COVERAGE AND CONNECTIVITY IN THREE DIMENSIONAL WIRELESS SENSOR NETWORKS

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

In Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Sheikh Mohammed Nazrul Alam

May 2010

COVERAGE AND CONNECTIVITY IN THREE DIMENSIONAL WIRELESS SENSOR NETWORKS

Sheikh Mohammed Nazrul Alam, Ph.D.

Cornell University 2010

Terrestrial wireless sensor networks are generally designed based on the assumption that sensor nodes are deployed on a two-dimensional (2D) plane. This assumption is usually invalid in an underwater sensor network, where sensor nodes may be deployed at various depths, thus creating a three-dimensional (3D) network. Other important applications of 3D networks also include future space and atmospheric networks. Consequently, new research challenges now exist in the field of wireless sensor networks, as several coverage and connectivity issues unique to 3D networks require resolution. For example, node placement strategies need to deploy the minimum number of sensor nodes and, at the same time, ensure that all points inside the network are within the sensing range of at least one sensor. All sensor nodes also need to communicate with each other, possibly over a multi-hop path. Establishment of this type of network will depend on the ratio of the communication range and the sensing range of each sensor. In this study, we use deterministic homogeneous sphere based communication and sensing range model to solve this issue, by placing a node at the center of each virtual cell created by truncated octahedron based tessellation of 3D space. This works well when the ratio of communication range and sensing range is greater than 1.7889. On the other hand, for smaller values of this ratio, the solution depends on the degree of communication redundancy needed by the network. We

provide solutions for both limited and full communication redundancy requirements. We also investigate coverage and connectivity issues in 3D networks where nodes were randomly deployed. Since node location can be random, redundant nodes have to be deployed to achieve 100% sensing coverage. However, at any particular time, not all nodes are needed to achieve full sensing coverage. Consequently, a subset of sensor nodes can be dynamically chosen to remain active at a given time to achieve sensing coverage based on their location at that time. One approach to achieve that goal in a distributed and scalable way is to partition the 3D network space into virtual regions or cells, and to keep one node active in each cell. Following this approach, we achieve a fully distributed and highly scalable solution that minimizes the number of active nodes that use cells created by truncated octahedral tessellation of 3D space. By adjusting the radius of each cell, this scheme can be used to achieve *k*-coverage where a point inside a network has to be within the sensing of *k* different sensor nodes with high probability. We analyze the performance of this scheme and found that performance improves significantly where value of *k* is larger than 1.

BIOGRAPHICAL SKETCH

S. M. Nazrul Alam was born and raised in the beautiful South Asian country of Bangladesh. He received his B.Sc. in Computer Science and Engineering from Bangladesh University of Engineering and Technology (BUET) in 2002. He won University Gold Medal and Sharfuddin Gold Medal for obtaining the highest cumulative grade point average in the entire university and in his department, respectively. After his graduation, he joined the Department of Computer Science and Engineering at BUET as a Lecturer. There he taught Operating Systems course for one semester before he left the country to start his graduate studies at the University of Toronto in Toronto, Canada. Before moving to Canada, he never saw snowfall in his life. During his time in Toronto, along with trying to adapt to bitter cold weather, he did research on Content Delivery Networks. He received his M.Sc. degree from University of Toronto in 2004 and moved to Cornell University to start his Ph.D. in Computer Science. Although weather in Ithaca is no better than that of Toronto, by the time he moved to Cornell, he learned how to stop worrying and love the snow. During his time at Cornell, he did his Ph.D. research in the exciting topic of three-dimensional wireless sensor networks. He received a Master of Science degree in Computer Science from Cornell University in January, 2008. He also completed his Ph.D. minor in Finance and spent a summer at Morgan Stanley working as a desk strategist in the securitized products group in New York City. Currently, he is working at a start-up company in Washington D.C. metro area.

Dedicated

To

My Family

ACKNOWLEDGMENTS

I feel pleased to have the opportunity of expressing my heart-felt and sincere gratitude to a number of people who have helped me in the course of this research work. First and foremost, I acknowledge my profound indebtedness to my supervisor Professor Zygmunt Haas. His constructive criticism, scholarly guidance, insightful advice and endless patience throughout the progress of the research work have made it possible to complete this research. I also thank Professor Johannes Gehrke and Professor Xiaoyan Zhang for their constructive suggestions. I am indebted to the members of the Wireless Network Laboratory with whom I have interacted during the course of my graduate studies. Special thanks to my family for giving me the encouragement, understanding, support, and love. Finally, to all who contributed in any way I express my appreciation and sincere feelings.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

A wireless sensor network (WSN) is a collection of spatially distributed sensor nodes that act in cooperation through wireless communication to monitor environmental or physical conditions. Application of WSNs is now virtually unlimited, occurring in a wide range of areas such as ecology, seismology, oceanography, industrial automation, structural monitoring, traffic control, health care, etc. According to a National Research Council report, the use of WSNs and other networked computer systems throughout society could "well dwarf previous milestones in the information revolution" [17].

1.1 MOTIVATION

Early work on WSNs focused primarily terrestrial applications. In general, these terrestrial WSNs are modeled as two-dimensional (2D) networks, where it is assumed that all nodes reside on a 2D plane. While this assumption is sufficient for most terrestrial form of these networks, recent innovations and technological breakthroughs have created many potential opportunities for use of WSNs in space, underwater, and in the atmosphere. In these locations, the 2D assumption is usually rendered invalid. The use of underwater sensor networks, in particular, has generated much recent research interest [1][44][35][31][12][36][16][18][22][29][39][40]. In some underwater applications, sensor nodes must be deployed at different water depths [1]. Consequently, these underwater networks can have significant height, meaning that

they must be modeled as three-dimensional (3D) networks where nodes are distributed over a 3D space. Many other potential applications, such as those used in atmospheric or space studies, share this need for 3D structure. For example, a recent article in Business 2.0 magazine [14] identified environmental sensor networks as one of eight technologies with potential for reducing the catastrophic consequences of global warming. The article described future atmospheric WSNs, where wireless sensors would float in the sky, collecting real-time data on climate change, hurricanes, air, etc. The sensor nodes in these types of atmospheric WSNs will clearly have to be distributed over a 3D space; therefore, these networks will have to be modeled as 3D networks.

Most networking issues are the same for both 2D and 3D networks. For example, the transmission control protocol (TCP) in either network type does not rely on the use of geographical coordinates of the nodes, so the same TCP can be used in both 2D and 3D networks. However, unlike 2D systems, 3D networks present some unique research challenges, especially in the area of topology control. Although coverage and connectivity issues have been thoroughly investigated in the context of terrestrial 2D sensor networks, many of these results cannot be directly applied to 3D networks. In fact, many widely used coverage analysis and placement strategies developed for 2D networks become NP-Hard in 3D [54]. As a result, prior research solutions used to solve these types of problems in terrestrial 2D WSNs cannot be fully utilized in the design and implementation of 3D networks. In this dissertation, we investigate some of the important research problems that arise specifically in relation to the coverage and connectivity issues unique to 3D networks. Our research results are expected to have a significant positive impact on the field of design and implementation of 3D networks.

One fundamental problem in the area of coverage and connectivity is the current lack of a strategy for sensor node placement that can achieve full coverage and full connectivity while using a minimal number of nodes. In this problem, a sensor node can monitor a point if and only if the distance between that point and the sensor node is less than or equal to its sensing range r_s. A network is considered to have full coverage, if every point inside that network is monitored by at least one sensor node. Two sensor nodes can communicate between themselves if and only if the distance between them is less than or equal to their communication range r_c.

For a given network, we can create virtual Voronoi cells using all sensor nodes. In this case, two sensor nodes are designated as geographically neighboring nodes if their Voronoi cells share a plane, a line, or a point. In this problem, all sensor nodes have the same sensing range and their communication ranges are also identical. Therefore, the objective is to deploy the minimum number of sensor nodes in such a way that any point inside the network is monitored by at least one sensor node (a.k.a. maintaining full coverage), while every sensor node can also communicate with any of its geographically neighboring nodes (a.k.a. full connectivity). This ability to communicate with any of the geographically neighboring nodes allows any sensor node to communicate with any other sensor node of the network, possibly over a multi-hop path.

Minimizing the number of sensor nodes in this type of system is a major goal, since doing so essentially minimizes the deployment cost of the network. Achieving full coverage and full connectivity in 3D networks with the minimum number of nodes has both civilian and military applications. For example, one can envision a scenario where the Air Force uses unmanned aerial vehicles with limited sensing range to form

a 3D network for surveillance of airspace. Similarly, the Navy can use a 3D network of underwater autonomous vehicles for ocean surveillance. In either case, it is always desirable to find the optimal placement of vehicles in three dimensions, in order to minimize the number of vehicles required, while still guaranteeing 100% coverage in 3D. Civilian applications, such as environment and climate monitoring in ocean and atmosphere using ad hoc and sensor networks, would also require this type of optimization.

This optimization problem is similar to the problem that arises when minimizing the number of base stations in a terrestrial cellular network. Dividing the 2D plane into hexagonal virtual cells and placing a base station (or a sensor node in the case of a WSN) at the center of each cell is a well-known method for solving this problem [37]. Clearly, however, this strategy is not applicable in a 3D context, as a cell in a 3D network must be a polyhedron whereas a hexagon is a polygon. This problem has largely remained unexplored for 3D networks. Since 3D networks are deployed in harsh environments, such as underwater, in space, or in the atmosphere, the sensor nodes in these types of networks are also likely to be more expensive than are sensor nodes used in terrestrial WSNs. Consequently, minimizing the number of sensor nodes is even more important for a 3D network, and solving this coverage and connectivity problem with minimum number of nodes becomes a particularly important research challenge.

Above mentioned problem is useful only when a sensor node can be deployed and maintained at any arbitrary location. We also investigate coverage and connectivity issues that arise in 3D networks where this assumption is not valid. Instead, we assume that we have no control over the movement of a node. As a result, the position

of a node can be random, which means that a large number of redundant nodes have to be deployed in order to ensure that every point of the network is within the sensing range of at least one sensor node. However, at any instant, all nodes are not usually needed for full sensing coverage. Thus, if a subset of nodes can sustain full sensing coverage, this should be sufficient. The challenge is to find a distributed and scalable scheme that dynamically selects a suitable subset of nodes to remain active based on their location and puts other nodes into sleep mode. Since energy consumption during sleep mode is insignificant, this approach has the added advantage of significantly prolonging network lifetime. Although many possible approaches exist for solving this complex problem, a fast, distributed, and scalable scheme that adjusts quickly with any change of the network (e.g., movement of nodes) has remained elusive. A trivial solution that depends on passing of many messages clearly cannot achieve the desired objective.

One promising approach is geographic adaptive fidelity (GAF), which has as yet only been applied in the context of 2D networks [25][51]. The idea underlying GAF is that the network space is divided into identical regions based on the sensing range and communication range of the sensor nodes. Among the sensor nodes located in each region, one sensor node is dynamically and locally selected as the active node. Its role is then to do the sensing for that region and to maintain connectivity with the active nodes of geographically neighboring regions. While this idea is simple, the difficult part lies in determining the best possible region that will minimize the number of regions (and thus minimize the number of active nodes at a given time). There are two constraints here. First, the diameter of the circumsphere of the region cannot be greater than the sensing range of any sensor node. Since we do not have any control on the position of the node, the selected active node can be located in one corner of the

region in the worst case, but this sensor node must still be able to sense all points of the region. Second, the maximum distance between the two furthest points of the neighboring regions cannot be greater than the communication range of any sensor node. This constraint ensures that active nodes of two neighboring region are able to communicate, irrespective of their positions inside each region. These two constraints ensure that full coverage and connectivity are maintained, even though active nodes are selected locally by sensor nodes inside each region.

Since 3D networks are still in their infancy, these problems have not yet received sufficient attention. Nevertheless, all of these problems are important considerations for designers of underwater, space and atmospheric 3D networks. This dissertation therefore focuses on these important problems.

1.2 RESULTS

Our analysis indicates that that the problem of achieving full coverage and full connectivity with minimum number of nodes in 3D networks can be solved by dividing the 3D space into identical truncated octahedron virtual cells. The radius of the circumsphere of each cell is made equal to the sensing range and a sensor is then placed at the center of each virtual cell. We define a metric called the *volumetric quotient* (*V.Q.*), which is the ratio of the volume of a polyhedron to the volume of its circumsphere. The higher the *V.Q.* of the shape of Voronoi cell, the smaller the number of sensor nodes is required. We show that the *V.Q.* of a truncated octahedron is 0.68329, much higher than other possible space-filling polyhedrons. For example, the *V.Q.* of rhombic dodecahedron is 0.477, while a hexagonal prism has *V.Q.* of 0.477 and a cube *V.Q.* is just 0.36755. These results imply that if the shape of the cell is

rhombic dodecahedron or hexagonal prism, then we need 43.25% more nodes than for the case where the cell shape is a truncated octahedron. When we compare energy consumption, we find that the model based on a truncated octahedron consumes more energy per node, because nodes are placed further apart to minimize the total number of nodes. However, when we take into account the number of nodes deployed and calculate the total energy consumption of the entire network, the truncated octahedron based model actually consumes less energy than the cube, hexagonal prism or rhombic dodecahedron based models. However, this solution using truncated octahedral cells will only satisfy the full connectivity constraint if the communication range is at least $4/\sqrt{5}$ times the sensing range. We investigated this problem further to find the best node placement strategy for any ratio of communication range and sensing range. If we denote the sensing range by r_s and the communication range by r_c, then the following node placement strategy will solve the full coverage and full connectivity problem for any ratio of communication range and sensing range:

- **Case I:** $r_c/r_s \geq 1.587401$

  Tessellate the 3D space into truncated octahedron cells such that the radius of each cell is $R = \min\left(r_c\sqrt{5}/4, r_s\right)$, and then place a node at the center of each virtual cell.

- **Case II:** $1.587401 > r_c/r_s \geq 1.211414$

  Create a hexagonal prism tessellation of a 3D space where each side of the hexagon is $a = \min\left(r_c/\sqrt{3}, r_s\sqrt{2}/\sqrt{3}\right)$ and the height of each hexagonal prism is $h = \min\left(2\sqrt{r_s^2 - a^2}, r_c\right)$, and then place a node at the center of each virtual cell.

- **Case III:** $r_c/r_s < 1.211414$

  Create a regular cube tessellation of a 3D space where the radius of each cell is $R = \min\left(r_c\sqrt{3}/2, r_s\right)$, and place a sensor at the center of each virtual cell.

After solving the node placement problem for full coverage and full connectivity in a 3D network for all values of r_c/r_s, we investigated the scenario where full connectivity is not desirable. When $r_c/r_s \geq 4/\sqrt{5}$, full coverage automatically ensures full connectivity with all neighboring nodes and the overhead of full connectivity is zero. However, for smaller values of r_c/r_s, a significant number of extra nodes have to be deployed to ensure full connectivity, even after full coverage is already achieved. In this type of network, fewer nodes are needed if the requirement of full connectivity with all neighbors is relaxed. However, the disadvantage of relaxing the full connectivity requirement is that communication among distant nodes, in general, takes a longer route. If nodes are failure prone, there is a chance that some nodes may be totally disconnected. Consequently, we need to make a tradeoff between faster communication and the required number of nodes, when $0 \leq r_c/r_s \leq 4/\sqrt{5}$. If nodes are robust and expensive, then the added cost to ensure full connectivity is not desirable. For these scenarios, we provide a strip based node placement strategy which provides full coverage and 1-connectivity[1] with fewer nodes when $r_c/r_s \leq 4/\sqrt{5}$. This approach provides full coverage and full connectivity with all neighboring nodes with minimum number of nodes when $r_c/r_s \geq 4/\sqrt{5}$. The placement strategy is as follows: Deploy nodes as strips, such that the distance between any two neighboring nodes in a strip is $\alpha = \min\left\{r_c, 4r_s/\sqrt{5}\right\}$ and the distance between two parallel strips in a plane is $\beta = 2\sqrt{r_s^2 - (\alpha/4)^2}$. Set the distance between two planes of strips as $\beta/2 = \sqrt{r_s^2 - (\alpha/4)^2}$ and deploy the strips such that a strip of one plane is placed between two strips of a neighboring plane. Unless $\beta \leq r_c$ or $\sqrt{\beta^2/2 + \alpha^2/4} \leq r_c$, this strip-based approach only ensures connectivity among nodes in the same strip. In

[1] We say that a network has *k-connectivity* if every node can communicate with every other nodes of the network along at least *k* different paths.

order to ensure connectivity between strips, additional nodes are deployed between strips.

For the problem of achieving full coverage and connectivity in 3D networks with random node deployment, we provide a highly distributed and scalable scheme that dynamically determines the active node locally. We extend the geographic adaptive fidelity (GAF) approach used in 2D wireless sensor networks to 3D networks. In this case, we randomly and uniformly deploy a large number of nodes in the network space. We exploit redundancy to improve network lifetime by partitioning the 3D network space into identical cells, such that only one node remains active in each cell, while full coverage and connectivity are maintained. We found that if the shape of each cell is a truncated octahedron, then the number of nodes needed to remain active is minimized. In this way, we can maintain a maximum number of nodes in sleep mode and thus increase network lifetime. We provide a very simple mechanism that allows instant identification of cell id by each sensor node, using a simple equation based on sensor node position. This requires only a constant number of arithmetic operations to compute the cell id for each node; hence, computation becomes very efficient. Once cell id is known, choice of the active node can be made locally using any standard leader selection algorithm. Note that the manner by which the virtual cells are created automatically ensures that all sensors inside a virtual cell can directly communicate with each other. If one node of a virtual cell broadcasts any message, this is heard by all other nodes in that virtual cell.

Our scheme is highly distributed and scalable and active nodes are dynamically selected locally in each cell without any message passing between nodes in different cells. However, on occasion, more nodes will be maintained in an active state than

would be with a centralized scheme that has global knowledge about the position of all nodes. We have therefore studied the efficiency of our scheme with that of a centralized scheme that can deploy nodes at any arbitrary location. Since the centralized scheme can control the position of nodes, it requires even fewer active nodes than does the optimal scheme. In order to highlight this distinction, we call this scheme *SuperOpt*. We compare our scheme with *SuperOpt* for k-coverage where a point is monitored by k sensor nodes rather than just one sensor node. We found that the gap between our scheme and *SuperOpt* decreases significantly when k is greater than 1. While the ratio of number of active nodes between the distributed scheme and *SuperOpt* decreases both in 2D and 3D, only in 3D can the k-coverage be maintained with high probability. According to our analysis, 2D-GAF requires four times the number of nodes needed by *SuperOpt*. On the other hand, the requirement of number of nodes in 3D-GAF is eight times of that of *SuperOpt*. However, the performance of the proposed scheme improves significantly in 3D for larger values of k. For example, our scheme can provide 4-coverage with a probability of 0.9971 with twice the number of nodes needed by the optimal scheme. Clearly, we get a substantial savings (the number of nodes decreases from 8 times to 2 times) when we have $k = 4$ instead of $k = 1$, with negligible loss in probability of coverage (<=0.0029).

1.3 OUTLINE

The rest of the dissertation is organized as follows. Chapter 2 formally describes the problems investigated in this dissertation. Background and related work are also presented in chapter 2. Coverage and connectivity issues imposed by using the minimum number of nodes for 3D networks are investigated in chapter 3. This includes full coverage and full connectivity as well as full coverage and 1-connectivity

using a minimum number of sensor nodes. Chapter 4 analyzes coverage and connectivity problem in 3D networks where nodes are randomly deployed. Performance comparison of the proposed scheme with a super optimal scheme is also presented. The dissertation is concluded in chapter 5.

CHAPTER 2

PROBLEM DESCRIPTION, BACKGROUND, AND RELATED WORK

In this chapter, first we formally describe our problem. Details of our assumptions and goals for each problem are also presented. We then provide necessary background, definition of important terms and work related to our problems. Actual analyses of our problems are presented in chapter 3 and chapter 4.

2.1 PROBLEM DESCRIPTION

The key problems investigated in this dissertation have been created by recent technological breakthroughs that allow building of 3D networks. Although extensive research has been done on networking problems in the context of 2D terrestrial networks, these remain as yet unexplored for 3D networks. Detailed descriptions of these problems are provided below.

2.1.1 Coverage and Connectivity in 3D Networks with the Minimum Number of Nodes

One of the fundamental problems in any sensor network is to determine the node placement strategy that achieves the required level of coverage and connectivity while using the minimum number of nodes. Although this is a fundamental and very important problem, no serious investigation has yet been made for this problem in the context of 3D networks. With recent interest in 3D underwater sensor networks,

solving fundamental problems for 3D networks has now become an important research challenge. In order to formally describe the problem, first we need to define a few terms.

Definition 2.1: *A sensor network has full coverage if every point inside the network is within the sensing range of at least one sensor node.*

Many detection and tracking applications of sensor networks require full coverage so that no event goes undetected.

Definition 2.2: *If we create virtual Voronoi cells using locations of sensor nodes in a 3D network, then we consider two sensor nodes to be geographically neighboring sensor nodes when their corresponding Voronoi cells share a common plane, a line, or a point. In other words, if Voronoi cells of two sensor nodes touch each other, then those two sensor nodes are designated as geographically neighboring nodes.*

The definition of a Voronoi cell is provided later in this chapter, where we discuss other issues related to solid geometry.

Definition 2.3: *A sensor network has full connectivity if any two geographically neighboring sensor nodes can communicate with each other.*

Note that our definition of full connectivity is less restrictive than other possible definitions, such as the one where any sensor node of a network can directly communicate with any other sensor node of the network. However, there is no point in using such a restrictive definition of full connectivity, because communication over a multi-hop path is fundamental to the nature of sensor networks. Our definition of full

connectivity provides a reasonable redundancy in communication, while at the same time it allows us to build a robust sensor network that can achieve full coverage with the minimum number of nodes.

After defining full coverage and full connectivity, we can describe our problem using the following two questions:

- What is the best way to place the sensor nodes in three-dimension, such that the number of sensor nodes required for surveillance of a 3D space is minimized while at the same time ensuring full coverage?

- What is the minimum ratio of the communication range and the sensing range of this type of a placement strategy that achieves full connectivity?

In order to make this problem more concrete, we have to make a number of assumptions, as follows:

- *Sphere-based sensing:* We assume a sphere based sensing model such that each sensor has a sensing range of r_s. A sensor can reliably detect any object that is located within a distance of r_s from that sensor.

- *Sphere-based communication:* We assume a spherical communication model whereby each sensor has a communication range of r_c. If the distance between two sensors is less than or equal to r_c, then they can communicate reliably with each other.

- *Homogeneous sensing and communication range:* We assume that all sensors have the same sensing range. The communication range of all sensors is also assumed to be identical.

- *No boundary effect:* We assume that the network is very large and that there is no boundary effect. Therefore, the number of nodes required for a placement strategy is inversely proportional to the volume of a Voronoi cell of sensor nodes.

- *Adjustable node position:* We assume that any node can be deployed in any position (or moved to that position), as required by the positioning algorithm. One major criticism of this assumption is that GPS does not work underwater. While there is research in underwater localization [12], we do not as yet have any robust positioning mechanism for underwater networks. However, this assumption ensures that our solution provides the lower bound of the number of nodes needed to achieve full coverage and connectivity. Knowing the lower bound of the number of nodes is always useful for a designer of an underwater sensor network. Adjustment of the node placement is also possible to accommodate small positioning errors. Finally, after solving this problem for an ideal scenario, we drop this assumption and we investigate coverage and connectivity problems in 3D networks where nodes are randomly deployed.

After formally describing our assumptions, our original two questions can be paraphrased into the following two goals:

- Given any sensing range r_s, find the number of nodes and their locations, such that the number of nodes required to achieve full coverage in a 3D network is minimized.

- For placement strategies that achieve the above goal, find the minimum communication range r_c in terms of sensing range r_s, so that full connectivity is ensured.

In sensor networks, nodes are usually battery operated, which makes energy consumption a concern. This leads us to our third goal:

- Find the comparative energy efficiencies of different possible node deployment strategies.

Our analysis of this problem reveals that the overhead in terms of the number of nodes needed to achieve full connectivity with all geographically neighboring nodes is prohibitively high when the value of r_c/r_s is small. In this type of network, full coverage can be achieved with a smaller number of nodes if the requirement of full connectivity with all geographically neighboring nodes is relaxed. However, communication among distant nodes, in general, takes a longer route. Thus, if nodes are prone to failure, there will be an increased chance that some nodes might become totally disconnected if we relax the full connectivity requirement. Therefore, we must impose a tradeoff between quick reliable communication and the number of nodes needed. Relaxing full connectivity makes sense when sensor nodes are expensive, when probability of node failure is low, and when the value of r_c/r_s is small. This leads us to the problem of finding a node deployment strategy that achieves full coverage and 1-connectivity in a 3D network with the minimum number of nodes.

Definition 2.4: *A network has k-connectivity if every node can communicate with every other node in the network along at least k different node disjoint paths.*

All assumptions made for full coverage and full connectivity problem, including sphere-based sensing, sphere-base communication, homogeneous sensing and communication range, no boundary effect, and adjustable node positions, also hold for this problem. While we can solve the problem of full coverage and full connectivity

for any value of r_c/r_s, it is less useful for smaller values of r_c/r_s. As we will see in chapter 3, maintaining full coverage automatically ensures full connectivity with all neighboring nodes, when $r_c/r_s \geq 4/\sqrt{5}$.

2.1.2 Coverage and Connectivity in 3D Networks with Random Node Deployment

Previous problem assumes that a sensor node can be deployed at any arbitrary location. In many environments, however, it is very difficult, if not impossible, to accomplish this. Therefore, for this problem, we drop the assumption of adjustable node position. Since node location can be random, we must deploy redundant nodes in order to achieve 100% sensing coverage. However, at any particular time, not all nodes are needed to achieve full sensing coverage. As a result, a subset of nodes can be dynamically chosen to remain active at any given time, to achieve sensing coverage based on their location at that time. One approach to achieve this goal in a distributed and scalable way is to partition the 3D network space into virtual regions or cells, and then to keep one node active in each cell. Extensive research has been done along this direction using 2D networks, but it has not yet been explored the context of 3D networks.

The assumptions of sphere-based sensing, sphere-base communication, homogeneous sensing and communication range, and no boundary effect made for the previous problem also hold for this problem. However, we now drop the assumption of adjustable node position and instead make the following assumption:

- *Random node position*: We make no assumption about the location at which any particular node is deployed. However, sensor node density must be high enough so that full coverage can be maintained.

For this problem, we need to define a couple of terms.

Definition 2.4: *A sensor node is designated as active sensor node when its sensing and communication mechanism is fully operational.*

When a node goes to sleeping mode, it turns off its sensing and communication mechanism to conserve energy. So a sensor node is active when it is not in sleeping mode.

Definition 2.5: *A network has k-coverage if every point inside the network is within the sensing range of at least k different active sensor nodes.*

The main goal is to find a distributed scalable scheme that dynamically determines the subset of nodes that remain active. As shown in chapter 4, we achieve this goal by achieving the following sub-goals.

- Given any fixed sensing range r_s, find the best partitioning scheme that maintains a minimum number of nodes active at a given time. In addition, determine the best portioning scheme, such that the lifetime of a cell (i.e., the time until the last node in a cell dies out) is maximized.

- Find a distributed and efficient algorithm for the determination of the cell to which a sensor node belongs.

- Find a solution for *k*-coverage problem, such that any point is within the sensing range of at least *k* nodes. Determine the efficiency of the scheme, compared to an optimal scheme where an oracle determines which nodes to keep active and where node position can be adjusted as needed.

Before we present our analysis of these two problems in chapter 3 and chapter 4, it is important to provide necessary background of the techniques that we use in our analysis.

2.2 BACKGROUND

The problems investigated in this dissertation require knowledge of solid geometry as well as familiarity with communication and networking. First, we provide some relevant background on solid geometry that is needed to understand our work. Later in this chapter, we describe communication and networking issues in related work.

2.2.1 Polyhedron

Any three-dimensional shape consisting of a finite number of polygonal faces[2] is called a ***polyhedron***. The faces of a polyhedron meet in straight line segments called edges, and the edges meet at points called vertices. Cubes, prisms, and pyramids are all examples of polyhedrons. A polyhedron surrounds a bounded volume in 3D. The two-dimensional analog of a polyhedron is called a ***polygon*** and the general term for any dimension is ***polytope***.

2.2.2 Space-Filling Polyhedron

A ***space-filling polyhedron*** is a polyhedron that can be used to fill a volume without any overlap or gap (a.k.a. tessellation or tiling). Since the sensing region of a node is spherical and spheres do not tessellate in 3D, we want to find a space-filling polyhedron that best approximates a sphere. In other words, we want to find a space-

[2] A *face* is part of a plane, i.e., all the points of a face lie on the same plane.

filling polyhedron whereby if each cell is modeled by that polyhedron, then the number of cells required to cover a volume is minimized. As well, the distance from the center of a cell to its farthest corner (i.e., the radius of a cell) will not be greater than the sensing range *R*.

Showing that a polyhedron has space-filling property is not easy. For example, although Aristotle claimed that the tetrahedron fills space [4], his claim was incorrect [21], and that mistake remained unnoticed until the 16th century [30].

Some of the important facts regarding space-filling polyhedrons are as follows: There are exactly five regular polyhedrons (a.k.a. ***platonic solids*** or ***regular solids***) [38]: the cube, dodecahedron, icosahedron, octahedron, and tetrahedron. This was proved by Euclid in the last proposition of *The Elements*[3]. Among these shapes, only the cube has the space-filling property [19]. There are only five convex polyhedrons with regular faces that have the space-filling property: the triangular prism, hexagonal prism, cube, truncated octahedron [38][47], and gyrobifastigium [23]. The rhombic dodecahedron, elongated dodecahedron, and squashed dodecahedron are also space-fillers. A combination of tetrahedrons and octahedrons fills space. In addition, octahedrons, truncated octahedrons, and cubes, combined in the ratio 1:1:3, can also fill space.

For our coverage and connectivity problem, we impose the restriction that the shape of the Voronoi cells should be identical, i.e., only one type of polyhedron is used to fill

---

[3] *The Elements* was written by Euclid about 2300 years ago. The earliest copy, located in Oxford, England, dates 888 AD. *The Elements* contains just theorems and their proofs, without examples, motivations, calculations, or introductions.

the space. The motivation for this requirement is twofold:

- Algorithms, especially distributed algorithms, to find the location of nodes are far simpler when only one type of polyhedron is used, and
- Since the radius of the polyhedron is fixed, it is unlikely that any significant improvement can be achieved by using two or more types of polyhedrons to fill the space.

2.2.3 Kelvin's Conjecture

Now, we describe the century-old Kelvin's conjecture that we use to justify why the truncated octahedron is the most likely building block for the optimal solution of our coverage and connectivity problem.

In 1887, Lord Kelvin asked the following question [41]: *"What is the optimal way to fill a three dimensional space with cells of equal volume, so that the surface area (interface area) is minimized?"* This is essentially the problem of finding a space-filling structure having the highest *isoperimetric quotient*. If the volume and surface area of a structure are V and S, respectively, then in three-dimensional space its isoperimetric quotient can be defined as $36\pi V^2/S^3$. A sphere has the highest isoperimetric quotient; this is 1. Kelvin's answer to his question was a 14-sided truncated octahedron having a very slight curvature of the hexagonal faces and its isoperimetric quotient is 0.757. However, Kelvin was unable to prove that the structure was optimal. The uncurved truncated octahedron has an isoperimetric quotient of 0.753367. For more than a century, Kelvin's solution was generally accepted as correct [48] and it has been widely known as *Kelvin's conjecture*. However, in 1994, two physicists, Denis Weaire and Robert Phelan, came up with

another space-filling structure. It consists of six 14-sided polyhedrons and two 12-sided polyhedrons with irregular faces of equal volume: this structure has 0.3% less surface area than the truncated octahedron [45][46]. The isoperimetric quotient of this structure is 0.764. Nevertheless, any proof that the structure of Weiare and Phelan is optimal or that Kelvin's solution is optimal for identical cells case has yet to be found.

2.2.4 Voronoi Tessellation

In three dimensions, for any (topologically) discrete set *S* of points in Euclidean space, the set of all points closer to a point *c* of *S* than to any other point of *S* is the interior of a convex polyhedron called the *Voronoi cell* of c. The set of these polyhedrons tessellate the whole space, and is called the *Voronoi tessellation* corresponding to the set *S*. The Voronoi tessellation of any solution to our problem of the optimal location of sensor nodes gives the optimal shape of each cell.

2.2.5 Kepler's Conjecture

Another closely related problem is Kepler's sphere packing problem. This problem involves finding the most efficient way to pack equal-sized spheres. In 1611, Kepler speculated that the face-centered cubic (FCC) lattice was the most efficient of all arrangements, but was unable to prove it. After four hundred years of failed efforts, Kepler's conjecture was finally proved to be correct by Thomas Hales in 1998 [20]. The proof extensively uses methods from the theory of global optimization, linear programming, and interval arithmetic. The computer code and data files used for the proof required more than 3 gigabytes of space for storage. The Voronoi tessellation of the FCC lattice is a rhombic dodecahedron. Although the FCC lattice is the optimal solution for sphere packing, we will show that the truncated octahedron, which is the

Voronoi tessellation of body-centered cubic (BCC) lattice, actually requires 43.25% fewer nodes in solving our problem. This significant difference is not very intuitive. Note that the FCC lattice has a packing density of 74.048% (the optimal solution for sphere packing), while the BCC lattice has packing density of about 68%.

2.3 RELATED WORK

The full coverage problem in two-dimensional networks has been investigated in the context of cellular networks [37]. In two-dimensional cellular systems, the cells are modeled as regular hexagons, such that the radius of each hexagon is equal to the maximum range of a base station. The problem of providing sensing coverage in two-dimensional sensor networks has received significant attention [13][10][34][55]. Maximizing the sensing coverage is a fundamental requirement for many critical applications of sensor networks; e.g., detection [43], monitoring, tracking, and classification [32]. The impact of sensing coverage on the performance of greedy geographic routing has been studied in [49] for 2D wireless sensor networks.

Few references exist in the literature regarding 3D networks; the work presented in [9] and [15] studied 3D cellular networks. In [9], each cell is represented as a rhombic dodecahedron, while in [15], each cell is represented as hexagonal prism. However, in this dissertation, we show that if a truncated octahedron is used to model the shape of a cell, then the required number of nodes to monitor a 3D space is 43.25% fewer than the case where the cell is represented as a hexagonal prism or a rhombic dodecahedron, when $r_c/r_s \geq 4/\sqrt{5}$. No prior work has dealt with the problem when $r_c/r_s < 4/\sqrt{5}$. Some of our work described in this dissertation has been published (see [2] and [3]).

We are not aware of any work on full coverage and 1-connectivity in 3D networks. However, this problem has been thoroughly investigated for 2D networks [5][27][50]. Under a disc-based model for communication and sensing range nodes are placed at the centers of the cells of a regular hexagonal tessellation of a 2D-plane such that each hexagon has radius equal to r_s (or, equivalently, at the vertices of a triangular lattice). This provides both full coverage and full connectivity with minimum number of sensors if $r_c/r_s \geq \sqrt{3}$. However, if $r_c/r_s < \sqrt{3}$, then full coverage does not automatically imply connectivity, and ensuring connectivity by adjusting the radius of the hexagons requires more nodes than the minimum number of nodes required for full coverage. Strip-based deployment achieves full coverage and 1-connectivity with minimum number of nodes in 2D when $r_c / r_s < \sqrt{3}$ [5].

The aim of our work with random node deployment is mainly to conserve energy by keeping a subset of sensor nodes active in a dense network, while putting the rest of the sensor nodes to sleep. This is a very common approach in terrestrial sensor networks [51] [11][55][53][7]. In an attempt to maximize the lifetime of a sensor network, energy conservation protocols dynamically maintain sensing coverage by keeping active only a subset of nodes at any particular time [42][50][52][53]. One important work in this context is geographic adaptive fidelity (GAF) [51], which is only applicable to 2D networks. In this dissertation, we solve this problem for 3D networks. GAF is not very efficient in terms of number of nodes being kept active. However, we show that the efficiency of our 3D solution is better than the efficiency of GAF in 2D for *k*-coverage when the value of *k* is large.

CHAPTER 3

COVERAGE AND CONNECTIVITY IN 3D NETWORKS WITH THE MINIMUM NUMBER OF NODES

In this chapter, we first investigate the problem of finding a node placement strategy that achieves full coverage and full connectivity in a 3D network, while minimizing the number of nodes required. We then investigate the similar problem where, instead of full connectivity, the goal is to achieve 1-connectivity.

3.1 FULL COVERAGE AND FULL CONNECTIVITY WHEN $r_c/r_s \geq 4/\sqrt{5}$

In this section, we analyze our problem from the point of view of the shape of Voronoi cells corresponding to the placement of nodes in the network. If each Voronoi cell is identical and the boundary effect is negligible, then the total number of nodes required for 3D coverage is simply the ratio of the volume of the 3D space to be covered to the volume of one Voronoi cell. Therefore, minimizing the number of nodes can be achieved if the Voronoi cells have the highest volume for the given sensing radius r_s. Clearly, the radius of the circumsphere of a Voronoi cell must be less than or equal to the sensing range r_s. Since achieving the highest volume is the goal, the radius of circumsphere must always be equal to the sensing range r_s. As well, since r_s is fixed, the volumes of the circumspheres of all Voronoi cells are the same and equal to $4\pi r_s^3/3$. Finally, the shape of any Voronoi cell in 3D is always a polyhedron. The restriction of identical Voronoi cells implies that the shape of all cells will be the same

polyhedron. Clearly, the polyhedron must have the space-filling property, so our problem reduces to the problem of finding the space-filling polyhedron that has the highest ratio of its volume to the volume of its circumsphere. We call this ratio the *volumetric quotient* of the space-filling polyhedron. A more formal definition is as follows:

Definition 3.1: *For any polyhedron, if the maximum distance from its center to any vertex is r_s and the volume of that polyhedron is V, then the volumetric quotient (V.Q.) of that polyhedron is defined as* $3V/4\pi r_s^3$**.**

Since the volume of the circumsphere is the upper bound on the volume of any polyhedron, the value of *V.Q.* will always be between 0 and 1. Clearly, for a given sensing range r_s, the number of nodes required to cover a 3D space is inversely proportional to the *V.Q.* of the space-filling polyhedron used as a Voronoi cell. Thus, our problem reduces to the problem of finding the space-filling polyhedron that has the highest *V.Q.*

One possible approach is to check all possible space-filling polyhedrons and to determine which space-filling polyhedron has the highest *V.Q.* However, a rigorous proof that considers all possible space-filling polyhedrons is intractable, as is evident from the fact that Kelvin's problem for the case of a single cell shape is still open after more than a century. Therefore, at first we instead provide some intuition as to why the truncated octahedron is the most likely solution, by drawing a similarity between our problem and Kelvin's conjecture. We then choose three other different space-filling polyhedrons that have been used by other researchers in similar problems and that are reasonable contenders of the truncated octahedron. We provide a detailed

comparison among these four space-filling polyhedrons and we show that the truncated octahedron has a much higher *V.Q.* than do the others.

3.1.1 Similarity with Kelvin's Conjecture

Kelvin's problem simplifies to essentially finding a space-filling polyhedron that has the highest isoperimetric quotient. On the other hand, our problem is essentially finding a space-filling polyhedron that has the highest *V.Q.* Among all 3D shapes, the sphere has the highest isoperimetric quotient and the highest *V.Q.* and, in both cases, that value is 1. We conjecture that for any two space-filling polyhedrons, P_1 and P_2, if P_1 has a higher isoperimetric quotient than P_2, then P_1 also have a higher *V.Q.* than P_2. Clearly, this is true if we compare any 3D shape with a sphere. If this conjecture is true, then the solution to Kelvin's problem is essentially the solution to our problem. Since, until now, the truncated octahedron has been the best solution for Kelvin's problem for the case of identical cells, we conjecture that the truncated octahedron is also the most likely solution to our problem. Note that we will consider the uncurved version of the truncated octahedron, because it is mathematically more tractable than the curved version and because the difference between the curved version and the uncurved version is negligible. Since the argument given above is not sufficiently rigorous, we also choose other likely contenders to the truncated octahedron to increase the confidence in our solution, and we provide comparison of the truncated octahedron with these other space-filling polyhedrons.

3.1.2 Choice of Polyhedrons

Kepler's conjecture for sphere packing problem has been proven recently, after five centuries of efforts, with the FCC lattice being the solution to that problem [20]. The

Voronoi tessellation of the FCC lattice is a rhombic dodecahedron. One can try to solve our problem using Kepler's problem in the following way:

Find the maximal packing of spheres and then take the Voronoi tessellation corresponding to the centers of the spheres. Define the radius of spheres, such that the maximum distance from a center to any vertex of the corresponding Voronoi cell is the sensing range r_s. Phrasing our problem in terms of Kepler's problem suggests that we should choose the rhombic dodecahedron as one of the contenders to the truncated octahedron.

The solution to our problem in 2D is the hexagon. The polyhedron that has a hexagon as its cross section in all the three axes (i.e., *x, y,* and *z* axes) does not have a space-filling property. Two polyhedrons that have a space-filling property with at least one hexagonal cross section are the rhombic dodecahedron and the hexagonal prism. Therefore, we include the hexagonal prism in our comparison as well. In fact, two previous studies used the rhombic dodecahedron [9] and the hexagonal prism [15] as the shape of the cell in the context of a 3D cellular network. Finally, the most simplistic choice is the cube and it is also the only regular polyhedron that tessellates a 3D space. For notational convenience, we designate the tessellation and node placement based on the cube, hexagonal prism, rhombic dodecahedron and truncated octahedron as the *CB*, *HP*, *RD* and *TO* models, respectively. In the next subsection, we compare all four models and find that the truncated octahedron has, indeed, a higher *V.Q.* than the other choices. Hence, the *TO* model requires the fewest number of nodes to cover a particular 3D volume. (We note that [6] provides a proof that implies that *TO* model actually requires the fewest number of nodes. However, [6] is written in a different context and so does not consider the connectivity issue. As a

result, [6] is not directly relevant for solving the problem as a function of r_c/r_s.)

3.1.3 Volumetric Quotient and Number of Nodes Needed

Here, we calculate the *V.Q.*s of our chosen polyhedrons and also provide a comparison of the number of nodes required when each of the polyhedrons is used as a Voronoi cell.

3.1.3.1 Cube (CB)

If the length of each side of a cube is *a*, then the radius of its circumsphere is $\sqrt{3}a/2$. The *V.Q.* of a cube is:

$$\frac{a^3}{\frac{4}{3}\pi\left(\frac{\sqrt{3}}{2}a\right)^3} = \frac{2}{\sqrt{3}\pi} = 0.36755.$$

3.1.3.2 Hexagonal Prism (HP)

The *V.Q.* of a hexagonal prism depends on its height. First, we need to find out the optimal height of a hexagonal prism so that we have the largest possible *V.Q.* Suppose that the length of each side of the hexagon is *a* and the height of the hexagonal prism is *h*. Then the radius of the circumsphere of the hexagonal prism then is $\sqrt{a^2 + h^2/4}$, the volume of the hexagonal prism is $3\sqrt{3}a^2h/2$, and the *V.Q.* is:

$$\frac{\frac{3\sqrt{3}}{2}a^2h}{\frac{4}{3}\pi\left(\sqrt{a^2+\frac{h^2}{4}}\right)^3}.$$

If we set the first derivative of the *V.Q.* to zero, then we obtain:

$$\frac{\frac{3\sqrt{3}}{2}a^2}{\frac{4}{3}\pi\left(\sqrt{a^2+\frac{h^2}{4}}\right)^3}-\frac{3}{2}\frac{\frac{3\sqrt{3}}{2}a^2h\cdot\frac{2h}{4}}{\frac{4}{3}\pi\left(\sqrt{a^2+\frac{h^2}{4}}\right)^5}=0$$

i.e.,

$$a^2+\frac{h^2}{4}=\frac{3h^2}{4}$$

Thus, the optimum value of *h* is $a\sqrt{2}$ and the optimum *V.Q.* of hexagonal prism is:

$$\frac{\frac{3\sqrt{3}}{2}a^2a\sqrt{2}}{\frac{4}{3}\pi\left(\sqrt{a^2+\frac{a^2}{2}}\right)^3}=\frac{6}{4\pi}=0.477.$$

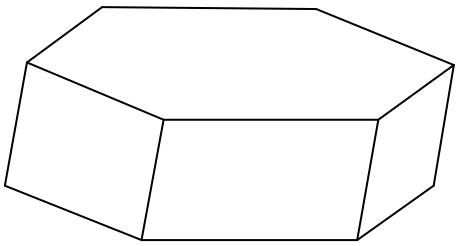


Figure 1: A hexagonal prism

3.1.3.3 Rhombic Dodecahedron (RD)

A rhombic dodecahedron can be constructed from two identical cubes, as shown in Figure 2:

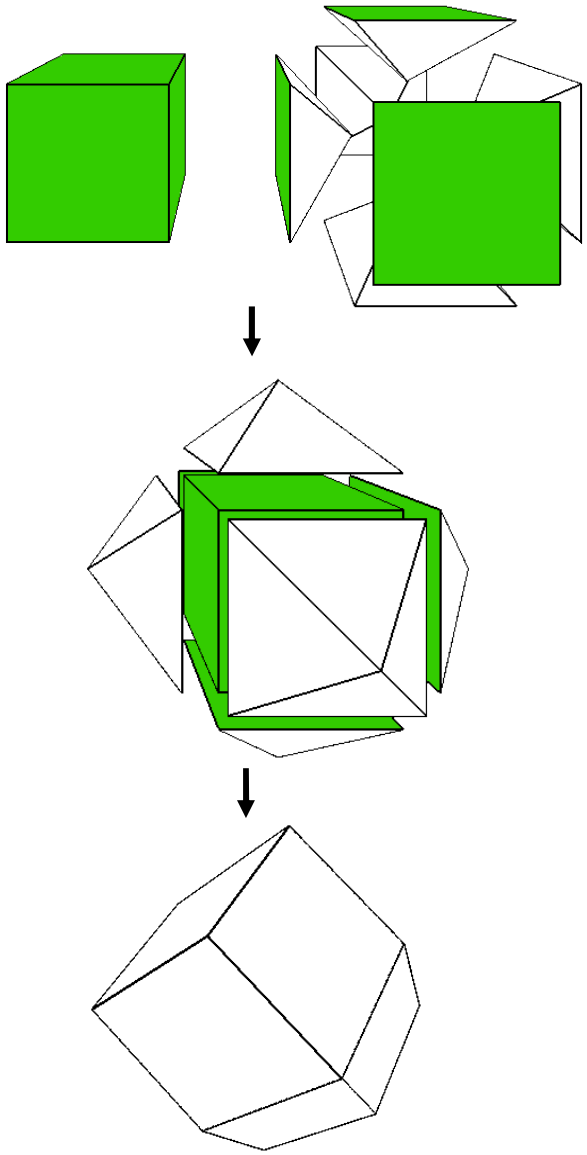


Figure 2: Construction of a rhombic dodecahedron from two identical cubes

Take one cube and cut it into six equal pyramids, such that the base of each pyramid consists of one face of the cube. Take another similar cube and place each pyramid on the cube, such that the base of the pyramid is on one side of the cube. This creates a rhombic dodecahedron. If each side of the two original cubes is a (i.e., the length of each edge of the rhombic dodecahedron is $\sqrt{3}a/2$), then the total volume of the rhombic dodecahedron is $2a^3$. The center of the rhombic dodecahedron is the center of the second (intact) cube. Eight vertices of the intact cube form eight vertices of the rhombic dodecahedron and their distance from the center is the radius of the circumsphere of the cube, equal to $\sqrt{3}a/2$. The other six vertices of the rhombic dodecahedron are formed by the six pieces of the first cube. The distance from the center of the second cube to its surface is $a/2$ and the height of each of the six pyramids is $a/2$. Therefore, the distance from the center of a rhombic dodecahedron to each of these six vertices is a, and this is also the radius of the circumsphere of the rhombic dodecahedron. Thus, the *V.Q.* of rhombic dodecahedron is:

$$\frac{2a^3}{\frac{4}{3}\pi a^3} = \frac{6}{4\pi} = 0.477 .$$

Note that, *V.Q.* of rhombic dodecahedron is exactly the same as that of a hexagonal prism.

3.1.3.4 Truncated Octahedron (TO)

The truncated octahedron has 14 faces, of which 8 are hexagonal and 6 are square faces: the length of the edges of hexagons and squares are the same. Suppose that the length of each edge is a. The distance between two opposite hexagonal faces is $\sqrt{6}a$ and the distance between two opposite square faces is $2\sqrt{2}a$. The radius of the

circumsphere of the truncated octahedron is $\sqrt{10}a/2$. The volume of the truncated octahedron is $8\sqrt{2}a^3$ and the *V.Q.* is:

$$\frac{8\sqrt{2}a^3}{\frac{4}{3}\pi\left(\frac{1}{2}\sqrt{10}a\right)^3} = \frac{24}{5\sqrt{5}\pi} = 0.68329.$$

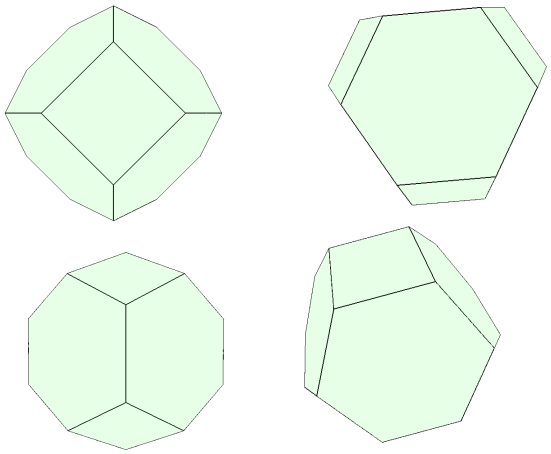


Figure 3: Truncated octahedrons

3.1.3.5 Comparison of Different Models

Among all the polyhedrons considered, the truncated octahedron has the largest *V.Q.* Since the *V.Q.* is inversely proportional to the number of nodes required to cover a 3D space, we can also compare the number of nodes required by each model. Table I shows these numbers.

Table I: Volumetric quotient and number of nodes needed by different models

Model	*Volumetric quotient*	*Number of nodes needed*
CB	0.36755	185.9%
HP	0.477	143.25%
RD	0.477	143.25%
TO	0.68329	100%

It is interesting to see how this result relates to 2D networks. The hexagon has the optimal tiling in 2D. The ratio of the area of a hexagon and the area of its circumcircle is $3\sqrt{3}/2\pi = 0.82699$. It is not difficult to see why the quotient in 3D is lower than that of 2D. In 1D tiling, we can achieve a quotient of 1 by using straight line tiling (actually, there is only one possible tiling in 1D). One can observe that the proportional loss in the value of the quotient remains roughly the same as we go to the higher dimensions. If we assume that the truncated octahedron is indeed the best shape for 3D tiling, then its quotient is 82.623% (0.68329/0.82699=0.82623) of the quotient achieved by hexagon in 2D, which is again 82.699% of the quotient of 1D tiling.

3.1.3.6 Explanation of Why the Truncated Octahedron is Better

Cross sections of the rhombic dodecahedron and the hexagonal prism are hexagons, but the vertices of this hexagon do not lie on the great circle of the circumsphere. As a result, the radius of the hexagon is smaller than the sensing range. In the case of the truncated octahedron, the two dimensional tiling is afforded by octagons that lie on the great circle. However, 2D tiling by regular octagon has square gaps (See Figure 4).

These square gaps are filled in 3D by cells from one level above and one level below. As a result, for a given sensing range and sufficiently high communication range (i.e., $r_c / r_s \geq 4/\sqrt{5}$), the *TO* model requires the smallest number of nodes to cover a given 3D space.

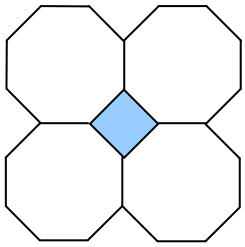


Figure 4: Octagons do not tile a plane.

3.1.4 Node Placement Strategies

In this subsection, we provide strategies (algorithms) to pinpoint the location where sensor nodes should be placed, using the Voronoi cells as our chosen space-filling polyhedrons. We choose an arbitrary point and place a node there as our center node. Then we find the locations of other nodes relative to this center node, so that the input to our algorithm is the sensing range r_s (so the radius of a cell is $R = r_s$) and the co-ordinates of a point, say (c*x, cy, cz*), which act as a seed for the growing lattice. This can be done in a distributed manner by first selecting a leader using any standard leader selection algorithm [33] and the location of the leader can then be used as the seed. Here, we assume that nodes already have information about the orientation of the *x*, *y*, *z*-axis from their localization component.

3.1.4.1 Cube (CB)

In each direction parallel to the *x*, *y*, and *z* axes, the distance between any two neighboring nodes is $2R/\sqrt{3}$. If a node is placed on each integer coordinate of the following coordinate system, then we obtain cube tessellation. Suppose that the coordinate system is defined by three axes: u, v, and w, which are parallel to the *x*, *y*, and *z* axes, respectively. The center of the coordinate system is (*cx*, *cy*, *cz*) and the unit distance in each axis is $2R/\sqrt{3}$. Therefore, a node at (u_1, v_1, w_1) in the new co-ordinate system should be placed in the original *x*,*y*,*z*-co-ordinate system at:

$$\left(cx + u_1 \times \frac{2R}{\sqrt{3}}, cy + v_1 \times \frac{2R}{\sqrt{3}}, cz + w_1 \times \frac{2R}{\sqrt{3}} \right) \quad (1)$$

In general, the real distance d_{12}^{cb} between any two points with coordinates (u_1, v_1, w_1) and (u_2, v_2, w_2) in the *u,v,w*- coordinate system is:

$$d_{12}^{cb} = \frac{2}{\sqrt{3}} R \sqrt{(u_2 - u_1)^2 + (v_2 - v_1)^2 + (w_2 - w_1)^2}\,.$$

In order to illustrate how this information can be used to place nodes efficiently, suppose that we want to cover a volume $\left(\frac{100\sqrt{3}}{2R} \times \frac{100\sqrt{3}}{2R} \times \frac{100\sqrt{3}}{2R} \right)$ with a center at (cx, cy, cz). The following pseudo-code segment could then be used:

For *u* = -50 to 50 do

For *v* = -50 to 50 do

For *w* = -50 to 50 do

put_node_at $\left(cx + u \times \frac{2R}{\sqrt{3}}, cy + v \times \frac{2R}{\sqrt{3}}, cz + w \times \frac{2R}{\sqrt{3}} \right)$

To save space, rather than providing pseudo-code for all polyhedrons, from this point

on we will only provide the necessary information about the node placement.

3.1.4.2 Hexagonal Prism (HP)

Suppose that the hexagons are parallel to the *xy* plane. The distance between neighboring nodes along the *z*-axis is then $2R/\sqrt{3}$. Note that the optimal height of the hexagonal prism is $\sqrt{2}$ times the radius of the hexagon. If a node is placed at every integer coordinate of the following coordinate system, then we have hexagonal prism tessellation. The axis *v* is parallel to the axis *y*. The angle between the *u* and the *v* axes is 60^0 in the positive half, and the unit distance along each axis is equal to $R\sqrt{2}$. Therefore, the angle between the axis *u* and the axis *x* is 30^0 and the angle between the axis *u* and the axis *y* is 60^0. The *w* axis is orthogonal to the *uv* plane and the unit distance along the *w* axis is $2R/\sqrt{3}$. Thus, the axis *w* is parallel to the axis *z*. Finally, the center of the *u,v,w*-coordinate system is at the (*cx, cy, cz*) point of the *x,y,z*-coordinate system.

A node at (u_1, v_1, w_1) in the new *u,v,w*-coordinate system should be placed in the original *x,y,z*-coordinate system at:

$$\left(cx + u_1 \times R\sqrt{2}\sin 60^0, cy + u_1 \times R\sqrt{2}\cos 60^0 + v_1 \times R\sqrt{2}, cz + w_1 \times \frac{2R}{\sqrt{3}} \right)$$

$$= \left(cx + u_1 R\sqrt{\frac{3}{2}}, cy + (u_1 + 2v_1)\frac{R}{\sqrt{2}}, cz + \frac{2Rw_1}{\sqrt{3}} \right) \quad (2)$$

The real distance between two points with coordinates (u_1, v_1, w_1) and (u_2, v_2, w_2) in the *uvw*-coordinate system is:

$$d_{12}^{hp} = R\sqrt{2}\sqrt{(u_2 - u_1)^2 + (u_2 - u_1)(v_2 - v_1) + (v_2 - v_1)^2 + \frac{2}{3}(w_2 - w_1)^2}\ .$$

3.1.4.3 Rhombic Dodecahedron (RD)

If a node is placed at every integer coordinate of the following *u,v,w* coordinate system, then we get rhombic dodecahedron tessellation. The axes *u* and *v* are parallel to the axes *x* and *y*, respectively. The angle between the *u* and the *w* axis is 60^0 in the positive half and the angle between the *v* and the *w* axes is also 60^0 in the positive half. The unit distance along each axis is $R\sqrt{2}$. The angle between the *w* axis and the *z* axis is 45^0. Finally, the center of the *u,v,w*-coordinate system is at the (*cx, cy, cz*) point of the *x,y,z*-coordinate system.

A node at (u_1, v_1, w_1) in the new *u,v,w*-coordinate system should be placed in the original *x,y,z*-coordinate system at:

$$\begin{pmatrix} cx + u_1 \times R\sqrt{2} + w_1 \times R\sqrt{2}\cos 60^0, \\ cy + v_1 \times R\sqrt{2} + w_1 \times R\sqrt{2}\cos 60^0, \\ cz + w_1 \times R\sqrt{2}\cos 45^0 \end{pmatrix}$$

$$= \begin{pmatrix} cx + (2u_1 + w_1)\dfrac{R}{\sqrt{2}}, \\ cy + (2v_1 + w_1)\dfrac{R}{\sqrt{2}}, \\ cz + w_1 R \end{pmatrix} \qquad (3)$$

The real distance between two points with coordinates (u_1, v_1, w_1) and (u_2, v_2, w_2) in the *u,v,w*-coordinate system is:

$$d_{12}^{rd} = R\sqrt{2}\sqrt{\begin{matrix}(u_2-u_1)^2 + (v_2-v_1)^2 + (w_2-w_1)^2 + \\ (u_2-u_1)(w_2-w_1) + (v_2-v_1)(w_2-w_1)\end{matrix}}.$$

Note that in the case of space-filling by identical rhombic dodecahedron, the distance

between the centers of any two neighboring rhombic dodecahedrons is the same. However, this is not the case for the hexagonal prism or for the truncated octahedron.

3.1.4.4 Truncated Octahedron (TO)

If a node is placed at every integer coordinate of the following *u,v,w*-coordinate system, then we get the truncated octahedron tessellation. The center of *u,v,w*-coordinate system is at the (*cx, cy, cz*) point of the *x,y,z*-coordinate system. The axes *u* and *v* are parallel to the axes *x* and *y*, respectively. The unit distance in both the *u* and *v* axes is $4R/\sqrt{5}$. The *w* axis is such that $\angle uv = \angle vw = \cos^{-1}\left(\sqrt{1/3}\right) = 54.73^0$ in the positive quadrant and the unit distance in *w* direction is $2\sqrt{3}R/\sqrt{5}$. The axis *w* creates an angle of $\cos^{-1}\left(\sqrt{1/3}\right) = 54.73^0$ with the *z* axis.

A node at (u_1, v_1, w_1) in the new *u,v,w*-coordinate system should be placed in the original *x,y,z*-coordinate system at:

$$\begin{pmatrix} cx + u_1\dfrac{4R}{\sqrt{5}} + w_1\dfrac{2\sqrt{3}R}{\sqrt{5}}\cos\alpha, \\ cy + v_1\dfrac{4R}{\sqrt{5}} + w_1\dfrac{2\sqrt{3}R}{\sqrt{5}}\cos\alpha, \\ cz + w_1\dfrac{2\sqrt{3}R}{\sqrt{5}}\cos\alpha \end{pmatrix}, \text{ where } \alpha = \cos^{-1}\left(\sqrt{\frac{1}{3}}\right).$$

After simplifying, we obtain:

$$\left(cx + (2u_1 + w_1)\frac{2R}{\sqrt{5}}, cy + (2v_1 + w_1)\frac{2R}{\sqrt{5}}, cz + w_1\frac{2R}{\sqrt{5}}\right) \qquad (4)$$

The real distance between two points with coordinates (u_1, v_1, w_1) and (u_2, v_2, w_2) in the *u,v,w*-coordinate system is:

$$d_{12}^{to} = \frac{4}{\sqrt{5}} R \sqrt{\begin{matrix}(u_2 - u_1)^2 + (v_2 - v_1)^2 + (u_2 - u_1)(w_2 - w_1) + \\ (v_2 - v_1)(w_2 - w_1) + \frac{3}{4}(w_2 - w_1)^2\end{matrix}} .$$

The location of any node can be found by using equations (1), (2), (3) and (4) for the cube, the hexagonal prism, the rhombic dodecahedron, and the truncated octahedron placement strategies, respectively.

3.1.5 Communication Range vs. Sensing Range

The minimum communication range r_c required to maintain connectivity among geographically neighboring nodes is different for different models. In each case, if we consider a new 3D co-ordinate system where integer coordinates (u,v,w) represent the location of each node, then the distance between two geographically neighboring nodes is essentially the unit distance along u, v and w-axes. In the *CB* model, the distance between any two neighboring nodes along the u, v and w-axes is $2r_s/\sqrt{3}$. Thus, we must have $r_c/r_s \geq 2/\sqrt{3} = 1.1547$. In the *HP* model, we must have $r_c/r_s \geq \sqrt{2} = 1.4142$ to maintain connectivity with the neighbors along the axes u and v, and $r_c/r_s \geq 2/\sqrt{3} = 1.1547$ along the w-axis. In the *RD* model, we need $r_c/r_s \geq \sqrt{2} = 1.4142$ along all three axes. Finally, in the *TO* model, we need $r_c/r_s \geq 4/\sqrt{5} = 1.7889$ along the axes u and v, and $r_c/r_s \geq 2\sqrt{3}/\sqrt{5} = 1.5492$ along the axis w. Since we need to communicate along all three axes in order to maintain connectivity, the minimum communication range must be the maximum of the minimum values along all three axes. These results are summarized in Table II.

Table II: Minimum communication range r_c for different models

Model	*Minimum* r_c			*Max of Min* r_c
	u-*axis*	v-*axis*	w-*axis*	
CB	1.1547 r_s	1.1547 r_s	1.1547 r_s	1.1547 r_s
HP	1.4142 r_s	1.4142 r_s	1.1547 r_s	1.4142 r_s
RD	1.4142 r_s	1.4142 r_s	1.4142 r_s	1.4142 r_s
TO	1.7889 r_s	1.7889 r_s	1.5492 r_s	1.7889 r_s

3.1.6 Illustrations

In this section, we provide a few illustrations of our model from a computer program that we wrote in C using OpenGL and employing the strategies described earlier in this chapter. The graphical outputs of our programs show that placing nodes according to equations (1), (2), (3), and (4) indeed covers the whole space, while the Voronoi cells have corresponding shapes. Our program outputs are animation videos from different viewing perspectives, which give the viewer a clear understanding of the placement strategies.

Figure 5 shows the placement of sensor nodes according to our proposed algorithm, based on the truncated octahedron model in a volume where the length in each dimension is 20m and the sensing range is R=5m. Each black dot represents a node. A truncated octahedron having radius 5m is drawn around each node in order to show that our placement strategy indeed provides 100% coverage. Axes *x, y,* and *z* are

represented by red, green and blue lines, respectively, in the figures to give the reader a perspective of the actual placement.

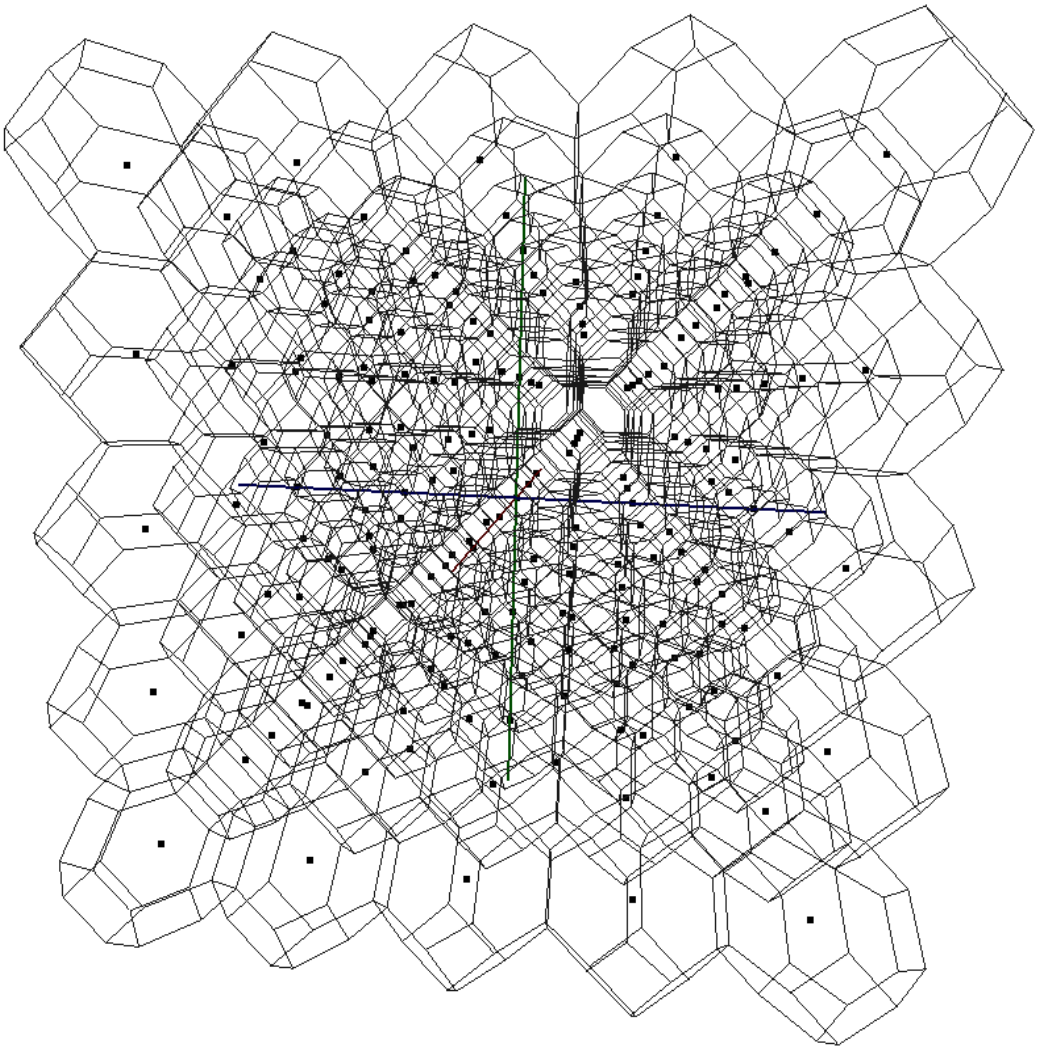


Figure 5: Truncated octahedron node placement strategy (cell surfaces are transparent).

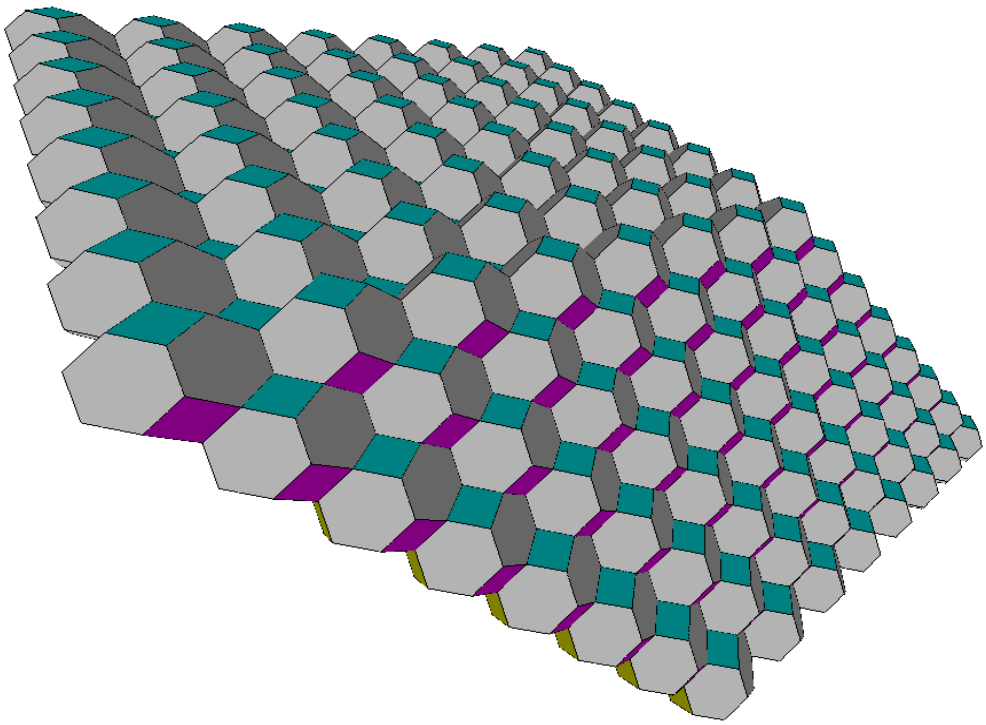


Figure 6: Truncated octahedron node placement strategy (cell surfaces are opaque).

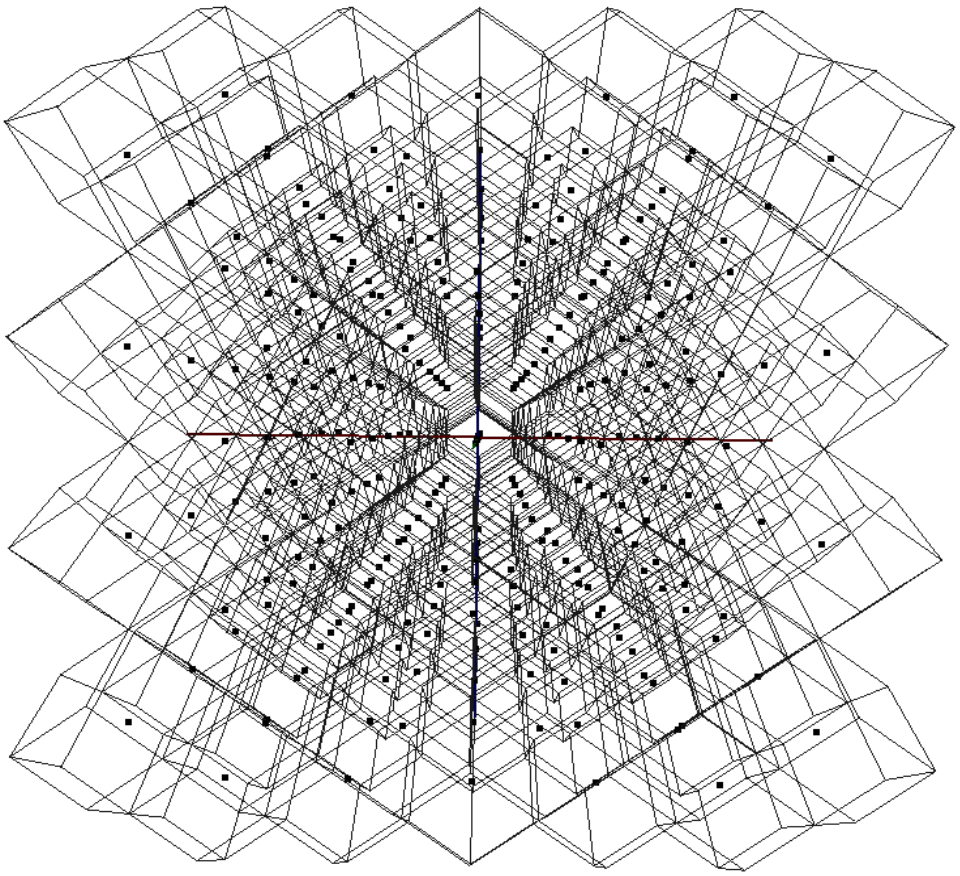


Figure 7: Rhombic dodecahedron node placement strategy.

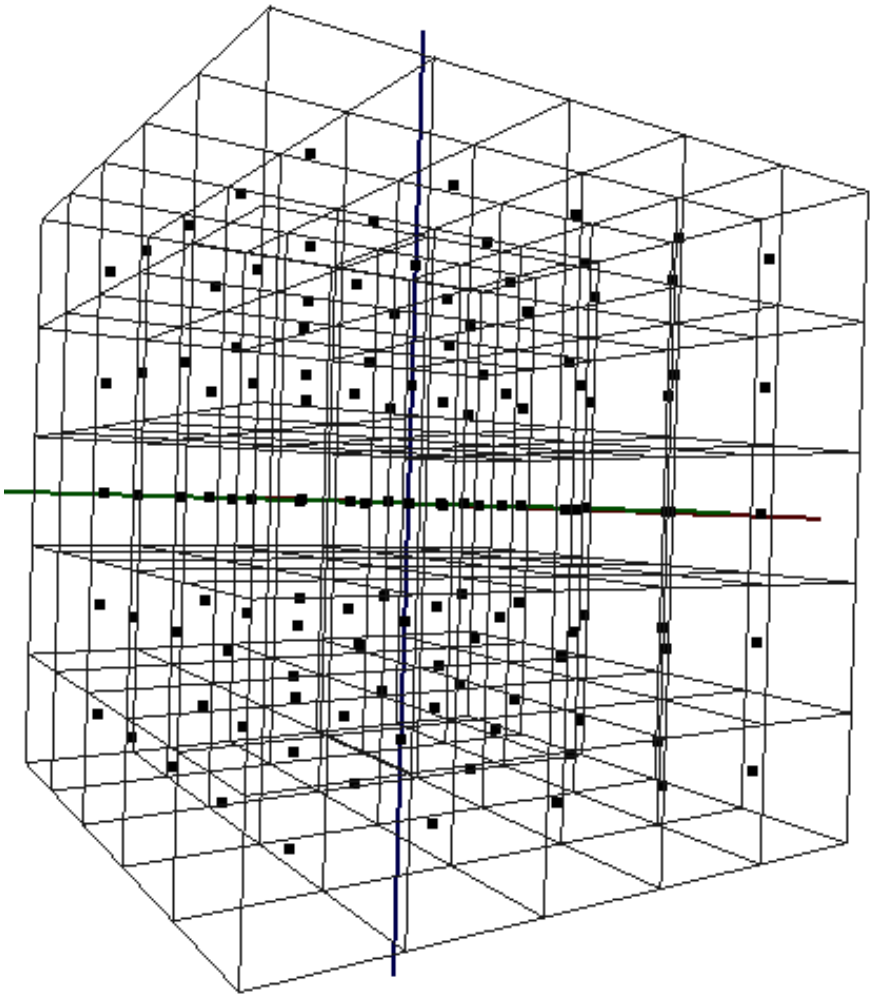


Figure 8: Node placement based on the cube model.

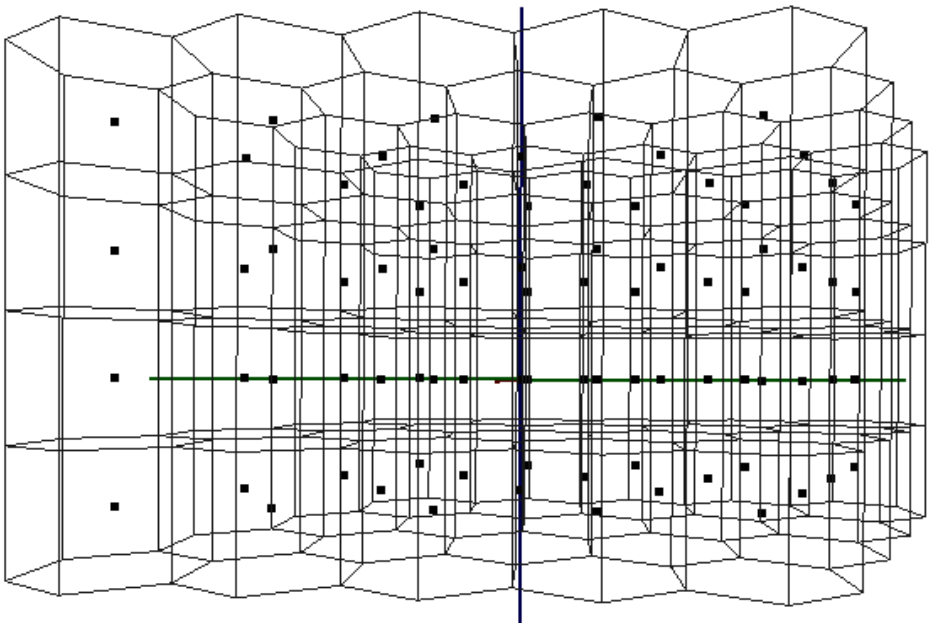


Figure 9: Node placement based on the hexagonal prism model.

Figure 6 shows the node placement strategy based on the truncated octahedron, where the cell surfaces are opaque. This network consists of 8x8x8 nodes. Figure 7 shows the placement of nodes according to the rhombic dodecahedron model. Placement of nodes according to the cube and the hexagonal prism models are shown in Figure 8 and Figure 9, respectively.

3.1.7 Comparison of Energy Efficiency of Various Models

Since energy consumption is critical for many sensor networks, we compare energy efficiency for all four models. We assume that $r_c / r_s \geq 4/\sqrt{5}$, so that we can maintain full connectivity using the minimum number of nodes in each model.

We assume that a node uses different signal strength in different models, such that transmission range is equal to the distance between two neighboring nodes in a particular model. We also assume that power consumption is primarily due to communication and that the differences in energy requirements in different models depend on the transmission range used by a node.

First, we compare the relative energy requirements to send a packet to the sink over a multi-hop path in each model. If the distance between two neighboring nodes is r_c, then for each packet generated at distance D from the sink, the total number of intermediate hops plus the source nodes (i.e., number of transmissions) is $\lceil D/r_c \rceil$. For simplicity of calculation, we use D/r_c instead, which is a reasonable approximation for large D and small r. For two models with transmission ranges of $r_{c'}$ and $r_{c''}$, the per packet power consumption ratio in each hop is $P^{h'} / P^{h''} = r_{c'}^2 / r_{c''}^2$. To send each packet to the sink, the power consumption ratio in the two models is $P' / P'' = r_{c'} / r_{c''}$. Table III

summarizes the power consumption ratio for each of the four models with respect to the truncated octahedron model.

Table III: Power consumption per packet

Model	*Power consumption ratio per packet*
Cube (*CB*)	0.64548
Hexagonal Prism (*HP*)	0.79054
Rhombic dodecahedron (*RD*)	0.79054
Truncated Octahedron (*TO*)	1.00000

If we assume that the total number of packets generated by each model is the same, then clearly the cube model has the smallest power consumption. This result is obvious, because it is a well known fact that the lower the transmission range, the lower the power consumption. However, this answer is misleading, given that we are not considering costs associated with the increase in the number of nodes used by cube model (85.9% more nodes than that of truncated octahedron model).

An alternative model can assume that each source node can aggregate information and send one packet, irrespective of the size of cell it covers, i.e., the number of packets generated by each model is proportional to the number of cells in that model. The ratio of power consumption of the entire network in each model is then essentially the power consumption ratio per packet times the ratio of the number of cells in each model. Power consumption of the entire network in each model with respect to the truncated octahedron model is shown in Table IV.

Table IV: Power consumption of the entire network

Model	*Power consumption ratio of entire network*
Cube (*CB*)	0.64548×1.859 = 1.1999
Hexagonal Prim (*HP*)	0.79054×1.4325=1.1325
Rhombic Dodecahedron (*RD*)	0.79054×1.4325=1.1325
Truncated Octahedron (*TO*)	1.0000

Power consumption per node is the largest in the truncated octahedron model, as expected, because this model deploys far fewer nodes than does any other model. Consequently, it must place nodes further apart, which leads to a higher transmission range. However, when we take into account the number of nodes deployed in each model by comparing power consumption of the entire network, the truncated octahedron model is the most energy efficient among the four models.

3.2 FULL COVERAGE AND CONNECTIVITY FOR ALL VALUES OF r_c/r_s

Based on the results of the previous section, the *TO* model clearly provides full connectivity only when $r_c/r_s \geq 4/\sqrt{5}$. None of the four models has full connectivity with all first-tier geographically neighboring nodes for all values of r_c/r_s. In order to ensure full coverage and full connectivity for all values of r_c/r_s, we modify our models as follows:

3.2.1 Modified TO Placement Strategy

In the original *TO* model, the maximum distance between any two neighboring nodes is $4R/\sqrt{5}$, where *R* is the radius of a cell. Therefore, in order to achieve connectivity along all three axes for all values of r_c/r_s, in the *modified TO* model, we set $R = \min\left(r_c\sqrt{5}/4, r_s\right)$.

3.2.2 Modified HP Placement Strategy

In the original *HP* model, the maximum distance between any two neighbors along the plane that has hexagonal tessellation is $a\sqrt{3}$, where a is the length of each side of a hexagonal face of a hexagonal prism. Now, if $r_c / r_s \geq \sqrt{2}$, we must have $a = r_s\sqrt{2}/\sqrt{3}$ in order to maintain connectivity. Thus, in the *modified HP* model, we set $a = \min\left(r_c/\sqrt{3}, r_s\sqrt{2}/\sqrt{3}\right)$ for the general case. In the original *HP* model, when $r_c / r_s \geq \sqrt{2}$, the distance between any two neighbors along the height axis is $h = 2\sqrt{r_s^2 - a^2}$. In order to ensure connectivity for all values of r_c/r_s, we must have $h = \min\left(r_c, 2\sqrt{r_s^2 - a^2}\right)$ in the *modified HP* model. We can achieve this *modified HP* placement in the following way:

Take any arbitrary point (*x, y, z*) as the center of the network and deploy one node in co-ordinate:

$$\left(x + u \times a\sqrt{3}\sin 60^0,\ y + u \times a\sqrt{3}\cos 60^0 + v \times a\sqrt{3}, z + w \times h\right)$$

where $u \in Z, v \in Z, w \in Z$ and *Z* is the set of all integers.

3.2.3 Modified RD Placement Strategy

In the original *RD* model, the distance between any two first-tier neighbors is always $R\sqrt{2}$, where *R* is the radius of a cell. Connectivity with all neighbors can be achieved for all values of r_c/r_s, if we set $R = \min\left(r_c/\sqrt{2}, r_s\right)$ in the *modified RD* model.

3.2.4 Modified CB Placement Strategy

In the original *CB* model, the distance between any two first-tier neighbors is always $2R/\sqrt{3}$, where *R* is the radius of a cell. Therefore, connectivity with all neighbors can be achieved for all values of r_c/r_s, if we set $R = \min\left(r_c\sqrt{3}/2, r_s\right)$ in the *modified CB* model.

The volume and placement strategy for *modified CB*, *modified RD* and *modified TO* are same as those of the *CB*, *RD*, *TO*, respectively, with the values of *R* being set as discussed above.

3.2.5 Comparison Among Different Models

Since all values are exact, we can just plug in the values of r_c/r_s into the equations and immediately generate a graph that nicely shows the performance of each placement strategy for various values of r_c/r_s (See Figure 10). In all cases, the volume of a cell (in terms of r_s^3) is a monotonically non-decreasing function of r_c/r_s. In particular, for the *modified TO* model, the volume of a cell is a monotonically increasing function of r_c/r_s, for all $r_c/r_s < 4/\sqrt{5} = 1.7889$. For $r_c/r_s \geq 4/\sqrt{5}$, it is constant and has the value $32r_s^3/5\sqrt{5} = 2.862r_s^3$. For both the *modified HP* model and

the *modified RD* model, the volume of a cell is a monotonically increasing function of r_c / r_s, for all $r_c / r_s < \sqrt{2} = 1.4142$. For $r_c / r_s \geq \sqrt{2}$, it is constant and has the value $2r_s^3$. However, as we see in Figure 10, the *modified HP* model has higher volume than does the *modified TO* model for $r_c / r_s < \sqrt{2}$, because unlike the *modified RD* model, the distance from all neighbors is not uniform in the *modified HP* model. Finally, for the *modified CB* model, the volume of a cell is a monotonically increasing function of r_c / r_s, for all $r_c / r_s < 2/\sqrt{3} = 1.1547$. For $r_c / r_s < 1.1547$, it is constant and has a value of $8r_s^3 / 3\sqrt{3} = 1.5396 r_s^3$.

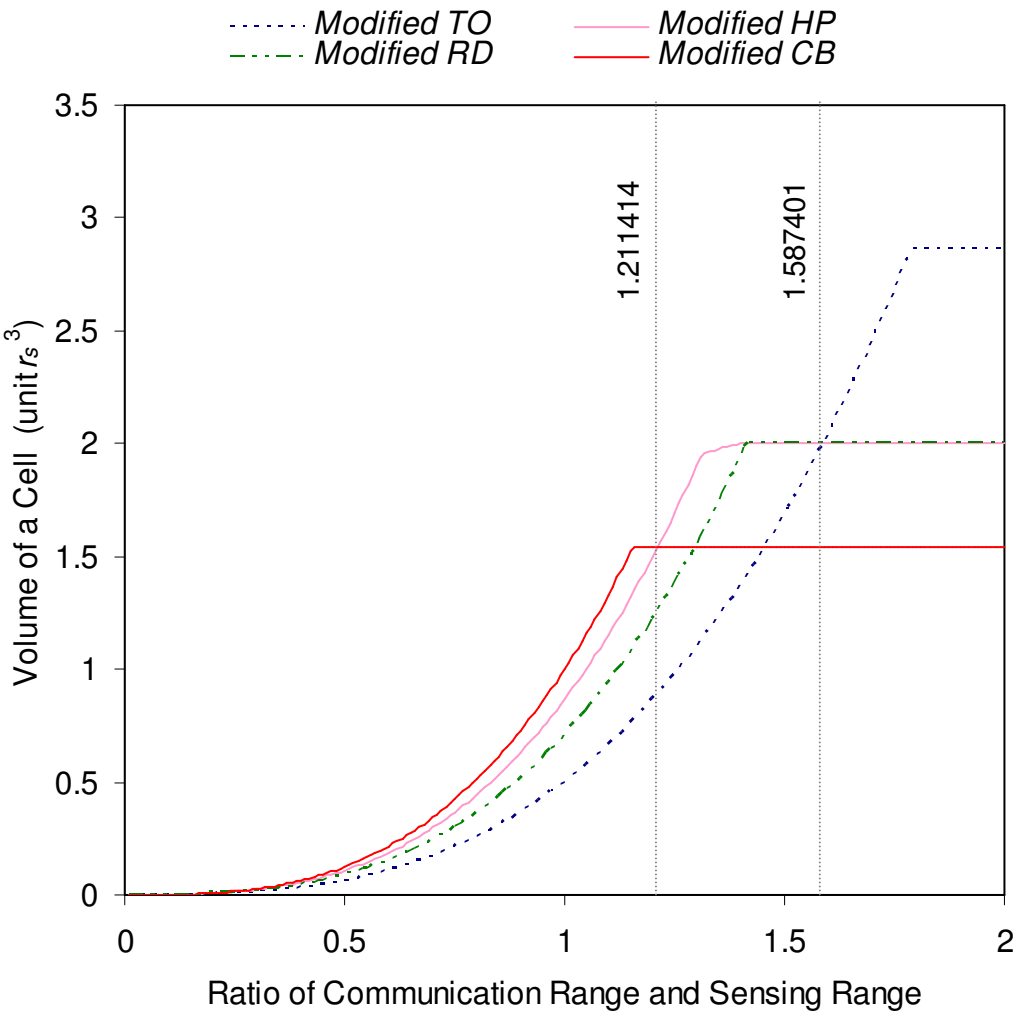


Figure 10: Performance of various placement strategies

Since the number of nodes required by any placement strategy is inversely proportional to the volume of a cell, we can conclude the following:

- If $r_c/r_s \geq 1.587401$ then the *modified TO* model is the best strategy,
- If $1.587401 > r_c/r_s \geq 1.211414$ then the *modified HP* model is the best strategy,
- If $r_c/r_s < 1.211414$ then the *modified CB* model (3D grid based node placement) is the best strategy.

3.3 FULL COVERAGE AND 1-CONNECTIVITY

In some scenarios, it is not feasible to maintain full connectivity with all geographically neighboring nodes. The overhead in terms of the number of nodes needed to achieve full connectivity with all geographically neighboring nodes is prohibitively high when the value of r_c/r_s is small. In this type of network, full coverage can be achieved with a smaller number of nodes if the requirement of full connectivity with all geographically neighboring nodes is relaxed. However, communication among distant nodes in general takes a longer route and, if nodes are failure prone, there is a chance that some nodes may be totally disconnected if we relax the full connectivity requirement. Here we face a tradeoff between quick reliable communication and the number of nodes needed. Relaxing full connectivity with all geographically neighboring nodes makes sense when the nodes are expensive and robust, with very low probability of failure, and the value of r_c/r_s is small. Consequently, finding a solution that provides full coverage and 1-connectivity with the minimum number of nodes becomes an important problem for this type of network.

Inspired by the work published in [5] for the 2D network, we provide a strip based node deployment strategy that achieves this goal for the 3D network. Since the 2D case is simpler than the 3D scenario, understanding how strip based node deployment works in 2D is important, before we describe our solution for 3D. If we assume disc based sensing range and communication range model, then creating a regular hexagonal tessellation of a 2D-plane with a cell radius equal to r_s and then placing a node at the center of each cell provides both full coverage and full connectivity with the minimum number of sensors if $r_c / r_s \geq \sqrt{3}$. Note that $\sqrt{3}$ is the critical number in 2D, while $4/\sqrt{5}$ is the critical number in 3D. A strip based deployment (see Figure 11 (a)) achieves full coverage and 1-connectivity with the minimum number of nodes in 2D, when $r_c / r_s < \sqrt{3}$ [5]. The distance between any two neighboring nodes on a strip is set to $\alpha = \min\{r_c, \sqrt{3} r_s\}$. The value of α comes from the fact that, in an optimal hexagonal based node placement, the distance between two geographically neighboring nodes is $\sqrt{3} r_s$. Figure 11(b) shows how to determine the value of β, the distance between two strips. Clearly:

$$\beta = r_s + d = r_s + \sqrt{r_s^2 - \alpha^2 / 4}\ .$$

In this deployment pattern, connectivity is ensured only among the neighboring nodes in a strip. If $r_c / r_s < \sqrt{3}$, then nodes of two horizontal strips cannot communicate and a vertical strip is deployed to achieve 1-connectivity in the whole network. This can easily be extended to 2-connectivity by deploying two vertical strips: one in the left boundary of the network and the other in the right boundary of the network. However, 3-connectivity cannot be achieved because, in general, a node in a horizontal strip can only communicate with two nodes.

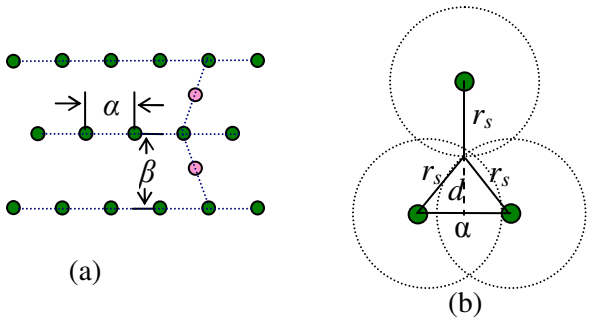


Figure 11: (a) Strip based deployment in 2D (b) Calculation of *β*

The difference between this strip based approach and the optimal hexagon-based tessellation approach is that the distance along one axis is adjusted to maintain connectivity when $r_c / r_s < \sqrt{3}$. When $r_c / r_s \geq \sqrt{3}$, the strip based node placement degenerates into optimal hexagonal tessellation based approach and no auxiliary node is needed. We extend this approach for 3D network in next subsection.

3.3.1 Strip Based Node Deployment in 3D

In this subsection, we provide a strip based placement strategy for 3D network that provides full coverage and 1-connectivity when $r_c / r_s < 4/\sqrt{5}$. This approach automatically provides full coverage and full connectivity with all neighboring nodes with minimum number of nodes when communication range is sufficiently high (i.e., $r_c / r_s \geq 4/\sqrt{5}$).

We can achieve full coverage and 1-connectivity with minimum number of nodes in the following strip-based node placement in 3D: Set the distance between any two nodes in a strip as $\alpha = \min\left\{r_c, 4r_s/\sqrt{5}\right\}$ and keep distance between two parallel strips in a plane as $\beta = 2\sqrt{r_s^2 - (\alpha/4)^2}$ (See Figure 12). Set the distance between two planes

of strips as $\beta/2 = \sqrt{r_s^2 - (\alpha/4)^2}$ and deploy strips such that a strip of one plane is placed between two strips of a neighboring plane. Here distance between two neighboring nodes that reside in two different planes is $\gamma = \sqrt{\beta^2/2 + \alpha^2/4}$. Figure 13 shows the horizontal and vertical projection of this type of deployment. For clarity, nodes from the same place are drawn in the same color.

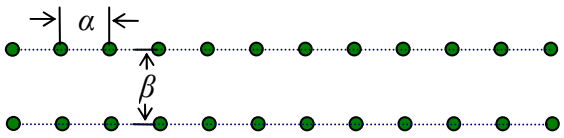
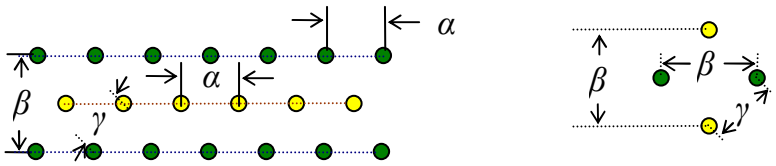


Figure 12: Nodes in a particular plane of a 3D strip based deployment



(a) Horizontal projection (b) Vertical projection

Figure 13: Horizontal and vertical projection of nodes in two different planes in a 3D strip based deployment

This deployment of sensors can be achieved by taking any arbitrary co-ordinate (*x*, *y*, *z*) as the center of the network and placing a node at each of the coordinates:

$$\begin{pmatrix} x + u\alpha + w\gamma\cos\theta, \\ y + v\beta + w\gamma\cos\theta, \\ z + w\gamma\cos\theta \end{pmatrix}$$

where $\theta = \cos^{-1}\left(\sqrt{\frac{1}{3}}\right)$ and $u \in Z, v \in Z, w \in Z$; Z is the set of all integers. A number of additional nodes have to be deployed to achieve connectivity among different strips, as described at the end of this section.

If $r_c/r_s \geq 4/\sqrt{5}$, we have $\alpha = \beta = 4r_s/\sqrt{5}$ and $\gamma = 2\sqrt{3}r_s/\sqrt{5}$, which is identical to that of the optimal *TO* model. If $r_c/r_s < 4/\sqrt{5}$, we have, $\alpha = r_c$, $\beta = 2\sqrt{r_s^2 - (r_c/4)^2}$ and $\gamma = \sqrt{2r_s^2 + r_c^2/8}$. We then have to place a node at each coordinate:

$$\begin{pmatrix} x + ur_c + w\sqrt{2r_s^2/\sqrt{3} + r_c^2/8\sqrt{3}}, \\ y + 2v\sqrt{r_s^2 - (r_c/4)^2} + w\sqrt{2r_s^2/\sqrt{3} + r_c^2/8\sqrt{3}}, \\ z + w\sqrt{2r_s^2/\sqrt{3} + r_c^2/8\sqrt{3}} \end{pmatrix}$$

where $u \in Z, v \in Z, w \in Z$; Z is the set of all integers.

The values of the parameters α, β and γ are determined using the following reasoning: We can ensure 1-connectivity by ensuring each node in a strip is connected to both of its neighbors in the strip. Since the maximum distance between any two neighbors in the original *TO* model is $4r_s/\sqrt{5}$, we can ensure connectivity as well as optimality (when $r_c \geq 4r_s/\sqrt{5}$) by setting $\alpha = \min\{r_c, 4r_s/\sqrt{5}\}$.

Once we ensure connectivity along the strip, we do not need to worry about connectivity when we determine the value of β. We rather want β to be as large as possible, as long as we have full coverage. In a 2D network, the calculation of the value of β is simpler, as all points between two strips must be within the sensing range of at least one node residing on the two strips. However, in a 3D network, this is no longer true and nodes residing on other planes (both above and below) provide coverage for some area between the two strips. Figure 14 shows a cross-section along

a plane. The area inside the dotted inner circle with diameter d_2 is covered by nodes placed in one level above and one level below. From Figure 14, we have $\alpha = d_1 + d_2 + d_3$, where $d_1 = d_3 = \sqrt{r_s^2 - \beta^2/4}$.

Next, we use a symmetry argument to find the value of d_2. If we rotate the network by 90^0, then the network looks identical to the network prior to rotation. However, nodes that were in one level above and one level below would now be in the same plane, so that the distance between them must be β (the two nodes are in the same plane but in different strips). Now, let us go back to our network before rotation. Since the distance between nodes in one level above and one level below is β, again, by symmetry, the height of the node in the above level is $\beta/2$.

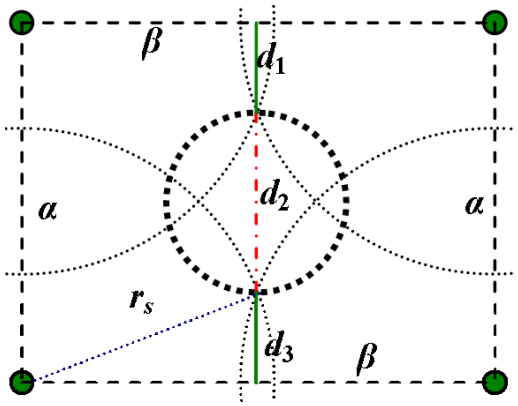


Figure 14: Cross-section along a plane of a 3D strip based deployment

From Figure 15, we have $d_2 = 2\sqrt{r_s^2 - \beta^2/4}$. Substituting the value of d_2, we have $\alpha = d_1 + d_2 + d_3 = 4\sqrt{r_s^2 - \beta^2/4}$, which implies $\beta = 2\sqrt{r_s^2 - (\alpha/4)^2}$.

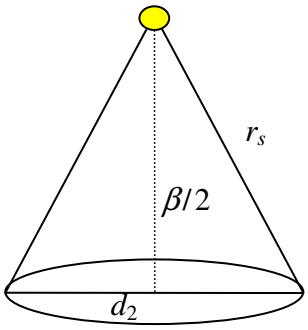


Figure 15: Calculation of d_2 for a 3D strip based deployment

We can calculate the distance between two neighboring nodes that reside in two different planes (e.g., the distance between any one of the four nodes in Figure 14 and the node in one level above that covers the inner circle) as follows: the distance from any of the four nodes to the center of the inner circle is $\sqrt{\alpha^2+\beta^2}/2$. We then have $\gamma=\sqrt{\beta^2/2+\alpha^2/4}$ (See Figure 16).

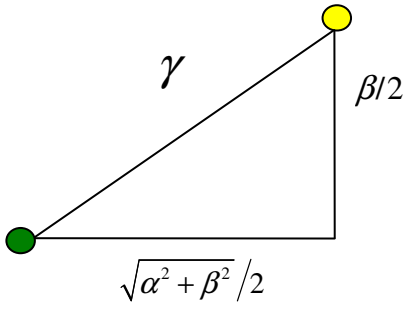


Figure 16: Calculation of γ for a 3D strip based deployment

This strip based approach only ensures connectivity among nodes in the same strip unless $\beta \le r_c$ or, $\gamma \le r_c$,. In order to ensure connectivity between strips, we need to place additional nodes between strips. We can achieve 1-connectivity by placing auxiliary nodes so that any two neighboring nodes in two strips are connected. However, 2-connectivity can only be achieved by placing auxiliary nodes at the two

endpoints of the strips along the boundary of the network. Unless $\beta \leq r_c$ or, $\gamma \leq r_c$, there is no way to achieve 3-or higher connectivity without deploying a large number of auxiliary nodes.

3.4 DISCUSSION

Our node placement strategies for both full connectivity and 1-connectivity are based on some ideal assumptions. This work, based on these ideal assumptions, is important because the minimum number of nodes required under these assumptions can be treated as the lower bound of the number of nodes required by any 3D sensor network to achieve similar level of coverage and connectivity. A sensor network designer can start with the node placement strategy provided here and then adjust it to adapt to a real world situation, where some of those assumptions are violated. At a high level, some of these adjustments can be as follows:

- Our assumptions of sphere-based sensing and communication (disc based in 2D) and the homogenous sensing and communication range of each sensor are standard assumptions in most network-modeling works. To adjust to a real world situation, the network designer can conservatively estimate the sensing range and communication range (i.e., set sensing range and communication range at some fractional level of the actual sensing and communication range) so that above assumptions remain true.

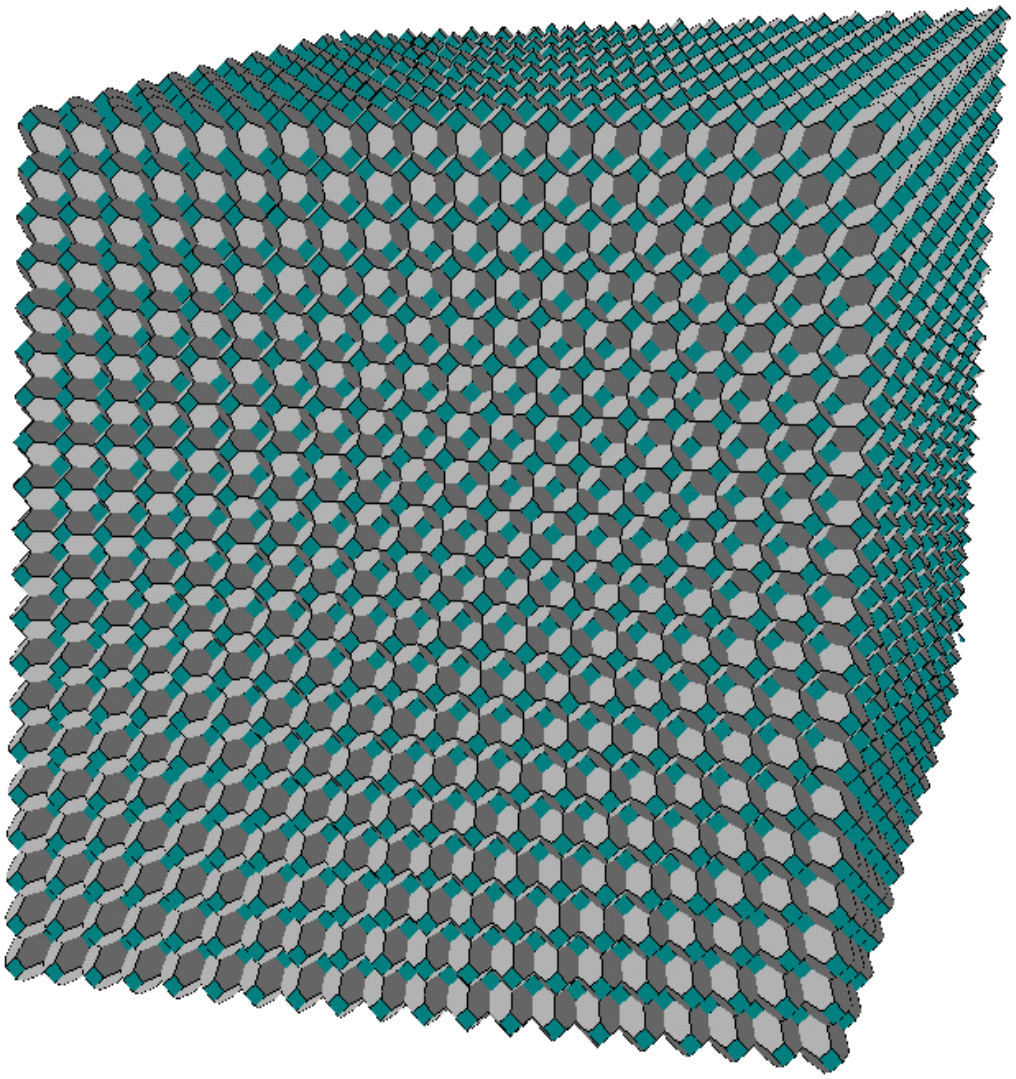


Figure 17: Coverage of a cube shaped 3D space in the TO model

- Our assumption of no boundary effect cannot be valid in practice, as all real networks will be finite in size. However, if the height, width, and length of the network are significantly larger than the sensing range of each node, then 3D space of any shape can be covered with any type of virtual cells, with small overhead near the boundary. The smaller the sensing range, the smaller the boundary effect; the boundary effect vanishes when the sensing range become infinitesimally small.

- Figure **17** shows how a 3D cube shaped space is covered by a network consisting of 20x20x20 nodes placed under the *TO* model. This illustration is made in C using OpenGL following the *TO* model node deployment strategy.

- The adjustable node position assumption is unlikely to be valid in a real underwater sensor network. Precise positioning underwater is extremely difficult, as GPS does not work underwater. However, our work does not require an absolute positioning mechanism: rather, any relative positioning mechanism where a node knows its position relative to the seed node is sufficient. Again, in many sensor network applications (e.g., detection, monitoring), knowledge of the origin of any piece of information important. Any underwater sensor network that is deployed for these types of applications must have some positioning component. Our node placement strategy can obtain the position information from this type of component without adding any extra overhead. Localization and positioning underwater is an active area of research and we may have a good positioning mechanism for underwater sensor networks in the near future. However, any such positioning mechanism is likely to have some errors in determining node positions. An underwater sensor network designer can accommodate any potential error by setting the

sensing range and communication range of each sensor to a fraction of the actual sensing and communication range. Clearly, the value of this fraction depends on the magnitude of the positioning error. Thus, by setting the sensing and communication range to a value lower than the actual value, we can accommodate small violations of sphere-based sensing and communication, homogeneous sensing and communication, and adjustable node position assumptions.

This work is applicable to both fixed and mobile networks. If sensor nodes are immobile, then the solution should be used during the initial node deployment. If nodes are mobile, then sensor nodes should dynamically compute their desired locations in a distributed fashion and move to the appropriate location to achieve the above stated goals.

A distributed version of the placement strategies can be devised in the following way: First, sensor nodes choose a leader by any standard leader election algorithm [33]. The structure can then grow relative to the location of the leader. For example, the location of the leader, say (cx,cy,cz), can be used as the center of the u,v,w-coordinate system and then the same placement strategy described here can be used by other nodes to determine their appropriate location and they can move accordingly. The placement of nodes can grow like a lattice, using the location of the leader as the seed. This approach works best if the sensor nodes know their precise location; e.g., using a GPS-like system. However, local estimation of distances to neighbors could also be used if sensor nodes can reach a consensus on the frame of reference (e.g., x,y,z-axes). For networks in an ocean or in the atmosphere, a good frame of reference can be as follows: the xy-plane is parallel to the earth's surface and the y-axis is parallel to the

axis going through earth's north-south pole, with the north-pole in the positive direction. Compass-like instrumentation can be used to find this direction by using the earth's magnetic field. Finally, the *z*-axis can be taken as perpendicular to the *xy*-plane with the positive direction away from the earth's surface, pointing into the space. This approach works if the network is not very large in the *x*- and the *y*-directions, so that the earth's surface can be approximated by a plane. A much simpler solution for this frame of reference problem is likely to exist.

In 2D, routing based on location information has been explored in [28][7][8][26] and [24]. In a 3D network, if placement of sensor nodes follows any of our placement strategies, then the locations of all other nodes are instantly available from our *u,v,w*-coordinate system. If the *u,v,w*-coordinate of each node is used as its id, then possible routes between two nodes can be easily determined. Location-based routing protocols can exploit this location-id information.

3.5 SUMMARY

In this chapter, we provide a solution for coverage and connectivity problem of 3D networks. Motivated by results of 2D cellular networking and using the century-old Kelvin's conjecture, we conjecture that dividing the 3D space into identical truncated octahedron virtual cells of radius equal to sensing range, r_s and then placing a sensor node at the center of each virtual cell would minimize the number of sensor nodes in the network. We defined a metric called the *volumetric quotient* (*V.Q.*) that is a measure of the quality of the competing space-filling polyhedrons as the shape of the virtual cell. The truncated octahedron turns out to be the best choice, with *V.Q.* of 0.68329. This is much better than the *V.Q.*s of all of the other possible choices considered here (both optimized hexagonal prism and rhombic dodecahedron have

*V.Q.*s of 0.477, while the cube *V.Q.* is just 0.36755). The number of nodes required for coverage of a large 3D space depends on the shape of a cell created by the Voronoi tessellation of that space by those nodes. If the shape of each cell is a space-filling polyhedron with a high *V.Q.*, then the number of nodes is small. For example, the number of nodes for the rhombic dodecahedron or the hexagonal prism placement requires 43.25% more nodes than does the truncated octahedron placement. After finding the optimal placement strategy, we examined the connectivity issues and we found that the best placement strategy (the truncated octahedron) requires the communication range, r_c, to be at least $4/\sqrt{5}$ times the sensing range, r_s in order to maintain full connectivity.

Since the regular truncated octahedron based model satisfies the full connectivity constraints only when $r_c/r_s \geq 4/\sqrt{5}$, this solution may not be applicable in all scenarios. We then find a generalize solution that works for any ratio of communication range, r_c, and sensing range, r_s. In particular, we establish that if $r_c/r_s \geq 1.587401$, then the truncated octahedron model with cell radius as $R = \min\left(r_c\sqrt{5}/4, r_s\right)$; if $1.587401 > r_c/r_s \geq 1.211414$ then hexagonal prism model with height of a cell as $h = \min\left(r_c, 2\sqrt{r_s^2 - a^2}\right)$ and the length of each side of a hexagonal face as $a = \min\left(r_c/\sqrt{3}, r_s\sqrt{2}/\sqrt{3}\right)$; if $r_c/r_s < 1.211414$, then cube model with cell radius $R = \min\left(r_c\sqrt{3}/2, r_s\right)$ achieve full coverage and full connectivity with all geographically neighboring nodes, by deploying minimum number of sensor nodes among the four models considered. This result is most significant when $r_c/r_s < 1.211414$. Since the cube model is the simplest and has the least boundary effect, in practice, network designers might be tempted to use this grid-like placement of sensor nodes in 3D. The good news for them is that this simple placement strategy requires a minimum number of nodes to achieve full coverage and full connectivity, if

r_c/r_s<1.211414.

We then investigate the coverage and connectivity problem in 3D network for a special case. When the value of r_c/r_s is very small, instead of maintaining full connectivity with all geographically neighboring nodes, full coverage and 1-connectivity with the minimum number of nodes are more cost effective and desirable. Inspired by a strip based solution for a similar problem in 2D networks, we provide a strip based node placement mechanism. If the value of r_c/r_s is smaller than $4/\sqrt{5}$, then this strategy achieves full coverage and 1-connectivity with a smaller number of nodes than used in the strategy that achieves full coverage and full connectivity. Savings in the number of nodes increases with the decrease in the value of r_c/r_s.

CHAPTER 4

COVERAGE AND CONNECTIVITY IN 3D NETWORKS WITH RANDOM NODE DEPLOYMENT

In this chapter, we analyze coverage and connectivity problem in 3D networks where we have no control over the location of any particular node. A large number of nodes are deployed randomly and uniformly in the network space. We exploit redundancy to improve network lifetime by partitioning the 3D network space into identical cells. Only one node remains active in each cell, and full coverage and connectivity are maintained. We then extend our work to *k*-coverage, where any point in the network has to be within the sensing range of at least *k* nodes. We also provide comparison between our proposed scheme and the scheme in which an oracle can decide where to place sensor nodes. Since that scheme can control the position of sensor nodes, it requires even fewer active nodes than does the optimal scheme. In order to highlight this distinction, we call this scheme *SuperOpt*. The performance of our scheme improves for larger values of *k*.

4.1 ANALYSIS

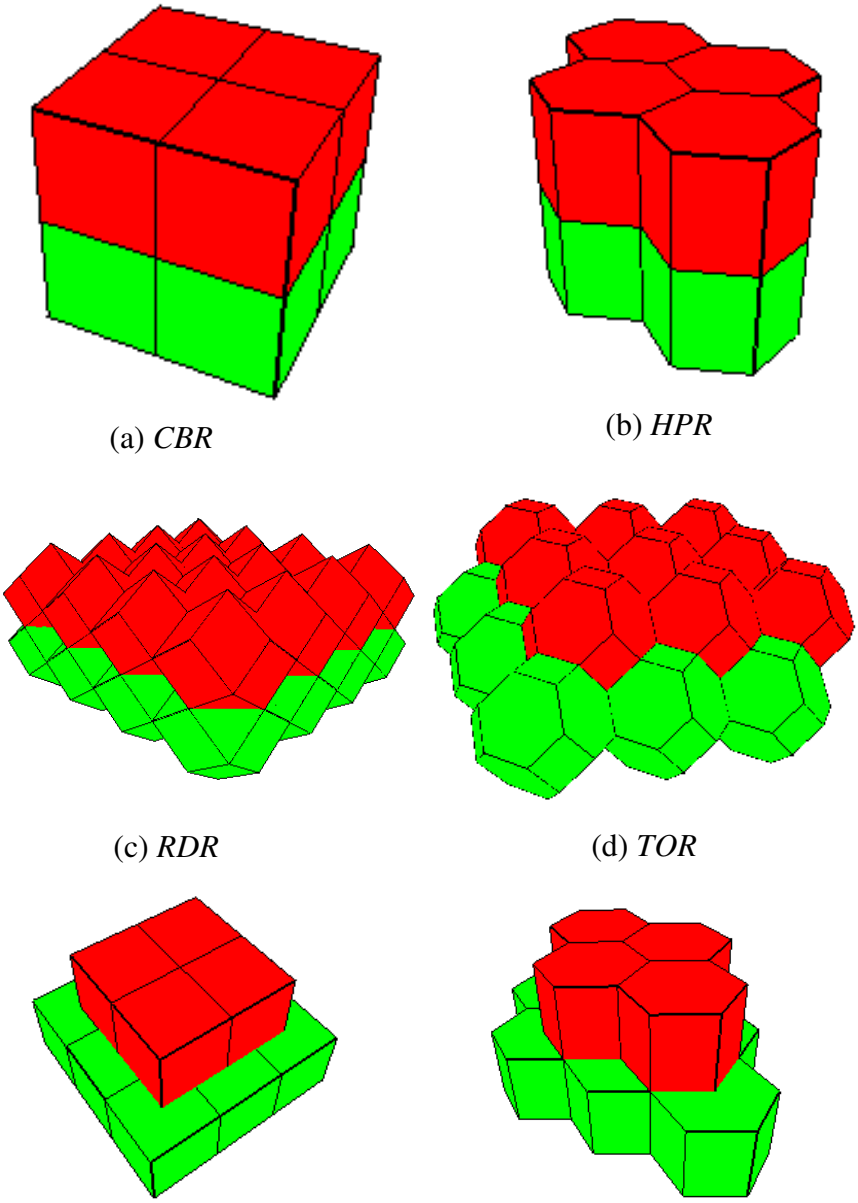
Here, we consider the scenario where deploying and maintaining every node in a carefully planned location is not feasible. This may happen if the sensor nodes cannot be deployed in pre-determined positions and/or they cannot maintain predetermined positions due to water current, gravity, marine animals, etc. The topology control algorithm has to assume that the sensor nodes are randomly deployed. However, due

to this random deployment, full coverage and connectivity can be ensured only if many redundant nodes are densely deployed. Of course, keeping redundant nodes active increases the consumption of energy and also may increase congestion by sending redundant messages. Therefore, it is important to find a dynamic mechanism that decreases the redundant active nodes by selecting a subset of sensor nodes to act as active nodes at any time. One simple way to do this is to partition the network space into cells and to keep one node active in each cell. In order to make the selection process distributed, we also impose the restriction that cells are identical. Clearly, the smaller the number of active nodes is at a time, the larger is the energy savings. However, maintaining full connectivity requires that the maximum distance between the active nodes of any two first-tier neighboring cells cannot exceed the transmission radius (a.k.a. the communication range). Since the active node can be located anywhere inside a cell, the maximum distance between any two points of the two first-tier neighboring cells must be less than or equal to the transmission radius. One major work in this context is the geographic adaptive fidelity (GAF) [51]. GAF divides a 2D network into squared virtual cells (a.k.a. grids) and keeps one node active in each cell. GAF can be shown to perform better when the shape of the virtual cell is a hexagon, rather than a square. The energy savings of GAF depends heavily on the choice of the partitioning scheme, because the number of active nodes at any time is equal to the total number of virtual cells. Clearly, the hexagonal partitioning scheme of the 2D networks is not applicable in 3D networks. In this section, we investigate and provide a solution for this partitioning problem in 3D.

It should be noted that any criticism of GAF in 2D also applies to the scheme in 3D. For example, even the best possible partitioning scheme may require more than an optimal number of active nodes to achieve full coverage and connectivity. We

investigate this in detail later in this section. Our scheme treats all nodes in a virtual cell as equivalent from the points of view of coverage and the connectivity. This scheme works well only in a network that has dense and uniform node deployment. Another criticism of GAF is that, for a network, where no node is physically located in a cell, the selection of which node is to be active in each cell makes a big difference [7]. However, we assume node density is high enough so that this problem does not occur.

Like our previous problem of coverage and connectivity with the minimum number of nodes, our focus here is on the four most common polyhedrons that tessellate a 3D space: the cube, hexagonal prism, rhombic dodecahedron, and truncated octahedron. However, unlike our previous problem, the arrangement of cells is now important, as the distance between any two points of two neighboring cells must be within the transmission radius. For a truncated octahedron and a rhombic dodecahedron, only one arrangement of cells is possible – the regular 3D space tessellation. On the other hand, for a cube and a hexagonal prism, an alternate arrangement of cells is possible that asymptotically requires fewer nodes than regular 3D space tessellation. We designate these alternate arrangements of cube and hexagonal prism as *Alt-CBR* and *Alt-HPR* (See Figure 18). The regular 3D space tessellation of a cube, hexagonal prism, rhombic dodecahedron, and truncated octahedron shaped cells are referred to as the *CBR*, *HPR*, *RDR*, and *TOR* models, respectively. The last character "*R*" in the name of each model is to emphasis the random position of the active node inside each cell. This is different from our previous work, where a node was always located at the center of each cell.



(a) *CBR* (b) *HPR*

(c) *RDR* (d) *TOR*

(e) *Alt-CBR* (f) *Alt-HPR*

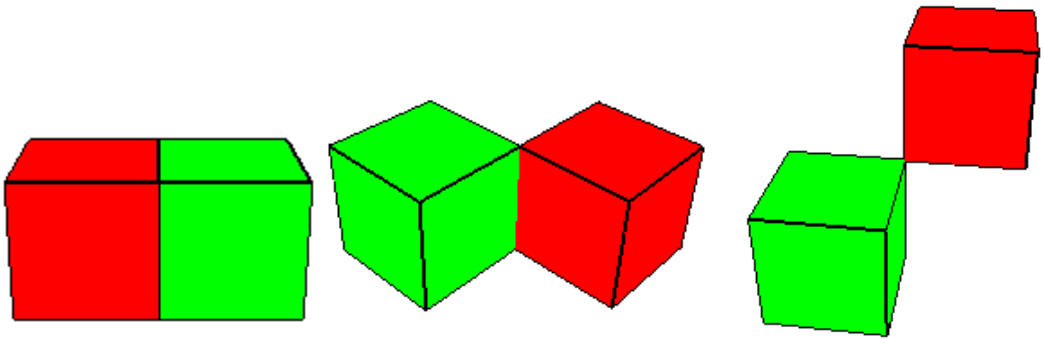
Figure 18: 3D Partitioning Schemes

4.1.1 Minimum Transmission Range Needed in Different Models

Since an active node can be located anywhere inside a cell and yet it still must be able to sense any point inside the cell, the sensing range must be at least equal to the distance between the two furthest points of a cell. This maximum distance is essentially the diameter of a cell and equals to twice the radius of the cell. Consequently, in all models, the radius of a cell, $R = r_s/2$. Given a fixed sensing radius r_s, the minimum required transmission ranges in the *CBR*, *Alt-CBR, HPR, Alt-HPR, RDR,* and *TOR* models are calculated below.

4.1.1.1 CBR Model

A cell has 26 geographically neighboring cells: 6 *Type* 1_{CBR} neighboring cells each share whole one side of a cube, 12 *Type* 2_{CBR} neighboring cells each share a common line, and 8 *Type* 3_{CBR} neighboring cells each share just a single common point with the cell (See Figure 19).



(a) *Type* 1_{CBR} Neighbors

(b) *Type* 2_{CBR} Neighbors

(c) *Type* 3_{CBR} Neighbors

Figure 19: Different types of neighbors in the *CBR* model

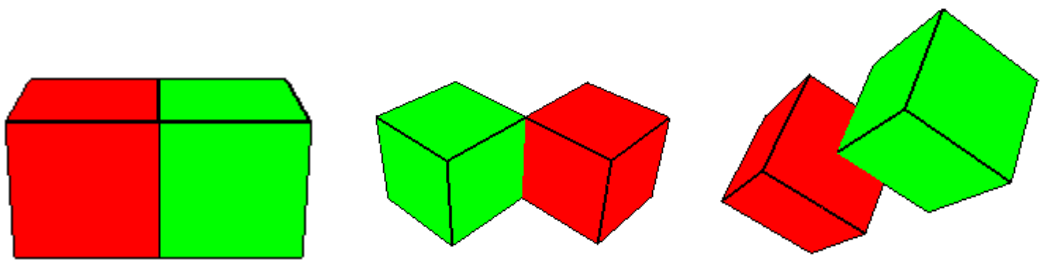
The largest distance between any point in the cell and any point in a *Type* 1_{CBR}

neighboring cells is $r_s\sqrt{2}$; while for *Type* 2_{CBR} and *Type* 3_{CBR} neighbors, it is $r_s\sqrt{3}$ and $2r_s$, respectively. The active node of a cell can communicate with active nodes of all first-tier neighboring cells if the minimum transmission range is:

$$r_t = \max\left(r_s\sqrt{2}, r_s\sqrt{3}, 2r_s\right)$$
$$= 2r_s$$

4.1.1.2 Alt-CBR Model

A cell has 16 geographically neighboring cells: 4 *Type* $1_{Alt\text{-}CBR}$ neighboring cells each share whole one side of a cube, 4 *Type* $2_{Alt\text{-}CBR}$ neighboring cells each share a common line, and 8 *Type* $3_{Alt\text{-}CBR}$ neighboring cells each share one quarter of one side of the cell (See Figure 20).



(a) *Type* $1_{Alt\text{-}CBR}$ Neighbors

(b) *Type* $2_{Alt\text{-}CBR}$ Neighbors

(c) *Type* $3_{Alt\text{-}CBR}$ Neighbors

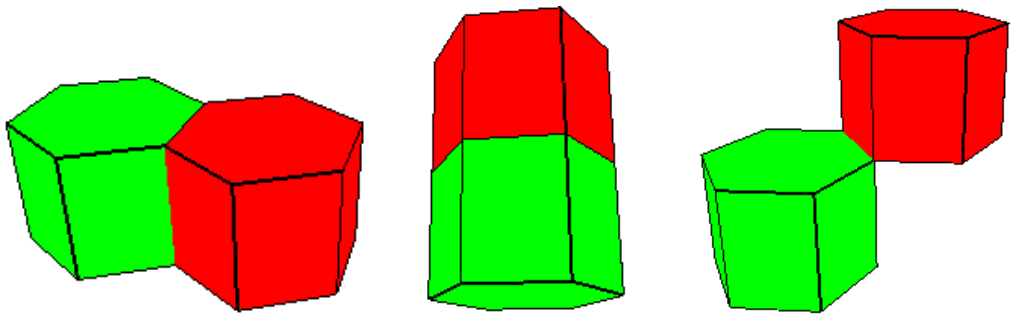
Figure 20: Different types of neighbors in the *Alt-CBR* model

The largest distance for *Type* $1_{Alt\text{-}CBR}$, *Type* $2_{Alt\text{-}CBR}$ and *Type* $3_{Alt\text{-}CBR}$ cells is $r_s\sqrt{2}$, $r_s\sqrt{3}$, and $r_s\sqrt{\frac{17}{6}}$, respectively. The minimum required transmission range in the *Alt-CBR* model is:

$$r_t = \max\left(r_s\sqrt{2}, r_s\sqrt{3}, r_s\sqrt{\frac{17}{6}}\right)$$
$$= r_s\sqrt{3}$$

4.1.1.3 HPR Model

A cell has 20 geographically neighboring cells: 6 *Type* 1_{HPR} neighboring cells each share a common square plane, 2 *Type* 2_{HPR} neighboring cells each share a common hexagonal plane, and 12 *Type* 3_{HPR} neighboring cells each share a common line with the cell (See Figure 21).



(a) *Type* 1_{HPR} Neighbors

(b) *Type* 2_{HPR} Neighbors

(c) *Type* 3_{HPR} Neighbors

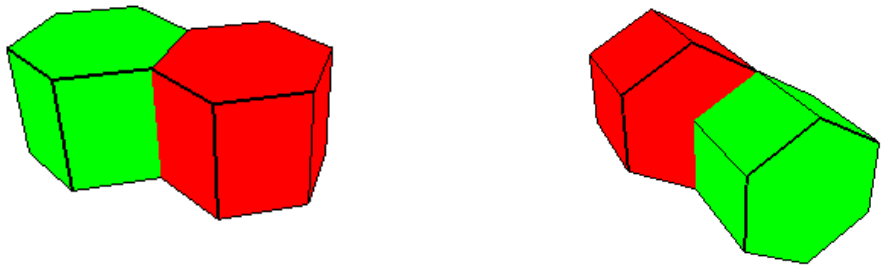
Figure 21: Different types of neighbors in the *HPR* model

Suppose that each side of the hexagonal face of a *HPR* cell has a length *a*, and its height is *h.* In a *HPR* cell with optimal height, $h = a\sqrt{2}$. Thus, the radius of the *HPR* cell is $\frac{r_s}{2} = \sqrt{a^2 + \left(a^2/2\right)} = a\sqrt{3}/\sqrt{2}$. The maximum distance from any point of the cell to any point of a *Type* $1_{HPR,}$ *Type* 2_{HPR} and *Type* 3_{HPR} neighbor is $\sqrt{\left(a\sqrt{13}\right)^2 + h^2} = r_s\sqrt{\frac{5}{2}}$, $\sqrt{(2a)^2 + (2h)^2} = r_s\sqrt{2}$ and $\sqrt{\left(a\sqrt{13}\right)^2 + (2h)^2} = r_s\sqrt{\frac{7}{2}}$, respectively. The active node of a cell can communicate with active nodes of all neighboring cells if minimum transmission range is:

$$r_t = \max\left(r_s\sqrt{\frac{5}{2}}, r_s\sqrt{2}, r_s\sqrt{\frac{7}{2}} \right) = r_s\sqrt{\frac{7}{2}}.$$

4.1.1.4 Alt-HPR Model

A cell has 12 first-tier neighboring cells: 6 *Type* $1_{Alt\text{-}HPR}$ neighboring cells each share a square plane and 6 *Type* $2_{Alt\text{-}HPR}$ neighboring cells each share one third of a hexagonal plane with the cell (See Figure 22).



(a) *Type* $1_{Alt\text{-}HPR}$ Neighbors (b) *Type* $2_{Atl\text{-}HPR}$ Neighbors

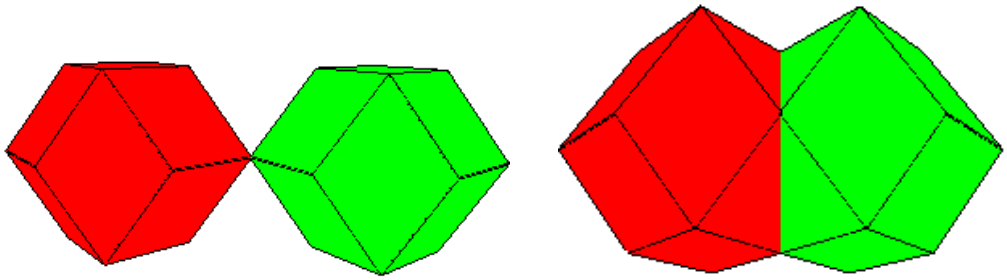
Figure 22: Different types of neighbors in the *Alt-HP* model

The maximum distance for *Type* $1_{Alt\text{-}HPR}$ and *Type* $2_{Alt\text{-}HPR}$ neighbors is $\sqrt{\left(a\sqrt{13}\right)^2 + h^2} = r_s\sqrt{\frac{5}{2}}$ and $\sqrt{(3a)^2 + (2h)^2} = r_s\sqrt{\frac{17}{6}}$, respectively. Thus, the minimum transmission range needed is:

$$r_t = \max\left(r_s\sqrt{\frac{5}{2}}, r_s\sqrt{\frac{17}{6}} \right) = r_s\sqrt{\frac{17}{6}}$$

4.1.1.5 RDR Model

A cell has 18 geographically neighboring cells: 6 *Type* 1_{RDR} neighboring cells each share just a point and 12 *Type* 2_{RDR} neighboring cells each share a plane with the cell (See Figure 23).



(a) *Type* 1_{RDR} Neighbors (b) *Type* 2_{RDR} Neighbors

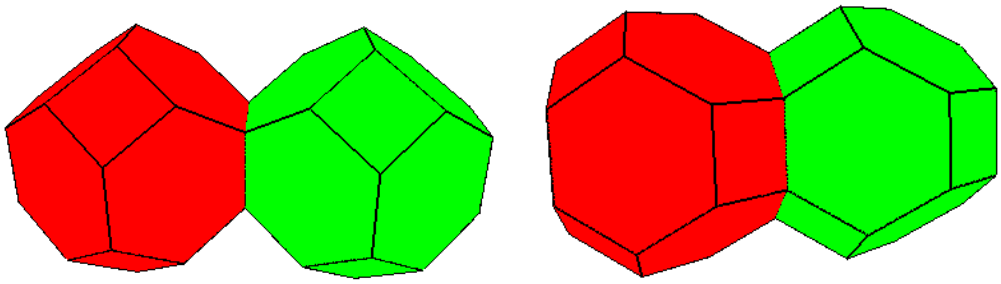
Figure 23: Different types of neighbors in the *RDR* model

The maximum distance for *Type* 1_{RDR} and *Type* 2_{RDR} neighbor is $2r_s$ and $r_s\sqrt{\frac{5}{2}}$, respectively. Thus, the minimum transmission range required in the *RDR* model is:

$$r = \max\left(2r_s, r_s\sqrt{\frac{5}{2}}\right).$$
$$= 2r_s$$

4.1.1.6 TOR Model

A cell has 14 geographically neighboring cells: 6 *Type* 1_{TOR} neighboring cells each share a common square plane and 8 *Type* 2_{TOR} neighboring cells each share a common hexagonal plane with the cell (See Figure 24).



(a) *Type* 1_{TOR} Neighbors (b) *Type* 2_{TOR} Neighbors

Figure 24: Different types of neighbors in the *TOR* model

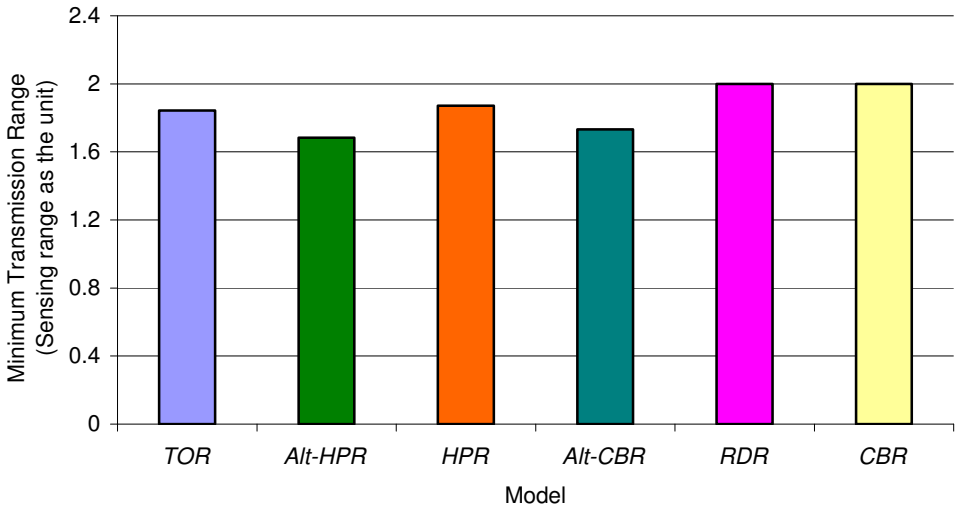


Figure 25: Minimum transmission range required in different models

The maximum distance for *Type* 1_{TOR} and *Type* 2_{TOR} neighbor is $r_s\sqrt{17}/\sqrt{5}$ and $r_s\sqrt{14}/\sqrt{5}$, respectively. The active node of a cell can communicate with active nodes of all neighboring cells if transmission range is at least:

$$r_t = \max\left(r_s\sqrt{\frac{17}{5}}, r_s\sqrt{\frac{14}{5}} \right)$$
$$= r_s\sqrt{\frac{17}{5}} \quad .$$

The minimum transmission range required for maintaining connectivity in each model is shown in Figure 25.

4.1.2 A Distributed Partitioning Scheme

This network architecture can easily be created in a distributed fashion if all nodes know their cell id. Since the technique is similar for all models, here we provide a calculation only for the *TOR* model. Suppose that the information sink (*IS*), where all data are gathered, resides in the center of a virtual cell and its coordinate (*x, y, z*) is known. For the *TOR* model, the center of a virtual cell can then be expressed by the general equation:

$$f(u,v,w) = \left(x + \frac{(2u+w)\,r_s}{\sqrt{5}},\ y + \frac{(2v+w)\,r_s}{\sqrt{5}},\ z + \frac{w r_s}{\sqrt{5}} \right).$$

Three integers (*u,v,w*) can be used as a unique cell id. The cell that contains *IS* has cell id (0, 0, 0). As an example, cell id (-1, -1, 2) has its center in coordinate $\left(x,\ y,\ z + \frac{2r_s}{\sqrt{5}} \right)$.

A sensor node can determine its own coordinate (x_s, y_s, z_s) using its localization component. *IS* can broadcast its coordinate (*x, y, z*) to all nodes and the sensing range r_s can be embedded in the sensor before deployment. In order to determine its cell id (u_s, v_s, w_s), a brute force method checks all possible values of (u_s, v_s, w_s) and

choose the cell whose center has the minimum Euclidean distance from the node, i.e.:

$$(u_s, v_s, w_s) = \arg \min_{\substack{u \in Z, \\ v \in Z, \\ w \in Z}} \left(x_s - x - (2u + w)\frac{r_s}{\sqrt{5}} \right)^2 + \left(y_s - y - (2v + w)\frac{r_s}{\sqrt{5}} \right)^2 + \left(z_s - z - w\frac{r_s}{\sqrt{5}} \right)^2$$

where Z is the set of all integers. However, an exhaustive search can easily be avoided. Since the value of a square term is never negative, we can set the value of the square terms to zero to get the values of u_s, v_s, and w_s. Since these values must be integers, we can get two possible integral values for each variable by taking the ceiling (denoted by subscript *h*) and floor (subscript *l*):

$$u_l = \left\lfloor (x_s - x - z_s + z)\sqrt{5}/2r_s \right\rfloor,\ u_h = \left\lceil (x_s - x - z_s + z)\sqrt{5}/2r_s \right\rceil,$$

$$v_l = \left\lfloor (y_s - y - z_s + z)\sqrt{5}/2r_s \right\rfloor,\ v_h = \left\lceil (y_s - y - z_s + z)\sqrt{5}/2r_s \right\rceil,$$

$$w_l = \left\lfloor (z_s - z)\sqrt{5}/r_s \right\rfloor,\ w_h = \left\lceil (z_s - z)\sqrt{5}/r_s \right\rceil.$$

Thus, we have eight possible values of (u_s, v_s, w_s). Each node has to calculate its distance from each of the eight centers and then choose the minimum one as its cell id, i.e.:

$$(u_s, v_s, w_s) = \arg \min_{\substack{u \in \{u_l, u_h\}, \\ v \in \{v_l, v_h\}, \\ w \in \{w_l, w_h\}}} \sqrt{\left(x_s - x - (2u + w)\frac{r_s}{\sqrt{5}} \right)^2 + \left(y_s - y - (2v + w)\frac{r_s}{\sqrt{5}} \right)^2 + \left(z_s - z - w\frac{r_s}{\sqrt{5}} \right)^2}$$

As cell id is a straightforward function of the location of a sensor, if a sensor knows the location of another sensor, it can readily calculate the cell id of that sensor. We use simulation to validate that each sensor node can determine its cell id correctly, according to above technique. In a very large number of trials, we found that, in every case, our equations (i.e., the ceiling and floor approach) were able to predict the cell id correctly. However, further effort to simplify the prediction process was not successful. For example, instead of calculating the distance from each of the eight centers, if we simply take the nearest integer value for u_s,v_s, w_s, then this approximation leads to incorrect prediction of cell id in almost one quarter of the cases (See Figure 26).

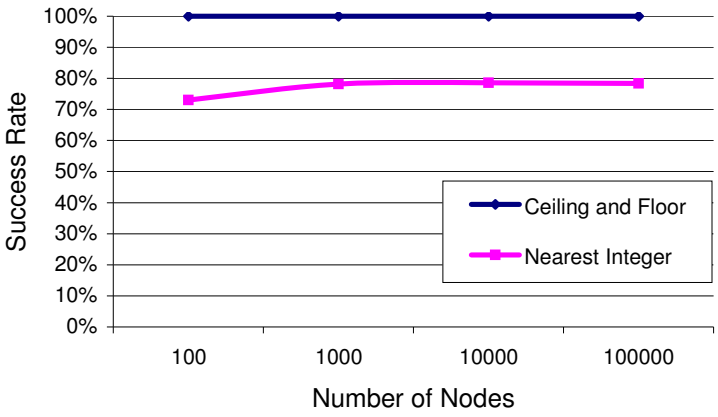


Figure 26: Cell id prediction accuracy

However, since there are only eight possible combinations, the calculations involved in our technique to find the cell id is just a small constant number of local arithmetic operations.

Once sensors have their cell ids, then sensors with same cell ids can use any standard

leader selection algorithms [33] to choose a leader among them, which then acts as the active node of that cell. All nodes that have same cell id are within the communication range of each other and the mechanism of keeping one node active among all the sensors with same cell id is essentially same for both 2D and 3D networks. Since the main focus of this dissertation is problems that are unique to 3D networks, we choose not to explore the issues that have already been studied in the context of 2D networks.

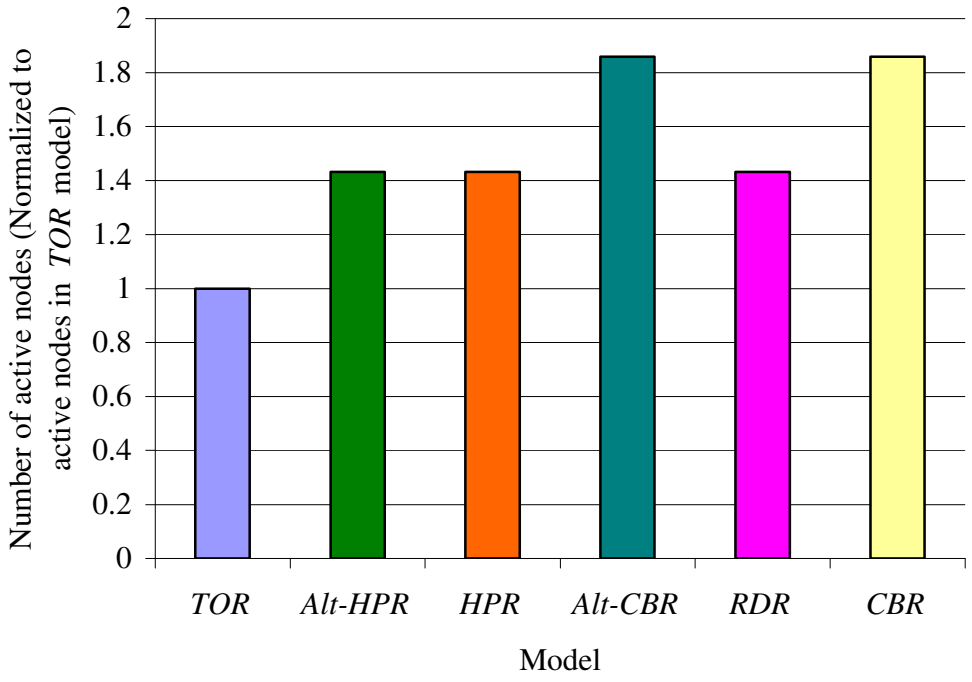


Figure 27: Number of active nodes in various models

4.1.3 Number of Active Nodes and Cell Lifetime

Ignoring boundary effect, the number of cells in a network is inversely proportional to the volume of the network. Since, at any time, the number of active nodes in a cell is one, the total number of active nodes in a network is equal to the number of cells in the network. The volume of a cube, hexagonal prism, rhombic dodecahedron and

truncated octahedron of radius R is $8R^3/3\sqrt{3}$, $2R^3$, $2R^3$ and $32R^3/5\sqrt{5}$, respectively.

Since we have $R = r_s/2$ in all models, the volume of a cell in each model is:

CBR and *Alt-CBR*: $8\left(\frac{r_s}{2}\right)^3 \Big/ 3\sqrt{3} = \frac{r_s^3}{3\sqrt{3}}$

HPR and *Alt-HPR*: $2\left(\frac{r_s}{2}\right)^3 = \frac{r_s^3}{4}$,

RDR: $2\left(\frac{r_s}{2}\right)^3 = \frac{r_s^3}{4}$,

TOR: $32\left(\frac{r_s}{2}\right)^3 \Big/ 5\sqrt{5} = \frac{4}{5\sqrt{5}} r_s^3$.

Consequently, the active nodes required by the *CBR, Alt-CBR, HPR, Alt-HPR* and *RDR* models are, respectively, $12\sqrt{3}/5\sqrt{5}$, $12\sqrt{3}/5\sqrt{5}$, $16/5\sqrt{5}$, $16/5\sqrt{5}$ and $16/5\sqrt{5}$ times of that of the *TOR* model. Numbers of active nodes in various models with respect to that of the *TOR* model are shown in Figure 27.

We use a simplified model to calculate the network lifetime for different partitioning schemes. We assume that the number of packets transmitted and relayed by a cell is the same in each model. The lifetime of the cell then depends on the transmission range used by a model and the number of nodes that resides inside a cell in that model. Since the assumption is that the sensor nodes are uniformly distributed, the number of nodes in a cell is proportional to the volume of the cell. Finally, in a radio network, power consumption to transmit a packet is proportional to the square of the transmission range. Suppose that two models *A* and *B* have transmission ranges r_A and r_B, respectively. The volume of a cell in model *A* and *B* is V^A and V^B, respectively. If the cell lifetime of models *A* and *B* is denoted by L^A and L^B, respectively, then we

have:

$$\frac{L^A}{L^B} = \frac{r_B^2}{r_A^2} \times \frac{V^A}{V^B}.$$

Using this equation, the cell lifetime of each model, as compared to the cell lifetime of the *TOR* model, is calculated below:

$$\frac{L^{CBR}}{L^{TOR}} = \frac{\left(r_s\sqrt{\frac{17}{5}}\right)^2}{\left(2r_s\right)^2} \times \frac{\frac{r_s^3}{3\sqrt{3}}}{\frac{4r_s^3}{5\sqrt{5}}} = \frac{17\sqrt{5}}{48\sqrt{3}}, \quad \frac{L^{Alt-CBR}}{L^{TOR}} = \frac{\left(r_s\sqrt{\frac{17}{5}}\right)^2}{\left(r_s\sqrt{3}\right)^2} \times \frac{\frac{r_s^3}{3\sqrt{3}}}{\frac{4r_s^3}{5\sqrt{5}}} = \frac{17\sqrt{5}}{36\sqrt{3}},$$

$$\frac{L^{HPR}}{L^{TOR}} = \frac{\left(r_s\sqrt{\frac{17}{5}}\right)^2}{\left(r_s\sqrt{\frac{7}{2}}\right)^2} \times \frac{\frac{r_s^3}{4}}{\frac{4r_s^3}{5\sqrt{5}}} = \frac{17\sqrt{5}}{56}, \quad \frac{L^{Alt-HPR}}{L^{TOR}} = \frac{\left(r_s\sqrt{\frac{17}{5}}\right)^2}{\left(r_s\sqrt{\frac{17}{6}}\right)^2} \times \frac{\frac{r_s^3}{4}}{\frac{4r_s^3}{5\sqrt{5}}} = \frac{3\sqrt{5}}{8},$$

$$\frac{L^{RDR}}{L^{TOR}} = \frac{\left(r_s\sqrt{\frac{17}{5}}\right)^2}{\left(2r_s\right)^2} \times \frac{\frac{r_s^3}{4}}{\frac{4r_s^3}{5\sqrt{5}}} = \frac{17\sqrt{5}}{64}$$

The cell lifetime of various models, as compared to the cell lifetime of the *TOR* model, is shown in See Figure 28.

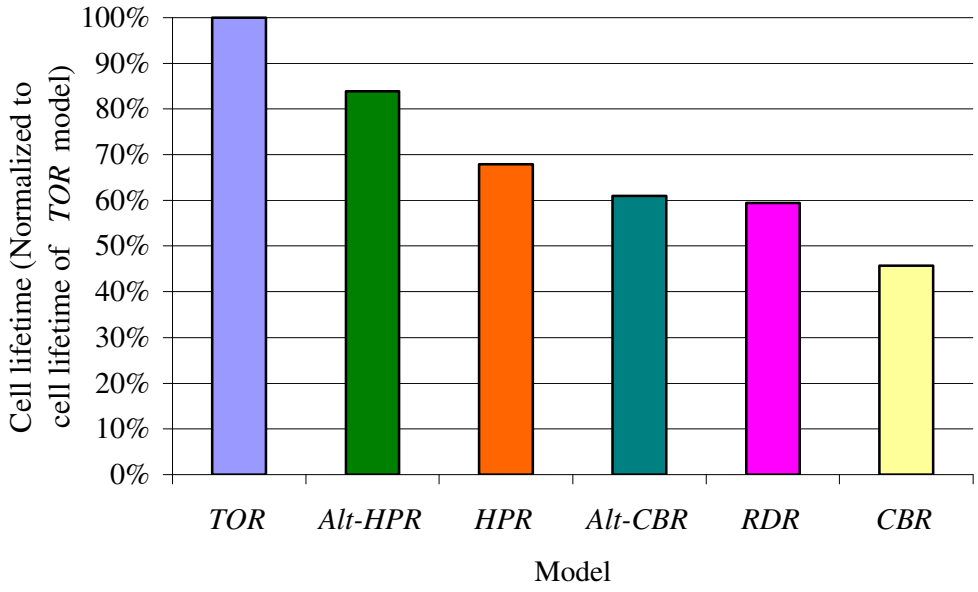


Figure 28: Cell lifetime of various models

4.2 *k*-COVERAGE AND COMPARISON WITH *SuperOpt*

Clearly, our GAF like approach of dividing a network into cells and keeping one node active in each cell is suboptimal. The number of nodes required by 2D-GAF can be shown to be 4 times of that the optimal number and, in the 3-D case, this value is 8 times. Although we cannot improve this scenario for 1-coverage, the GAF-like approach may require significantly fewer nodes for *k*-coverage with high probability.

4.2.1 2D GAF

Let us first explore how to improve 2D-GAF for *k*-coverage. For 1-coverage, we have to keep one node active in a hexagonal cell with radius $r = r_s/2$, where r_s is the sensing range of each sensor. For *k*-coverage, we propose to extend the 2D GAF by setting the radius of each cell as $r = r_s \Big/ \left(2\sqrt{\lceil k/4 \rceil}\right)$ and to keep one node active in each

cell. We want to answer the following two questions:

1. What is the probability that this scheme has *k*-coverage?
2. How many nodes will this scheme require compared to *SuperOpt*?

To answer these two questions, we need the help of the following theorem:

Theorem 4.1: Suppose that we have two areas and, in each area, nodes are randomly distributed, based on a 2D *Poisson* distribution. The sum of the number of nodes in two (independently) selected sub-areas is then *Poisson*, with a parameter equal to the sum of the expected number of nodes in each individual area.

Proof:

Assume that we have two areas, A_1[m^2] and A_2 [m^2], where nodes are randomly distributed in each area based on 2D *Poisson* distribution, with parameters ρ_1 [nodes/m^2] and ρ_2 [nodes/m^2], respectively. Within the areas A_1 and A_2, there are sub-areas, a_1 [m^2] and a_2 [m^2], respectively, which are chosen independently from one another. Consequently, the expected numbers of nodes in the two sub-area a_1 and a_2 are: $a_1\rho_1$ and $a_2\rho_2$, respectively, and the number of nodes within each sub-area is also *Poisson* with the parameters $\lambda_1 = a_1\rho_1$ and $\lambda_2 = a_2\rho_2$, respectively (this can be easily shown).

Let us label K_1 and K_2 as the random variable indicating the number of nodes in the areas a_1 and a_2, respectively. We can write the probabilities of finding k nodes in area a_1 and area a_2 are, respectively, as:

$$P(K_1 = k) = e^{-\lambda_1}\frac{\lambda_1^{\,k}}{k!} \text{ and } P(K_2 = k) = e^{-\lambda_2}\frac{\lambda_2^{\,k}}{k!}.$$

We are trying to show that $P(K_1 + K_2 = k) = e^{-(\lambda_1+\lambda_2)} \frac{(\lambda_1 + \lambda_2)^k}{k!}$; i.e., the probability that the total number of nodes in both sub-areas, $K = k$, is also *Poisson* with the parameter equal to the expected total number of nodes in the two areas, $\lambda = \lambda_1 + \lambda_2 = a_1\rho_1 + a_2\rho_2$.

$$P(K_1 + K_2 = k)$$

$$= \sum_{i=0}^{k} P(K_1 + K_2 = k \mid K_2 = i) \times P(K_2 = i)$$

$$= \sum_{i=0}^{k} P(K_1 = k - i \mid K_2 = i) \times P(K_2 = i)$$

$$= \sum_{i=0}^{k} P(K_1 = k - i) \times P(K_2 = i)$$

$$= \sum_{i=0}^{k} \left[\left(e^{-\lambda_1} \frac{\lambda_1^{k-i}}{(k-i)!} \right) \times \left(e^{-\lambda_2} \frac{\lambda_2^{i}}{i!} \right) \right]$$

$$= e^{-(\lambda_1+\lambda_2)} \sum_{i=0}^{k} \left[\left(\frac{1}{(k-i)!\,i!} \right) \lambda_1^{k-i} \lambda_2^{i} \right]$$

$$= \frac{e^{-(\lambda_1+\lambda_2)}}{k!} \sum_{i=0}^{k} \left[\left(\frac{k!}{(k-i)!\,i!} \right) \lambda_1^{k-i} \lambda_2^{i} \right]$$

$$= \frac{e^{-(\lambda_1+\lambda_2)}}{k!} \sum_{i=0}^{k} \left[\binom{k}{i} \lambda_1^{k-i} \lambda_2^{i} \right]$$

$$= \frac{e^{-(\lambda_1+\lambda_2)}}{k!} (\lambda_1 + \lambda_2)^k$$

$$= \frac{e^{-(a_1\rho_1+a_2\rho_2)}}{k!} (a_1\rho_1 + a_2\rho_2)^k$$

Thus, if we have two independent *Poisson* random variables, the sum of the two variables is *Poisson* as well, with a parameter equal to the sum of the two individual parameters.

Note #1: In the proof above, we used the fact that the two random variables are independent:

$P(K_1 = k - i \mid K_2 = i) = P(K_1 = k - i)$.

Note #2: By repeating the process n-1 times, we can prove that the sum of n independent *Poisson* random variable is *Poisson* as well, with the parameter equal to the sum of the n parameters of the individual random variables.

In particular, for our application, the sum of the number of nodes in two (independently) selected sub-areas is *Poisson*, with parameter equal to the sum of the expected number of nodes in each individual area. Note that this is a valid statement, even if the two sub-areas are in the same area, as long as there is no overlap between the two sub-areas.

If the two areas have the same node density $\rho_1 = \rho_2 = \rho$, then $P(K_1 + K_2 = k) = e^{-\rho(a_1+a_2)} \left[\rho(a_1 + a_2)\right]^k \Big/ k!$; i.e., one can just simply "combine" the areas for the purpose of calculating the parameter of the *Poisson* distribution. Note that this would be the case when the two sub-areas are both within the same area (and are, of course, non-overlapping).

□

Now, regarding for our proposed 2D-GAF with k-coverage; the area of each cell

is $3\sqrt{3}r^2/2 = 3\sqrt{3}r_s^2/8\lceil k/4 \rceil$. Since we keep one node active in each such cell, the active node density is $\rho = 1 \Big/ \frac{3\sqrt{3}}{2}\frac{r_s^2}{4\lceil k/4 \rceil}$ node per unit area. Within r_s distance of any point, the number of active nodes is a *Poisson* random variable K with parameter

$$\lambda_k = \frac{\pi r_s^2}{\frac{3\sqrt{3}}{2}\frac{r_s^2}{4\lceil k/4 \rceil}} = \frac{8\pi\lceil k/4 \rceil}{3\sqrt{3}} .$$

The probability that any point is within the sensing radius of at least k nodes is given by:

$$\begin{aligned} P(K \geq k) &= 1 - P(K < k) \\ &= 1 - \sum_{i=0}^{k-1} P(K = i) \\ &= 1 - \sum_{i=0}^{k-1} e^{-\lambda_k} \frac{\lambda_k^i}{i!} \\ &= 1 - \sum_{i=0}^{k-1} e^{-\left(\frac{8\pi\lceil k/4 \rceil}{3\sqrt{3}}\right)} \frac{\left[\frac{8\pi\lceil k/4 \rceil}{3\sqrt{3}}\right]^i}{i!} \\ &= 1 - e^{-\left(\frac{8\pi\lceil k/4 \rceil}{3\sqrt{3}}\right)} \sum_{i=0}^{k-1} \frac{\left[8\pi\lceil k/4 \rceil\right]^i}{\left[3\sqrt{3}\right]^i i!} \end{aligned}$$

Table V: Probability of *k*-coverage and node requirement for 2D GAF

K	λ_k	*P(K>=k)*	*Number of nodes vs. Optimal*
1	4.8367983	1	400%
2	4.8367983	0.9616325	200%
3	4.8367983	0.8688446	133%
4	4.8367983	0.7192460	100%
5	9.6735966	0.9639949	160%

Now, it can be shown that one optimal solution for *k*-coverage is to divide the 2D plane into hexagonal cells of radius r_s and keep *k* nodes active at the center of each cell (this scheme is not applicable when nodes are randomly deployed: we mention it here only to find a lower bound on the number of nodes needed for *k*-coverage). Thus, the number of nodes needed by our proposed scheme is at most $\frac{3\sqrt{3}r_s^2/2k}{3\sqrt{3}r_s^2/8\lceil k/4\rceil} = \frac{4\lceil k/4\rceil}{k}$ times the number of nodes needed by *SuperOpt*.

From Table V, we see that our proposed scheme provides 1-coverage with probability 1, but the active node requirement is 4 times of that of *SuperOpt*. One the other hand, 4-coverage requires the same number of nodes as that of *SuperOpt*, but the probability of at least *k*-coverage falls to 0.72. Note that *SuperOpt* assumes that nodes can be deployed at any desired place, so the actual lower bound can be higher. This, in turn, means our scheme is probably better than what we indicate above.

4.2.2 3D GAF

For 1-coverage, we have to keep one node active in a truncated octahedron cell with $r = r_s/2$, where r_s is sensing range of each sensor. For *k*-coverage, we propose the following scheme: set the radius of each truncated octahedron cell $r = r_s \Big/ 2\sqrt[3]{\lceil k/8 \rceil}$ and keep one node active in each cell. The volume of each cell is then $32r^3/5\sqrt{5} = 4r_s^3 \Big/ \left(5\sqrt{5}\lceil k/8 \rceil\right)$. Since we keep one node active in each cell, the active node density is $\rho = 5\sqrt{5}\lceil k/8 \rceil \Big/ \left(4r_s^3\right)$ node per unit volume. Within r_s distance of any point, the number of active nodes is a *Poisson* random variable *K* with parameter:

$$\lambda_k = \frac{\frac{4}{3}\pi r_s^3}{\frac{4}{5\sqrt{5}}\frac{r_s^3}{\lceil k/8 \rceil}} = \frac{5\sqrt{5}\pi\lceil k/8 \rceil}{3}.$$

The probability that any point is within the sensing radius of at least *k* nodes is given by:

$$\begin{aligned} P(K \geq k) &= 1 - P(K < k) \\ &= 1 - \sum_{i=0}^{k-1} P(K = i) \\ &= 1 - \sum_{i=0}^{k-1} e^{-\lambda_k} \frac{\lambda_k^i}{i!} \\ &= 1 - \sum_{i=0}^{k-1} e^{-\left(\frac{5\sqrt{5}\pi\lceil k/8 \rceil}{3}\right)} \frac{\left[\frac{5\sqrt{5}\pi\lceil k/8 \rceil}{3}\right]^i}{i!} \\ &= 1 - e^{-\left(\frac{5\sqrt{5}\pi\lceil k/8 \rceil}{3}\right)} \sum_{i=0}^{k-1} \frac{\left[5\sqrt{5}\pi\lceil k/8 \rceil\right]^i}{3^i i!} \end{aligned}$$

Table VI: Probability of *k*-coverage and node requirement for 3D GAF

k	λ_k	$P(K>=k)$	*Number of nodes vs. Optimal*
1	11.70802455	1	800%
2	11.70802455	0.9999	400%
3	11.70802455	0.9994	233%
4	11.70802455	0.9971	200%

Now, it can be shown that one optimal solution for *k*-coverage is to divide the 3D space into hexagonal cells of radius r_s and to keep *k* nodes active at the center of each cell (this scheme is not applicable when nodes are randomly deployed: we mention it here only to find a lower bound on the number of nodes needed for *k*-coverage). Therefore, the number of nodes needed by our proposed scheme is at most $\frac{\frac{32}{5\sqrt{5}}\frac{r_s^3}{k}}{\frac{32}{5\sqrt{5}}\frac{r_s^3}{8\lceil k/8\rceil}} = \frac{8\lceil k/8\rceil}{k}$ times the number of nodes needed by *SuperOpt*.

From Table VI, we see that our 3D-GAF scheme achieves 4-coverage with a probability 0.9971 using twice the optimal number of nodes. Unlike 2D-GAF, 3D-GAF can provide *k*-coverage with very high probability for higher values of *k*. Therefore, 3D GAF is more promising than 2D GAF for higher values of *k*.

4.3 DISCUSSION

Nodes can use their cell ids as their addresses. A greedy geographic routing scheme

can work here, as follows: the source node writes its cell id and destination node's cell id in the packet. Suppose that the source cell id is (u_s, v_s, w_s) and the destination cell id is (u_d, v_d, w_d). Then the source sends this packet to a neighbor with cell id (u_i, v_i, w_i) such that:

$$(u_d - u_i)^2 + (v_d - v_i)^2 + (w_d - w_i)^2 < (u_d - u_s)^2 + (v_d - v_s)^2 + (w_d - w_s)^2.$$

Then the node with cell id (u_i, v_i, w_i) sends this packet to a neighbor with cell id (u_j, v_j, w_j) such that:

$$(u_d - u_j)^2 + (v_d - v_j)^2 + (w_d - w_j)^2 < (u_d - u_i)^2 + (v_d - v_i)^2 + (w_d - w_i)^2.$$

If more than one neighbor satisfies these criteria (which most often actually the case), then the least loaded node, the node with the highest energy or just one random node can be chosen. When the shape of each cell is a truncated octahedron, each cell has 14 neighboring cells. The neighboring cells of a cell having cell id (u_1, v_1, w_1) have the following ids: $(u_1 + 1,\ v_1,\ w_1)$, $(u_1 - 1,\ v_1,\ w_1)$; $(u_1,\ v_1 + 1,\ w_1)$, $(u_1,\ v_1 - 1,\ w_1)$; $(u_1 - 1,\ v_1 - 1,\ w_1 + 2)$, $(u_1 + 1,\ v_1 + 1,\ w_1 - 2)$; $(u_1,\ v_1,\ w_1 + 1)$, $(u_1,\ v_1,\ w_1 - 1)$; $(u_1 - 1,\ v_1,\ w_1 + 1)$, $(u_1 + 1,\ v_1,\ w_1 - 1)$; $(u_1,\ v_1 - 1,\ w_1 + 1)$, $(u_1, v_1 + 1,\ w_1 - 1)$; $(u_1 - 1,\ v_1 - 1,\ w_1 + 1)$, $(u_1 + 1, v_1 + 1,\ w_1 - 1)$. Therefore, it requires a small constant number of arithmetic operations to choose the optimal neighboring node to forward a packet. This simple approach works well when all nodes are always connected with all of their neighboring nodes. However, this greedy scheme might not work in all possible scenarios. In the presence of an obstacle, there is a possibility that the packet reaches a dead end where there is no neighboring node that satisfies the criteria mentioned above and the packet is yet to reach the destination. Routing, in such cases, for 3D network has been investigated in [16][18].

4.4 SUMMARY

In this chapter, we investigate the coverage and connectivity issues in three-dimensional networks, where deploying and maintaining nodes in predetermined positions are difficult. Consequently, a large number of nodes have to be deployed randomly and uniformly, such that full sensing coverage can still be achieved. However, at any instant, all nodes are not needed for full sensing coverage. It is important to dynamically put those redundant nodes into sleep mode to increase the network lifetime based on the current positions of the sensor nodes. We provide a highly distributed and scalable scheme to achieve that goal in 3D networks. While a similar solution exists for 2D networks, the transition from 2D to 3D is not always easy, given that many problems in 3D are more difficult than are their 2D counterparts, by several orders of magnitude. In order make the solution highly distributed and scalable, we partition the 3D network space into identical regions (or, cells) and keep one node active in each of these cells.

Finding the right partitioning scheme for 3D networks was one of the most challenging problems. We analyze six partitioning schemes in 3D and find that partitioning the 3D space into truncated octahedron shaped cells is the best approach. In this case, full coverage can be achieved if the sensing range is at least 0.542326 times the transmission radius. We also compare different partitioning schemes based on their energy consumption and find that the truncated octahedron based partitioning scheme has a higher cell lifetime than do the other schemes. We describe a mechanism for each sensor node to determine in which cell it belongs, by using a small constant number of arithmetic operations that can be applied if the sensor node knows its own position. No message passing between nodes in different cells is needed to choose the

active node. While this scheme is highly distributed and scalable, sometime it is not as good as a centralized scheme that has global information about the position of all nodes. We extend our work for *k*-coverage, where sensing by *k* sensor nodes are needed. Our scheme can provide *k*-coverage in 3D with high probability, while significantly decreasing the gap with the centralized scheme with respect to the number of active nodes required. While the relative number of active nodes can be decreased in both 2D and 3D, only in 3D can *k*-coverage be ensured with high probability.

CHAPTER 5

CONCLUSIONS

Although terrestrial wireless sensor networks are usually modeled as 2D networks, wireless sensor networks that are deployed underwater, in the atmosphere and in space must be modeled as 3D networks. Since the existence of 3D networks is a relatively new phenomenon, many fundamental problems for these networks have not yet been investigated. In this dissertation, we investigate a number of previously unexplored but important problems for 3D networks.

One fundamental problem is to find a node placement strategy in 3D that achieves full coverage and full connectivity using the minimum number of sensor nodes. Both sensing range and communication range of each sensor node are assumed to be homogeneous, deterministic, and sphere based. The solution of this problem depends on the ratio of communication range, r_c and sensing range, r_s. When $r_c/r_s \le 4/\sqrt{5}$, dividing the 3D space into identical truncated octahedron virtual cells with radius r_s and placing a sensor node at the center of each cell can solve the problem. This result can also be explained by using crystal lattice structures. Since the Voronoi tessellation of a body-centered cubic (BCC) lattice is a truncated octahedron, placing a node at each lattice point in a BCC lattice also solves the problem. However, when $r_c/r_s \ge 4/\sqrt{5}$, the above solution no longer satisfies the full connectivity constraints. A generalized solution that works for all values of r_c/r_s is as follows:

When $r_c/r_s \ge 1.587401$, create a tessellation of truncated octahedron virtual cells of radius $R = \min\left(r_c\sqrt{5}/4, r_s\right)$ and then place a sensor node at the center of each virtual

cell. When the value of r_c/r_s is less than 1.211414, tessellate the 3D space into regular cube virtual cells, where the radius of a cell is $R = \min\left(\sqrt{3}r_c/2, r_s\right)$ and place a sensor node at the center of each virtual cell. Finally, when the value of r_c/r_s falls between these two thresholds (i.e., $1.587401 > r_c/r_s \geq 1.211414$), then create a hexagonal prism tessellation of a 3D space, such that each side of the hexagon is $a = \min\left(r_c/\sqrt{3}, r_s\sqrt{2}/\sqrt{3}\right)$ and the height of each hexagonal prism is $h = \min\left(2\sqrt{r_s^2 - a^2}, r_c\right)$, and then place a node at the center of each virtual cell.

When the ratio of communication range and sensing range is sufficiently high (i.e., $r_c/r_s \geq 4/\sqrt{5}$), ensuring full coverage automatically provides full connectivity with all neighboring nodes. Consequently, the overhead of full connectivity is zero. However, if the ratio is smaller, a placement strategy that provides full coverage with the minimum number of nodes no longer provides full connectivity. In fact, the overhead can be significant if the ratio is much smaller than $4/\sqrt{5}$. In this type of network, relaxing the requirement of full connectivity with all neighbors can save a significant number of nodes. For this network, finding the node placement strategy that provides full coverage and 1-connectivity with the minimum number of nodes is an important problem. The following strip-based node placement strategy solves this problem:

Deploy nodes as strips, such that the distance between any two nodes in a strip is $\alpha = \min\left\{r_c, 4r_s/\sqrt{5}\right\}$ and keep the distance between two parallel strips in a plane as $\beta = 2\sqrt{r_s^2 - (\alpha/4)^2}$. Set the distance between two planes of the strips as $\beta/2 = \sqrt{r_s^2 - (\alpha/4)^2}$ and deploy strips such that a strip of one plane is placed between two strips of a neighboring plane. Unless $\beta \leq r_c$ or, $\sqrt{\beta^2/2 + \alpha^2/4} \leq r_c$, this strip-based approach only ensures connectivity among nodes in the same strip. In order to ensure connectivity between strips, additional nodes are placed between

strips.

Since energy efficiency is an important consideration for any sensor network, different network topologies for node placement are compared in terms of their energy efficiency. The truncated octahedron model of node placement, which minimizes the number of nodes, is found to require more energy per node, as the transmission range is higher in this network topology. However, the overall energy consumption of the entire network is lower as the number of nodes is significantly smaller in the truncated octahedron model.

In many practical scenarios, it is not possible to deploy and maintain every node in a desired location. We provide a highly distributed and scalable solution to maintain full coverage and connectivity in networks faced with these issues. The idea is to overcome these limitations by randomly deploying a large number of redundant sensor nodes so that any point is within the sensing range of at least one sensor node. However, if all redundant nodes remain active at all times, energy efficiency suffers. Energy efficiency can be improved by partitioning the 3D network space into identical cells, such that only one node remains active in each cell, while full coverage and connectivity is still maintained. We investigate the problem for 3D networks and provide a distributed solution that uses only local information. We then solve an extension of the problem, called *k*-coverage, where the application requires that any point in the network must be within the sensing range of at least *k*-nodes with high probability. We then compare our scheme with a super optimal scheme, where an oracle can decide where to place sensor nodes using global information. We call this oracle scheme *SuperOpt*, because it does not have any constraints on node location and a node can be deployed at any position. Consequently, *SuperOpt* may require

fewer nodes than the optimal scheme, where nodes are randomly deployed. Our analysis shows that the number of extra nodes needed by our scheme decreases significantly for larger values of k. For example, the number of active nodes needed in our scheme is eight times the number nodes needed by *SuperOpt* for 1-coverage. However, our scheme achieves 4-coverage with a probability of 0.9971 with only twice the number of nodes needed for *SuperOpt*. Note that our scheme is totally distributed, whereas *SuperOpt* must be centralized where an oracle decides where to put a node.

REFERENCES

[1] I. F. Akyildiz, D. Pompili, and T. Melodia, Underwater Acoustic Sensor Networks: Research Challenges, *Ad Hoc Networks Journal*, *(Elsevier)*, March 2005.

[2] S. M. N. Alam and Z. J. Haas, Coverage and Connectivity in Three-Dimensional Networks, In *Proc of ACM MobiCom*, 2006.

[3] S. M. N. Alam and Z. J. Haas, Coverage and Connectivity in Three-Dimensional Underwater Sensor Networks, *Wireless Communication and Mobile Computing (WCMC)*, Volume 8 Issue 8, pages 995-1009, October, 2008.

[4] Aristotle, *On the Heaven*, Vol. 3, Chapt. 8, 350BC.

[5] X. Bai, S. Kumar, Z. Yun, D. Xuan, and T. H. Lai, Deploying Wireless Sensors to Achieve Both Coverage and Connectivity, In *Proc. of ACM MobiHoc*, 2006.

[6] E. S. Barnes and N. J. A. Sloane, The Optimal Lattice Quantizer in Three Dimensions, *SIAM J. Algebraic Discrete Methods* 4, 30-41, 1983.

[7] S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward, A distance routing effect algorithm for mobility (DREAM), in *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom)*, 1998, pp.76–84.

[8] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, *Routing with guaranteed delivery in ad hoc wireless networks*. Wireless Networks, 7(6):609--616, 2001.

[9] J. Carle, J.F. Myoupo, and D. Semé, A Basis for 3-D Cellular Networks. In *Proc. of the 15th International Conference on Information Networking*, 2001.

[10] K. Chakrabarty, S. S. Iyengar, H. Qi, and E. Cho, Grid coverage for surveillance and target location in distributed sensor networks. *IEEE Transactions on Computers,* 51(12):1448–1453, December 2002.

[11] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. *Wireless Networks*, 8(5), 2002.

[12] W. Cheng, A.Y. Teymorian, L. Ma, X. Cheng, X. Lu and Z. Lu, Underwater Localization in Sparse 3D Acoustic Sensor Networks, pp.236-240, *INFOCOM 2008.*

[13] T. Couqueur, V. Phipatanasuphorn, P. Ramanathan, and K. K. Saluja, Sensor deployment strategy for target detection. In *Proceeding of The First ACM International Workshop on Wireless Sensor Networks and Applications*, pages 169–177, Sep 2002.

[14] S. Datta and T. Woody, Go Green, Get Rich: 8 Technologies to Save the World, *Business 2.0 Magazine*, January 2007. Available online at http://money.cnn.com/galleries/2007/biz2/0701/gallery.8greentechs/2.html

[15] C. Decayeux and D. Semé, A New Model for 3-D Cellular Mobile Networks*, ISPDC/HeteroPar* 2004.

[16] S. Durocher, D. Kirkpatrick and L. Narayanan, On Routing with Guaranteed Delivery in Three-Dimensional Ad Hoc Wireless Networks, *ICDCN 2008*.

[17] D. Estrin (Editor), *Embedded, Everywhere: A Research Agenda for Networked Systems of Embedded Computers* (Report of National Research Council Committee on Networked Systems of Embedded Computers). National Academy Press, Washington, DC, USA, 2001.

[18] R. Flury and R. Wattenhofer, Randomized 3D Geographic Routing, *INFOCOM 2008.*

[19] M. Gardner, *The Sixth Book of Mathematical Games from Scientific American*, Chicago, IL: University of Chicago Press, 1984.

[20] T. C. Hales, A Proof of the Kepler Conjecture. *Ann. Math.* 162, 1065-1185, 2005

[21] D. Hilbert and S. Cohn-Vossen, *Geometry and the Imagination.* New York: Chelsea, 1999.

[22] J. Heidemann, W. Ye, J. Wills, A. Syed, and Y. Li, Research Challenges and Applications for Underwater Sensor Networking*, IEEE Wireless Communications and Networking Conference*, p. to appear. Las Vegas, Nevada, USA, IEEE. April, 2006.

[23] N. W. Johnson, *Uniform Polytopes.* Cambridge, England: Cambridge University Press, 2000.

[24] R. Jain, A. Puri, and R. Sengupta, Geographical routing using partial information for wireless ad hoc networks, *IEEE Personal Communications*, pp.48–57, Feb.2001.

[25] H. Karl and A. Willig, *Protocols and Architectures for Wireless Sensor Networks*, John Wiley & Sons Ltd, 2006.

[26] B. Karp and H. T. Kung, GPSR: Greedy perimeter stateless routing for wireless networks, *ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom)*, 2000, pp.243–254.

[27] R. Kershner. The Number of Circles Covering a Set. *American Journal of Mathematics*. 61:665-671, 1939.

[28] Y. Ko and N. H. Vaidya, Location-aided routing (LAR) in mobile ad hoc networks, *ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom)*, 1998, pp.66–75.

[29] J. Kong, J. Cui, D. Wu and M. Gerla, Building Underwater Ad-hoc Networks and Sensor Networks for Large Scale Real-time Aquatic Applications, *IEEE Military Communications Conference (MILCOM'05),* October 17-20, 2005. Atlantic City, New Jersey, USA.

[30] M. Křížek, Superconvergence phenomena on three-dimensional meshes, *International Journal of Numerical Analysis and Modeling*. Vol. 2, No. 1, 2005, pp. 43-56.

[31] R. Lei, L. Wenyu and G. Peng, A coverage algorithm for three-dimensional large-scale sensor network, *ISPACS*, pp 420-423, 2007.

[32] D. Li, K. Wong, Y. H. Hu, and A. Sayeed, Detection, classification and tracking of targets in distributed sensor networks, *IEEE Signal Processing Magazine*, 19(2), Mar. 2002.

[33] N. Lynch, *Distributed Algorithms*, Morgan Kaufmann Publishers, Wonderland, 1996.

[34] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava. Coverage problems in wireless ad-hoc sensor networks. *INFOCOM'01*, pages 1380–1387, 2001.

[35] D. Pompili and T. Melodia, Three-dimensional Routing in Underwater Acoustic Sensor Networks, in *Proc. of ACM PE-WASUN*, Montreal, Canada, October 2005.

[36] D. Pompili, T. Melodia and I. F. Akyildiz, Three-dimensional and two-dimensional deployment analysis for underwater acoustic sensor networks, *Ad Hoc Networks*, 7(4), 778-790, June 2009.

[37] T. S. Rappaport, *Wireless Communications: Principles and Practice*, Prentice Hall, 2002.

[38] H. Steinhaus, *Mathematical Snapshots*, 3rd edition, Oxford University Press, 1969.

[39] M. Stojanovic, Design and Capacity Analysis of Cellular Type Underwater Acoustic Networks, *IEEE Journal of Oceanic Engineering*, 33(2), pp.171-181, April 2008.

[40] M. Stojanovic, Frequency Reuse Underwater: Capacity of an Acoustic Cellular Network, *ACM WUWNeT*'07, Montreal, Canada, September 2007.

[41] W. Thomson (Lord Kelvin), On the division of space with minimum partition area. *Philosophical Magazine*, 24 (1887)503-514.

[42] D. Tian and N. D. Georganas, A coverage-preserved node scheduling scheme for large wireless sensor networks. In *Proceedings of First International Workshop on Wireless Sensor Networks and Applications (WSNA'02)*, pages 169–177, Atlanta, USA, Sep 2002.

[43] P. Varshney. *Distributed Detection and Data Fusion*. Spinger-Verlag, New York, NY, 1996.

[44] I. Vasilescu, K. Kotay, D. Rus, M. Dunbabin and P. Corke, Data Collection, Storage, and Retrieval with an Underwater Sensor Network, *SenSys'05,* November 2–4, 2005, San Diego, California, USA.

[45] D. Weaire, *The Kelvin Problem: Foam Structures of Minimal Surface Area.* London: Taylor and Francis, 1996.

[46] D. Weaire and R. Phelan, A Counter-Example to Kelvin's Conjecture on Minimal Surfaces. *Philosophical Magazine Letters*, 69, 107-110, 1994.

[47] D. Wells, *The Penguin Dictionary of Curious and Interesting Geometry,* London: Penguin, 1991

[48] H. Weyl, *Symmetry.* Princeton, NJ: Princeton University Press, 1952.

[49] G. Xing, C. Lu, R. Pless, and Q. Huang, On Greedy Geographic Routing Algorithms in Sensing-Covered Networks, In *Proc. of MobiHoc'04*, Tokyo, Japan, May 2004.

[50] G. Xing, X. Wang, Y. Zhang, C. Lu, R. Pless and C. D. Gill, Integrated Coverage and Connectivity Configuration for Energy Conservation in Sensor Networks, *TOSN*, Vol. 1 (1), 2005.

[51] Y. Xu, J. Heideman and D. Estrin. Geography-informed energy conservation in ad hoc routing. In *Proc. of the 7th ACM MobiCom*, July, 2001.

[52] T. Yan, T. He, and J. A. Stankovic, Differentiated surveillance for sensor networks, *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*, 2003.

[53] F. Ye, G. Zhong, S. Lu, and L. Zhang, Peas: A robust energy conserving protocol for long-lived sensor networks, *23rd International Conference on Distributed Computing Systems (ICDCS'03)*, pages 169–177, May 2003.

[54] M. Younis and K. Akkaya, "Strategies and techniques for node placement in wireles sensor networks: A survey", *Ad Hoc Networks* 6 (2008) 621–655.

[55] H. Zhang and J. C. Hou, Maintaining sensing coverage and connectivity in large sensor networks, *Wireless Ad Hoc and Sensor Networks: An International Journal*, vol. 1, no. 1-2, pp. 89-124, 2005.