introduce the quantity  $\eta := p - q = 1 - \epsilon' > 1 - \epsilon$ , and thus write  $c(\eta)$  rather than  $c(\epsilon)$ . We can choose  $\epsilon'$  to be very close to  $\epsilon$ , and approximate  $\eta \approx 1 - \epsilon$ . So,  $\eta$  gives the accuracy in the proportionality between hop distance ( $h$ ) and Euclidean distance ( $d$ ). By Theorem 1, for two fixed points  $b_1$  and  $b_2$ , the  $\frac{d}{h}$  ratio will be within  $(\eta r(n), r(n))$  with high probability (since  $h$  being in the interval  $\left[\frac{d}{r(n)}, \frac{d}{\eta r(n)}\right]$ ) implies that  $\frac{d}{h}$  ratio will be within  $(\eta r(n), r(n))$ , if  $c^2(\eta) \geq \frac{1}{2(p-\eta)\sqrt{1-p^2}}$ , where  $0 < \eta < p < 1$  and  $\eta \approx 1 - \epsilon$ .

For a given  $\epsilon$ ,  $\eta$  is fixed, and we choose  $p$ , such that,

$$p = \arg \max_{p > \eta} (p - \eta) \sqrt{1 - p^2} = \frac{\eta + \sqrt{\eta^2 + 8}}{4} \quad (3.15)$$

Which minimises  $c(\eta)$ . The minimum  $c(\eta)$  is denoted by  $c_{min}(\eta)$  and we get,

$$c_{min}^2(\eta) = \frac{8}{(\sqrt{\eta^2 + 8} - 3\eta)\sqrt{8 - 2\eta^2 - 2\eta\sqrt{\eta^2 + 8}}} \quad (3.16)$$

We use this  $c_{min}(\eta)$  in the simulations and the corresponding RGG radius is referred to as  $r(n, \eta)$  and the RGG as  $\mathcal{G}(\mathbf{V}, r(n, \eta))$ .

We take,  $\eta = 0.8$ , which gives  $c_{min}(\eta) = 3.23$ . With  $\mathcal{A}$  being the unit square  $[0, 1]^2$ , we chose two fixed points  $b_1 = (0.1, 0.1)$  and  $b_2 = (0.9, 0.9)$ , hence  $d = \overline{b_1 b_2} = 1.1314$ . Our simulation



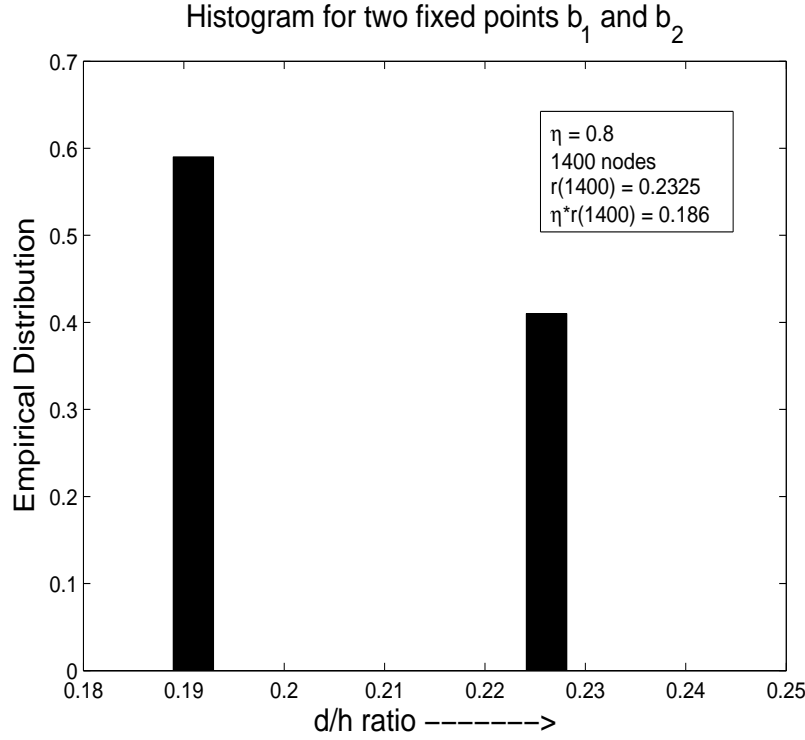


Figure 3.13: Histogram of  $\frac{d}{h}$  ratio for 1400 nodes (100 runs)

deploys  $n$  nodes in the uniform i.i.d. fashion ( $n$  varies as shown in Table 3.1), and builds the RGG with radius  $r(n, \eta)$ . We then obtain the hop distance ( $h$ ) between  $b_1$  and  $b_2$  on this RGG. This is repeated 100 times for each  $n$ , and the histogram of the  $\frac{d}{h}$  ratio, is plotted, taking 200 bins in the interval  $[0, 1]$ . An example histogram for 1400 nodes is shown in Figure 3.13.

We see that the  $\frac{d}{h}$  ratio falls in the bin centred at 0.190, in 59% of the deployments, and in the bin centred at 0.225, in 41% of the deployments. The frequency of occurrences (or the empirical probability mass) of the corresponding hop distances for each  $n$  are summarised in Table 3.1.

From Table 3.1 we see that for small  $n$ ,  $H_{b_1 b_2}(\mathbf{V})$  falls outside the range  $\left[ \frac{d}{r(n, \eta)}, \frac{d}{\eta r(n, \eta)} \right)$  with some positive probability. This happens for  $n = 200, 400, 500, 800$ . But as  $n$  grows,  $H_{b_1 b_2}(\mathbf{V})$  falls within the predicted interval with probability 1. For all  $n \geq 900$ , this was observed in this particular experiment. We may infer that, for all  $n \geq 900$ , we can approximate hop distance between  $b_1$  and  $b_2$  to be proportional to Euclidean distance between them with an accuracy factor 0.8. Of course, this value of  $n$  depends on the value of  $\eta$  and the locations of the points  $b_1$  and  $b_2$ , e.g., if  $\eta$  is chosen to be close to 1, i.e., giving much accurate proportionality between ED and HD,

$n$	$r(n, \eta)$	$\frac{d}{r(n, \eta)}$	$H_{b_1 b_2}$	probability mass	$\frac{d}{\eta r(n, \eta)}$
100	0.6935	1.6314	2	1	2.0393
200	0.5260	2.1510	3	1	2.6887
300	0.4456	2.5390	3	1	3.1736
400	0.3955	2.8607	3	0.87	3.5758
			4	0.13	
500	0.3603	3.1402	4	1	3.9257
600	0.3337	3.3905	4	1	4.2390
700	0.3126	3.6193	4	1	4.5238
800	0.2954	3.8300	4	0.87	4.7879
			5	0.13	
900	0.2810	4.0277	5	1	5.0352
1000	0.2686	4.2122	5	1	5.2672
1100	0.2579	4.3886	5	1	5.4869
1200	0.2484	4.5547	5	1	5.6940
1300	0.2400	4.7141	5	1	5.8927
1400	0.2325	4.8683	5	0.41	6.0860
			6	0.59	
1500	0.2256	5.0150	6	1	6.2681
1600	0.2194	5.1567	6	1	6.4467
1700	0.2138	5.2943	6	1	6.6164
1800	0.2085	5.4264	6	1	6.7830
1900	0.2037	5.5542	6	1	6.9411
2000	0.1992	5.6797	6	1	7.1023
2100	0.1950	5.8021	6	1	7.2526
2200	0.1911	5.9204	6	0.03	7.3996
			7	0.97	

Table 3.1: Simulation Results

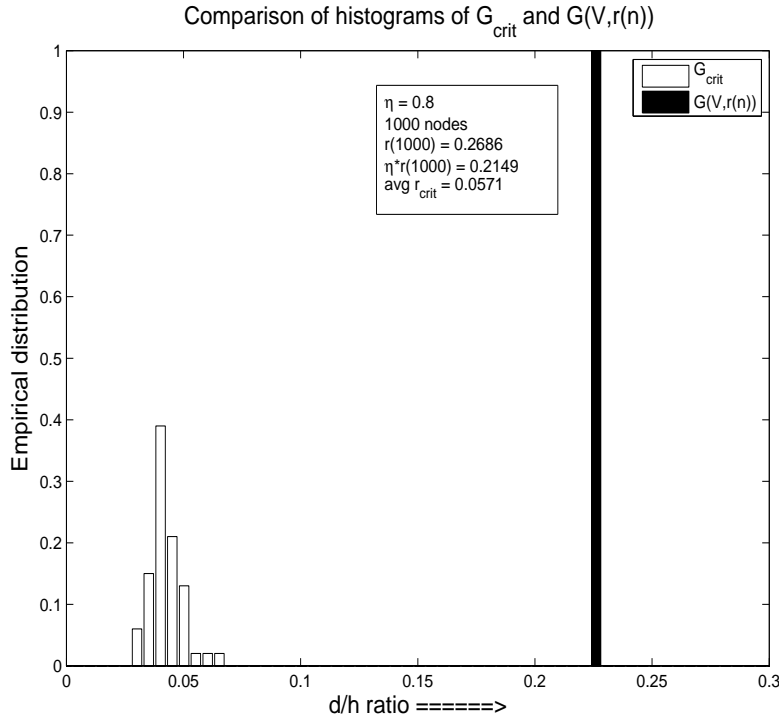


Figure 3.14: Comparing the histograms (100 runs) of  $\frac{d}{h}$  ratio on  $\mathcal{G}(\mathbf{V}, r(n))$  and  $\mathcal{G}_{crit}(\mathbf{V})$  for 1000 nodes

we will need larger values of  $n$ .

**Comparison with other Geometric Graphs:** For a given node deployment  $\mathbf{v}$  define  $r_{crit}(\mathbf{v})$  as the least value of  $r$  that makes the  $\mathcal{G}(\mathbf{v}, r)$  connected. We denote  $\mathcal{G}(\mathbf{v}, r_{crit}(\mathbf{v}))$  by  $\mathcal{G}_{crit}(\mathbf{v})$ . Figure 3.14 compares the  $\frac{d}{h}$  histogram for the proposed RGG  $\mathcal{G}(\mathbf{V}, r(n, \eta))$  with that for  $\mathcal{G}_{crit}(\mathbf{V})$ , for  $n = 1000$ . We observe the variability in  $\frac{d}{h}$  for  $\mathcal{G}(\mathbf{V}, r(n, \eta))$  to be small and lies in the theoretical interval  $(\eta r(n, \eta), r(n, \eta)]$  given by Theorem 1. Whereas, the  $\frac{d}{h}$  histogram for  $\mathcal{G}_{crit}(\mathbf{V})$  has positive mass over an interval of 0.03 to 0.065 (see the enlarged Figure 3.15) for  $d = 1.1314$ .

When  $\mathcal{G}(\mathbf{V}, r(n, \eta))$  is used,  $\eta r(n, \eta) < \frac{d}{h} \leq r(n, \eta)$  w.h.p.. Consider two pairs of points  $(a_1, a_2)$  and  $(b_1, b_2)$  with  $\overline{a_1 a_2} = d_1, H_{a_1 a_2}(\mathbf{V}) = h_1$  and  $\overline{b_1 b_2} = d_2, H_{b_1 b_2}(\mathbf{V}) = h_2$  respectively. Theorem 1 implies that for  $r(n, \eta)$  chosen appropriately,  $\mathbb{P}^n \left( \bigcap_{i=1}^2 \left\{ \frac{d_i}{r(n, \eta)} \leq h_i < \frac{d_i}{\eta r(n, \eta)} \right\} \right) \xrightarrow{n \rightarrow \infty} 1$  (as the probability of both the events  $\rightarrow 1$  individually as  $n \rightarrow \infty$ ). Thus for  $n$  large enough,  $\eta r(n, \eta) < \frac{d_i}{h_i} \leq r(n, \eta); i = 1, 2$ . So,  $\eta < \frac{d_1/d_2}{h_1/h_2} < \frac{1}{\eta}$  w.h.p.. Hence, for  $\eta = 0.8$ ,  $0.8 < \frac{d_1/d_2}{h_1/h_2} < 1.25$  w.h.p., i.e., if  $h_1/h_2$  is used as an approximation to  $d_1/d_2$ , then the ratio of these quantities

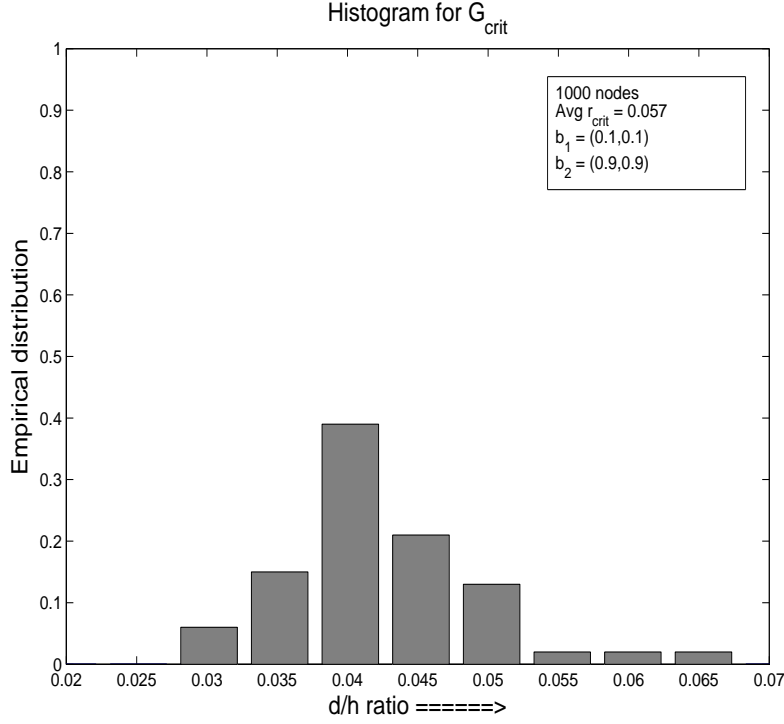


Figure 3.15: Histogram of  $\frac{d}{h}$  ratio on  $\mathcal{G}_{crit}(\mathbf{V})$  for 1000 nodes (100 runs): points  $b_1$  and  $b_2$

will lie in the interval  $(0.8, 1.25)$  w.h.p. if  $r(n, \eta)$  is chosen according to Theorem 1. Such a question is of interest in the context of the GPS-free localisation approach presented in [27]. We know, proportionality between  $d$  and  $h$  implies that the ratio of  $d_1/d_2$  to  $h_1/h_2$  should be as close to unity as possible. This can be achieved by choosing  $\eta$  closer to unity. Of course, the value of  $n$  will accordingly need to be larger for the asymptotics to work.

On the other hand, consider the same ratio for  $\mathcal{G}_{crit}(\mathbf{V})$ . Simulation with  $\mathcal{G}_{crit}(\mathbf{V})$  yielded,  $0.03 \leq \frac{d_2}{h_2} \leq 0.065$  w.h.p. for  $b_1 = (0.1, 0.1)$  and  $b_2 = (0.9, 0.9)$  (see Figure 3.15). Similarly for  $a_1 = (0.1, 0.9)$  and  $a_2 = (0.9, 0.1)$ , we have,  $0.03 \leq \frac{d_1}{h_1} \leq 0.07$  w.h.p. (see Figure 3.16). Hence, for  $\mathcal{G}_{crit}(\mathbf{V})$ ,  $0.46 \leq \frac{d_1/d_2}{h_1/h_2} \leq 2.33$ , which is substantially inaccurate compared to  $\mathcal{G}(\mathbf{V}, r(1000, 0.8))$ . These simulations show that if the proportionality between Euclidean and hop-distance is concerned,  $\mathcal{G}(\mathbf{V}, r(n, \eta))$  is better than  $\mathcal{G}_{crit}(\mathbf{V})$ .

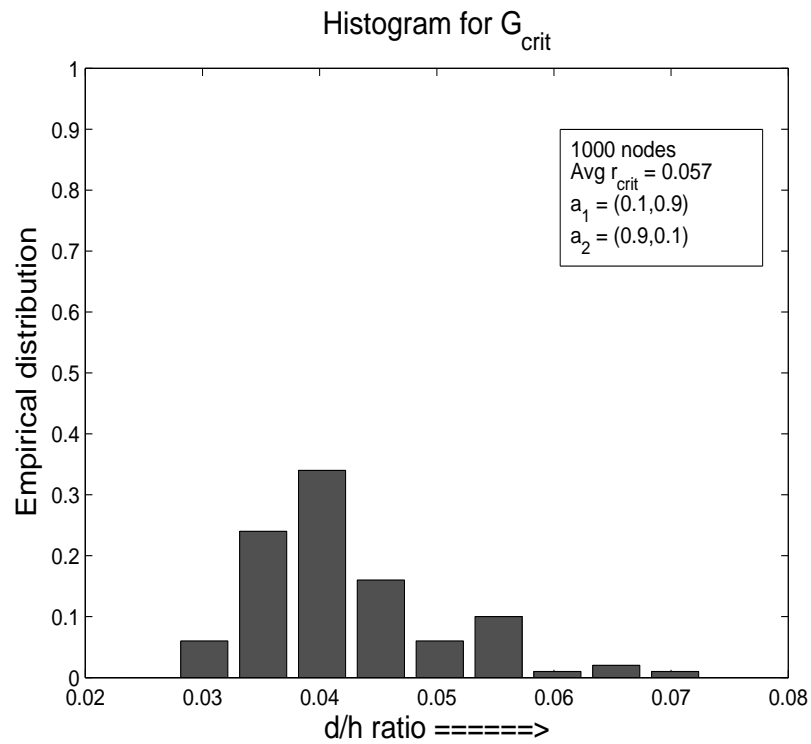


Figure 3.16: Histogram of  $\frac{d}{h}$  ratio on  $\mathcal{G}_{crit}(\mathbf{V})$  for 1000 nodes (100 runs): points  $a_1$  and  $a_2$

# Chapter 4

## Distance Discretisation between Two Random Nodes

In this chapter, we look into the relationship of Euclidean and hop-distance between the nodes of a random geometric graph. The proof techniques and the constructions are similar to what we got for the case of point to point (Chapter 3) distance discretisation. As shown in Theorem 4, we can show that the Euclidean distance lies within  $[(1 - \epsilon)hr(n), hr(n)]$  for this setup too, given that the hop-distance  $h$  between any pair of nodes is large enough.

**Setting:** We take a unit square area  $\mathcal{A}$ .  $n$  nodes are deployed Uniform i.i.d. on it, the node location random vector is denoted by  $\mathbf{V} \in \mathcal{A}^n$ , with a certain realisation being  $\mathbf{v} = [v_1, v_2, \dots, v_n] \in \mathcal{A}^n$ , where  $v_i$  is the location of the  $i^{th}$  node. The associated probability law is given by  $\mathbb{P}^n(\cdot)$ . After deployment, the nodes form a geometric graph with radius  $r(n)$ ,  $r(n) = \Theta\left(\sqrt{\frac{\ln n}{n}}\right)$ . We will call this geometric graph  $\mathcal{G}(\mathbf{v}, r(n))$ . Here we try to find a bound on the Euclidean distance between any pair of nodes on  $\mathcal{G}(\mathbf{v}, r(n))$  having hop distance  $h$  between them.

**Notation:**

- $\mathcal{N} = [n] = \{1, 2, \dots, n\}$ , the index set of the nodes, i.e., node  $i \in \mathcal{N}$  has a location  $v_i$  on  $\mathcal{A}$ .
- $D_{a,b}(\mathbf{v})$ : The Euclidean distance on  $\mathcal{A}$  between two nodes  $a$  and  $b$ ,  $a, b \in \mathcal{N}$ , for the sample

deployment  $\mathbf{v}$ .

- $H_{a,b}(\mathbf{v})$ : The hop distance on  $\mathcal{G}(\mathbf{v}, r(n))$  between two nodes  $a$  and  $b$ ,  $a, b \in \mathcal{N}$ , for the sample deployment  $\mathbf{v}$ .

$$\begin{aligned}\overline{D}(\mathbf{v}, h) &= \max_{\{(a,b) \in \mathcal{N}^2 : H_{a,b}(\mathbf{v})=h\}} D_{a,b}(\mathbf{v}) \\ \underline{D}(\mathbf{v}, h) &= \min_{\{(a,b) \in \mathcal{N}^2 : H_{a,b}(\mathbf{v})=h\}} D_{a,b}(\mathbf{v})\end{aligned}$$

Here, we will take  $r(n) = c\sqrt{\frac{\ln n}{n}}$ . Now we have the construction as shown in Figure 4.1. We have split the unit square  $\mathcal{A}$  into squarelets with diagonal  $r(n)$ . We construct a circle of radius  $hr(n) + \frac{r(n)}{2}$  centred at the centre of each squarelet. Then split it with the blades so that it covers the entire circumference of the circle. Let us define the following.

- $K(n)$  : Number of squarelets that split  $\mathcal{A}$ .
- $J(n)$  : Number of blades to cover the part of circle within  $\mathcal{A}$ .
- $A_{i,j,k} = \{\mathbf{v} : \exists \text{ at least one node in the } i^{\text{th}} \text{ strip of } j^{\text{th}} \text{ blade centred at the } k^{\text{th}} \text{ squarelet}\}$

If for a deployment  $\mathbf{v}$ , there exists at least one node in each of the  $h - 1$  strips for each of the  $J(n)$  blades centred at the centre of each of the  $K(n)$  squarelets, we have  $\underline{D}(\mathbf{v}, h) \geq (p - q)(h - 1)r(n) - \frac{r(n)}{2}$ , since all nodes that fall within this distance can be reached in  $\leq h - 1$  hops for that  $\mathbf{v}$ . Hence,

$$\begin{aligned}& \{\cap_{k=1}^{K(n)} \cap_{j=1}^{J(n)} \cap_{i=1}^{h-1} A_{i,j,k}\} \\ & \subseteq \{\mathbf{v} : (p - q)(h - 1)r(n) - \frac{r(n)}{2} \leq \underline{D}(\mathbf{v}, h) \leq \overline{D}(\mathbf{v}, h) \leq hr(n)\}\end{aligned} \quad (4.1)$$

Since  $1 > p > q > 0$ , we can choose  $p - q = 1 - \epsilon$ , for any given  $1 > \epsilon > 0$ . For such a choice of  $p$  and  $q$ , the event

$$\begin{aligned}& \{\cap_{k=1}^{K(n)} \cap_{j=1}^{J(n)} \cap_{i=1}^{h-1} A_{i,j,k}\} \\ & \subseteq \{\mathbf{v} : \left((1 - \epsilon)(h - 1) - \frac{1}{2}\right)r(n) \leq \underline{D}(\mathbf{v}, h) \leq \overline{D}(\mathbf{v}, h) \leq hr(n)\}\end{aligned}$$

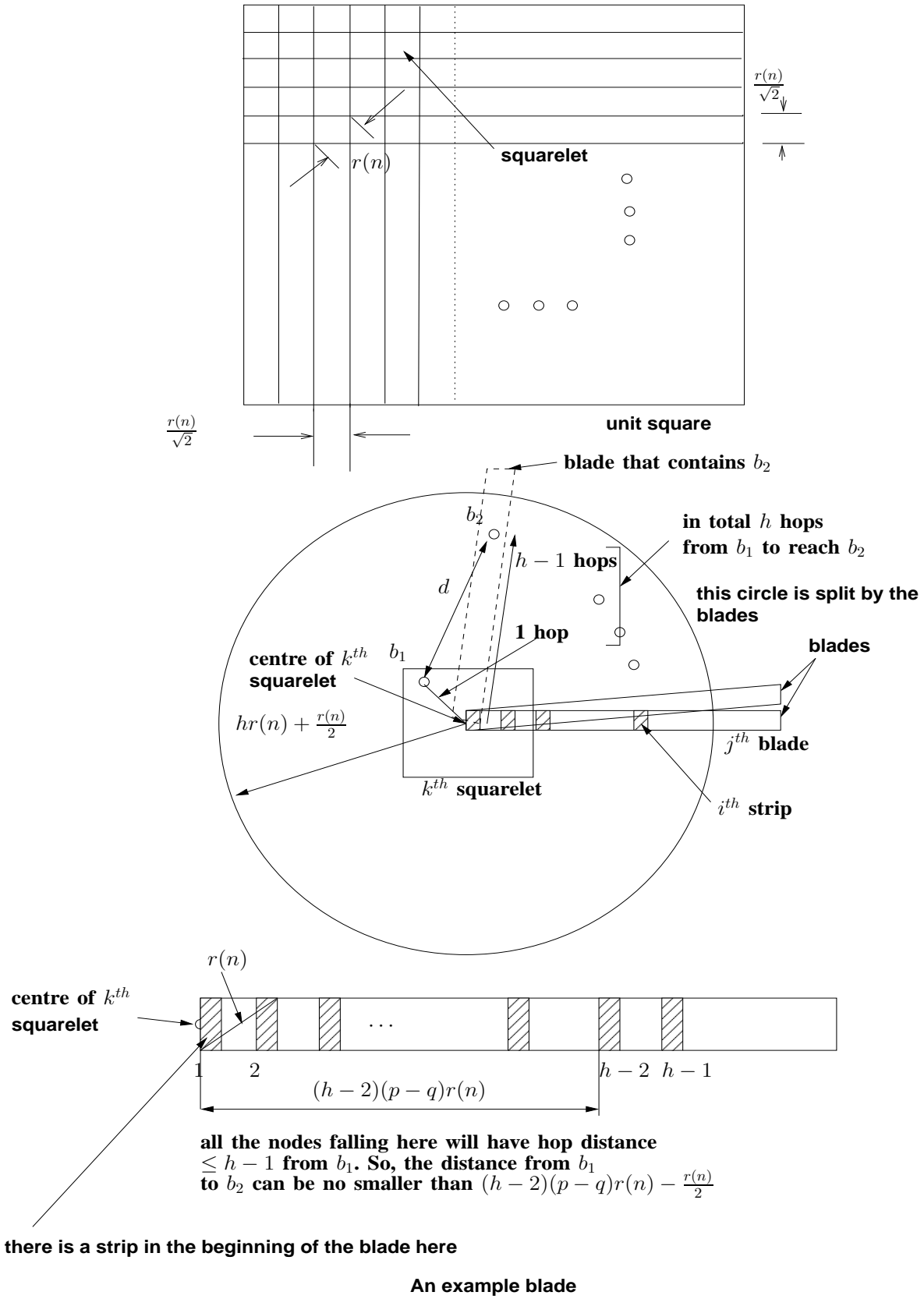
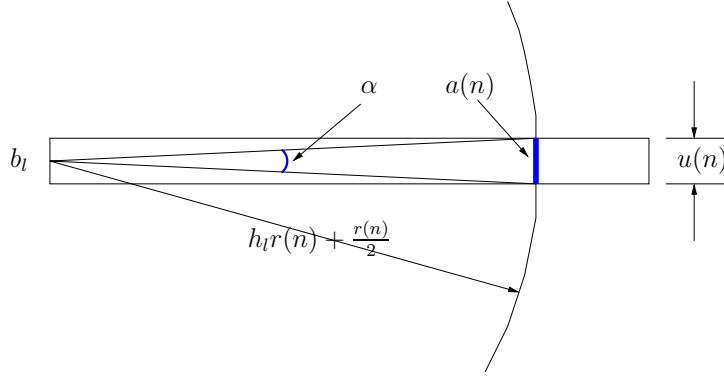


Figure 4.1: Construction using the blades cutting the circumference of the circle of radius  $hr(n) + \frac{r(n)}{2}$  and the squarelet splitting the region  $\mathcal{A}$ .



Figure 4.2: Construction to find  $J(n)$ .

$$\begin{aligned} & \mathbb{P}^n \left( \cap_{k=1}^{K(n)} \cap_{j=1}^{J(n)} \cap_{i=1}^{h-1} A_{i,j,k} \right) \\ & \leq \mathbb{P}^n \{ \mathbf{v} : \left( (1 - \epsilon)(h - 1) - \frac{1}{2} \right) r(n) \leq \underline{D}(\mathbf{v}, h) \leq \overline{D}(\mathbf{v}, h) \leq hr(n) \} \end{aligned} \quad (4.2)$$

Now, we need to find the values of  $J(n)$  and  $K(n)$ . To find  $J(n)$ , we need to define the following.

- $a(n)$  is the length of the arc of radius  $hr(n) + \frac{r(n)}{2}$  that lies within a blade, drawn taking  $k$  as centre ( $k$  is the centre of the  $k^{th}$  squarelet), as shown in Figure 4.2.
- $\alpha(n)$  : angle subtended by  $a(n)$  at  $k$ , see Figure 4.2.

Now we have,  $J(n) = \left\lceil \frac{2\pi}{\alpha(n)} \right\rceil$ . We also have from Figure 4.2,  $\left( hr(n) + \frac{r(n)}{2} \right) \alpha(n) = a(n) \geq u(n) = \sqrt{1 - p^2} r(n)$ . Hence,  $\alpha(n) \geq \frac{\sqrt{1 - p^2}}{h + \frac{1}{2}}$ . So,  $J(n) \leq \left\lceil \frac{2\pi(h + \frac{1}{2})}{\sqrt{1 - p^2}} \right\rceil = \left\lceil \frac{\pi(2h + 1)}{\sqrt{1 - p^2}} \right\rceil$ .

We can find out  $K(n) = \left\lceil \frac{\sqrt{2}}{r(n)} \right\rceil^2$ . Now let us compute,

$$\begin{aligned}
\mathbb{P}^n \left( \bigcap_{k=1}^{K(n)} \bigcap_{j=1}^{J(n)} \bigcap_{i=1}^{h-1} A_{i,j,k} \right) &= 1 - \mathbb{P}^n \left( \bigcup_{k=1}^{K(n)} \bigcup_{j=1}^{J(n)} \bigcup_{i=1}^{h-1} A_{i,j,k}^c \right) \\
&\geq 1 - \sum_{k=1}^{K(n)} \sum_{j=1}^{J(n)} \sum_{i=1}^{h-1} \mathbb{P}^n (A_{i,j,k}^c) \\
&\geq 1 - (h-1) \left\lceil \frac{\sqrt{2}}{r(n)} \right\rceil^2 \left\lceil \frac{\pi(2h+1)}{\sqrt{1-p^2}} \right\rceil (1 - u(n)t(n))^n \\
&\geq 1 - (h-1) \frac{2}{c^2} \frac{n}{\ln n} \left\lceil \frac{\pi(2h+1)}{\sqrt{1-p^2}} \right\rceil e^{-nu(n)t(n)} \\
&= 1 - (h-1) \frac{2}{c^2} \left\lceil \frac{\pi(2h+1)}{\sqrt{1-p^2}} \right\rceil \frac{1}{\ln n} n^{1-c^2q\sqrt{1-p^2}} \\
&\xrightarrow{n \rightarrow \infty} 1 \quad \text{iff } c^2 \geq \frac{1}{q\sqrt{1-p^2}}
\end{aligned} \tag{4.3}$$

For brevity, we have carried out the calculation above without the ceiling on  $K(n)$ , since the result remains unaltered even with the ceiling. So, we get from Equations 4.2 and 4.3,

$$\begin{aligned}
\mathbb{P}^n \{ \mathbf{v} : \left( (1-\epsilon)(h-1) - \frac{1}{2} \right) r(n) \leq \underline{D}(\mathbf{v}, h) \leq \overline{D}(\mathbf{v}, h) \leq hr(n) \} \\
&\geq \mathbb{P}^n \left( \bigcap_{k=1}^{K(n)} \bigcap_{j=1}^{J(n)} \bigcap_{i=1}^{h-1} A_{i,j,k} \right) \\
&\geq 1 - (h-1) \frac{2}{c^2} \left\lceil \frac{\pi(2h+1)}{\sqrt{1-p^2}} \right\rceil \frac{1}{\ln n} n^{1-c^2q\sqrt{1-p^2}}
\end{aligned} \tag{4.4}$$

Let us define the following,  $E_h(n) = \{ \mathbf{v} : ((1-\epsilon)(h-1) - \frac{1}{2}) r(n) \leq \underline{D}(\mathbf{v}, h) \leq \overline{D}(\mathbf{v}, h) \leq hr(n) \}$ . Then Equation 4.4 implies, as  $n \rightarrow \infty$ ,  $\mathbb{P}^n(E_h(n)) = 1 - \mathcal{O} \left( \frac{n^{1-c^2q\sqrt{1-p^2}}}{\ln n} \right)$ . So, we want,  $c^2 \geq \frac{1}{q\sqrt{1-p^2}}$ , for  $\mathbb{P}^n(E_h(n)) \xrightarrow{n \rightarrow \infty} 1$ . We have already chosen  $p - q = 1 - \epsilon \Rightarrow q = p - (1 - \epsilon)$ . To keep  $c$  as small as possible (this is needed because it will keep the  $r(n)$  to be the smallest, which is the radius of the geometric graph), we need  $q\sqrt{1-p^2}$  as large as possible. So, we choose,  $p = \arg \max_p (p - (1 - \epsilon))\sqrt{1-p^2} = \frac{1-\epsilon+\sqrt{(1-\epsilon)^2+8}}{4}$ , and  $q = \frac{-3(1-\epsilon)+\sqrt{(1-\epsilon)^2+8}}{4}$ . By this choice,  $p$ ,  $q$  and  $c$  becomes function of  $\epsilon$ . We will denote them by  $p(\epsilon)$ ,  $q(\epsilon)$  and  $c(\epsilon)$  respectively. The radius of the geometric graph,  $r(n, \epsilon) = c(\epsilon)\sqrt{\frac{\ln n}{n}}$  in turn also becomes a function of  $\epsilon$ . Define,  $E_h(n, \epsilon) = \{ \mathbf{v} : ((1-\epsilon)(h-1) - \frac{1}{2}) r(n, \epsilon) \leq \underline{D}(\mathbf{v}, h) \leq \overline{D}(\mathbf{v}, h) \leq hr(n, \epsilon) \}$ . Hence, for a

given  $1 > \epsilon > 0$ , writing  $g(\epsilon) = q(\epsilon)\sqrt{1 - p^2(\epsilon)}$ , we get the following theorem.

**Theorem 4** For  $1 > \epsilon > 0$ , if  $c^2(\epsilon) \geq \frac{1}{g(\epsilon)}$ , where  $g(\epsilon) = q(\epsilon)\sqrt{1 - p^2(\epsilon)}$ , and  $p(\epsilon) = \frac{1-\epsilon+\sqrt{(1-\epsilon)^2+8}}{4}$ ,  
 $q(\epsilon) = \frac{-3(1-\epsilon)+\sqrt{(1-\epsilon)^2+8}}{4}$ ,

$$\mathbb{P}^n(E_h(n, \epsilon)) = 1 - \mathcal{O}\left(\frac{n^{1-c^2(\epsilon)g(\epsilon)}}{\ln n}\right)$$

$$\text{Thus,} \quad \lim_{n \rightarrow \infty} \mathbb{P}^n(E_h(n, \epsilon)) = 1$$

■

A plot of  $g(\epsilon)$  versus  $\epsilon$  is given in Figure 5.4, which shows that for smaller  $\epsilon$ , we need larger values of  $c(\epsilon)$  and  $r(n, \epsilon)$ .

## Chapter 5

# Distance Discretisation between A Fixed Point and A Random Node

In this chapter, we will focus on the relationship of Euclidean and Hop distance between a node, whose location is random, to a fixed point, e.g., an anchor. In a large sensor field, the anchors (also called *beacons*) are usually placed at fixed locations, e.g. at the 4 corners of a unit square, and the nodes are deployed in some random fashion over the square. A high probability bound on the Euclidean distance given a hop distance will help in determining a region where a certain node is expected to lie, and it will be helpful for finding out the location of the node. We used this Euclidean distance information yielded by the hop distance to propose an algorithm for localisation in the next chapter.

In Section 3.3, we find bounds on Euclidean distance for a given hop distance on *arbitrary* geometric graphs in 2 dimensions, and show that for 2-D, the hop distance (HD) is not a good measure of Euclidean distance (ED). However, when the node deployment is spatially homogeneous and random, thus yielding an RGG, HD does become proportional to ED in an approximate sense. This has been explored in the sections to follow. For the RGG, nodes are distributed in a uniform i.i.d. fashion over  $\mathcal{A} \subset \mathbb{R}^2$ , i.e., the location of each node is uniformly distributed over  $\mathcal{A}$ , independent of the locations of the other nodes. On such a deployment of nodes, we consider the RGG with radius  $r(n) = c\sqrt{\frac{\ln n}{n}}$ ,  $c > \frac{1}{\sqrt{\pi}}$ , which ensures connectedness of the RGG with probability

approaching 1, as  $n \rightarrow \infty$  (by Gupta and Kumar [9]). We find in Section 5.1 that given a HD  $h$  from an anchor node (location fixed) on this RGG, the ED  $d$  between the anchor and the node lies within an interval  $[(1 - \epsilon)(h - 1)r(n), hr(n)]$  w.h.p.<sup>1</sup>, for  $\epsilon > 0$ , with the convergence rate dictated by the  $\epsilon$  chosen. We note that this result is different from the approximations of [20] or [27] where  $d$  is assumed to be exactly proportional to  $h$ . Rather, nodes having hop distance  $h$  from the anchor lies within an interval described by the two bounds mentioned above. In Section 5.2, we show that the rate of convergence can be improved if the radius  $r$  does not vary with  $n$ . Of course, we need to choose  $n$  large enough so that the radius for connectivity according to [9] is smaller than  $r$ . We extend both the results for the case of *randomised lattice deployment*<sup>2</sup> in Section 5.3.

Our results are via bounds and provide a sufficient condition for the rate of convergence. However in Section 5.4, we have considered Poisson deployment in 1-dimension for which these conditions are necessary and sufficient for the construction considered. Finally in Section 5.5, we provide simulation results to illustrate the theorems.

## 5.1 HD-ED Relationship in Random Geometric Graphs

In this section we will provide theoretical results concerning distance-hop proportionality in an RGG. The setting and few notations are as follows. The setting and the notations are as follows.

### Setting:

- $n$  nodes are deployed on a unit area  $\mathcal{A}$  in the uniform i.i.d. fashion. The node locations are random, and are denoted by the random vector  $\mathbf{V} \in \mathcal{A}^n$ , with a particular realisation being denoted by  $\mathbf{v} = [v_1, v_2, \dots, v_n] \in \mathcal{A}^n$ , where  $v_i$  is the location of the  $i^{th}$  node. We denote by  $\mathbb{P}^n(\cdot)$  the probability measure on  $\mathcal{A}^n$  so obtained.
- We form the RGG  $\mathcal{G}(\mathbf{v}, r(n))$  by connecting the nodes that are within the radius  $r(n)$  of each other, where  $r(n)$ , the radius of the geometric graph is chosen so that the network

---

<sup>1</sup>w.h.p. (with high probability) means that the probability of the said event  $\rightarrow 1$  as  $n \rightarrow \infty$

<sup>2</sup>A randomised lattice deployment is obtained as follows. The area  $\mathcal{A}$  is partitioned into  $n$  equal area “cells”, e.g., squares, and one node is placed at a uniformly distributed random location in each cell.

remains asymptotically connected. We take  $r(n) = c\sqrt{\frac{\ln n}{n}}$ ,  $c > \frac{1}{\sqrt{\pi}}$ , a constant; this ensures asymptotic connectivity (see [9]).

**Notation:**

- $\mathcal{N} = [n] = \{1, 2, \dots, n\}$ , the index set of the nodes, i.e., node  $i \in \mathcal{N}$  has a location  $v_i$  on  $\mathcal{A}$ .
- $b_l$  = Location of the  $l^{th}$  anchor node,  $l = 1, \dots, L$ .
- $H_{l,i}(\mathbf{v})$  = Minimum number of hops of node  $i$  from anchor  $b_l$  on the graph  $\mathcal{G}(\mathbf{v}, r(n))$  for a sample deployment  $\mathbf{v}$ .
- $D_{l,i}(\mathbf{v})$  = Euclidean distance of node  $i$  from anchor  $b_l$  for the deployment  $\mathbf{v}$ .

$$\overline{D}_l(\mathbf{v}, h_l) = \max_{\{i \in \mathcal{N} : H_{l,i}(\mathbf{v}) = h_l\}} D_{l,i}(\mathbf{v})$$

$$\underline{D}_l(\mathbf{v}, h_l) = \min_{\{i \in \mathcal{N} : H_{l,i}(\mathbf{v}) = h_l\}} D_{l,i}(\mathbf{v})$$

Graphical illustration of the above two quantities is similar to Figure 3.2, with  $r$  replaced by  $r(n)$ .

In Section 5.1.1, we analyse the distribution of distance from one anchor node and in Section 5.1.2, we generalise it for  $L$  anchors.

The choice of the radius,  $r(n) = c\sqrt{\frac{\ln n}{n}}$ ,  $c > \frac{1}{\sqrt{\pi}}$ , does not only guarantee asymptotic connectivity among the nodes, but also ensures connectivity of the nodes with all the anchors. The following lemma states that there will be at least a node within a distance  $r(n)$  of each anchor  $b_l$ ,  $l = 1, \dots, L$  w.h.p. and so the nodes are connected to all the anchors in a dense network. Define,  $B_l = \{\mathbf{v} : \exists \text{ at least one node within a radius of } r(n) \text{ from } b_l\}$ ,  $l = 1, \dots, L$ .

**Lemma 6**  $\lim_{n \rightarrow \infty} \mathbb{P}^n (\cap_{l=1}^L B_l) = 1$

**Proof:**  $\mathbb{P}^n (\cap_{l=1}^L B_l) = 1 - \mathbb{P}^n (\cup_{l=1}^L B_l^c) \geq 1 - \sum_{l=1}^L \mathbb{P}^n \{B_l^c\} = 1 - \sum_{l=1}^L (1 - \pi r^2(n))^n \geq 1 - L e^{-n\pi r^2(n)} \xrightarrow{n \rightarrow \infty} 1$ , since  $r(n) = c\sqrt{\frac{\ln n}{n}}$  and  $1 - x \leq e^{-x}$ . ■

### 5.1.1 Distance distribution from a fixed anchor $b_l$ : Uniform i.i.d. deployment

We make the construction as shown in Figure 5.1. From  $b_l$  (without loss of generality, we can choose  $l = 1$ ), we draw a circle of radius  $h_l r(n)$  centred at  $b_l$ , this is the maximum distance reachable in  $h_l$  hops, by triangle inequality, since each hop can be of maximum length  $r(n)$ . All the nodes  $\{i \in \mathcal{N} : H_{l,i}(\mathbf{v}) = h_l\}$  lie within this disk. So,  $\overline{D}_l(\mathbf{v}, h_l) \leq h_l r(n)$  for all  $\mathbf{v}$ . To obtain a lower bound on  $\underline{D}_l(\mathbf{v}, h_l)$ , we construct blades as shown in Figure 5.1. We start with one blade. It will cover some portion of the circumference of the circle of radius  $h_l r(n)$ ; see Figure 5.1. Construct the next blade so that it covers the adjacent portion of the circumference that has not been covered by the previous blade. We go on constructing these blades until the entire portion of the circle lying inside the unit square  $\mathcal{A}$  is covered (see Figure 5.1). Let us define,

- $J(n)$  : Number of blades required to cover the part of the circle within  $\mathcal{A}$ .
- $\mathcal{B}_j^l$  :  $j^{th}$  blade drawn from the point  $b_l$  as shown in Figure 5.1,  $1 \leq j \leq J(n)$ .

On each of these blades, we construct  $h_l$  strips, shown shaded in Figure 5.2,  $u(n)$  being the width of the blade and  $t(n)$  the width of the strip. We define the following event.

- $A_{i,j}^l = \{\mathbf{v} : \exists \text{ at least one node in the } i^{th} \text{ strip of } \mathcal{B}_j^l\}$

If a  $\mathbf{v} \in A_{i,j}^l$ ,  $\forall 1 \leq i \leq h_l - 1, 1 \leq j \leq J(n)$ , i.e., there exists at least one node in each of the  $h_l - 1$  strips (see Figure 5.2) for all the blades  $\mathcal{B}_j^l$ , then for that  $\mathbf{v}$ , all nodes at a distance  $< (p - q)(h_l - 1)r(n)$  from  $b_l$  are reachable in at most  $h_l - 1$  hops, hence will have a hop distance  $\leq h_l - 1 < h_l$ . So, we have  $\underline{D}_l(\mathbf{v}, h_l) \geq (p - q)(h_l - 1)r(n)$ , for such a deployment  $\mathbf{v}$ ; see Figure 5.2. Hence,

$$\begin{aligned} & \left\{ \bigcap_{j=1}^{J(n)} \bigcap_{i=1}^{h_l-1} A_{i,j}^l \right\} \\ & \subseteq \left\{ \mathbf{v} : (p - q)(h_l - 1)r(n) \leq \underline{D}_l(\mathbf{v}, h_l) \leq \overline{D}_l(\mathbf{v}, h_l) \leq h_l r(n) \right\} \end{aligned} \quad (5.1)$$

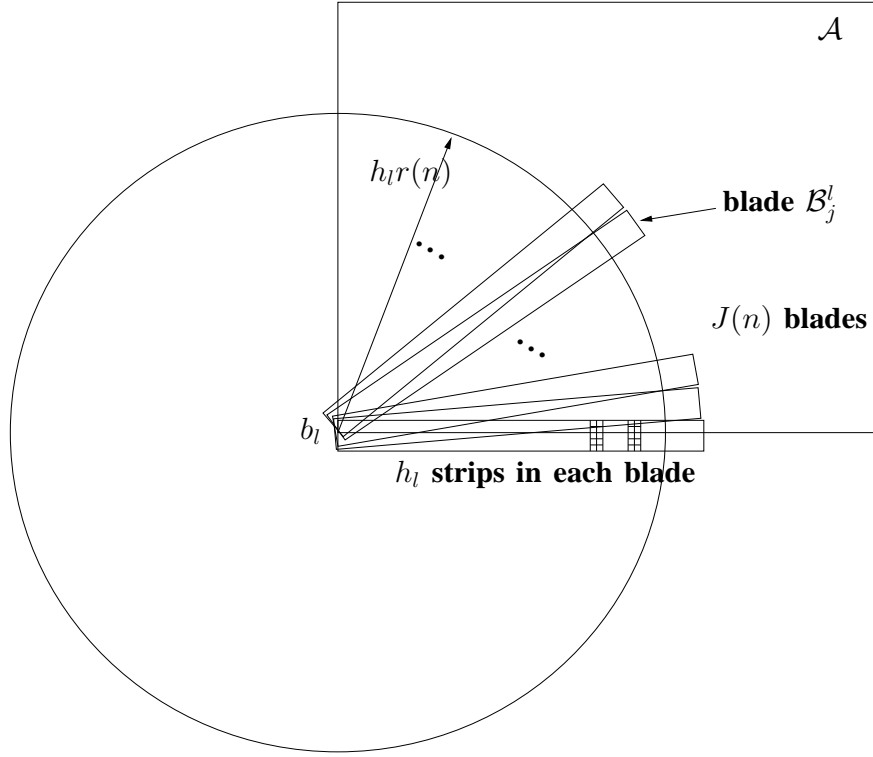


Figure 5.1: Construction using the blades cutting the circumference of the circle of radius  $h_l r(n)$ .

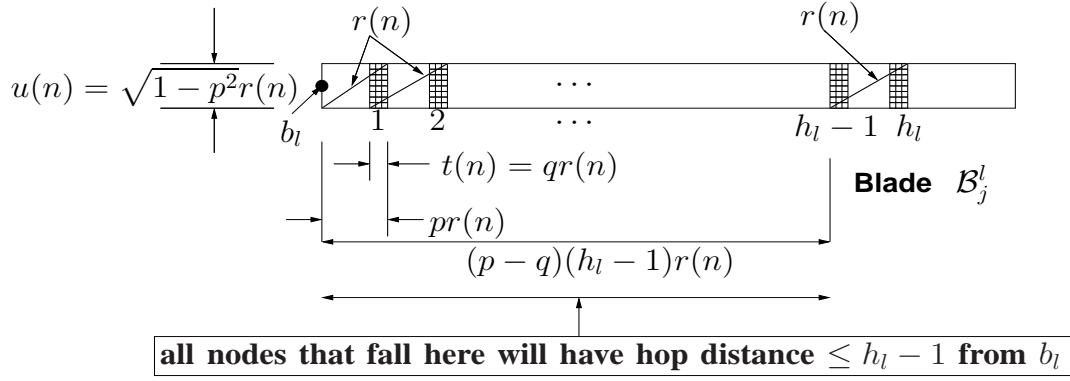
Since  $1 > p > q > 0$ , we can choose  $p - q$  to be equal to  $1 - \epsilon$ , for the given  $\epsilon > 0$ . So the lower bound in Equation 5.1 becomes,  $(p - q)(h_l - 1)r(n) = (1 - \epsilon)(h_l - 1)r(n)$ .

To find the value of  $J(n)$ , we need to define the following.

- $a(n)$  is the length of the arc of radius  $h_l r(n)$  that lies within a blade, drawn taking  $b_l$  as centre, as shown in Figure 5.3.
- $\alpha(n)$  : angle subtended by  $a(n)$  at  $b_l$ , see Figure 5.3.

Now from Figure 5.1, we have,  $J(n) = \left\lceil \frac{\pi}{2\alpha(n)} \right\rceil$ . We also have from Figure 5.3,  $h_l r(n)\alpha(n) = a(n) \geq u(n) = \sqrt{1 - p^2}r(n)$ . Hence,  $\alpha(n) \geq \frac{\sqrt{1 - p^2}}{h_l}$ . So,  $J(n) \leq \left\lceil \frac{\pi h_l}{2\sqrt{1 - p^2}} \right\rceil$ .

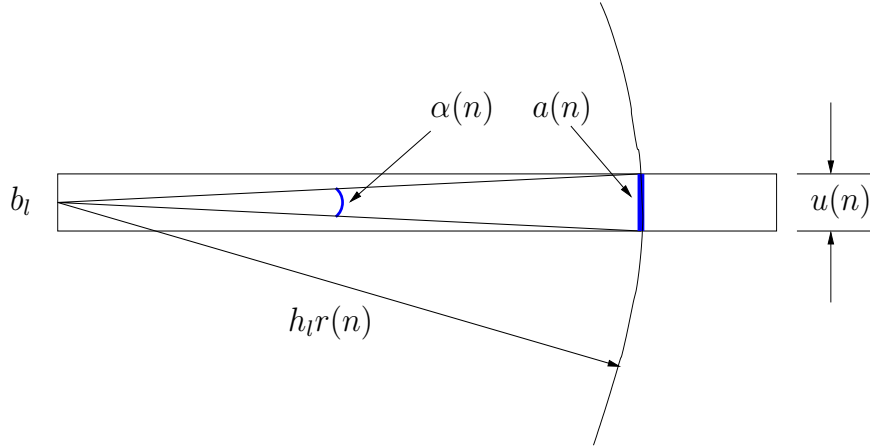


Figure 5.2: The construction with  $h_l$  hops.

Now we compute,

$$\begin{aligned}
 & \mathbb{P}^n \left( \bigcap_{j=1}^{J(n)} \bigcap_{i=1}^{h_l-1} A_{i,j}^l \right) \\
 &= 1 - \mathbb{P}^n \left( \bigcup_{j=1}^{J(n)} \bigcup_{i=1}^{h_l-1} A_{i,j}^{l,c} \right) \\
 &\geq 1 - \sum_{j=1}^{J(n)} \sum_{i=1}^{h_l-1} \mathbb{P}^n (A_{i,j}^{l,c}) \\
 &\geq 1 - (h_l - 1) \left[ \frac{\pi h_l}{2\sqrt{1-p^2}} \right] (1 - u(n)t(n))^n \\
 &\geq 1 - (h_l - 1) \left[ \frac{\pi h_l}{2\sqrt{1-p^2}} \right] e^{-nu(n)t(n)} \\
 &= 1 - (h_l - 1) \left[ \frac{\pi h_l}{2\sqrt{1-p^2}} \right] e^{-nq\sqrt{1-p^2}r^2(n)} \\
 &\xrightarrow{n \rightarrow \infty} 1
 \end{aligned} \tag{5.2}$$

The first inequality comes from the union bound, the second inequality, from the upper bound on  $J(n)$ . The third inequality uses the result  $1 - x \leq e^{-x}$ . We see that if the node distribution was non-homogeneous with positive density over all points in  $\mathcal{A}$ , the term  $(1 - u(n)t(n))^n$  could have been replaced by  $(1 - f_{\min}u(n)t(n))^n$ , where  $f_{\min}$  is the minimum density over  $\mathcal{A}$  and as  $f_{\min} > 0$ , the same convergence result would have been true even for non-homogeneous node placements.

Figure 5.3: Construction to find  $J(n)$ .

Let us define,  $E_{h_l}(n) = \{\mathbf{v} : (1 - \epsilon)(h_l - 1)r(n) \leq \underline{D}_l(\mathbf{v}, h_l) \leq \overline{D}_l(\mathbf{v}, h_l) \leq h_l r(n)\}$ . So, we have, for the given  $\epsilon > 0$ , and using Equations 5.1 and 5.2,

$$\begin{aligned}
 1 &\geq \mathbb{P}^n(E_{h_l}(n)) \\
 &\geq \mathbb{P}^n\left(\bigcap_{j=1}^{J(n)} \bigcap_{i=1}^{h_l-1} A_{i,j}^l\right) \\
 &\geq 1 - (h_l - 1) \left[ \frac{\pi h_l}{2\sqrt{1-p^2}} \right] e^{-nq\sqrt{1-p^2}c^2 \frac{\ln n}{n}}
 \end{aligned} \tag{5.3}$$

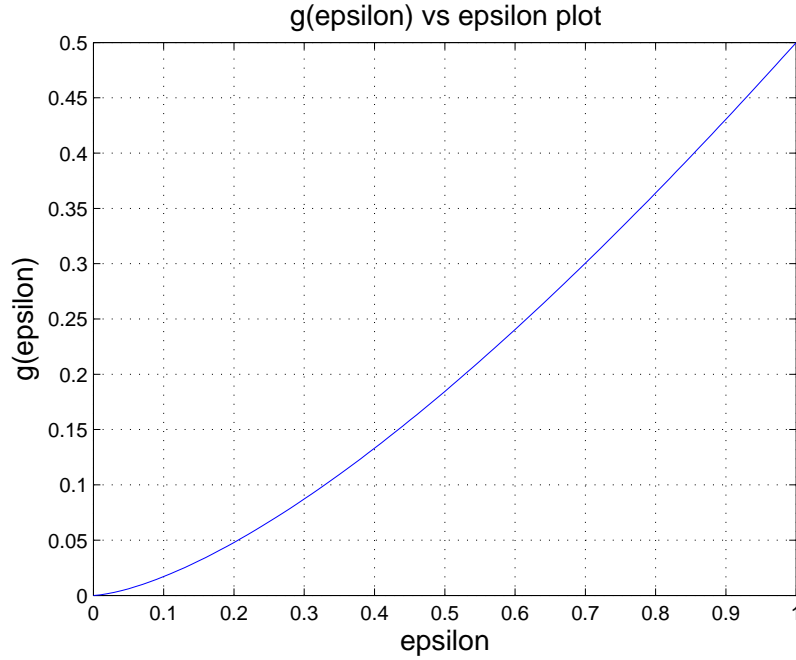
which implies,

$$0 \leq 1 - \mathbb{P}^n(E_{h_l}(n)) \leq (h_l - 1) \left[ \frac{\pi h_l}{2\sqrt{1-p^2}} \right] e^{-q\sqrt{1-p^2}c^2 \ln n} \tag{5.4}$$

And as  $n \rightarrow \infty$ ,

$$\begin{aligned}
 1 - \mathbb{P}^n(E_{h_l}(n)) &= \mathcal{O}\left(e^{-q\sqrt{1-p^2}c^2 \ln n}\right) \\
 &= \mathcal{O}\left(\frac{1}{n^{q\sqrt{1-p^2}c^2}}\right)
 \end{aligned} \tag{5.5}$$

This result is true for any  $p$  and  $q$ . But we can choose these constants so that the convergence  $\rightarrow 0$  of the bound in Equation 5.5 is the most rapid, i.e.,  $p$  and  $q$  are chosen so as to maximise

Figure 5.4:  $g(\epsilon)$  vs  $\epsilon$  plot.

$q\sqrt{1-p^2}$ , thus making the upper bound to reduce at the fastest rate. For the given  $\epsilon > 0$ ,  $p - q = 1 - \epsilon \Rightarrow q = p - (1 - \epsilon)$ . We can show that,  $p = \arg \max_p (p - (1 - \epsilon))\sqrt{1-p^2} = \frac{1-\epsilon+\sqrt{(1-\epsilon)^2+8}}{4}$ ,  $q = \frac{-3(1-\epsilon)+\sqrt{(1-\epsilon)^2+8}}{4}$ . Then writing,  $g(\epsilon) = q(\epsilon)\sqrt{1-p^2(\epsilon)}$ , we obtain the following theorem,

**Theorem 5** For a given  $1 > \epsilon > 0$ , and  $r(n) = c\sqrt{\frac{\ln n}{n}}$ ,  $c > \frac{1}{\sqrt{\pi}}$ ,  $\mathbb{P}^n(E_{h_l}(n)) = 1 - \mathcal{O}\left(\frac{1}{n^{g(\epsilon)c^2}}\right)$ ,

where  $g(\epsilon) = q(\epsilon)\sqrt{1-p^2(\epsilon)}$ ,

$$p(\epsilon) = \frac{1-\epsilon+\sqrt{(1-\epsilon)^2+8}}{4}, \quad q(\epsilon) = \frac{-3(1-\epsilon)+\sqrt{(1-\epsilon)^2+8}}{4}.$$

■

**Remark:** A plot of  $g(\epsilon)$  vs  $\epsilon$  is given in Figure 5.4. We see that  $g(\epsilon) \downarrow 0$  as  $\epsilon \downarrow 0$ . Hence Theorem 5 says that  $\lim_{n \rightarrow \infty} \mathbb{P}^n(E_{h_l}(n)) = 1$ , for any  $1 > \epsilon > 0$ , so we can expect a node having a HD of  $h_l$  from anchor  $b_l$  to be within a distance  $[(1 - \epsilon)(h_l - 1)r(n), h_l r(n)]$  from  $b_l$  in a dense network. We notice that the width of this band of uncertainty is roughly  $r(n)$ , which is the unit of distance measurement on  $\mathcal{G}(\mathbf{v}, r(n))$ . The theorem also says that the rate of convergence is governed by the  $\epsilon$  chosen, i.e., the smaller the  $\epsilon$ , the slower the rate of convergence.

### 5.1.2 Distance distribution from fixed anchors $b_l, l = 1, \dots, L$ : Uniform i.i.d. deployment

For  $L$  anchors, the question arises whether the hop-distances from the  $L$  anchors are *feasible* or not, e.g., if we denote a disk with centre  $a$  and radius  $r$ , by  $C(a, r) = \{z \in \mathcal{A} : \|z - a\| \leq r\}$ , then a necessary condition for a *feasible*  $\mathbf{h}$  vector ( $\mathbf{h} = [h_1, \dots, h_l, \dots, h_L] \in \mathbb{N}^L$  is the hop distance vector) is that  $\cap_{l=1}^L C(b_l, h_l r(n)) \neq \phi$  (there will be other feasibility conditions also). We denote the set of all *feasible*  $\mathbf{h}$  vectors by  $\mathcal{H}(n)$  (note that *feasible*  $\mathbf{h}$  vector depends on  $n$ ). We see that  $\forall \mathbf{h} \in \mathcal{H}(n)$ ,  $\cap_{l=1}^L E_{h_l}(n) \supseteq \cap_{l=1}^L \cap_{j=1}^{J(n)} \cap_{i=1}^{h_l-1} A_{i,j}^l$ , which implies that (analysing similar to Equation 5.2),

$$\begin{aligned} \mathbb{P}^n \left( \cap_{l=1}^L E_{h_l}(n) \right) &\geq \mathbb{P}^n \left( \cap_{l=1}^L \cap_{j=1}^{J(n)} \cap_{i=1}^{h_l-1} A_{i,j}^l \right) \\ &\geq 1 - \sum_{l=1}^L (h_l - 1) \left[ \frac{\pi h_l}{2\sqrt{1-p^2}} \right] n^{-q\sqrt{1-p^2}c^2} \\ &\Rightarrow \mathbb{P}^n \left( \cap_{l=1}^L E_{h_l}(n) \right) = 1 - \mathcal{O} \left( n^{-q\sqrt{1-p^2}c^2} \right) \end{aligned}$$

Hence we get the following theorem,

**Theorem 6** For a given  $1 > \epsilon > 0$ , and  $r(n) = c\sqrt{\frac{\ln n}{n}}$ ,  $c > \frac{1}{\sqrt{\pi}}$ ,  $\forall \mathbf{h} = [h_1, \dots, h_l, \dots, h_L] \in \mathcal{H}(n)$ ,

$$\mathbb{P}^n \left( \cap_{l=1}^L E_{h_l}(n) \right) = 1 - \mathcal{O} \left( \frac{1}{n^{g(\epsilon)c^2}} \right)$$

where  $g(\epsilon) = q(\epsilon)\sqrt{1-p^2(\epsilon)}$ ,

$$p(\epsilon) = \frac{1-\epsilon+\sqrt{(1-\epsilon)^2+8}}{4}, q(\epsilon) = \frac{-3(1-\epsilon)+\sqrt{(1-\epsilon)^2+8}}{4}$$

■

This theorem tells us that for a feasible  $\mathbf{h}$ , the node lies within the intersection of the annuli of inner and outer radii  $(1-\epsilon)(h_l-1)r(n)$  and  $h_l r(n)$  respectively, centred at anchors  $b_l$ ,  $1 \leq l \leq L$ , with a probability that scales as shown in the above theorem. A graphical illustration of this is shown in Figure 5.5 for  $L = 4$ .

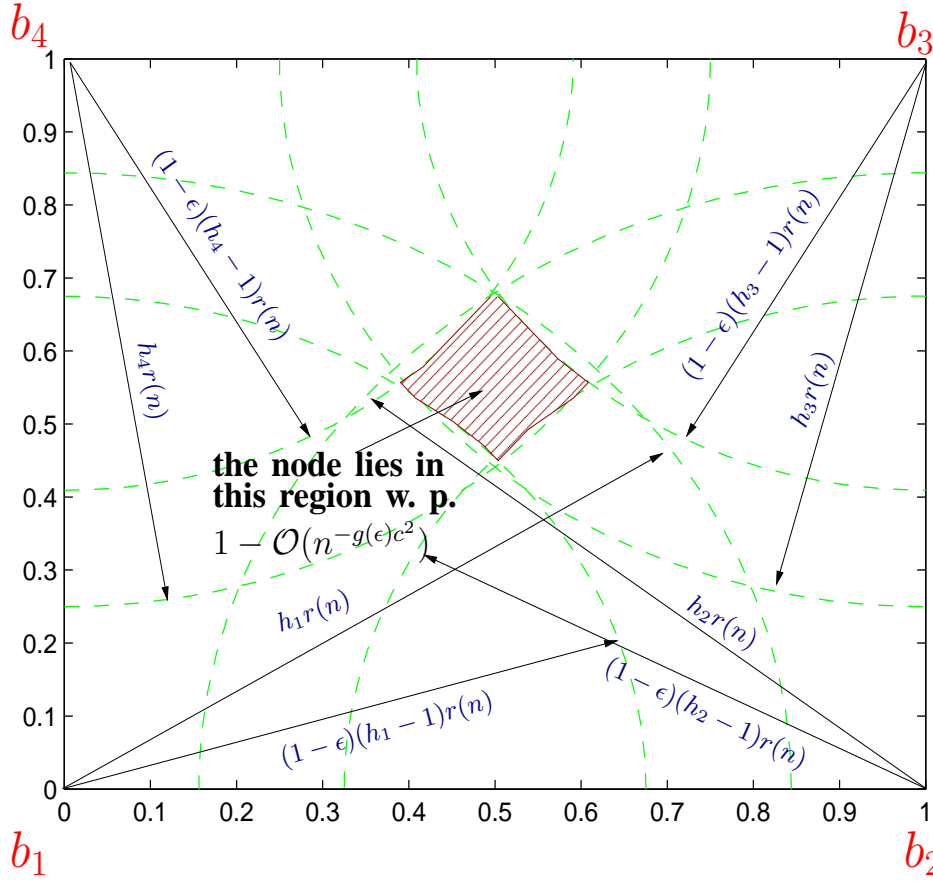


Figure 5.5: Graphical illustration of how Theorem 6 yields a location region for a node that is at a HD  $h_l$  from anchor  $b_l$ ,  $1 \leq l \leq 4$ .

## 5.2 HD-ED Relationship in Random Geometric Graphs: Fixed Radius

The scaling of  $r(n)$  with  $n$  as shown in the previous section ensures asymptotic connectivity and increases the precision in localisation as  $n \rightarrow \infty$ . But in a wireless sensor network the radius  $r$  of the RGG on which hop-distances are measured often corresponds to the radio range for a given transmit power, and hence does not decrease with  $n$ . So, it is meaningful to use a fixed radius  $r$  for the RGG and it is denoted by  $\mathcal{G}(\mathbf{v}, r)$ . But for connectivity, we need to use number of nodes sufficient to make the network connected (i.e., the radius should scale with  $n$  like  $r(n) = c\sqrt{\frac{\ln n}{n}}$ ,  $c > \frac{1}{\sqrt{\pi}}$ , a constant; see [9]), i.e., need at least  $n_0 = \inf\{n : r(n) \leq r\}$  nodes. Using a constant value for radius  $r$ , and redefining  $E_{h_l} = \{\mathbf{v} : (1 - \epsilon)(h_l - 1)r \leq \underline{D}_l(\mathbf{v}, h_l) \leq \overline{D}_l(\mathbf{v}, h_l) \leq h_l r\}$ ,

where the hop distance is measured on the RGG  $\mathcal{G}(\mathbf{v}, r)$ , we can show (along similar lines as for Equation 5.2),

$$\begin{aligned}
1 &\geq \mathbb{P}^n(E_{h_l}) \\
&\geq \mathbb{P}^n\left(\bigcap_{j=1}^J \bigcap_{i=1}^{h_l-1} A_{i,j}^l\right) \\
&\geq 1 - (h_l - 1) \left[ \frac{\pi h_l}{2\sqrt{1-p^2}} \right] e^{-nq\sqrt{1-p^2}r^2}
\end{aligned} \tag{5.6}$$

where  $J \leq \left\lceil \frac{\pi h_l}{2\sqrt{1-p^2}} \right\rceil$ . Which implies, as  $n \rightarrow \infty$ ,

$$\begin{aligned}
1 - \mathbb{P}^n(E_{h_l}) &= \mathcal{O}\left(e^{-nq\sqrt{1-p^2}r^2}\right) \\
\mathbb{P}^n(E_{h_l}) &= 1 - \mathcal{O}\left(e^{-nq\sqrt{1-p^2}r^2}\right)
\end{aligned} \tag{5.7}$$

So, we can state the following theorem,

**Theorem 7** For a given  $1 > \epsilon > 0$ , and a fixed  $r$ ,  $\mathbb{P}^n(E_{h_l}) = 1 - \mathcal{O}\left(e^{-ng(\epsilon)c^2r^2}\right)$ ,

where  $g(\epsilon) = q(\epsilon)\sqrt{1-p^2(\epsilon)}$ ,

$$p(\epsilon) = \frac{1-\epsilon+\sqrt{(1-\epsilon)^2+8}}{4}, \quad q(\epsilon) = \frac{-3(1-\epsilon)+\sqrt{(1-\epsilon)^2+8}}{4}.$$

■

Hence,  $\lim_{n \rightarrow \infty} \mathbb{P}^n(E_{h_l}) = 1$ . For  $L$  anchors, we will get  $\forall \mathbf{h} \in \mathcal{H}$  (note that the set of feasible  $\mathbf{h}$  vectors,  $\mathcal{H}$ , does not scale with  $n$  in this case),

$$1 - \mathbb{P}^n(\bigcap_{l=1}^L E_{h_l}) = \mathcal{O}\left(e^{-nq\sqrt{1-p^2}r^2}\right) \tag{5.8}$$

Hence we get the following theorem.

**Theorem 8** For a given  $1 > \epsilon > 0$ , and  $r$  fixed,  $\forall n \geq n_0 = \inf\{n : r(n) \leq r\}$ ,  $\forall \mathbf{h} = [h_1, \dots, h_l, \dots, h_L] \in \mathcal{H}$ ,

$$\mathbb{P}^n(\bigcap_{l=1}^L E_{h_l}) = 1 - \mathcal{O}\left(e^{-ng(\epsilon)r^2}\right)$$

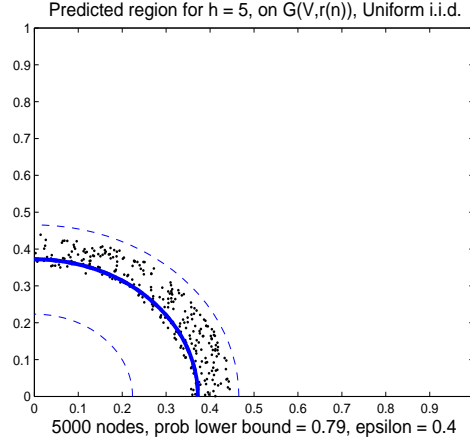
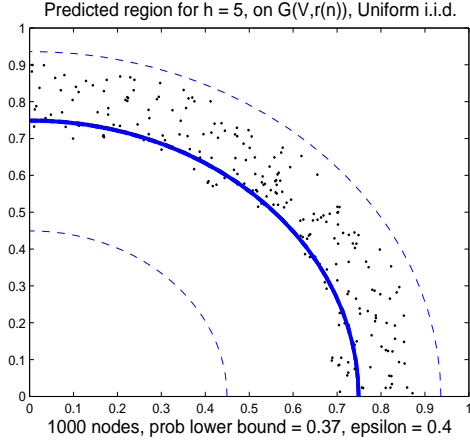


Figure 5.6: 1000 nodes, 5 hops,  $\epsilon = 0.4$ ,  $\mathbb{P}^n(E_1(n)) \geq 0.37$ .  $r(n) = 0.1876$ .  
Figure 5.7: 5000 nodes, 5 hops,  $\epsilon = 0.4$ ,  $\mathbb{P}^n(E_1(n)) \geq 0.79$ .  $r(n) = 0.0931$ .

The thin dashed lines show the ED bounds given by Theorem 5, the thick solid line shows ED  $(h_1 - 1)r(n)$  from  $b_1 \cdot r(n) = \frac{4}{\sqrt{\pi}} \sqrt{\frac{\ln n}{n}}$ .

where  $g(\epsilon) = q(\epsilon) \sqrt{1 - p^2(\epsilon)}$ ,

$$p(\epsilon) = \frac{1 - \epsilon + \sqrt{(1 - \epsilon)^2 + 8}}{4}, \quad q(\epsilon) = \frac{-3(1 - \epsilon) + \sqrt{(1 - \epsilon)^2 + 8}}{4}$$

■

**Remark:** We see that, for all  $\mathbf{h} \in \mathcal{H}$ ,  $\lim_{n \rightarrow \infty} \mathbb{P}^n(\cap_{l=1}^L E_{h_l}) = 1$ , but with an exponential convergence rate compared to the power law scaling in the previous section. But it also says that the precision of localisation remains fixed at  $r$  rather than increasing with  $n$  like in the previous section.

### 5.3 Extension to Randomised Lattice Deployment

In the previous sections we analysed the performance of ED-HD proportionality approximation for uniform i.i.d. deployment. In this section we will prove a similar result for the randomised lattice deployment. In randomised lattice node deployment, the unit area is split into  $n$  cells each of area  $\frac{1}{n}$ , and in each cell exactly one node is deployed, uniformly over the cell area. The locations of the nodes in two different cells are independent of each other. We denote by  $\mathbb{P}_{RL}^{(n)}(\cdot)$  the probability

We have the following notation,



Hence,  $\lim_{n \rightarrow \infty} \mathbb{P}_{RL}^n(E_{h_l}(n)) = 1$ . Following a similar analysis as in Section 5.1.2 for  $L$  anchors, we can state the following theorem for randomised lattice node deployment.

**Theorem 10** *For a given  $1 > \epsilon > 0$ , and  $r(n) = c\sqrt{\frac{\ln n}{n}}$ ,  $c > \frac{1}{\sqrt{\pi}}$ ,  $\forall \mathbf{h} = [h_1, \dots, h_l, \dots, h_L] \in \mathcal{H}(n)$ ,*

$$\mathbb{P}_{RL}^n \left( \cap_{l=1}^L E_{h_l}(n) \right) = 1 - \mathcal{O} \left( \frac{1}{n^{g(\epsilon)c^2}} \right)$$

where  $g(\epsilon) = q(\epsilon)\sqrt{1-p^2(\epsilon)}$ ,

$$p(\epsilon) = \frac{1-\epsilon+\sqrt{(1-\epsilon)^2+8}}{4}, q(\epsilon) = \frac{-3(1-\epsilon)+\sqrt{(1-\epsilon)^2+8}}{4}$$

■

## 5.4 ED bound on a Single Blade with Poisson Deployment

Here we consider another kind of deployment, where we pick the number of nodes with the distribution  $\text{Poisson}(n)$  and deploy these nodes uniformly over the area  $\mathcal{A}$ . The number of nodes falling in  $\mathcal{A}$  is a random variable with mean  $n$ , and since we are throwing the picked nodes uniformly over  $\mathcal{A}$ , the nodes falling in disjoint areas are independent and Poisson distributed with rate proportional to the area considered. Hence, for disjoint strips with area  $u(n)t(n)$  each and the number of node selection being  $\text{Poisson}(n)$ , the number of nodes falling in each strip is  $\text{Poisson}(nu(n)t(n))$ , independent and identically distributed. Let the probability law associated with this kind of deployment be denoted by  $\mathbb{P}_{Po}^n(\cdot)$ . Let us focus our attention to a certain blade  $\mathcal{B}_j^l$  as shown in Figure 5.2 pivoted at the anchor location  $b_l$ . We also denote the maximum and minimum Euclidean distance travelled by a  $h_l$  hop path within this blade by  $\overline{D}_l^{\mathcal{B}_j^l}(\mathbf{v}, h_l)$  and  $\underline{D}_l^{\mathcal{B}_j^l}(\mathbf{v}, h_l)$  respectively. Now, ensuring at least one node in each of the  $h_l - 1$  strips of  $\mathcal{B}_j^l$  will ensure the event  $E_{h_l}^{\mathcal{B}_j^l}(n) = \{\mathbf{v} : (1 - \epsilon)(h_l - 1)r(n) \leq \underline{D}_l^{\mathcal{B}_j^l}(\mathbf{v}, h_l) \leq \overline{D}_l^{\mathcal{B}_j^l}(\mathbf{v}, h_l) \leq h_l r(n)\}$  also occurs. So, we

have for the given  $\epsilon > 0$ ,

$$\begin{aligned}
1 &\geq \mathbb{P}_{Po}^n(E_{h_l}^{\mathcal{B}_j^l}(n)) \\
&\geq \mathbb{P}_{Po}^n\left(\cap_{i=1}^{h_l-1} A_{i,j}^l\right) \\
&= \left(1 - e^{-nu(n)t(n)}\right)^{h_l-1} \\
&= \left(1 - n^{-c^2q\sqrt{1-p^2}}\right)^{h_l-1}
\end{aligned} \tag{5.10}$$

The second inequality comes because  $\{\mathbf{v} \in \cap_{i=1}^{h_l-1} A_{i,j}^l\} \subseteq \{\mathbf{v} \in E_{h_l}^{\mathcal{B}_j^l}(n)\}$  and the first equality comes because of the independence of the number of nodes due to Poisson deployment and disjoint strips. Since in this deployment we are not using the union bound, the expression for probability is exact. Hence the lower bound on the probability of the event  $E_{h_l}^{\mathcal{B}_j^l}(n)$  is tighter, yet the rate of convergence follows the power law ( $e^{-nu(n)t(n)} = n^{-q\sqrt{1-p^2}c^2}$ ), which shows that the rate of convergence is not affected by the union bound used in the case of uniform i.i.d. and randomised lattice deployments.

## 5.5 Simulation Results

In this section, we illustrate Theorem 5 through simulation. We deploy  $n$  nodes in uniform i.i.d. fashion on the unit square  $\mathcal{A}$ , and form the geometric graph  $\mathcal{G}(\mathbf{v}, r(n))$ , where  $r(n) = \frac{4}{\sqrt{\pi}} \sqrt{\frac{\ln n}{n}}$ . We also have 4 anchors at the 4 corners of  $\mathcal{A}$ .

### 5.5.1 Illustration of Theorem 5 with increasing $n$ for a fixed $\epsilon$ and HD

Recall from Theorem 5, the Euclidean distance of a node from a fixed anchor lies in  $[(1 - \epsilon)(h_1 - 1)r(n), h_1r(n)]$  with probability  $\geq 1 - (h_1 - 1) \left\lceil \frac{\pi h_1}{2\sqrt{1-p^2(\epsilon)}} \right\rceil n^{-g(\epsilon)c^2}$ , for a given  $\epsilon$  and a hop-distance  $h_1$  from anchor  $b_1$ .

In this section, we fix  $\epsilon = 0.4$  and hop-distance  $h_1 = 5$  from anchor  $b_1$  located at the bottom-left corner of the unit square  $\mathcal{A}$ . The results are summarised in Table 5.1 and illustrates how the theoretical bounds given in Theorem 5 becomes tighter as we increase the number of nodes  $n$ ,

$n$	$r(n)$	$(1 - \epsilon)(h_1 - 1)r(n)$	$\underline{D}_1$	$\overline{D}_1$	$h_1 r(n)$	PLB	EP
1000	0.1876	0.4494	0.6934	0.9053	0.9362	0.37	1
2000	0.1391	0.3336	0.5196	0.6678	0.6950	0.61	1
3000	0.1166	0.2796	0.4313	0.5590	0.5826	0.70	1
4000	0.1028	0.2465	0.3761	0.4929	0.5136	0.75	1
5000	0.0931	0.2235	0.3428	0.4559	0.4655	0.79	1
6000	0.0859	0.2062	0.3123	0.4191	0.4295	0.81	1

Table 5.1:  $(1 - \epsilon)(h_1 - 1)r(n)$  and  $h_1 r(n)$  are found from Theorem 5.  $\underline{D}_1$  and  $\overline{D}_1$  are the maximum and minimum EDs from anchor 1 given the hop-distance  $h_1 = 5$ . The theoretical Probability Lower Bound (PLB)  $= 1 - (h_1 - 1) \left[ \frac{\pi h_1}{2\sqrt{1-p^2(\epsilon)}} \right] e^{-ng(\epsilon)r^2(n)}$ , and the Empirical Probability (EP) is found from this experiment.  $r(n) = \frac{4}{\sqrt{\pi}} \sqrt{\frac{\ln n}{n}}$ ,  $\epsilon = 0.4$ .

keeping the hop-distance  $h_1$  and  $\epsilon$  fixed. In this simulation, we used  $h_1 = 5$  and  $\epsilon = 0.4$ .

Figures 5.6 and 5.7 show the theoretical bounds given by Theorem 5, and only those nodes are shown that have a hop-distance  $h_1 = 5$  from anchor  $b_1$ , for 1000 and 5000 nodes respectively.

### 5.5.2 Illustration of Theorem 5 with decreasing HD for a fixed $n$ and a fixed lower bound on probability

In this section, we have fixed the number of nodes  $n = 5000$  and also fixed the lower bound on probability that the node lies within the bound of  $[(1 - \epsilon)(h_1 - 1)r(n), h_1 r(n)]$  (as given by Theorem 5) at 0.80. Figures 5.9, 5.10 and 5.11 show that as we decrease the hop-distance  $h_1$ , the bound on the ED becomes tighter, which implies that if we keep the lower bound fixed, the  $\epsilon$  that achieves that lower bound will be smaller for smaller hop-distances, as predicted by Theorem 5.

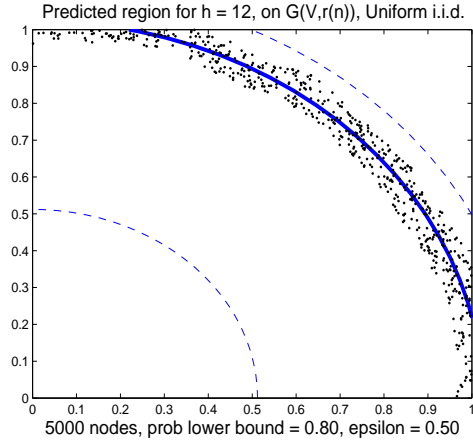


Figure 5.8: 5000 nodes, 12 hops.

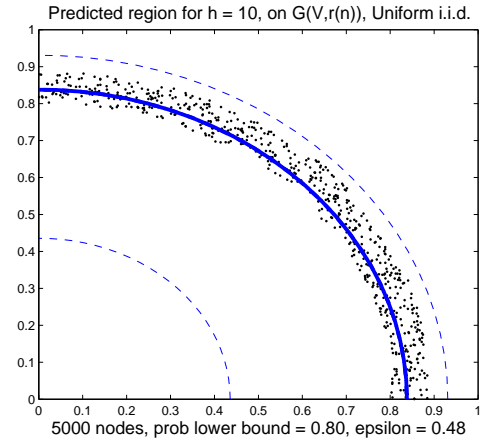


Figure 5.9: 5000 nodes, 10 hops.

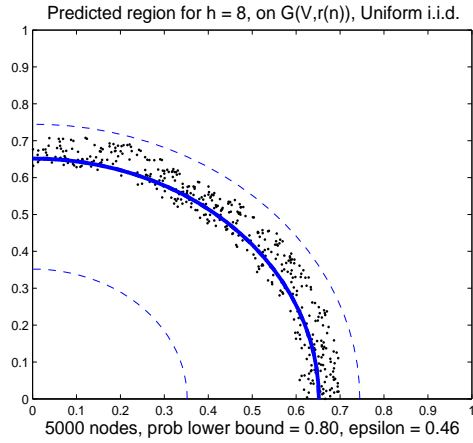


Figure 5.10: 5000 nodes, 8 hops.

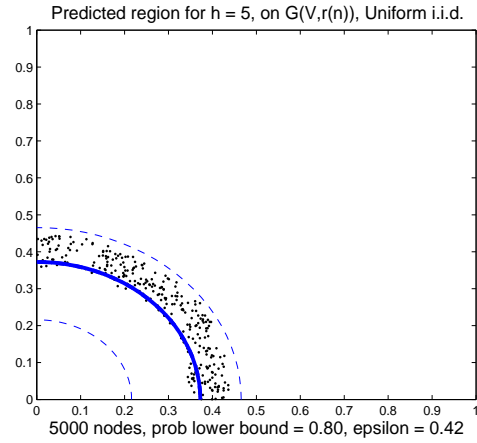


Figure 5.11: 5000 nodes, 5 hops.

The thin dashed lines show the ED bounds given by Theorem 5, the thick solid line shows ED  $(h_1 - 1)r(n)$  from  $b_1$ . For all the cases,  $\mathbb{P}^n(E_1(n)) \geq 0.80$ .  $r(n) = \frac{4}{\sqrt{\pi}} \sqrt{\frac{\ln n}{n}}$ . For 5000 nodes,  $r(n) = 0.0931$ .

$n$	$(1 - \epsilon)(h_1 - 1)r$	$\underline{D}_1$	$\overline{D}_1$	$h_1 r$	PLB	EP
1000	0.2560	0.3483	0.4574	0.5000	0.0000	1
2000	0.2560	0.3479	0.4677	0.5000	0.0000	1
3000	0.2560	0.3775	0.4835	0.5000	0.0000	1
4000	0.2560	0.3741	0.4843	0.5000	0.2906	1
5000	0.2560	0.3831	0.4897	0.5000	0.7733	1
6000	0.2560	0.3705	0.4826	0.5000	0.9275	1

Table 5.2: Radius  $r = 0.1$ ,  $h_1 = 5$ ,  $\epsilon = 0.36$ . The theoretical PLB  $= 1 - (h_1 - 1) \left[ \frac{\pi h_1}{2\sqrt{1-p^2(\epsilon)}} \right] e^{-ng(\epsilon)r^2}$ . Abbreviations are as defined in Table 5.1.

### 5.5.3 Illustration of convergence in probability for the geometric graph with fixed radius

Recall from Theorem 7, the Euclidean distance of a node from a fixed anchor lies in  $[(1 - \epsilon)(h_1 - 1)r, h_1 r]$  with probability  $\geq 1 - (h_1 - 1) \left[ \frac{\pi h_1}{2\sqrt{1-p^2(\epsilon)}} \right] e^{-ng(\epsilon)r^2}$ , for a given  $\epsilon$  and a hop-distance  $h_1$  from anchor  $b_1$ .

In this section, we fix the radius of the graph  $\mathcal{G}(\mathbf{v}, r)$ ,  $r = 0.1$  and take  $h_1 = 5$ ,  $\epsilon = 0.36$ . The simulation results are summarised in Table 5.2, which shows that for smaller  $n$ , the lower bound of probability (as given by Equation 5.6) is weak, but the convergence rate, due to its exponential nature, is very rapid with the increase in  $n$ .

# Chapter 6

## Application to Node Localisation

Localisation is defined as the procedure for estimating the location of a node. In a wireless sensor network setup, we use the information of hop distance of a node from the anchor nodes to estimate the location. Usually, the anchor nodes are fixed at some points on the region to be monitored and the nodes are deployed randomly over the area. So, for the localisation application, we will use the fixed point to random node theorems from Chapter 5.

The setup for this application is as follows.  $n$  nodes are deployed uniform i.i.d. on a unit area  $\mathcal{A} = [0, 1]^2$ . We consider the geometric graph with radius  $r(n)$  on this node placements. The hop-distance corresponds to the hops over this geometric graph  $\mathcal{G}(\mathbf{V}, r(n))$  in this chapter. We introduce an algorithm called Hop Count-derived Distance-based Localisation (HCDL), which is explained in detail in the next section, to estimate the location of each node. Also, to use this algorithm, we need the information of the radius  $r(n)$ . We provide a scheme to estimate  $r(n)$  using the ED and HD between the anchor nodes only. We compare the performance of this algorithm with HCRL, proposed by Yang et al. [27], and PDM, proposed by Lim and Hou [17]. Both techniques are discussed in Section 2.4.

## 6.1 Algorithm: Hop Count-derived Distance-based Localisation (HCDL)<sup>1</sup>

From Theorem 5, we know that given a hop distance  $h$  from a fixed anchor located at one of the corners of the unit square, the Euclidean distance lies in  $[(1 - \epsilon)(h - 1)r(n), hr(n)]$  w.h.p. Also, with multiple anchors, the node is expected to be in the intersection of all these annuli as given by Theorem 6 and shown in Figure 5.5. The algorithm HCDL exploits this fact. The algorithm is as follows.

### 6.1.1 The Algorithm

**STEP 1: (Initialisation)** Given the geometric graph  $\mathcal{G}(\mathbf{V}, r(n))$  and the corresponding radius  $r(n)$ , each node finds the hop-distances from the  $L$  anchors and sets up its own  $\mathbf{h} = [h_1, \dots, h_L]$  vector, where  $h_l$  is the hop distance of the node from  $l^{th}$  anchor (This can be carried out by a distributed Bellman-Ford algorithm).

**STEP 2: (Region of Intersection)** Pick a certain node, pick its  $\mathbf{h}$  vector as found from **STEP 1**, set an  $\epsilon$ , small enough, and find the region of intersection formed by the annuli of radii  $[(1 - \epsilon)(h_l - 1)r(n), h_l r(n)]$  centred at the  $l^{th}$  anchor location,  $l = 1, \dots, L$ , for that tagged node.

**STEP 3: (Terminating Condition)** Check if the region of intersection is non-empty, otherwise increase the value of  $\epsilon$ . For a finite number of nodes  $n$  and a small enough  $\epsilon$ , it is possible that the annuli do not have a common region. A graphical illustration is given in Figure 5.5 for 4 anchors. The value of  $\epsilon$  for which an intersection is found, can be different for different nodes. Hence, this step can be stated as follows.

**IF** there is an intersection, declare the centroid of the region of intersection as the estimate of the node. **GO TO STEP 4.**

**ELSE** increase  $\epsilon$  by an amount  $k$ ,  $0 < k < 1$ . **GO TO STEP 2.**

---

<sup>1</sup>This is a joint work with Venkatesan N.E. and Prof. P. Vijay Kumar

**STEP 4: (Repetition)** Repeat **STEP 2** to **STEP 3** for all  $n$  nodes.

**STEP 5: STOP**

**Time complexity of HCDL:** Assuming that given a region of intersection, the computation of the centroid is of constant time, the time complexity to find the hop distances of  $n$  nodes from the  $L$  anchors is  $\mathcal{O}(n^2)$ . Hence, **STEP 1** completes in  $\mathcal{O}(n^2)$  time. After getting the hop distances, we are increasing the  $\epsilon$  in each iteration and if there is an intersection in the annuli, we compute the centroid, which is an  $\mathcal{O}(1/k)$  ( $k$  is the step size of the increment of  $\epsilon$ ) computation. So, for each node the time complexity of the **STEPS 2 to 4** is  $\mathcal{O}(1/k)$ . For  $n$  nodes, it is  $\mathcal{O}(n/k)$ . Hence, the time complexity of HCDL algorithm is  $\mathcal{O}(n^2) + \mathcal{O}(n/k)$ .

### 6.1.2 Estimating the value of $r(n)$

We notice that the choice of  $r(n)$  only requires the network to be connected, and hence we can use the critical graph (the connected geometric graph with smallest radius for a certain node deployment) for localisation using HCDL. Algorithm that gives the critical graph or any geometric graph does not usually give the value of the radius of that geometric graph (e.g., the DISCRIT algorithm proposed by Acharya in [1] gives an approximation of  $\mathcal{G}_{crit}(\mathbf{V})$ , but does not give the  $r_{crit}$ ), thus, compelling us to find out an estimate of the true  $r(n)$  ( $\hat{r}(n)$ ) using the hop distance between the anchors and the Euclidean distance between them. In the following, we illustrate one such method to estimate the  $r(n)$  using the HD and ED between the anchors and the point-node theory (Chapter 5).

Assume two anchors  $b_{l_1}$  and  $b_{l_2}$  with their locations fixed and a hop-distance  $h$  between them. Being anchors, the Euclidean distance between them,  $d_{l_1 l_2}$  is known. From this information, we want to get an estimate of the radius  $r(n)$  of the geometric graph using the point-node theory of ED-HD relationship only (Chapter 5). For that, we pick an intermediate node  $k$  on the shortest path between  $b_{l_1}$  and  $b_{l_2}$ , such that the node is nearest to the straight line joining  $b_{l_1}$  and  $b_{l_2}$ , as shown in Figure 6.1. Denoting the vertical distance of  $k$  from the straight line as  $w$ , we claim that,  $w \leq r(n)$ . The argument is as follows, if we consider the node that is one hop away from  $b_{l_1}$  in the shortest path between  $b_{l_1}$  and  $b_{l_2}$ , it lies within a circle of radius  $r(n)$  of  $b_{l_1}$ . Hence its vertical distance from



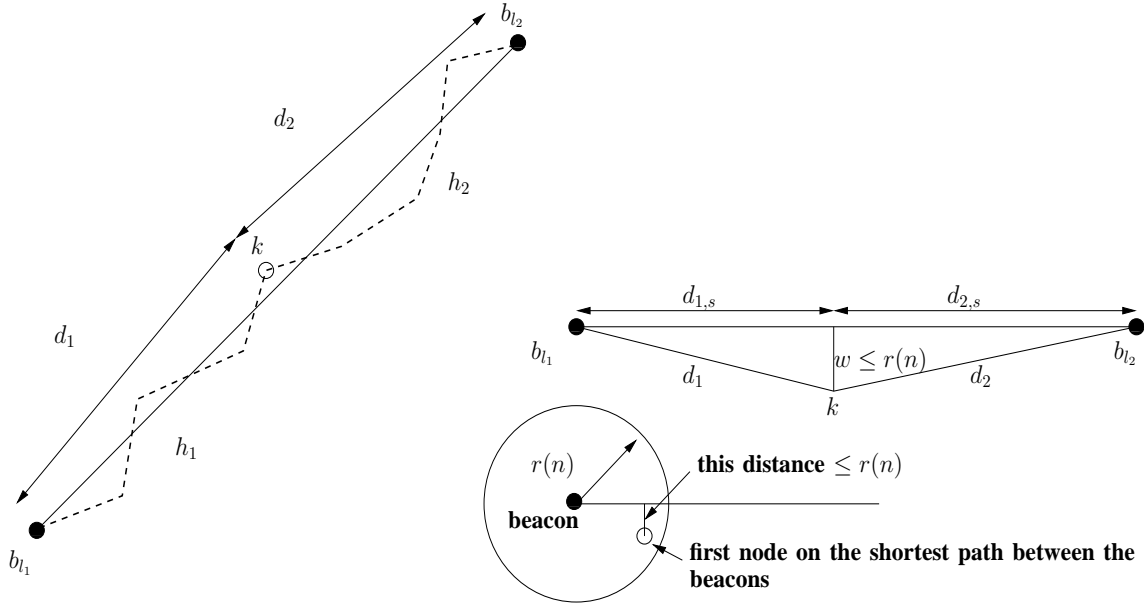


Figure 6.1: Construction to show how  $r(n)$  is estimated in  $\mathcal{G}(\mathbf{V}, r(n))$  using the ED and HD between the anchors only and the node-point theory.

the straight line cannot be more than  $r(n)$  (see Figure 6.1), and since we are considering the node that is closest to the line, hence,  $w \leq r(n)$ . We can state this result in the form of the following lemma.

**Lemma 7** *A node on the shortest path between two fixed anchors  $b_{l_1}$  and  $b_{l_2}$  and closest to the straight line joining them, is no farther than  $r(n)$  from the line.*

Let us denote the hop distance between  $b_{l_1}$  and  $k$  by  $h_1$  and that between  $k$  and  $b_{l_2}$  by  $h_2$ , and the Euclidean distances by  $d_1$  and  $d_2$  respectively. Since  $k$  is on the shortest path between  $b_{l_1}$  and  $b_{l_2}$ ,  $h = h_1 + h_2$ . The perpendicular from  $k$  on the straight line joining  $b_{l_1}$  and  $b_{l_2}$ , divides the line into two parts of length  $d_{1,s}$  and  $d_{2,s}$  as shown in Figure 6.1. We see that  $d_{l_1 l_2} = d_{1,s} + d_{2,s}$ , and by triangle inequality,  $d_{l_1 l_2} \leq hr(n)$ . Following is the calculation to find out a lower bound on  $d_{l_1 l_2}$ .

$$d_{1,s} = \sqrt{d_1^2 - w^2}, \quad d_{2,s} = \sqrt{d_2^2 - w^2}$$

Now, according to Theorem 5,  $d_i \geq (1 - \epsilon)(h_i - 1)r(n)$ ,  $i = 1, 2$  w.h.p. and we know,  $h = h_1 + h_2$

and  $w \leq r(n)$ . Hence,

$$d_{i,s} = \sqrt{d_i^2 - w^2} \geq \sqrt{[(1 - \epsilon)^2(h_i - 1)^2 r^2(n) - r^2(n)]^+}, i = 1, 2 \quad (6.1)$$

Where,  $x^+ = \max\{0, x\}$ , the above is true since distance cannot be negative.

$$\begin{aligned} \therefore d_{l_1 l_2} &= d_{1,s} + d_{2,s} \geq r(n) \left( \sqrt{[(1 - \epsilon)^2(h_1 - 1)^2 - 1]^+} + \sqrt{[(1 - \epsilon)^2(h_2 - 1)^2 - 1]^+} \right) \\ \Rightarrow hr(n) &\geq d_{l_1 l_2} \geq r(n) \left( \sqrt{[(1 - \epsilon)^2(h_1 - 1)^2 - 1]^+} + \sqrt{[(1 - \epsilon)^2(h_2 - 1)^2 - 1]^+} \right) \\ \Rightarrow \frac{d_{l_1 l_2}}{h} &\leq r(n) \leq \frac{d_{l_1 l_2}}{\sqrt{[(1 - \epsilon)^2(h_1 - 1)^2 - 1]^+} + \sqrt{[(1 - \epsilon)^2(h_2 - 1)^2 - 1]^+}} \end{aligned} \quad (6.2)$$

All the above statements are w.h.p. The quantity  $r(n)$  is highly probable to lie within this interval as  $n \rightarrow \infty$ , but we do not have the distribution of  $r(n)$  within this interval. So, with this information, if we need to estimate  $r(n)$ , we should choose a distribution that gives the largest entropy. For a continuous random variable with bounded support, the uniform distribution is the entropy maximiser. Hence, the estimate of  $r(n)$  is,

$$\hat{r}(n) \approx \frac{1}{2} \left( \frac{d_{l_1 l_2}}{h} + \frac{d_{l_1 l_2}}{\sqrt{[(1 - \epsilon)^2(h_1 - 1)^2 - 1]^+} + \sqrt{[(1 - \epsilon)^2(h_2 - 1)^2 - 1]^+}} \right)$$

We notice that, the choice of a node  $k$  explained above, only gives us  $w \leq r(n)$ , which is true even if we take  $k$  to be the node one hop away from  $b_{l_2}$  and  $h - 1$  hops away from  $b_{l_1}$  on the shortest path between  $b_{l_1}$  and  $b_{l_2}$  (instead of taking  $k$  to be the node nearest to the straight line joining  $b_{l_1}$  and  $b_{l_2}$ ). The inequality of Equation 6.1 still holds and the estimate of  $r(n)$  given by Equation 6.2 remains valid. Hence,  $h_1 = h - 1$  and  $h_2 = 1$ , and  $\hat{r}(n) = \frac{1}{2} \left( \frac{d}{h} + \frac{d}{\sqrt{(1 - \epsilon)^2(h - 2)^2 - 1}} \right)$ .

This is how we estimate the  $r(n)$  for a pair of anchors. For localisation we have  $L$  anchors and  $\binom{L}{2}$  pairs of anchors, each of which will yield an estimate of  $r(n)$ . In HCDL, we use an average of all these  $\binom{L}{2}$  values of  $\hat{r}(n)$  to get the final estimate  $\bar{r}(n)$  of  $r(n)$  for a given  $\epsilon$ .

Figures 6.2 and 6.3 show the error histograms of  $\hat{r}(n)$  for  $\epsilon = 0.02$  and  $0.2$  respectively. While simulating the HCDL with the true value of  $r(n)$ , it was found that the maximum value of  $\epsilon$  over

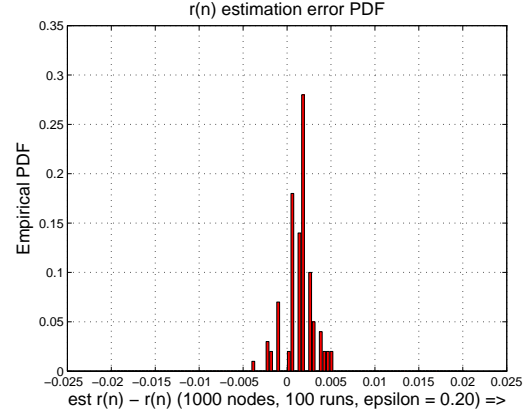
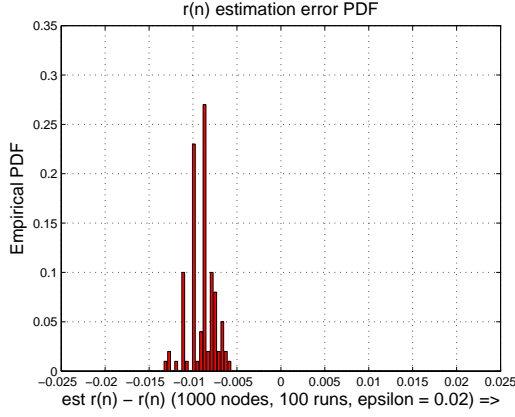


Figure 6.2:  $r(n)$  estimation error for  $\epsilon = 0.02$       Figure 6.3:  $r(n)$  estimation error for  $\epsilon = 0.20$

$$r(n) = \frac{2}{\sqrt{\pi}} \sqrt{\frac{\ln n}{n}}, \text{ 1000 nodes, 100 runs, 100 bins in } [-0.02, 0.02].$$

all nodes remains near the value of 0.2, and the minimum was close to 0.02 (the starting value of  $\epsilon$  was taken to be 0.01 and was increased in steps of 0.01 ( $= k$ )). This was the reason to choose these values for  $\epsilon$ . In both the cases, it turns out that the error in estimating  $r(n)$  was within  $\pm 0.010$ .

## 6.2 Simulations implementing HCDL on $\mathcal{G}_{crit}(\mathbf{V})$

In this section, we try out the HCDL algorithm on the critical graph formed by  $n$  nodes deployed uniform i.i.d. on  $\mathcal{A} = [0, 1]^2$ . We know from Chapter 5 that for the point-node theory, there is no restriction in choosing the radius  $r(n)$ , other than requiring the network to be connected. Hence we can choose  $r(n)$  to be the critical radius itself and hence can try out the algorithm on the critical graph  $\mathcal{G}_{crit}(\mathbf{V})$ . We take the number of anchors,  $L = 4$ , placed at the 4 corners of the unit square. We start with  $\epsilon = 0.01$  and increase with step size  $k = 0.01$ . In each step of increasing  $\epsilon$ , we compute  $\bar{r}(n)$  and use that value for computing the intersection of the annuli of radii  $[(1 - \epsilon)(h_l - 1)\bar{r}(n), h_l\bar{r}(n)]$ ,  $l = 1, \dots, L$ . Figure 6.4 shows the error pattern of the localisation of 1000 nodes using HCDL and Figures 6.5 and 6.6 show the error pattern for the same node deployment with the localisation strategy being PDM and HCRL respectively. Figure 6.7 shows the comparison between the CDFs of localisation errors in the three strategies.

The simulation results clearly show that for the same node deployment, the HCDL algorithm,

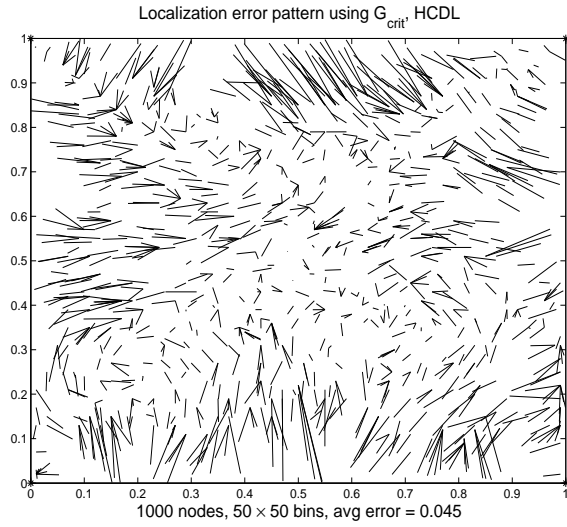


Figure 6.4: HCDL: 1000 nodes.

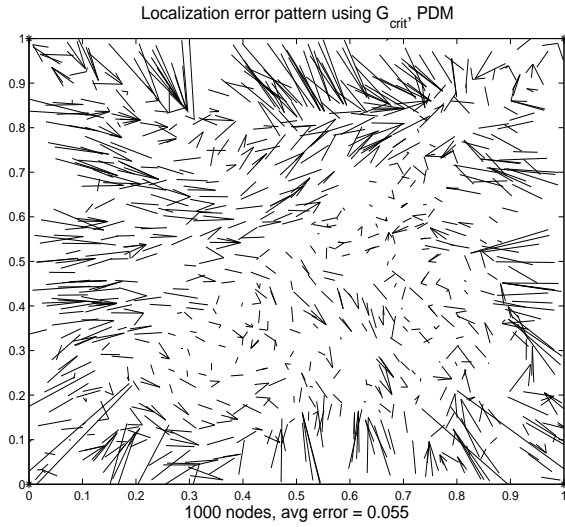


Figure 6.5: PDM: 1000 nodes.

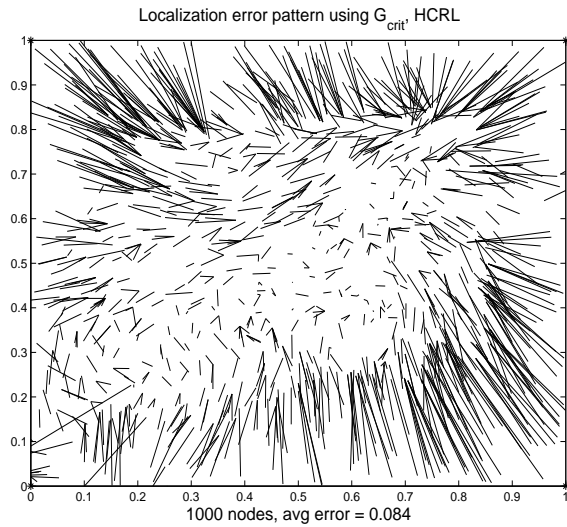


Figure 6.6: HCRL: 1000 nodes.

This error pattern is found by joining the true location of a node with its estimated location.  $r_{crit}$  was 0.0583 for the three error pattern plots.

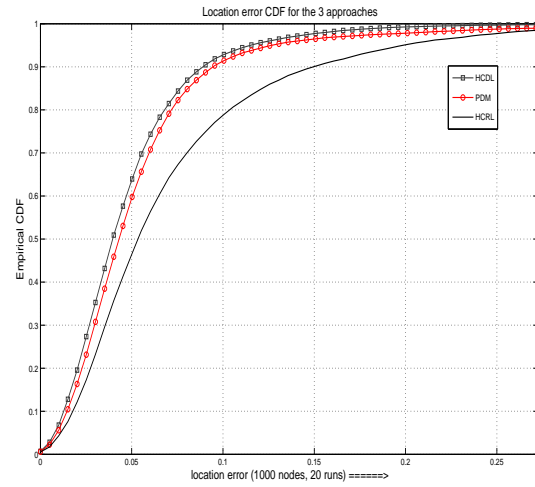


Figure 6.7: Comparison of error CDFs.

which uses the fact of the ED-HD relationship in a random geometric graph, performs better in estimating the node locations than HCRL, which uses the heuristic of ED being directly proportional to HD, or PDM, that approximates Euclidean distance distance vector of a node from anchor nodes to be a linear transformation of the corresponding hop-distance vector.

## 6.3 A Heuristic for Localisation

In this section, we present a heuristic way of localising nodes depending upon the empirical distribution of the  $\frac{d}{hr_{crit}}$ . To find the empirical distribution, we take the Euclidean distance  $d$  of a node from a fixed anchor, find the corresponding hop-distance  $h$  over the critical geometric graph, find the ratio  $\frac{d}{hr_{crit}}$ , repeat it for all nodes and plot the histogram. For plotting the histogram, we use the true  $r_{crit}$  value, whereas during the localisation, where we will use this empirical distribution, we estimate the value of  $r_{crit}$  as explained in Appendix B. Once we have this empirical distribution of  $\frac{d}{hr_{crit}}$ , we know how the ED  $d$  is distributed, given the hop-distance and the estimated critical radius  $r_{crit}$ . We use ED distribution from a finite number of anchors to get the location distribution of the nodes and then we do an MMSE estimate to locate the nodes. The approach has been described in detail in Appendix B. Despite having only empirical verification of the distribution and a simplifying assumption of independence of the distances from the 4 anchors, it performs better than the HCDL algorithm.

# Chapter 7

## Real Scenarios

The theorems presented in the Chapters 3, 4 and 5, use the geometric graph abstraction of the wireless sensor network. We proved results assuming the node deployment being either uniform i.i.d., randomised lattice or Poisson, all of which are homogeneous deployment over  $\mathcal{A}$ . In Section 7.1.1, we show that for positive non-homogeneous deployment density of nodes on a unit area, the theorems still remain valid.

In this chapter, we consider cases where the node deployment density hits zero at certain regions of the network, or the radiation pattern of the antennae of the sensor nodes are directional, or the radio propagation on the wireless media is susceptible to fading and shadowing. We note that, if it is possible to construct a geometric graph in a distributed asynchronous fashion, over a network with the above mentioned non-idealities, localisation could have been done using HCDL as before.

So, in such environments, one solution approach to localisation might be to construct the geometric graph by some means and apply HCDL, or another approach is to localise without constructing the geometric graph by some different algorithm.

In this chapter, we do not introduce any new algorithm for localisation in anisotropic environment. But in Section 7.2, we assume a model for radiation anisotropy and apply HCDL on the graph given by the anisotropic model. It turned out that the performance in terms of location error distribution is approaching that of the true geometric graph as number of nodes become large.

## 7.1 Non-homogeneous node deployment

### 7.1.1 Node deployment density is positive on all points

Let  $v_i$  be the location of the  $i^{th}$  node on  $\mathcal{A}$  and the distribution of the node  $f(v_i) > 0, \forall v_i \in \mathcal{A}$ . We have shown in Section 5.1.1, that even with this kind of a node placement, the Theorems 5 and 6 hold in the point-node case. This is because, in Equation 5.2, the probability  $\mathbb{P}^n(A_{i,j}^{l,c})$  can now be upper bounded by  $(1 - f_{min}u(n)t(n))^n$ , where  $f_{min}$  is the minimum density of a node over  $\mathcal{A}$ , and the upper bound still  $\rightarrow 0$  as  $n \rightarrow \infty$  as long as  $f_{min} > 0$ . Although, a small value of  $f_{min}$  will affect the rate of convergence of the probability  $\mathbb{P}^n\{\mathbf{v} : (1 - \epsilon)(h_l - 1)r(n) \leq \underline{D}_l(\mathbf{v}, h_l) \leq \overline{D}_l(\mathbf{v}, h_l) \leq h_l r(n)\}$  to 1 and we will need more nodes to guarantee the Euclidean distance to lie within the bounds given mentioned before with a high probability.

### 7.1.2 Node deployment density hits zero at some points

If  $f(v_i) = 0$ , for some  $v_i \in \mathcal{A}$  (regions with such points are also called *holes*), the ED and HD ceases to be proportional, and the localisation algorithms that uses the proportionality between them, performs worse. This is because, if there is a hole in the path between the node and the anchor, the relationship derived in the previous chapters are no longer valid and the hop-distance multiplied by the radius of the geometric graph is much larger than the true Euclidean distance. This phenomena has been illustrated in Figure 7.1.

This issue has been addressed in the paper by Li and Liu [16], where they propose a new algorithm called REndered Path (REP). This algorithm tries to figure out the true ED between two nodes from the path information and the shape of the holes. The assumption there is that the number of holes and their boundary information is known. Also they assume that outside the holes the node density is homogeneous and the ED-HD proportionality is valid there. When a hole comes in between the node and an anchor, the shortest path takes a route over the boundary of the hole, and using several geometric propositions, they prove that the true ED can be figured out in such a setup. Now, with a ED information about a node from 3 anchors, the same triangulation technique can be applied to estimate the location of the node. The complexity of REP is claimed to be  $\mathcal{O}(nL)$ , where  $n$  is

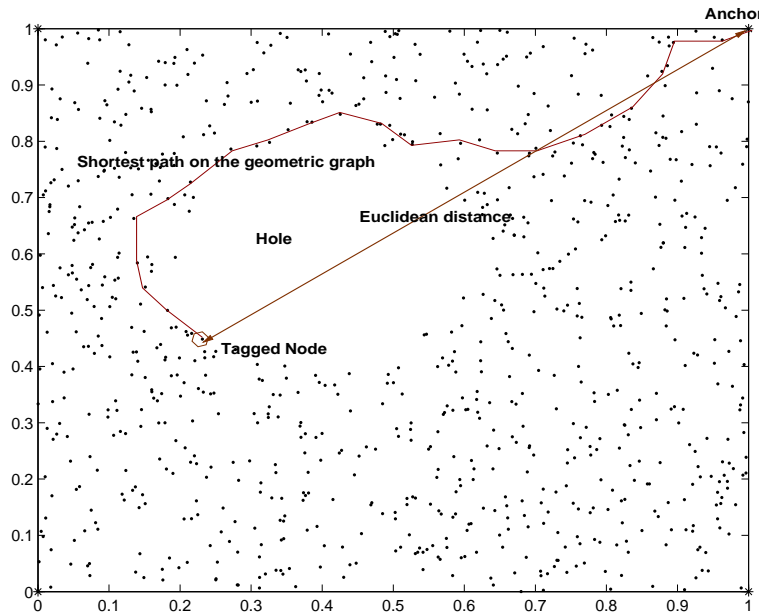


Figure 7.1: We see that the shortest path from the anchor to the tagged node takes a path along the boundary of the hole on the geometric graph. Hence, ED is no longer proportional to HD in this setup.

the number of nodes and  $L$  is the number of holes (This is the complexity when the hop distance information between the nodes is available beforehand).

In a node deployment with holes, we can have the following observations:

1. The number of anchor nodes need to be higher in the case of non-homogeneity. Our previous examples with 4 beacons would not suffice since there might be holes in the path. Hence, the anchors should be large in number and deployed uniformly over the non-hole region, and localisation using 3 nearest anchors is expected to give an accurate estimate of the location. In [16], the typical number of anchors is in the multiples of 10.
2. Regarding the ED-HD relationship in node-node case, we need the hop-distance to be larger. Hence, we need a dense deployment of nodes so that  $r(n)$  decreases and the hop-distance between the node and an anchor increases. But we recall that the result presented in Chapter 4 was a sufficient condition. We also need to find a necessary condition for node-node ED-HD relationship, which will be helpful in localisation using randomly placed anchor nodes.



## 7.2 Anisotropic radiation

In real networks, the radiation pattern of the antennae of the sensor nodes are anisotropic. So, the nodes experience different gains depending on the location and random orientation of the antenna. In this section, we have taken an easier model for the antenna anisotropy and tested the localisation algorithm HCDL using the network formed by such nodes and compared its performance with the corresponding isotropic network. The anisotropic antenna model is shown in Figure 7.2, and the orientation axis of the antenna of a node is picked uniformly over  $[0, 2\pi]$ . And two nodes share an edge if and only if they are within  $r(n)$  of each other and also one node is in the angular spread  $\theta$  of the other and vice-versa.

Figures 7.3 to 7.6 (on the left are the graphs formed for isotropic radiation pattern and on the left the same for anisotropic; we took  $\theta = \frac{\pi}{2}$  and  $r(n) = \frac{2}{\sqrt{\pi}} \sqrt{\frac{\ln n}{n}}$  for these simulations) show that the geometric graph formed by such a radiation pattern is sparser than that of the isotropic one. There are few isolated nodes for 1000 nodes, which diminishes as  $n$  increases to 5000. In Figures 7.7 to 7.10, we show the error patterns of the nodes with isotropic and anisotropic radiation patterns for a certain deployment for 1000 and 5000 nodes. In the anisotropic case, we have eliminated the isolated nodes from the error computing. The CDF of the location errors is given in Figures 7.11 and 7.12. We can observe from the simulation outputs that with the increase in  $n$ , not only the geometric graph becomes connected, but also the ED starts following some relationship with HD, similar to the theorems stated in previous chapters. That is why, the error CDF for the anisotropic radiation comes closer to that for isotropic as we increase  $n$  from 1000 to 5000.

**A proposition:** We can figure out the isotropic geometric graph from the anisotropic radiation pattern setup through collaborating with the first hop neighbours of each node. Even if a node is not connected to another node, which is within  $r(n)$  of it on  $\mathcal{A}$ , because of anisotropy, it is highly probable that some of its neighbours must be having a link with it when the node density is high. It is, therefore, seems reasonable to disseminate information about the neighbours among nodes to figure out the true geometry of the isotropic geometric graph.

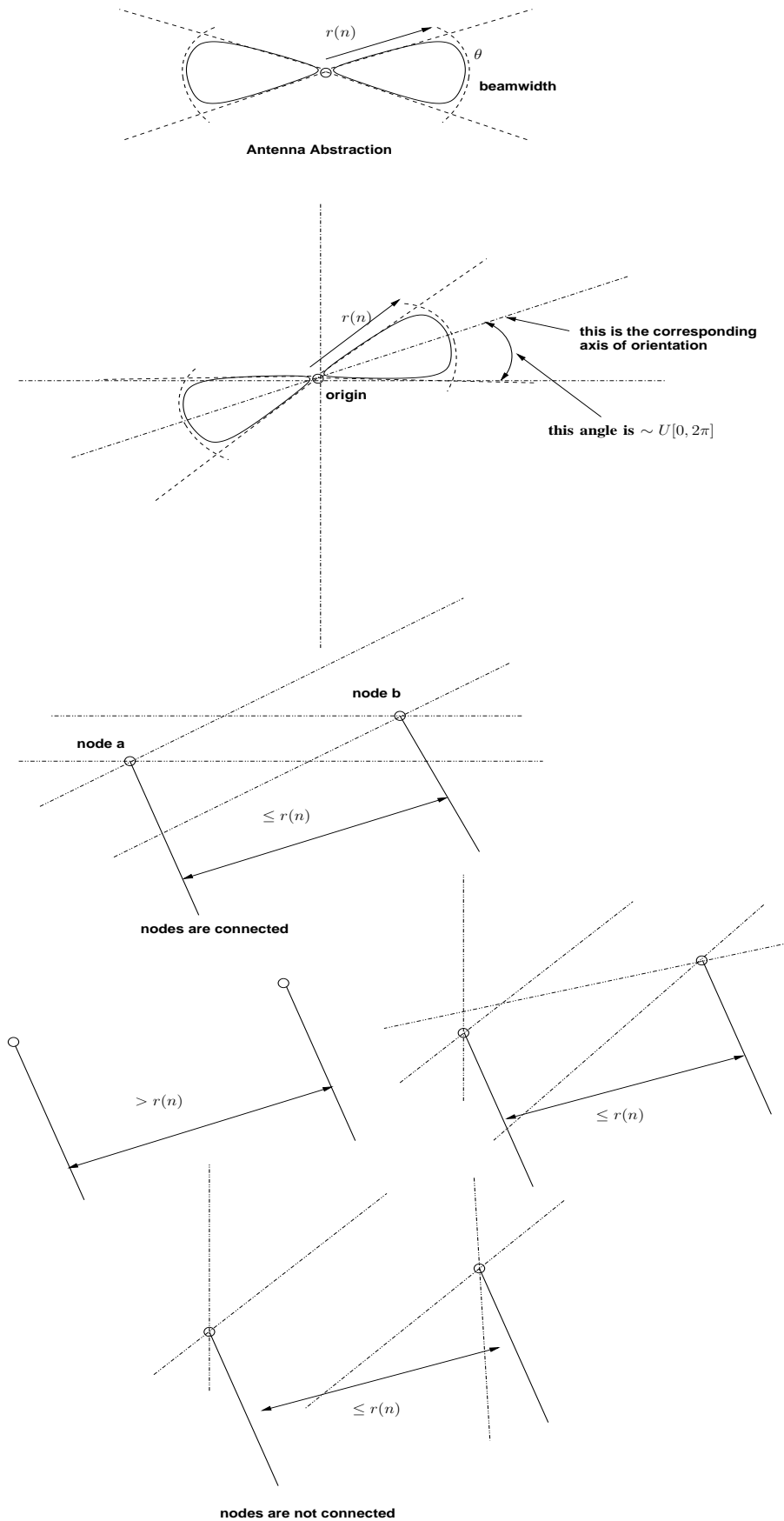


Figure 7.2: The setup for the Anisotropic Network.

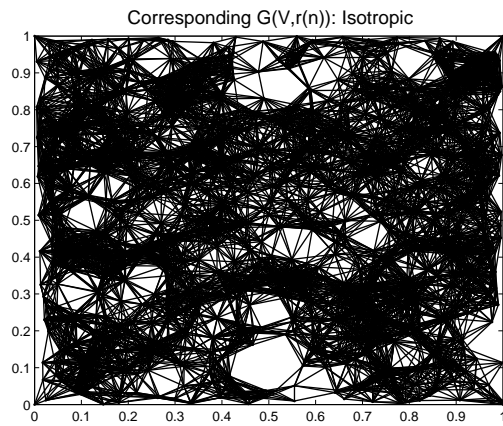


Figure 7.3: GG for 1000 nodes: Isotropic.

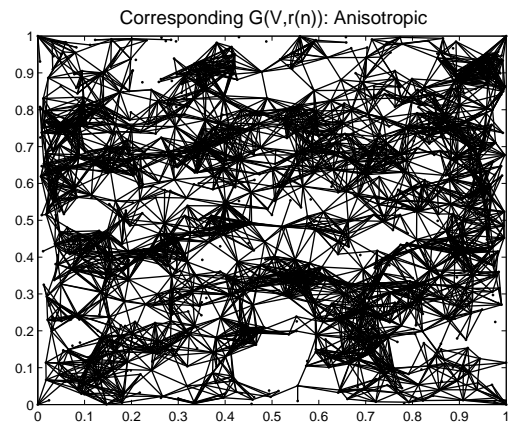


Figure 7.4: GG for 1000 nodes: Anisotropic.

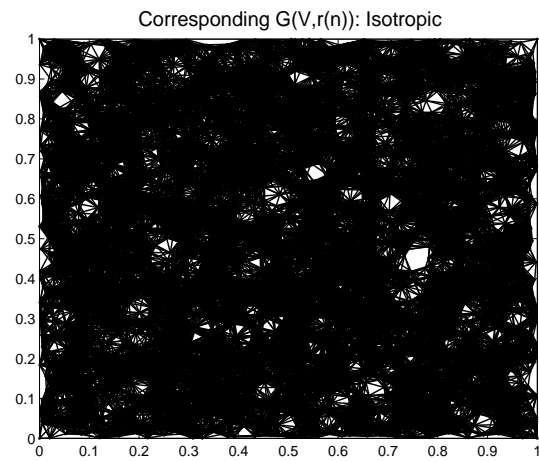


Figure 7.5: GG for 5000 nodes: Isotropic.

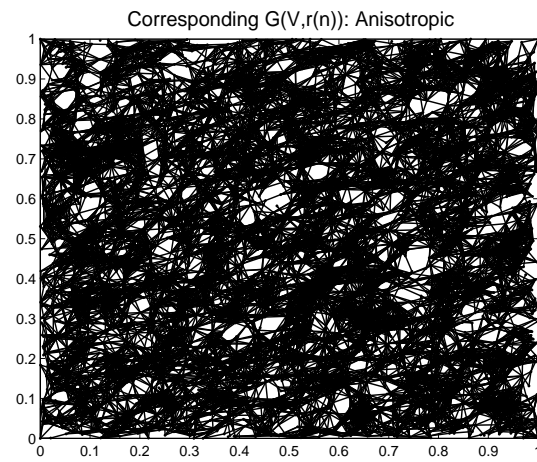


Figure 7.6: GG for 5000 nodes: Anisotropic.

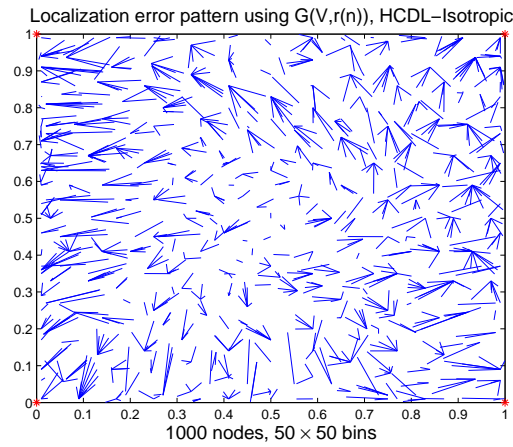


Figure 7.7: Error for 1000 nodes: Isotropic.

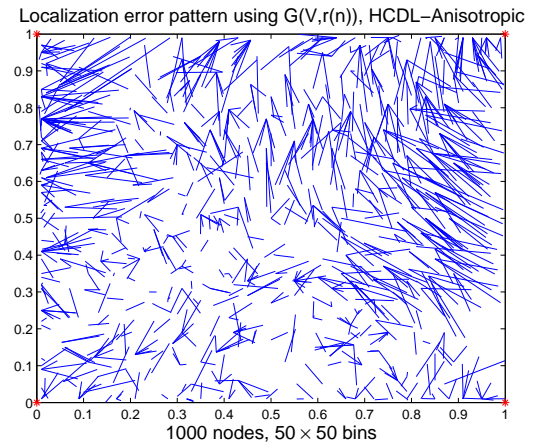


Figure 7.8: Error for 1000 nodes: Anisotropic.

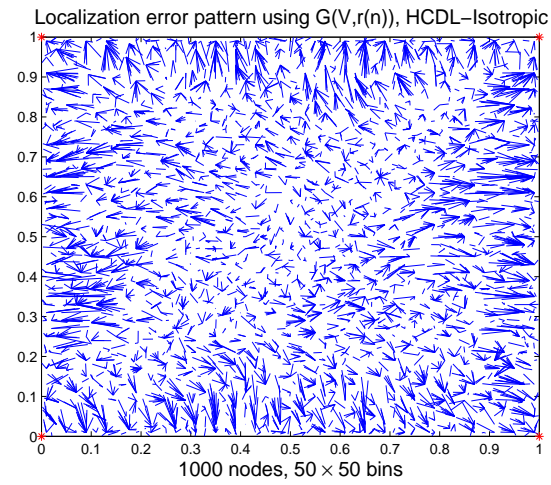


Figure 7.9: Error for 5000 nodes: Isotropic.

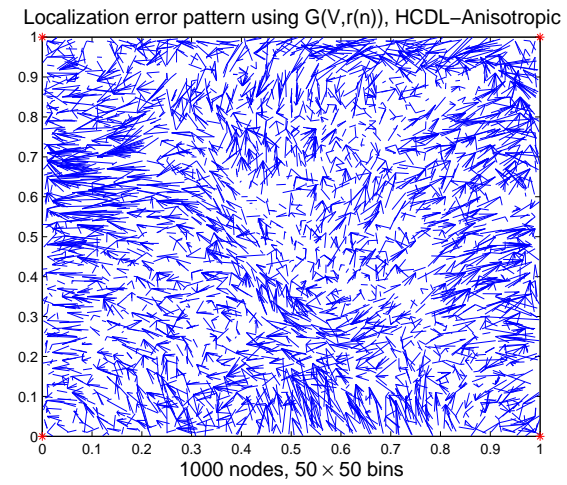


Figure 7.10: Error for 5000 nodes: Anisotropic.

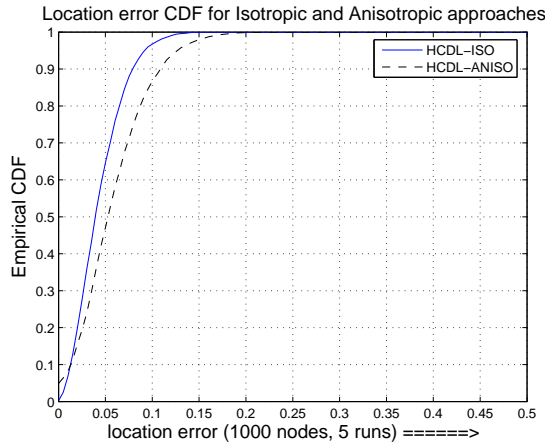


Figure 7.11: Error CDF comparison for 1000 nodes.

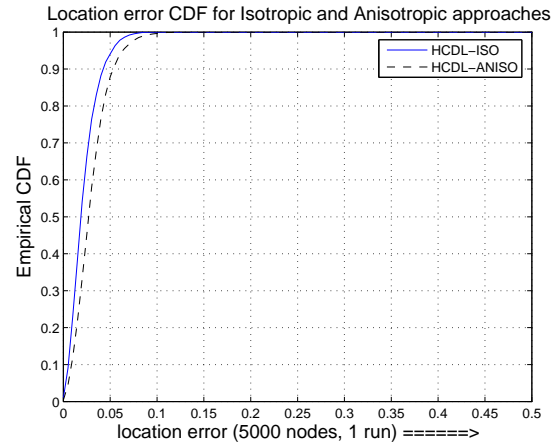


Figure 7.12: Error CDF comparison for 5000 nodes.

## 7.3 Anisotropic propagation

The propagation of radio signals in wireless networks are susceptible to non-idealities like fading, shadowing etc. In absence of all these the radio propagation is affected by only path loss, and the received power decreases monotonically with distance. This monotonicity is lost whenever the non-idealities are taken into consideration. The geometric graph will not be a good model of the communication graph for that kind of network, since a node far away in Euclidean distance may be connected whereas a nearer node may not be. Hence there will not be any upper bound on ED given the HD, as we had in a geometric graph.

# Chapter 8

## Conclusions and Work Ahead

From the discussions of the previous chapters, we can conclude the following.

1. We have proved hop-distance is approximately proportional to Euclidean distance in a dense random geometric graph with positive density of nodes on all points in an area.
2. We were able to prove a sufficient condition for the proportionality between the ED and HD for point-point (Chapter 3) and node-node (Chapter 4) cases, which requires the radius of the geometric graph to be scaled in a certain fashion. We see that this radius is larger than the radius for the network to be just connected (the critical radius). Future scope of work in this area is in finding a necessary condition for this proportionality.
3. In point-node case (Chapter 5), however, we proved that given  $HD = h$ ,  $(1 - \epsilon)(h - 1)r < ED \leq hr$  w.h.p. The parameter  $\epsilon$  provides a trade-off between ED-HD proportionality and the rate of convergence of the desired probability to 1. This theory does not require any specific scaling of the radius of the geometric graph and hence, can be used for the critical graph.
4. We used the node-point theory in localising nodes in Chapter 6, and proposed an algorithm HCDEL, which can be used on any geometric graph including the critical graph and uses only the information of the hop-distances of a node from the 4 anchors. This is the same information required for the HCRL algorithm, but HCDEL gives a much better location estimate than

HCRL.

We illustrated a heuristic algorithm in Appendix B using the  $\frac{d}{hr_{crit}}$  distribution, which performs better than both HCDL and HCRL in localisation. Future scope of work is in formalising this approach theoretically.

5. In Chapter 7, we discussed about some challenging issues in a real wireless sensor network, e.g., anisotropy in antenna radiation patterns and in radio propagation. We also looked into networks with *holes*, where the proposal of using more number of anchor nodes and revision of node-node theory can be looked into in future.

Simulation results show that HCDL performs well even in a dense network having nodes with anisotropic antenna radiation patterns. It compels to think that a result similar to node point theory is true even for this kind of a setup. Formalising this observation is also a future work.

# Appendix A

## Proof of Lemma 1

The setting for this lemma is as described in Section 3.3 and as depicted in Figure 3.5. We had an  $h_l + 1$  sided regular polygon with sides of length  $r$ . We deleted a certain edge  $sd$  and increased all angles except the two adjacent to the deleted edge by a small amount  $\delta$ . The resulting figure is as shown in Figure A.1. The length of edge  $sd$  in this new figure is  $r_1$ . We restate the lemma.

**Lemma:** For  $h_l > 2$  and  $0 < \delta < \frac{4\pi}{h_l+1}$ ,  $r_1 > r$

**Proof:** Each of the internal angles except the adjacent angles of  $sd$  is  $\phi = \frac{(h_l-1)\pi}{h_l+1} + \delta$  (see Figure A.1) and angles adjacent to  $sd$  are,

$$\begin{aligned}\phi' &= \frac{1}{2}[\text{total internal angle} - (h_l - 1) \times \phi] \\ &= \frac{1}{2} \left[ (h_l - 1)\pi - (h_l - 1) \left( \frac{(h_l - 1)\pi}{h_l + 1} + \delta \right) \right] \\ &= \frac{(h_l - 1)\pi}{h_l + 1} - \frac{(h_l - 1)}{2}\delta \quad h_l > 2\end{aligned}\tag{A.1}$$

It can be easily proved<sup>1</sup> that the angle bisectors of all these internal angles meet at the point  $c$  as shown in Figure A.2. Now, we apply sine rule in the triangles  $\triangle sdc$  and  $\triangle sp_1c$ , where  $p_1$  is the adjacent node of  $s$  other than  $d$  (see Figure A.2).  $z$  is the length of the line segment connecting  $s$  and  $c$ . For  $\triangle sdc$ ,

---

<sup>1</sup>It can be shown that  $(\pi - \phi') + 2(\pi - \frac{\phi}{2} - \frac{\phi'}{2}) + (h_l - 2)(\pi - \phi) = 2\pi$



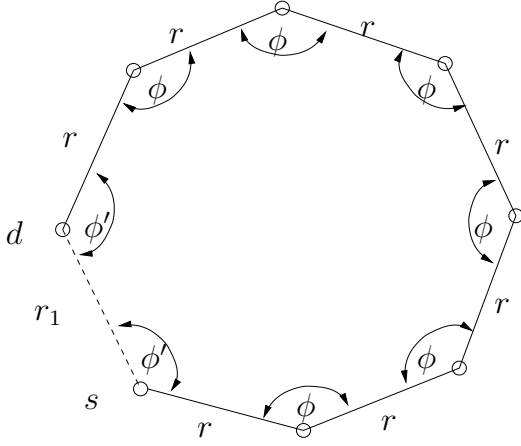


Figure A.1:  $\phi = \frac{(h_l-1)\pi}{h_l+1} + \delta$  and  $\phi' = \frac{(h_l-1)\pi}{h_l+1} - \frac{(h_l-1)}{2}\delta$ .

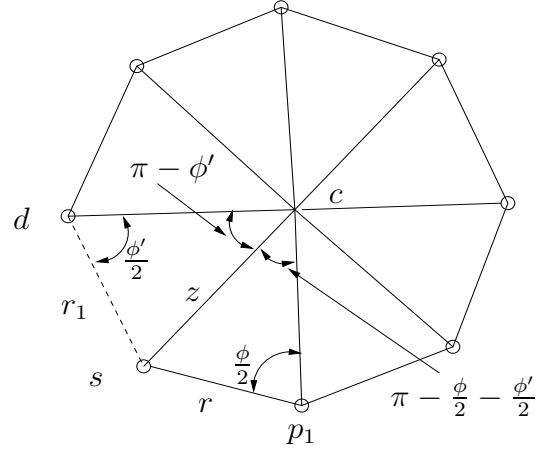


Figure A.2:  $c$  is the point where the angle bisectors meet.

$$\begin{aligned} \frac{r_1}{\sin(\pi - \phi')} &= \frac{z}{\sin \frac{\phi'}{2}} \\ z &= \frac{r_1}{2 \cos \frac{\phi'}{2}} \end{aligned} \quad (\text{A.2})$$

and for  $\triangle sp_1c$ ,

$$\begin{aligned} \frac{z}{\sin \frac{\phi}{2}} &= \frac{r}{\sin(\pi - \frac{\phi}{2} - \frac{\phi'}{2})} \\ z &= \frac{r \sin \frac{\phi}{2}}{\sin(\frac{\phi+\phi'}{2})} \end{aligned} \quad (\text{A.3})$$

Eliminating  $z$  from these two equations, we get,

$$r_1 = \frac{2r \sin \frac{\phi}{2} \cos \frac{\phi'}{2}}{\sin \frac{\phi}{2} \cos \frac{\phi'}{2} + \cos \frac{\phi}{2} \sin \frac{\phi'}{2}}$$

Now we have,  $\frac{\phi-\phi'}{2} = \frac{h_l+1}{4}\delta$  and  $\frac{\phi+\phi'}{2} = \frac{(h_l-1)\pi}{h_l+1} - \frac{(h_l-3)}{4}\delta$ . Hence, to show,

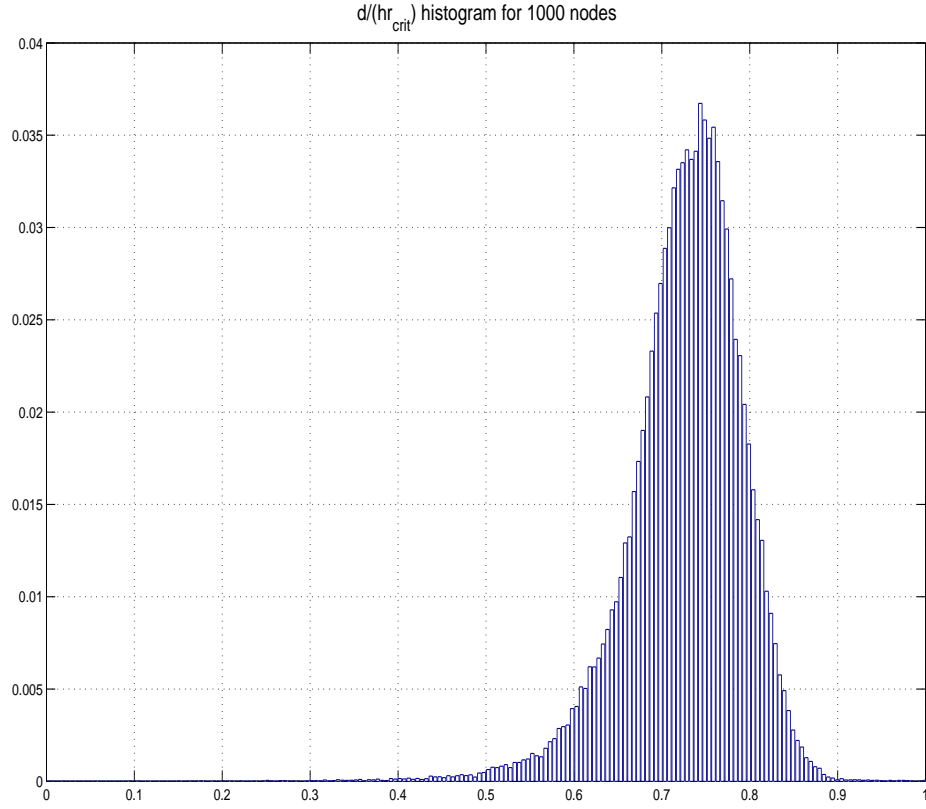
$$\begin{aligned}
 & r_1 > r \\
 \Rightarrow & r_1 - r > 0 \\
 \Rightarrow & r \frac{\sin \frac{\phi-\phi'}{2}}{\sin \frac{\phi+\phi'}{2}} > 0 \\
 \Rightarrow & r \frac{\sin \frac{h_l+1}{4}\delta}{\sin \left( \frac{(h_l-1)\pi}{h_l+1} - \frac{(h_l-3)}{4}\delta \right)} > 0 \tag{A.4}
 \end{aligned}$$

For the numerator to be positive,  $\pi > \frac{h_l+1}{4}\delta > 0 \Rightarrow \frac{4\pi}{h_l+1} > \delta > 0$ . For the denominator, the condition is  $\frac{4(h_l-1)\pi}{(h_l+1)(h_l-3)} > \delta > 0$ , for  $h_l \geq 3$ . Together, the condition on  $\delta$  is  $\frac{4\pi}{h_l+1} > \delta > 0$ , to have  $r_1 > r$ , for  $h_l > 2$  (since  $\frac{h_l-1}{h_l-3} > 1$  and so,  $\frac{4(h_l-1)\pi}{(h_l+1)(h_l-3)} > \frac{4\pi}{h_l+1}$ ,  $\forall h_l \geq 3$ ). Hence proved. ■

## Appendix B

### Localisation using the Empirical Distribution of $\frac{d}{hr_{crit}}$

This localisation algorithm depends on the histogram of  $\frac{d}{hr_{crit}}$  found by taking the ratio of Euclidean distance of 1000 nodes, deployed uniform i.i.d. over  $\mathcal{A} = [0, 1]^2$ , from the fixed anchor at the origin to the product of the hop-distance, measured on  $\mathcal{G}_{crit}(\mathbf{V})$ , of the node and the corresponding critical radius  $r_{crit}$ . Figure B.1 shows the histogram, where we took 200 bins in  $[0, 1]$ . From the critical graph, we find out the hop-distances of a node from the 4 anchors and call it the hop-distance vector  $\mathbf{h}_i = [h_{i,1} \ h_{i,2} \ h_{i,3} \ h_{i,4}] \in \mathbb{N}^4$ . If  $v_i$  is the location of the  $i^{th}$  node on  $\mathcal{A}$ , and  $\mathbf{h}_i$  be the hop-distance vector, we know that the location of that node is uniquely determined by the distances  $d_{i,l}$ ,  $l = 1, 2, 3, 4$  ( $d_{i,l}$  = Euclidean distance between  $i^{th}$  node location ( $v_i$ ) and  $l^{th}$  anchor location ( $b_l$ ), i.e.,  $||v_i - b_l||$ ) from the anchors. So, we can write the following.

Figure B.1:  $\frac{d}{hr_{crit}}$  Histogram.

$$\begin{aligned}
 f(v_i | \mathbf{h}_i, r_{crit}) &= f(v_i | h_{i,1}, \dots, h_{i,4}, r_{crit}) \\
 &= \frac{f(h_{i,1}, \dots, h_{i,4} | v_i, r_{crit}) f(v_i, r_{crit})}{f(h_{i,1}, \dots, h_{i,4}, r_{crit})} && \text{Bayes theorem} \\
 &\propto f(h_{i,1}, \dots, h_{i,4} | v_i, r_{crit}) \\
 &\approx \prod_{l=1}^4 f(h_{i,l} | v_i, r_{crit}) \\
 &= \prod_{l=1}^4 \frac{f(v_i | h_{i,l}, r_{crit}) f(h_{i,l}, r_{crit})}{f(v_i, r_{crit})} \\
 &\propto \prod_{l=1}^4 f(v_i | h_{i,l}, r_{crit}) && \text{(B.1)}
 \end{aligned}$$

The assumption in the proportionality of step 3 is the independence of  $r_{crit}$  with the node location  $v_i$  (then the factor  $f(v_i, r_{crit}) = f(v_i)f(r_{crit})$  becomes independent of  $v_i$ ). But we know, the critical radius is roughly  $c\sqrt{\frac{\ln n}{n}}$ ,  $c > \frac{1}{\sqrt{\pi}}$ , by Gupta-Kumar [9], which is independent of the node locations. The approximation in the fourth step can be argued as follows. If the anchors are spaced far apart, so that given the node location  $v_i$ , the shortest path of the node from the anchors are non-intersecting, the assumption of independence of hop-distance given  $v_i$  is reasonable. That is why we took the 4 anchors at the 4 corners of  $\mathcal{A}$ , and shortest path are most likely to be non-intersecting. Let us denote  $X = \frac{d}{hr_{crit}}$  and its distribution by  $f_X(\cdot)$ . Assuming the random variable  $X$  to be independent of the hop-distance, we compute,

$$f(v_i|h_{i,l}, r_{crit}) = \begin{cases} \frac{f_X\left(\frac{\|v_i - b_l\|}{h_{i,l}r_{crit}}\right)}{\int_{v_i \in \mathcal{R}_{i,l}} f_X\left(\frac{\|v_i - b_l\|}{h_{i,l}r_{crit}}\right) dv_i} & \forall v_i \in \mathcal{R}_{i,l} = \{v : \|v - b_l\| \leq h_{i,l}r_{crit}\} \cap \mathcal{A} \\ 0 & \text{otherwise} \end{cases} \quad (\text{B.2})$$

We see that the denominator of the first term above is independent of the location  $v_i$ . Therefore,

$$\begin{aligned} f(v_i|\mathbf{h}_i, r_{crit}) &\propto \prod_{l=1}^4 f(v_i|h_{i,l}, r_{crit}) \\ &\propto \begin{cases} \prod_{l=1}^4 f_X\left(\frac{\|v_i - b_l\|}{h_{i,l}r_{crit}}\right) & v_i \in \cap_{l=1}^4 \mathcal{R}_{i,l} \\ 0 & \text{otherwise} \end{cases} \\ &= \begin{cases} \prod_{l=1}^4 f_X\left(\frac{d_{i,l}}{h_{i,l}r_{crit}}\right) & v_i \in \cap_{l=1}^4 \mathcal{R}_{i,l} \\ 0 & \text{otherwise} \end{cases} \end{aligned} \quad (\text{B.3})$$

With this distribution of node location, we now use the minimum mean square error (MMSE) estimate of the node location. Hence the location of node  $i$  is estimated as,

$$\hat{v}_i = \mathbb{E}(v_i|\mathbf{h}_i, r_{crit}) = \int_{v_i \in \cap_{l=1}^4 \mathcal{R}_{i,l}} v_i f(v_i|\mathbf{h}_i, r_{crit}) dv_i \quad (\text{B.4})$$

$\mathcal{G}_{crit}(\mathbf{V})$  can be found using a distributed algorithm, but the information about  $r_{crit}$  may not be known. So, we use the following technique to estimate the critical radius. We use the ED and HD information between the anchors. Let  $D$  and  $H$  be the ED and HD between a pair of anchors and both are known apriori. Hence all the distributions and expectations used in the following analysis are given these two. Now we need to find the distribution of the critical radius  $R$  given  $D$  and  $H$ , using the distribution of  $X = \frac{d}{hR}$  (i.e.,  $f_X$ ). We assume,  $X$  to be independent of  $D$  and  $H$ . Hence, we have,

$$\begin{aligned}
 F_R(r|D \in [d, d + \Delta d), H = h) &= \mathbb{P}(R \leq r|D \in [d, d + \Delta d), H = h) \\
 &= \mathbb{P}\left(\frac{d}{hR} \leq r|D \in [d, d + \Delta d), H = h\right) \\
 &\approx \mathbb{P}\left(X \geq \frac{d}{hr}\right) \\
 &= \int_{\frac{d}{hr}}^1 f_X(y) dy \quad \text{since } X \in [0, 1] \\
 &= 1 - F_X\left(\frac{d}{hr}\right)
 \end{aligned} \tag{B.5}$$

The approximation on the third line is due to the independence assumption. Differentiating,

$$f_R(r|D \in [d, d + \Delta d), H = h) = \frac{d}{hr^2} f_X\left(\frac{d}{hr}\right) \tag{B.6}$$

Now, for MMSE estimate of  $R$ , we have,

$$\begin{aligned}
 \hat{R} = \mathbb{E}(R|D \in [d, d + \Delta d), H = h) &= \int_{\frac{d}{h}}^{\infty} r f_R(r|D \in [d, d + \Delta d), H = h) dr \\
 &\quad \text{since } 0 \leq \frac{d}{hr} \leq 1, \text{ hence } \frac{d}{h} \leq r \leq \infty \\
 &= \int_{\frac{d}{h}}^{\infty} r \frac{d}{hr^2} f_X\left(\frac{d}{hr}\right) dr \\
 &= \int_{\frac{d}{h}}^{\infty} \frac{d}{hr} f_X\left(\frac{d}{hr}\right) dr \\
 &= \int_1^0 z f_X(z) \left(-\frac{d}{hz^2}\right) dz \quad \text{let } \frac{d}{hr} = z, dr = -\frac{d}{hz^2} dz \\
 &= \int_0^1 \frac{d}{hz} f_X(z) dz
 \end{aligned} \tag{B.7}$$

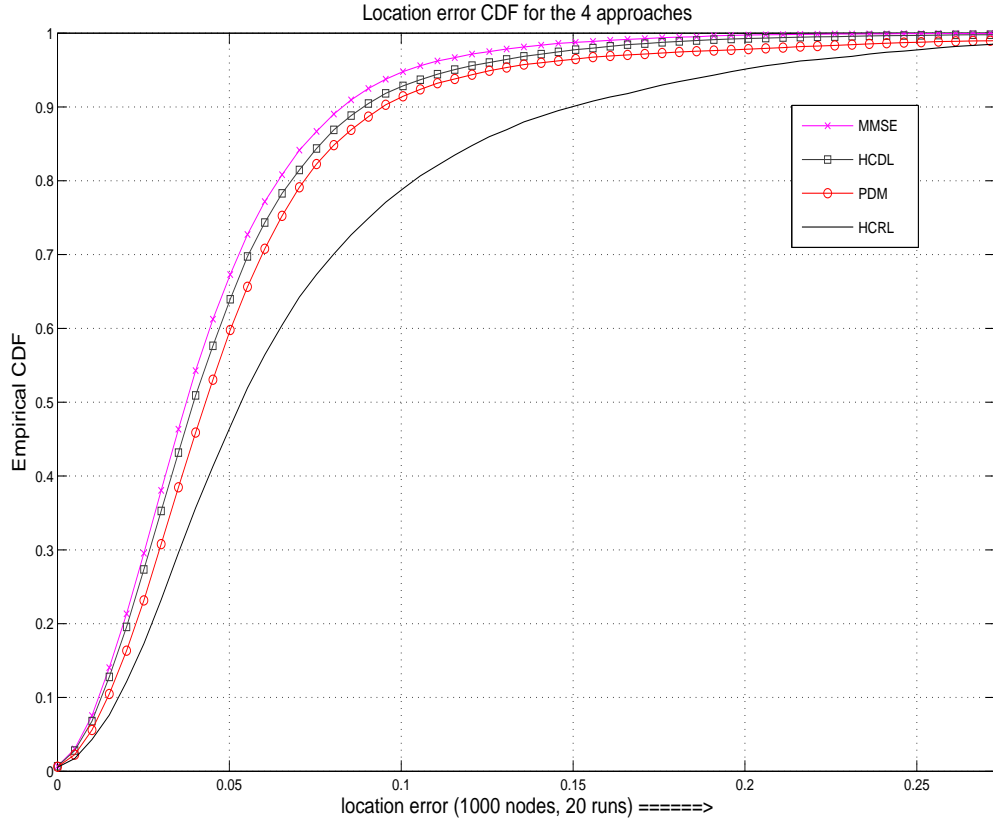


Figure B.2: Localisation error CDFs of MMSE, HCDL, PDM &amp; HCRL.

We carry out this integration numerically in the following way. For a certain pair of anchors  $p$  (there are 6 such pairs for 4 beacons), denoting the ED and HD between them by  $d_p$  and  $h_p$  respectively,

$$\hat{r}_{crit,p} = \mathbb{E}_p(r_{crit}|d_p, h_p) \quad (\text{B.8})$$

We have found the  $\frac{d}{hr_{crit}}$  distribution using 200 bins in  $[0, 1]$ , let the bin vector be represented by  $value_j, j = 1, 2, \dots, 200$ , and we have the corresponding probability masses  $\pi(value_j)$ . So, the above expression is calculated according to Equation B.7 as,

$$\begin{aligned} \hat{r}_{crit,p} &= \mathbb{E}_p(r_{crit}|d_p, h_p) \\ &= \sum_{j=1}^{200} \frac{d_p}{h_p value_j} \pi(value_j) \end{aligned} \quad (\text{B.9})$$

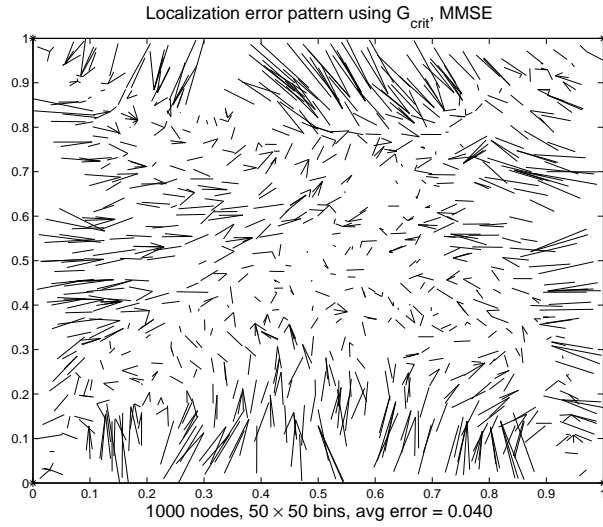


Figure B.3: Error for MMSE.

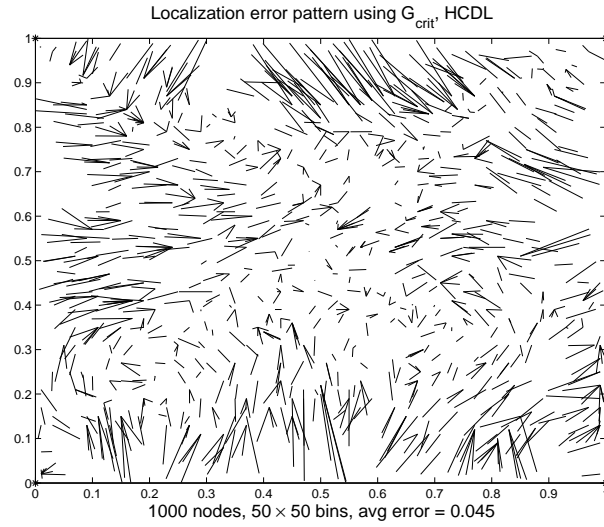


Figure B.4: Error for HCDL.

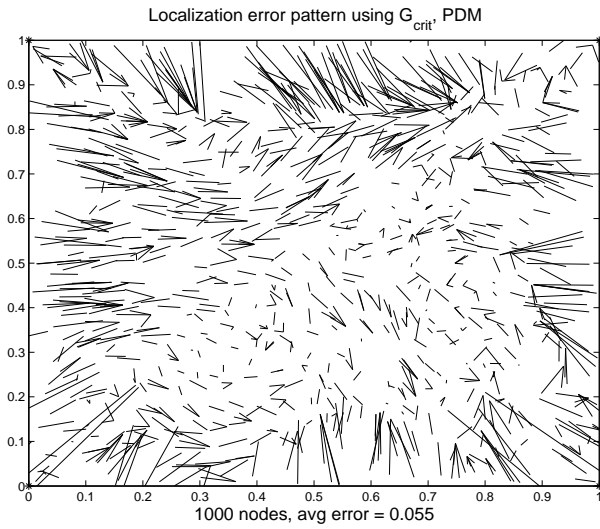


Figure B.5: Error for PDM.

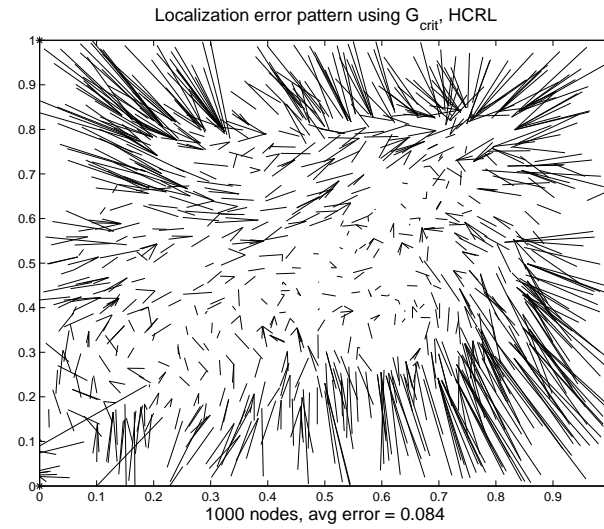


Figure B.6: Error for HCRL.

$r_{crit}$  was 0.0583 for this example deployment.



Finally,

$$\hat{r}_{crit} = \frac{1}{6} \sum_{p=1}^6 \hat{r}_{crit,p} \quad (\text{B.10})$$

This is our final estimate for  $r_{crit}$ . We will use this value of  $r_{crit}$  for location estimate.

Figure B.2 shows the error CDF of such an approach and compares it with the other three approaches, viz. HCDL, PDM and HCRL. Figures B.3 to B.6 show error pattern plots for these different schemes on one such deployment. It shows that this heuristic approach performs better than the other three approaches.

# Bibliography

- [1] Srivathsa Acharya. Distributed Self-Organisation in Dense Wireless Ad Hoc Sensor Networks. ME Thesis, Indian Institute of Science, Bangalore, June 2007.
- [2] W. B. Heinzelman, A. P. Chandrakasan, and Hari Balakrishnan. An application-specific protocol architecture for wireless microsensor networks. *IEEE Transactions on Wireless Communications*, 1(4), October 2002.
- [3] Lorna Booth, Jehoshua Bruck, Matthew Cook, and Massimo Franceschetti. Ad hoc wireless networks with noisy links. Technical report, California Institute of Technology, 2003.
- [4] David Braginsky and Deborah Estrin. Rumor routing algorithm for sensor networks. In *First ACM Int. Workshop on Wireless Sensor Networks and Applications (WSNA)*, pages 22–31, Sept. 28 2002. Atlanta.
- [5] Alberto Cerpa and Deborah Estrin. ASCENT: Adaptive self-configuring sensor networks topologies. In *IEEE Trans. on Mobile Computing, Special Issue on Mission-Oriented Sensor Networks*, volume 3, page 272285. IEEE, 2004.
- [6] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. In *Mobicom*, 2001.
- [7] Stefan Dulman, Michele Rossi, Paul Havinga, and Michele Zorzi. On the hop count statistics for randomly deployed wireless sensor networks. *Int. J. Sensor Networks*, I(1/2), 2006.
- [8] Deepak Ganesan, Ramesh Govindan, Scott Shenker, and Deborah Estrin. Highly-resilient, energy-efficient multipath routing in wireless sensor networks. In *Mobile Computing Commun. Rev.*, pages 11–25. ACM SIGMOBILE, 2001.

- [9] Piyush Gupta and P. R. Kumar. Critical power for asymptotic connectivity in wireless networks. *Stochastic Analysis, Control, Optimization and Applications*, 1998.
- [10] Piyush Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2), March 2000.
- [11] Jason Hill, Robert Szewczyk, Alec Woo, Seth Hollar, David Culler, and Kristofer Pister. System architecture directions for networked sensors. In *Proc. ACM Architectural Support for Programming Languages and Operating Systems (ASPLOS IX)*, 2000.
- [12] Chalermek Intanagonwiwat, Ramesh Govindan, and Deborah Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *Sixth Annual Int. Conf. Mobile Computing and Networking (Mobicom '00)*, 2000. Boston.
- [13] J. M. Kahn, R. H. Katz, and K. S. J. Pister. Next century challenges: Mobile networking for smart dust. In *IEEE Int. Conf. on Mobile Computing and Networking (Mobicom 99)*, pages 271–278. IEEE, 1999.
- [14] Nilesh Khude, Anurag Kumar, and Aditya Karnik. Time and energy complexity of distributed computation in wireless sensor networks. *IEEE/ACM Transactions on Networking*, to appear 2008.
- [15] L. Kirousis, E. Kranakis, D. Krizanc, and A. Pelc. Power consumption in packet radio networks. *Theoretical Computational Sciences*, pages 289 – 305, 2000.
- [16] Mo Li and Yunhao Liu. Rendered Path: Range-Free Localization in Anisotropic Sensor Networks with Holes. In *Mobicom*. ACM, September 2007.
- [17] Hyuk Lim and Jennifer C. Hou. Localization for Anisotropic Sensor Networks. volume 1, pages 138– 149. IEEE, Infocom, 2005.
- [18] Radhika Nagpal, Howard Shrobe, and Jonathan Bachrach. Organizing a global coordinate system from local information on an ad hoc sensor network. In *IPSN*. IEEE, 2003.

- [19] S. Narayanaswamy, V. Kawadia, R. Sreenivas, and P. R. Kumar. Power control in ad hoc networks: Theory, architecture, algorithm and implementation of the compow protocol. In *European Wireless*, 2002.
- [20] D. Niculescu and B. Nath. Ad hoc positioning system (aps). In *IEEE Globecom*. IEEE, Nov. 2001.
- [21] M. D. Penrose. *Random Geometric Graphs*. Oxford University Press, 2003.
- [22] Venkatesh Ramaiyan, Anurag Kumar, and Eitan Altman. Jointly optimal power control and hop length for a single cell,dense,ad hoc wireless network, 2006.
- [23] Curt Schurgers and Mani B. Srivastava. Energy efficient routing in wireless sensor networks. In *Military Communication Conference (MILCOM 2001)*, 2001.
- [24] Serdar Vural and Eylem Ekici. Analysis of hop-distance relationship in spatially random sensor networks. In *Mobihoc*. ACM, 2005.
- [25] Chen Wang and Li Xiao. Sensor Localization in Concave Environments. *ACM Transactions on Sensor Networks*, 4(1), January 2008.
- [26] Feng Xue and P. R. Kumar. The number of neighbors needed for connectivity of wireless networks. *ACM/Kluwer Wireless Networks 10*, pages 169–181, 2004.
- [27] Sungwon Yang, Jiyoung Yi, and Hojung Cha. HCRL: A Hop-Count-Ratio based Localization in wireless sensor networks. In *IEEE SECON*. IEEE, 2007.
- [28] Fan Ye, Gary Zhong, Songwu Lu, and Lixia Zhang. GRAdient Broadcast: A robust data delivery protocol for large scale sensor networks. *Wireless Networks*, 11:285–298, 2005.