國立交通大學

網路工程研究所

碩 士 論 文

將二度空間路由應用至三度空間之方法與條件

Slab Routing: Adapting Two-Dimensional Geographic Routing to Three-Dimensions

研 究 生：蔣易杉

指導教授：彭文志　教授

中 華 民 國 九 十 七 年 八 月

將二度空間路由應用至三度空間之方法與條件
Slab Routing: Adapting Two-Dimensional Geographic Routing to Three-Dimensions

研 究 生：蔣易杉　　Student：Paul I-Shan Chiang

指導教授：彭文志　　Advisor：Wen-Chih Peng

國 立 交 通 大 學

網 路 工 程 研 究 所

碩 士 論 文

A Thesis

Submitted to Institute of Network Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

August 2008

Hsinchu, Taiwan, Republic of China

中華民國九十七年八月

將二度空間路由應用至三度空間之方法與條件

學生：蔣易杉　　指導教授：彭文志 教授

國立交通大學網路工程研究所碩士班

摘　　要

地理位址路由是一種非常適合應用於無線隨意網路之路由方法，也在二度空間的環境中有許多深入的研究。然而針對三度空間中的地理位址路由提案卻寥寥無幾。在這篇論文中，我們發展出 Slab Routing，利用投影的方式將二度空間的地理路由演算法延伸至三度空間，並保留其原有的短路徑優點。Slab Routing 的做法是即時的切割出一個稱作 slab 的空間，將在其中之點投影至一個平面後擷取平面圖，再執行二度空間的地理路由演算法。雖然 Slab Routing 的操作方式無法確保所有訊息都可以抵達其目的地，我們推導出一個可以根據網路部署環境預測傳送機率的數學模型，並以實驗驗證之。

Slab Routing: Adapting Two-Dimensional Geographic Routing to Three-Dimensions

Student：Paul I-Shan Chiang　　Advisor：Prof. Wen-Chih Peng

Institute of Network Engineering
National Chiao Tung University

Abstract

Geographic routing, an attractive routing solution for wireless ad hoc networks, has been studied extensively in two-dimensional environments. However, three-dimensional space has proven to be much more challenging, with only a few existing proposals. In this paper, we present Slab Routing - a novel projection based method for adapting two-dimensional geographic face routing techniques to three-dimensional space, avoiding flooding and preserving their route optimality properties. Slab Routing accomplishes this by executing face routing over the planar projected graph of nodes contained within a dynamically created space partition, called a slab. While the adaption does not offer guaranteed delivery, we provide an analysis of the conditions required to achieve a desired delivery probability and verify the results through simulation.

誌　　謝

這篇論文的完成，首先要感謝彭文志教授所給予的指導與機會，不僅僅提升整體論文的品質更帶領我了解到整個研究過程。其次要感謝口試委員蔡明哲教授與黃俊龍教授，在口試的時候，提供了許多寶貴的意見。也感謝實驗室的同學們；學長蕭向彥給予的最初指導與許多忠告，同學駱嘉濠、郭員榕、蔡尚樺、傳道揚的陪伴與協助、討論與分享，以及博士班學長姐洪智傑、江孟芬、魏綾音的關懷與照顧。在實驗室外還有許多許多的人們，請原諒我不一一道出，感謝你們個個獨特的協助。最後但非最少，要感謝我的父母，始終站在我的背後支持我，給予我莫大的精神力量。因為有你們的付出，讓我能無後顧之憂的專注於研究，真的很感謝您們所作的一切。

Contents

List of Figures

Chapter 1

Introduction

Geographic routing, the forwarding of messages using only the position information of the destination and nodes in the local neighborhood, is a very attractive class of routing algorithms for MANETs and wireless sensor networks. Traditional forwarding-table based methods are not suitable for such wireless ad hoc networks due to the stringent memory and energy limitations of the nodes. While geographic routing has been studied extensively in two-dimensions, with many proposals such as GPSR [16] and GOAFR [18], scant few results have been published for three-dimensional space. Yet, with the growing maturity of wireless sensor technology and the community's experience with them, researchers are moving on from simple terrestrial deployments to more complicated and challenging environments that cannot be adequately modeled as a two-dimensional plane, such as underwater deployments [13], atmospheric projects [4], and indoor applications [20, 23]. As a consequence, the need for new geographic routing algorithms that are able to operate in three-dimensional space is ever increasing.

The concept of geographic routing was first proposed by [10], using a greedy forwarding strategy that was susceptible to getting stuck in local minima where a node does not have any neighbor that is closer to the destination than itself. This problem was solved with the invent of face routing, which forwards messages along

the faces of a planar subgraph and guarantees delivery if a path exists. Combining greedy forwarding with face routing produced the Greedy-Face-Greedy paradigm, routing greedily whenever possible and using face routing to escape local minima. The result was an approach that not only requires very little routing state, but is also energy efficient in that the message follows a single path to the destination (i.e. no flooding or extra copies are required) and that the path is competitive with the shortest path. The challenge of extending two-dimensional geographic face routing algorithms to three-dimensions lies in face routing's dependence on a planar graph; there is no direct analogy to planar graphs and their faces in three-dimensional space. As early as in the proposal of GPSR, the first Greedy-Face-Greedy protocol, the idea of extending geographic face routing to three-dimensional space was mentioned as a future work [16]. However, an acceptable solution has eluded researchers for all these years.

Recently, it was proven that for general three-dimensional graphs, a deterministic routing algorithm which uses only local information and which guarantees delivery does not exist [9]. Following this result, the first randomized algorithm utilizing a random walk strategy appeared in [11]. Random walks tend to suffer from inefficiency, especially when node density is high. A density control algorithm coined Dual Graph was proposed in the same paper to be used in conjunction, yet even so the cost is considerable. Prior to appearance of the proof of non-existence, there were several attempts to devise a deterministic routing algorithm, such as the works of Opatrny et al. [2, 3, 14, 15]. Their methods revolved around the concept of projecting the three-dimensional network onto a two-dimensional plane determined using various heuristics. However, the majority of their algorithms resort to duplicate message copies and/or limited flooding, and thus sacrificing energy efficiency. In contrast to the above approaches, this paper seeks neither to devise a new randomized algorithm nor attempt to accommodate all networks, but instead

we tackle the question: can we adapt two-dimensional geographic face routing algorithms to three-dimensional space while preserving their desirable route optimality properties, and under what network conditions will this adaption be able to achieve a desired delivery rate?

To operate two-dimensional face routing in three-dimensional space, we propose a novel projection-based local minima escape strategy called Slab Routing. When attempting to recover from a local minimum, Slab Routing dynamically creates a space partition called a slab. Neighboring nodes contained within the slab are projected onto a plane and face routing is invoked on the projected graph. Slab Routing differs from the projection based methods of [2,3,14,15] in two important aspects. First, only one projection plane and one message copy is ever used. Second, it is always possible to extract a planar graph from the projected graph. In contrast to the random walk method of [11] which performs well on sparse topologies, Slab Routing requires a sufficiently dense network in order to deliver the majority of messages. To answer the question of what density is "sufficient", we draw upon results from the study of topology control to derive a mathematical model for the relationship between density and delivery probability.

The rest of the paper is organized as follows. Section 2 provides a more detailed look into the related work and covers the required preliminaries. Section 3 describes our method for adapting two-dimensional algorithms to three-dimensional space. Section 4 answers the question of what network conditions can provide a given delivery rate. Section 5 verifies our analysis with simulation results. Finally, Section 6 concludes this paper.

Chapter 2

Preliminaries and Related Work

2.1 Geographic Routing

We define a *geographic routing algorithm* as an algorithm that is limited to the following information when nodes make their forwarding decisions:

1. The node's own position
2. The positions of the node's neighbors
3. The position of the destination
4. Control information about at most $O(1)$ nodes contained in the message

Nodes do not maintain any state information about messages that they process. It is also assumed that the sender of a message is able to obtain the location of the desired destination (via mechanisms such as location databases [19]).

The earliest proposal of geographic routing was of a purely greedy nature [10]. In its simplest form, messages were forwarded to a neighbor that is closer to the destination than the current node. Aside from measuring closeness using Euclidian distance, alternative approaches, such as using angles [17], were also proposed. However, regardless of the distance measure employed, all greedy-only approaches

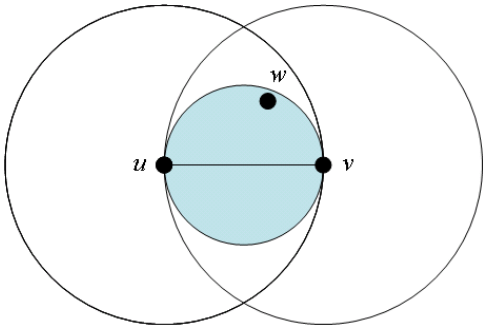


Figure 2.1: The Gabriel Graph. The edge (u, v) is removed if there is a witness node w in the shaded circle.

are vulnerable to falling into *local minima* where a neighbor that is closer to the destination cannot be found. Despite this defect, greedy geographic routing schemes were recognized to have the advantage of requiring little forwarding state information and yet arrive at the destination using a close to optimal path when they succeed.

In the search for a geographic routing algorithm that guarantees delivery, *face routing* was proposed in [17]. In face routing, a planar subgraph is first extracted from the network graph. Then, the face that has the current node as one of its vertices and which intersects with the line connecting the source and the destination is found. The message is then forwarded in a particular direction (decided for example using the right hand rule [16]) along the edges that form the perimeter of the face. After gathering enough information about the current face, the message switches to the next face and the process is repeated until the destination is reached. The success of face routing is dependant on the planarity of the graph that it routes over.

The most commonly used planar graph is the Gabriel Graph, which can be computed locally at each node. In the Gabriel Graph, the edge (u, v) between nodes u and v is retained if no other node w is present within the circle whose diameter is $\overline{uv}$. See Figure 2.1 for an illustration.

Combining greedy geographic routing with face routing yielded a model that later became known as Greedy-Face-Greedy - the message is forwarded greedily until a local minimum is encountered, whereupon the packet switches to face routing to escape. The packet later returns to greedy forwarding when a node that is closer to the destination than the point where the local minimum escape began is arrived at (we will refer to this as meeting the *return to greedy criterion* in the rest of this paper). Various refinements and studies on the basic protocol in GPSR [16] ensued. For example, GOAFR [18] limits the exploration area during face routing to obtain asymptotically optimal delivery times.

Of particular relevance to our work is the extension of the network model from Unit Disc Graphs (abbr. UDG) to quasi-UDG [5]. In the traditional UDG model, all nodes are assumed to have the same transmission radius, normalized to one. A node is connected to another node if and only if the distance between them is less than or equal to one unit distance. Throughout the rest of this paper, all distances are expressed in this normalized form - one unit distance refers to the maximum transmission radius of the nodes. To deal with heterogeneous transmission ranges or simply the irregularities often found in real radio hardware, the QUDG model allows a "uncertain" region wherein links possibly exist. Formally, in a QUDG model with parameter d ($0 \leq d \leq 1$), the edge (u, v) between two nodes u and v exists if and only if $\| u - v \| \leq d$ and does not exist if and only if $\| u - v \| > 1$. For nodes with distance $d < \| u - v \| \leq 1$, no conclusion can be drawn and they might or might not be connected. Face routing using the QUDG model with $d \geq 1/\sqrt{2}$ was also presented in [5], introducing an extra pre-processing phase which adds virtual edges to nodes in the uncertain region that will affect the result of the planar graph extraction process. Correctness is guaranteed only for d values greater than or equal to $1/\sqrt{2}$ because $1/\sqrt{2}$ is the minimum radius required to detect all crossing edges locally. Interested readers are referred to [5] for a detailed proof. Figure 2.2 is

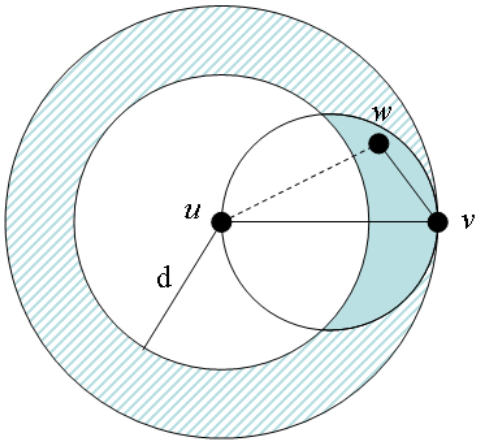


Figure 2.2: An example of the QUDG model.

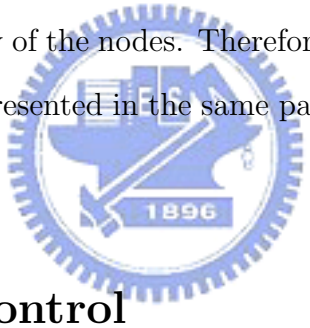
an illustration of the QUDG model and virtual edge addition. In Figure 2.2, the shaded region is an uncertain region where nodes are not necessarily connected to u. Also in figure 2.2, the node w will cause the edge (u, v) to be removed during planar graph extraction but u may not necessarily know of the existence of w, and therefore the virtual edge (u, w) must be added.

Moving on to algorithms for three-dimensional space, a series of works by Opatrny et al. use the idea of a projection plane [2, 3, 14, 15]. Nodes in the network are projected onto a given plane, on which face routing is performed. Various heuristics for selecting the projection plane were proposed, including using any arbitrary plane containing the line from the source to the destination [14], or the plane defined by the triple {forwarding node, destination, next hop} [1], or the xy, yz, xz coordinate planes [2], or the least squares projection plane of the current node's neighbors [15]. However, there is no clear winner amongst these heuristics, and techniques such as multiple projection planes and/or message duplication/flooding are employed to increase delivery rate.

In a recent paper, Durocher et al. proved that a deterministic routing algorithm which uses only local information and that guarantees delivery for general three-dimensional networks does not exist [9]. The proof has two parts. First, it is shown that if the height of the three-dimensional network does not exceed $1/\sqrt{2}$ a local

algorithm does exist. Then, they prove that for networks with height $1/\sqrt{2} + \epsilon$ where ϵ is any constant greater than zero, if a k-local routing algorithm which succeeds for any graph exists then a 1-local routing algorithm that succeeds for any connected graph exists. However, it is easily shown that such a 1-local routing algorithm will fail on even simple graph topologies and therefore a k-local algorithm does not exist.

Given the non-existence of deterministic algorithms for general graphs, using random walks to escape local minima was proposed [11]. The message is forwarded to a random neighbor until the return to greedy criterion is met. Similar to the idea of GOAFR, the random walk exploration is confined to a limited, exponentially expanding volume. The amount of random walk steps required to probabilistically cover all the nodes in each exploration region is dependant on the number of nodes in the region, i.e. the density of the nodes. Therefore, a topology control algorithm called Dual Graph, is also presented in the same paper to obtain a desirable graph density.



2.2 Topology Control

Before we delve into the details of our algorithm, we first provide a brief introduction to topology control, whose results we draw upon to analyze our algorithm's delivery rate under different conditions.

When studying wireless ad hoc networks, the network is typically modeled as a *geometric random graph* [8, 22]. A geometric random graph $G(n, r)$ is a graph where n nodes are independently and uniformly randomly distributed in a metric space and where two distinct nodes u and v have an edge (u, v) connecting them if and only if the distance $\| u - v \|$ between them is smaller or equal to r.

Given an input graph, topology control seeks to find a (minimal) subgraph that possesses certain properties. Of the properties that might be of interest, one

such is connectivity. A graph is said to be connected if for every pair of nodes there exists a path between them. In particular, we are interested in the range assignment problem for connectivity which asks: *given a set of nodes V, what is the minimum transmission radius for each node such that the induced graph is connected?* There are further two major variations of the range assignment problem - the homogeneous range assignment and the k-neighbor range assignment. In a homogeneous range assignment r, every node sets its transmission range to the same value r. Prominent results include [6, 12, 21]. On the other hand, in a kth-neighbor range assignment k, each node sets its transmission range to $max(1, d_k)$ where d_k is the distance to the kth-nearest-neighbor. An example of a study dealing with this problem is [25].

Chapter 3

Slab Routing

We now describe Slab Routing, our adaption of two-dimensional Greedy-Face-Greedy geographic routing algorithms to three-dimensional space. We assume a connected input graph in three-dimensional space adhering to the Unit Ball Graph model, which is the three-dimensional version of UDG. That is, the edge (u, v) exists if and only if the Euclidean distance $\| u - v \| \leq 1$. We call the nodes u and v that have an edge connecting them in the UBG *neighbors*.

Extending the greedy forwarding component is trivial - the two most common distance measures, Euclidean and angle, are both easily extended to three-dimensions. However, as mentioned previously, there is no direct analogy to planar faces in three-dimensions. To overcome this problem, our local minima escape strategy has the following steps:

1. Define a *forwarding slab*
2. Project nodes that are within the forwarding slab onto a *forwarding plane*
3. Extract the planar graph from the projected graph
4. Invoke a two-dimensional face routing algorithm

We shall now elaborate on the first three operations.

A slab is defined as the volume between two parallel planes. The thickness of the slab is the minimum distance between the planes. From [9] we have the following lemma:

Lemma 3.0.1 *Choose any $\lambda \leq 1$ and let P denote a set of points in R^3 contained in a slab of thickness λ. Let $f : R^3 \rightarrow R^2$ denote the projection onto a plane parallel to the slab. Let $G = (V, E)$ denote the embedded graph such that $V = f(v)|v \in P$ and $E = f(u), f(v)| \parallel u - v \parallel \leq 1, u, v \in P$ (V and E may be multisets). G is a $(\sqrt{1-\lambda^2})$-quasi unit disk graph.*

Since only the subset of nodes that are contained within the forwarding slab will be eligible for forwarding, intuitively, maximizing the volume of the slab will maximize the chances of successful and efficient delivery. However, the minimum value of the parameter d for which we can guarantee the correct extraction of a planar graph from is $1/\sqrt{2}$ [5]. Therefore, trying to maximize λ while observing that $\sqrt{1-\lambda^2} \leq 1/\sqrt{2}$, we set the thickness of the forwarding slab to $1/\sqrt{2}$.

When a message encounters a local minimum, the node at which it occurs must select one of the infinite possible slabs available as the forwarding slab and communicate its choice to subsequent nodes. In order to guarantee the correctness of face routing, it is imperative that all nodes involved in a particular local minimum escape attempt use the same slab. While each node along route does not necessarily have to use the same projection plane, if a deterministic relation between the plane and the slab is agreed upon, communicating the details of the plane would be sufficient for other nodes to infer the slab. Therefore, for simplicity, we define the forwarding plane to be the plane that is equidistance from the two planes which define the slab. The task of selecting a slab is now analogous to choosing a forwarding plane. Obviously, the destination and the current node must lie within the slab for a path between the two to exist. It is further advantageous to let the destination lie on the forwarding plane as information about the destination is a

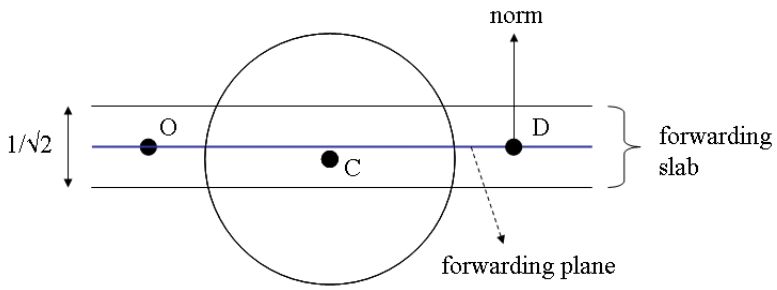


Figure 3.1: Side on view of the forwarding slab and forwarding plane. The origin O and the destination D are always on the forwarding plane while the current node C is not necessarily on it but is always within distance $1/2\sqrt{2}$.

necessary part of any message under any protocol, and coupling the destination with a norm vector we can communicate the chosen forwarding slab with minimal additional cost. The origin of the coordinate system is also a desirable point for the forwarding plane to pass through as it will simplify several calculations. Note that while it is convenient for the origin to be included, it is not necessary. Any arbitrary point that all nodes agree upon will suffice - the effect is the same as translating the coordinate system to said arbitrary point. The natural choice for the third point needed to define the forwarding plane is the current node. However, should the origin, the destination, and the current node lie upon a line, a random point within $1/2\sqrt{2}$ of the current node (so as to ensure that the current node is contained in the slab) should be chosen instead of the current node. See Figure 3.1 for an illustration of the forwarding slab selection process.

After defining a forwarding slab, a node projects those neighbors which are within the forwarding slab onto the forwarding plane. Since the projected graph is a QUDG, a graph completion phase similar to the one in [5] is required. The purpose of the completion phase is to ensure that both nodes at the ends of an edge will agree on whether to keep or remove the edge during the planar graph extraction phase. Disagreement can occur under the QUDG model when the witness node lies in the uncertain region of one of the end points (the dark shaded area in Figure 2.2). In the completion phase of [5], each node checks each incident edge for the existence

of witness nodes that will result in the removal of that edge. If a witness node is found, it is reported to the neighbor at the other end of the edge to ensure that the neighbor also knows of the existence of the witness node. When a node learns of a previously unknown witness node, it establishes a virtual edge to the witness node by adding the pair {witness node, informer node} to its *adjacency list*. After adding a virtual edge, the virtual edge is also checked for the existence of witness nodes. Forwarding messages over a virtual edge is done by sending the message via the informer node. While this graph completion strategy is efficient for two-dimensional planes as nodes only add virtual edges to nodes that will affect planar graph extraction, it is valid for only that particular plane. In our three-dimensional environment, there are an infinite amount of possible projection planes, and on each projection plane the length of a certain edge and the relative positions of nodes will differ. Thus, if we adopt the two-dimensional graph completion strategy, it will have to be executed for the forwarding plane of each message that the node processes. This however, is clearly infeasible as the message exchanges required for graph completion can result in a high communication overhead, and waiting for the procedure to finish would also incur a significant forwarding delay at each node. Instead, just as in two-dimensions, we would like to execute graph completion once and for all before any messages are routed.

In three-dimensions, after projection, the uncertain region of the QUDG model arises from vertical distance to the projection plane. Since the thickness of the forwarding slab is $1/\sqrt{2}$, the maximum distance between two nodes that might affect each other's decision over an edge is $\sqrt{1^2 + (1/\sqrt{2})^2} = \sqrt{3/2}$, see Figure 3.2. Therefore, if every node learns of all nodes within $\sqrt{3/2}$ that it shares a common adjacent node with, the planar graph computed by every node will be the same for a given forwarding plane. Thus the graph completion procedure executed at each node is as follows:

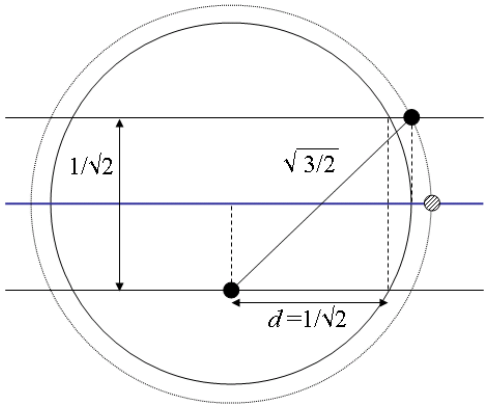


Figure 3.2: $\sqrt{3/2}$ is the maximum distance at which a witness node will affect planar graph extraction.

The adjacency list is initialized to contain all nodes that are in the neighbor list, and each neighbor is put into a processing queue. While the queue is not empty, dequeue the first node n and check for each node $a \neq n$ in the adjacency list whether $1 <\| n - a \| \leq \sqrt{3/2}$. For each pair of nodes (n, a) that satisfy the above condition, two "add virtual node" messages are sent - one to n and one to a, informing each of the other. Upon receiving an "add virtual node w" message from an adjacent node, if w is not already in the adjacency list, w is added to the adjacency list and pushed into the processing queue.

After constructing the adjacency list, when a node needs to forward a message over a given forwarding plane, all adjacent nodes that lie within the forwarding slab are projected. However, since the adjacency list was built to accommodate all possible slabs, some edges might have a projected length greater than one, e.g. the shaded node in Figure 3.2. These edges must be removed before extracting the underlying planar graph. We call the remaining set of nodes the *candidate list*.

In GPSR, the default criterion for returning to greedy forwarding mode is when the message arrives at a node that is closer to the destination than the node N_p where the message first entered face forwarding mode [16]. This ensures that the message will not return to previously visited nodes. However, in Slab Routing, the candidate list is only a subset of the neighbor list, and the alternative strategy

of returning to greedy forwarding when there is a neighbor that is closer to the destination than N_p should be employed in order to return to greedy mode as soon as possible.

Before we conclude this section, we first summarize Slab Routing. After the initial deployment of nodes, every node executes the graph completion protocol to construct their adjacency lists. Later, when a node attempts to forward a message but finds that the message has reached a local minimum, it selects a forwarding plane which is less than $1/2\sqrt{2}$ away from itself and which contains both the origin of the coordinate system and the destination of the message. The slab centered on the forwarding plane, i.e. the slab whose boundary planes are equidistance from the forwarding plane, is then the forwarding slab. Adjacent nodes that are within the forwarding slab are projected onto the forwarding plane, and those projected edges whose length is greater than one are removed to form the candidate list. The local planar graph is then extracted from the candidate nodes and a two-dimensional face routing algorithm is executed on the planar candidate node graph to determine the next hop. Similar to face routing, the message is marked as being in *slab mode* and in addition to the header fields required by the invoked face routing algorithm, an additional header field is transmitted along with the message - the norm vector of the forwarding slab. When a node receives a message that is in slab mode, it first checks whether it has a neighbor that satisfies the return to greedy criteria; if not, it then proceeds to forward the message using the two-dimensional face routing algorithm on the slab projection graph. Should the face routing algorithm report that the destination is unreachable, the message is dropped by Slab Routing.

Chapter 4

Analysis

In essence, Slab Routing extracts a two-dimensional planar graph from the three-dimensional input graph for face routing to run upon. We know that face routing is a correct algorithm given a planar graph - that is, the message will reach it's destination if a path to the destination exists. The question therefore, is whether the slab projection graph contains a path to the destination. While we can assume that the three-dimensional input graph is connected, there is no guarantee that the subset of nodes forming the slab projection graph is connected. This section investigates the problem: given what conditions will the subset of nodes in an arbitrary forwarding slab be connected with high probability?

To answer this question we turn to results from the study of the range assignment problem. Here we are not interested in finding a range assignment for connectivity but rather in the reverse question: given a set homogeneous transmission range (equal to one), how many nodes are required to produce a connected graph? Although there exists several variations of the basic range assignment problem and solutions for different settings, we adopt and refine the results of Bettstetter [6], which studies the homogeneous range assignment for general settings with assumptions similar to ours.

Bettstetter [6] derived a relationship between the minimum node degree d_{min},

node density ρ, and radio range r for two-dimensional graphs:

$$P(d_{min} \geq n_0) = \left(1 - \sum_{N=0}^{n_0-1} \frac{(\rho A)^N}{N!} \cdot e^{-\rho A}\right)^n \tag{4.1}$$

where $A = \pi r^2$ is the area covered by the radio range of a node. Using Penrose's theorem [21], it was also shown that for a random geometric graph G:

$$P(G\, is\, k - connected) = P(d_{min} \geq k) \tag{4.2}$$

for $P(d_{min} \geq k)$ almost one.

In the context of slab projection graph connectivity, we wish to find $P(d_{min} \geq 1)$ for a given ρ. However, the variables A and n in Equation (4.1) have different forms in three-dimensions. Firstly, in two-dimensions, the neighborhood area of a node u with radio range r is the circle πr^2 centered at u, i.e. the number of other nodes that lie within this area is the degree of u. Translated into three-dimensions, the neighborhood volume is the ball $\frac{4}{3}\pi r^3$ centered at u. However, during slab routing, the *effective* neighborhood volume is smaller - only those nodes that lie within the slab are projected. Thus, the effective neighborhood volume is the cylinder $\pi r^2 \lambda$, where λ is the thickness of the slab. Secondly, in the original context, n is the number of nodes being considered for connectivity, which is the total number of nodes in the system. During slab routing however, we wish only to find the probability of connectivity for a subset of nodes, i.e. those nodes that lie within the slab. Let c be the constant that represents the ratio of nodes we expect to find within an arbitrary slab to the total number of nodes in the system. Then, Equation (4.1) becomes:

$$P(d_{min} \geq 1) = \left(1 - e^{-\rho \pi r^2 \lambda}\right)^{cn} \tag{4.3}$$

We note that while this model calculates the probability that the slab graph is connected, during routing we are only interested in the existence of a path between a pair of nodes. Therefore, we expect that the actual probability of finding a path to the destination will be higher than the value arrived at using the above equation.

The constant c in Equation (4.3) depends on the dimensions of the deployment region. For irregular volumes this value can be difficult to calculate and would be best derived using computer experiments. To serve as an estimation, we provide calculations for a spherical deployment region of radius R below. Assuming uniform distribution of the nodes, the ratio of nodes within a slab to the total number of nodes is equivalent to the ratio of the volume of the slab to the total volume. The spherical region is perhaps the easiest to analyze since we can calculate the expected intersection volume for a particular fixed norm vector and know that the results will be the same for any other norm vector. Let the center of the sphere be the origin $(0, 0, 0)$ and let the x-axis be parallel to the arbitrary fixed norm vector. See Figure 4.1 and Figure 4.2 for an illustration. We can express the intersection volume of the slab centered at $(x, 0, 0)$ as:

$$V(x) = \begin{cases} \frac{4}{3}\pi R^3 - (\pi R h_l^2 - \frac{1}{3}\pi h_l^3) \\ \qquad\qquad R - \frac{1}{2\sqrt{2}} \leq x \leq R \\ \\ \frac{4}{3}\pi R^3 - (\pi R h_u^2 - \frac{1}{3}\pi h_u^3) - (\pi R h_l^2 - \frac{1}{3}\pi h_l^3) \\ \qquad\qquad -R + \frac{1}{2\sqrt{2}} < x < R - \frac{1}{2\sqrt{2}} \\ \\ \frac{4}{3}\pi R^3 - (\pi R h_u^2 - \frac{1}{3}\pi h_u^3) \\ \qquad\qquad -R \leq x \leq -R + \frac{1}{2\sqrt{2}} \end{cases} \tag{4.4}$$

where $h_u = R - 1/2\sqrt{2} - x$, $h_l = R - 1/2\sqrt{2} + x$, being the height of the upper

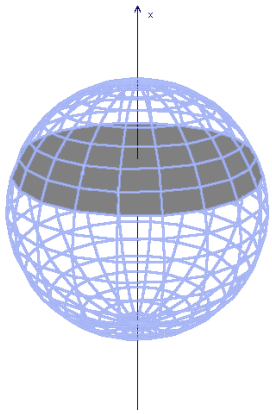


Figure 4.1: Intersection volume of a slab and a spherical deployment region.

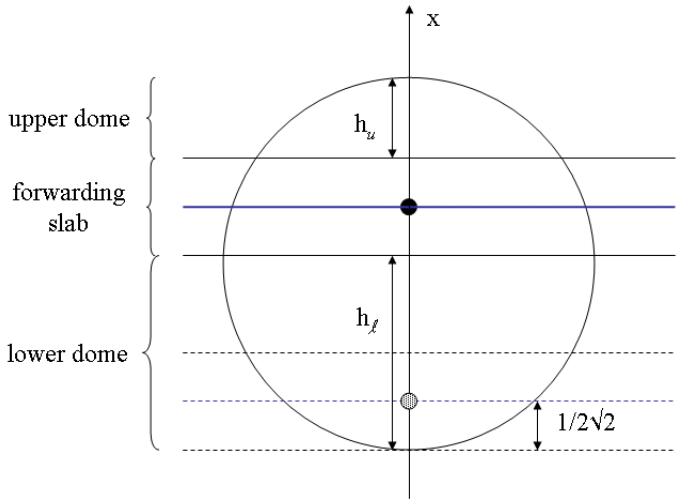


Figure 4.2: Side view of the intersection of a slab and a sphere. The slab centered at the solid point has both an upper dome and lower dome while the slab centered at the dotted point near the bottom does not have a lower dome.

and lower domes respectively. For slabs centered at the very top of the sphere ($R \geq x \geq R - 1/2\sqrt{2}$) and the very bottom of the sphere ($-R \leq x \leq -R + 1/2\sqrt{2}$), such as the dashed slab shown in Figure 4.2, there are no lower and upper domes respectively.

The slabs centered at any two points in the sphere that have the same x coordinate are identical and the area of the cross-sectional circle containing them is $A(x) = \pi(R^2 - x^2)$. Therefore, the expected slab intersection volume ratio to the

total deployment volume is:

$$\begin{aligned} c &= \left(\int_{-R}^{R} A(x)V(x)dx \right) / \left(\frac{4}{3}\pi R^3 \right) \\ &= \left(\int_{0}^{R} A(x)V(x)dx \right) / \left(\frac{2}{3}\pi R^3 \right) \\ &= \frac{\pi}{4} \left(\frac{8\sqrt{2}R^2}{5} - \frac{\sqrt{2}}{12} + \frac{1}{64R} - \frac{1}{15360R^3} \right) \end{aligned} \tag{4.5}$$

Using Equation (4.5), we can calculate an estimated value of c for a deployment region of volume V by assuming that the region is spherical, and in turn we can derive a value for n or $P(d_{min} \geq 1)$. For example, if we have a deployment region of one thousand cubic units, we can set $V = \frac{4}{3}\pi R^3 = 1000$ to find $R = \sqrt[3]{750/\pi}$ and thus $c \approx 0.0683$. Therefore, if there are a total of 5000 nodes in the deployment, the delivery probability is $P(d_{min} \geq 1) \approx 0.9949$.

Chapter 5

Simulation

We conducted a series of simulations using Sinalgo [24], to validate our analytical results. Sinalgo is a Java based simulation framework for wireless devices which focuses on testing and validating network algorithms, abstracting the underlying layers, and is thus suitable for our purposes. We set the simulation field to $10 \times 10 \times 10$ units and randomly placed ten void zones of random sizes ranging from $1 \times 1 \times 1$ to $2 \times 2 \times 2$ (not necessarily a cube) within the field. These void zones were regions where nodes could not be deployed and served the purpose of introducing network holes to the generated graph. Nodes were deployed randomly within the field but outside void zones and the largest connected component of each deployment was kept. Different densities were obtained by varying the number of nodes deployed.

For each network, 5000 messages with random sender/target pairs were generated, and up to 150 networks were simulated for each deployment density. The messages were routed through the network using Slab Routing and their final state - delivered or dropped, was recorded. We implemented OFR [18] as Slab Routing's two-dimensional face routing component and messages were dropped when OFR could not find a path to the destination on the forwarding slab. To measure our algorithm's performance, we define *delivery rate* to be the ratio $\parallel S_d \parallel / \parallel S \parallel$, where S is the set of messages that fell into a local minimum at least once during

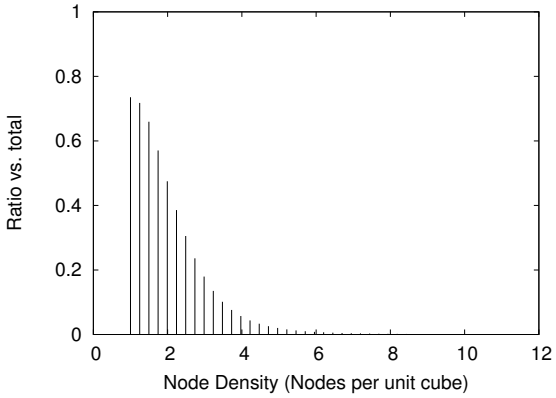


Figure 5.1: Ratio of messages that encountered at least one local minimum.

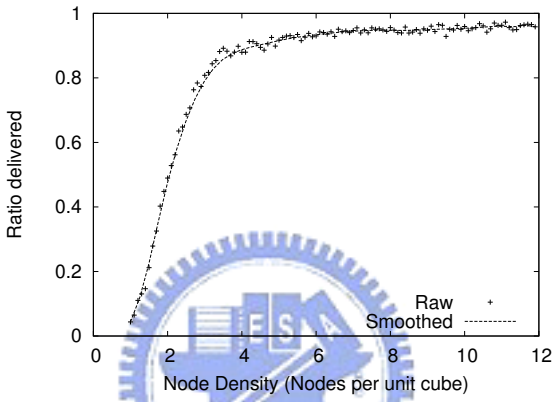


Figure 5.2: Delivery rate of Slab Routing under different densities.

forwarding and S_d is the subset of S that were successfully delivered to their destination. This measure ignores those messages that arrive at their destination using only greedy forwarding, which constitute the majority in higher density environments, as shown in Figure 5.1 which displays the relative size of the set S to the total number of messages.

Figure 5.2 plots the delivery rate under different deployment densities along with a smooth curve to approximate the data points.

Using Equation (4.5) with $V = \frac{4}{3}\pi R^3 = 10^3$ we have $c \approx 0.0683$. However, through repeated computer experiments we found that c for a deployment region with dimensions $10 \times 10 \times 10$ is approximately 0.0531. Figure 5.3 plots Equation (4.3) with $c = 0.0683$ and $c = 0.0531$ against the actual simulation results.

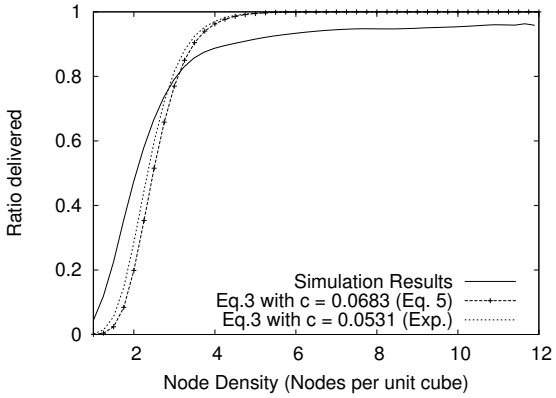


Figure 5.3: Theoretical delivery rate obtained with Equation (4.3).

From Figure 5.3 we observe that in our simulations the delivery rate converges to one at a much slower rate than it should in theory. While the analytical function rises smoothly to one, simulation results show that delivery rate reaches 0.9 at 3.5 nodes per unit cube, then grows slowly to 0.95 at 6.5 nodes per unit cube, and almost levels off after. Possible reasons include the fact that the nodes are not deployed perfectly uniformly in the region (especially with the artificial void zones), and the boundary effect of the deployment zone [7]. We also observe that the small difference in c value does not make a significant difference in the resulting curve.

Figure 5.4 verifies that our intuition in selecting the maximum thickness possible for the forwarding slab is correct. The delivery rates for slab thicknesses of $1/\sqrt{2}$, $1/\sqrt{3}$, and $1/\sqrt{5}$ are plotted.

We also conducted simulations with different deployment dimensions while holding the total volume constant in order to investigate the effect on delivery rate. Figure 5.5 plots the results obtained under dimensions of $20 \times 10 \times 5$ and $40 \times 5 \times 5$ with the original results. We observe that the the more symmetric the deployment region is, the better the delivery rate.

Finally, we measured the *transmission stretch* of Slab Routing. The transmission stretch of an algorithm is the factor by which the total cost incurred by an

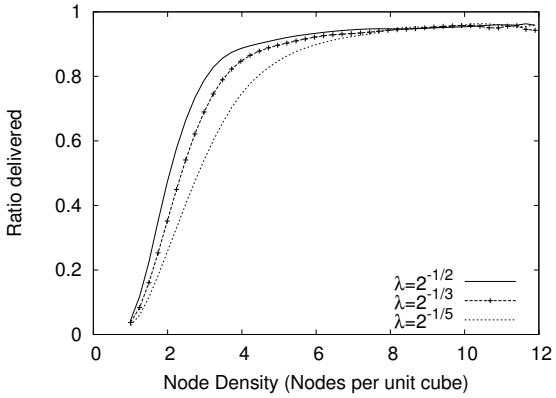


Figure 5.4: Effect of slab thickness on delivery rate.

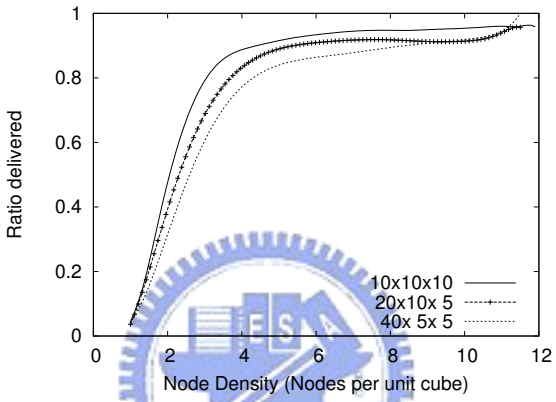


Figure 5.5: Effect of simulation field dimensions on delivery rate.

algorithm for routing a message to its destination exceeds the cost of the shortest path between the source and the destination. Here we use the hop metric in measuring the length of a path, as is common when studying wireless ad hoc networks. Figure 5.6 plots the average stretch of Slab Routing for several sets of messages. The solid line plots the average stretch of all messages. We observe that Slab Routing's stretch only slightly exceeds 2 at worst. We also notice that at the lowest density the measured stretch is lower than 1 - this is because at the lowest density hardly any messages are successfully delivered. Filtering out those messages that were not successfully delivered gives the second line. The third line further retains only those messages that encountered at least one local minimum. For this set of messages that had to escape from a local minimum at least once, Slab Routing has

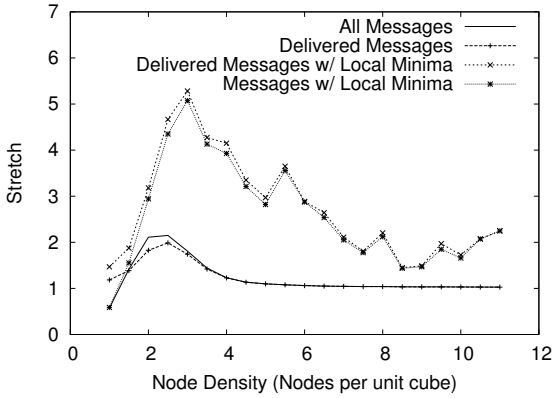


Figure 5.6: Routing stretch of Slab Routing.

a maximum stretch of slightly over 5. The last line includes those messages that encountered a local minimum but did not reach their destination. We notice that the average stretch for this set is lower than the previous set, implying that when Slab Routing is unable to deliver a message, it can quickly discover this fact and drop the message.

To give an idea of how Slab Routing's stretch compares with other algorithms, we implemented a bounded flooding local minima escape scheme (abbr. BF). On each simulated network, the exact same set of messages were routed through the network twice - once with Slab Routing and once using bounded flooding. When escaping a local minimum, bounded flooding increases it's flooding range exponentially until it finds a node that satisfies the local minimum escape criterion. The initial bound was set to 2 hops, the value that we found to yield the lowest stretch factor for BF. We note that bounded flooding will always succeed in delivering a message, but on the other hand it is also not a memoryless algorithm. Figure 5.7 plots the stretch of bounded flooding against Slab Routing. Two data plots for each algorithm are given: one with all messages, and one for the set of messages that fell into a local minimum and were delivered by Slab Routing. The stretch of BF for all messages starts out incredibly high and gradually declines as density increases until it approaches Slab Routing's stretch. This decline is mainly

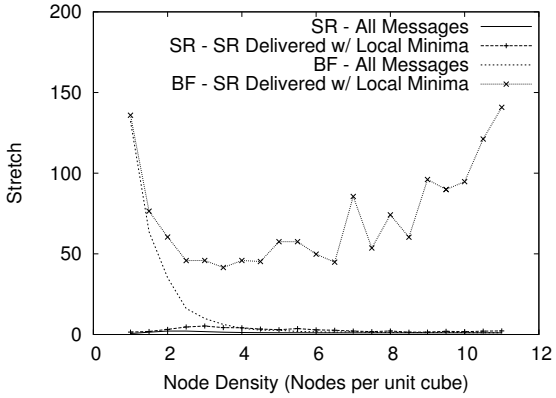


Figure 5.7: Slab Routing's stretch compared to Bounded Flooding.

because as density rises so does the probability that a message never falls into a local minimum increase. Examining only those messages that had to escape at least one local minimum shows that BF has a stretch of 40 to 140 for the simulated densities. Interestingly, the density at which BF performs best is approximately that at which Slab Routing shows it's worse performance. Yet, even at this density BF's stretch is still more than 8 times that of Slab Routing. The interested reader can compare these results to the simulation results of [11] which also uses bounded flooding as a basis for comparison. Their results show that random walk achieves at best a routing stretch half that of bounded flooding.

The above transmission stretch figures measure Slab Routing's performance while routing messages. In addition to the message overhead costs for routing a message, the completion protocol required for Slab Routing's operation after network deployment incurs an initialization cost as measured in Figure 5.8. As density increases, the number of nearby nodes that fall in the uncertainty region of the QUDG model increases, and therefore the average number of messages required for the completion phase increases. While the initialization overhead becomes quite significant as density increases, it is however a once per network deployment operation.

We were unable to present a formal analysis of the maximum number of hops

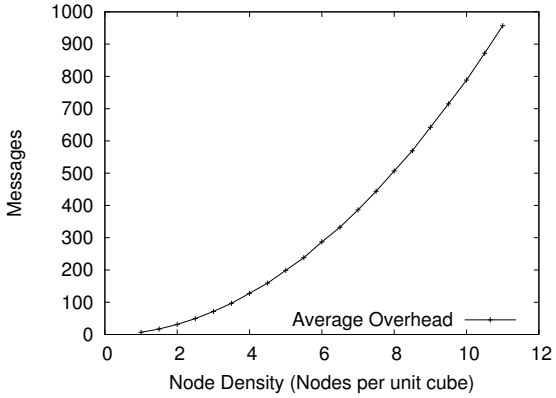


Figure 5.8: Overhead of Slab Routing's completion protocol.

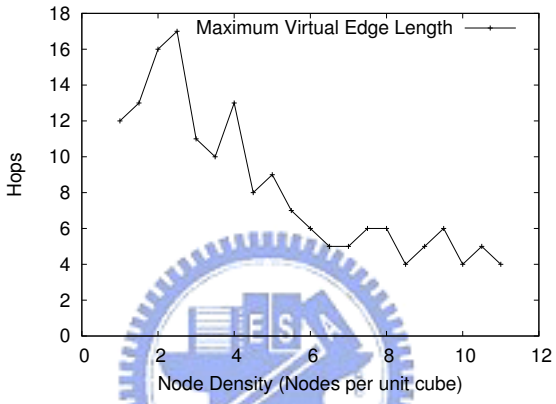


Figure 5.9: The maximum number of physical hops of virtual edges observed at each density.

that the completion protocol would span. However, Figure 5.9, which plots the maximum observed physical path length of all virtual edges at each density, gives us an indication of the value.

Chapter 6

Conclusion

In this paper we have presented Slab Routing, a local geographic routing algorithm that adapts two-dimensional geographic face routing to three-dimensional space. Like other local routing algorithms for three-dimensions, Slab Routing cannot guarantee the delivery of messages. However, unlike previous proposals, Slab Routing is efficient, employing only one message copy and traversing a path that is competitive with the shortest path. To help allay the problem of not being able to guarantee delivery, we provide an analysis of the relationship between delivery probability and system node density in the form of Equation (4.3). This relationship can be used to estimate the delivery rate achievable under a certain density, and thus can serve as a guideline for when using Slab Routing is appropriate or as a guideline for deciding the number of nodes that should be deployed in a region. Our simulation results show that while Equation (4.3) is indeed able to approximate actual performance, there is one major discrepancy - the delivery rate approaches one at a much slower rate. The cause for this phenomenon and a more accurate mathematical model to model delivery probability is a part of our future work.

Bibliography

[1] A. E. Abdallah, T. Fevens, and J. Opatrny. Hybrid position-based 3d routing algorithms with partial flooding. In *CCECE '06: Proceedings of the 19th Annual IEEE Canadian Conference on Electrical and Computer Engineering*, pages 227–230, 2006.

[2] A. E. Abdallah, T. Fevens, and J. Opatrny. Randomized 3d position-based routing algorithms for ad-hoc networks. In *MOBIQUITOUS '06: The 3rd Annual International Conference on Mobile and Ubiquitous Systems: Networks and Services*, pages 1–8, 2006.

[3] A. E. Abdallah, T. Fevens, and J. Opatrny. High delivery rate position-based routing algorithms for 3d ad hoc networks. *Computer Communications*, 31(4):807–817, 2008.

[4] J. Allred, A. B. Hasan, S. Panichsakul, W. Pisano, P. Gray, J. Huang, R. Han, D. Lawrence, and K. Mohseni. Sensorflock: an airborne wireless sensor network of micro-air vehicles. In *SenSys '07: Proceedings of the 5th International Conference on Embedded Networked Sensor Systems*, pages 117–129, 2007.

[5] L. Barriére, P. Fraigniaud, and L. Narayanan. Robust position-based routing in wireless ad hoc networks with unstable transmission ranges. In *DIALM '01: Proceedings of the 5th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, pages 19–27, 2001.

[6] C. Bettstetter. On the minimum node degree and connectivity of a wireless multihop network. In *MobiHoc '02: Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 80–91, 2002.

[7] H. Dette and N. Henze. The limit distribution of the largest nearest-neighbour link in the unit d-cube. *Journal of Applied Probability*, 26(1):67–80, 1989.

[8] J. Diaz, M. D. Penrose, J. Petit, and M. Serna. Convergence theorems for some layout measures on random lattice and random geometric graphs. *Combinatorics, Probability, and Computing*, 9(6):489–511, 2000.

[9] S. Durocher, D. Kirkpatrick, and L. Narayanan. On routing with guaranteed delivery in three-dimensional ad hoc wireless networks. In *ICDCN '08: Proceedings of the 9th International Conference on Distributed Computing and Networking*, pages 546–557, 2008.

[10] G. G. Finn. Routing and addressing problems in large metropolitan-scale internetworks. Technical Report ISI/RR-87-180, Information Sciences Institute, 1987.

[11] R. Flury and R. Wattenhofer. Randomized 3d geographic routing. In *INFOCOM '08: Proceedings of the 27th IEEE Conference on Computer Communications*, pages 834–842, 2008.

[12] P. Gupta and P. R. Kumar. Critical power for asymptotic connectivity. In *CDC '98: Proceedings of the 37th IEEE Conference on Decision and Control*, pages 1106–1110, 1998.

[13] J. Heidemann, W. Ye, J. Wills, A. Syed, and Y. Li. Research challenges and applications for underwater sensor networking. In *WCNC '06: Proceedings of IEEE Wireless Communications and Networking Conference*, pages 228–235, 2006.

[14] G. Kao, T. Fevens, and J. Opatrny. Position-based routing on 3-d geometric graphs in mobile ad hoc networks. In *CCCG '05: Proceedings of the 17th Canadian Conference on Computational Geometry*, pages 88–91, 2005.

[15] G. Kao, T. Fevens, and J. Opatrny. 3-d localized position-based routing with nearly certain delivery in mobile ad hoc networks. In *ISWPC '07: 2nd International Symposium on Wireless Pervasive Computing*, pages 344–349, 2007.

[16] B. Karp and H. T. Kung. Gpsr: greedy perimeter stateless routing for wireless networks. In *MobiCom '00: Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, pages 243–254, 2000.

[17] E. Kranakis, H. Singh, and J. Urrutia. Compass routing on geometric networks. In *CCCG '99: Proceedings of the 11th Canadian Conference on Computational Geometry*, pages 51–54, 1999.

[18] F. Kuhn, R. Wattenhofer, and A. Zollinger. Worst-case optimal and average-case efficient geometric ad-hoc routing. In *MobiHoc '03: Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 267–278, 2003.

[19] J. Li, J. Jannotti, D. S. J. De Couto, D. R. Karger, and R. Morris. A scalable location service for geographic ad hoc routing. In *MobiCom '00: Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, pages 120–130, 2000.

[20] M. S. Pan, C. H. Tsai, and Y. C. Tseng. Emergency guiding and monitoring applications in indoor 3d environments by wireless sensor networks. *International Journal of Sensor Networks*, 1(1/2):2–10, 2006.

[21] M. D. Penrose. On k-connectivity for a geometric random graph. *Random Structures and Algorithms*, 15(2):145–164, 1999.

[22] S. Ramanathan and E. L. Lloyd. Scheduling algorithms for multi-hop radio networks. *SIGCOMM Computing Communication Review*, 22(4):211–222, 1992.

[23] T. J. Shepard. A channel access scheme for large dense packet radio networks. *SIGCOMM Computer Communication Review*, 26(4):219–230, 1996.

[24] Sinalgo. *Simulator for Network Algorithms*. http://dcg.ethz.ch/projects/sinalgo, 2007.

[25] F. Xue and P. R. Kumar. The number of neighbors needed for connectivity of wireless networks. *Wireless Networks*, 10(2):169–181, 2004.