LOCALIZED SQUARE TEST GRAPH WITH
APPLICATIONS TO WIRELESS AD HOC NETWORKS

by

Sun, Hui

Submitted in partial fulfillment of the
requirements for the degree of
Master of Computer Science

at

Carleton University
Ottawa, Ontario
April 2008

© Copyright by Sun, Hui, 2008

Library and
Archives Canada

Bibliothèque et
Archives Canada

Published Heritage
Branch

Direction du
Patrimoine de l'édition

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 978-0-494-40651-9

Our file Notre référence

ISBN: 978-0-494-40651-9

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Table of Contents

List of Tables	v
List of Figures	vi
List of Symbols	viii
Abstract	xi
Acknowledgements	xii
Chapter 1 Introduction	1
1.1 Motivation	2
1.2 Statement of the Problem	3
1.3 Organization of Thesis	5
Chapter 2 Background and State of the Art	6
2.1 Introduction	6
2.2 Unit-disk Graph (UDG)	6
2.3 Proximity Graphs	8
2.3.1 Relative Neighborhood Graph (RNG) and Gabriel Graph (GG)	9
2.3.2 Delaunay Triangulation (Del)	10
2.3.3 Yao Graph	10
2.4 Use of Proximity Graphs to Simplify the Unit Disk Graph	11
2.4.1 Sparseness, Planarizations and Spanner	12
2.4.2 Efficient Localized Construction and Bounded Degree	13
2.4.3 Key Areas of Related Work	14
2.4.4 Del and GG in the L_1 Metric Plane	16
2.5 Summary	17
Chapter 3 Theoretical Treatment	18
3.1 Introduction	18

3.2	Short-edge Replacement Rule in the Reduction of UDG	18
3.3	A New Proximity Graph—the Square Test Graph	20
3.4	Use Unit-square Graph as the Original Network Model	44
3.5	Summary	47
Chapter 4	Description of Algorithmic Designs	49
4.1	Introduction	49
4.2	Discussion of Design Issues	49
4.3	A Distributed and Localized Algorithm for the Construction of LSTG	50
4.3.1	Two Main Data Structures	51
4.3.2	Explanations of Functions Used in Pseudocode	53
4.4	Running Time of the Algorithm	54
4.5	Summary	55
Chapter 5	Simulation and Presentation of Experimental Results	56
5.1	Introduction	56
5.2	About the Connectivity of the Unit-square Graph	57
5.3	Stochastic Geometry	58
5.3.1	Key-Node Set	58
5.3.2	Coupon Collector’s Problem Model	59
5.4	Discussion of the Parameter Space	61
5.5	Analysis of Experimental Results	62
5.5.1	Minimum Transmission Range vs. Stability of Edges of LSTG	63
5.5.2	Correctness of Formula (5.2) in Simulation	66
5.5.3	Comparisons of Spanning Ratios	70
5.6	Summary	72
Chapter 6	Summary, Conclusions, Contributions and Future Work	74
6.1	Summary of work and Conclusions	74
6.2	Contributions	75
6.3	Future Work	75

Appendix A	Part 1 of the parallel algorithm	76
Appendix B	Part 2 of the parallel algorithm	77
Appendix C		78
Bibliography		80

List of Tables

Table 5.1	A number n of nodes in a plane of size $a \times b$ ($n = 500, a = b = 500$ meters).	67
Table 5.2	A number n of nodes in a plane of size $a \times b$ ($n = 1000, a = b = 500$ meters).	68
Table 5.3	Comparisons of the averages and deviations of the all-pairs shortest-path length ratios of GG-to-UDG, GG-to-UDG and RNG-to-UDG ($r \geq 40$).	71

List of Figures

Figure 2.1	Definition and an example of the UDG.	7
Figure 2.2	Definitions of RNG, GG, Del and Yao graph.	9
Figure 2.3	The L_1 plane and the Euclidean plane define the “planar” graph in different ways.	17
Figure 3.1	The short-edge replacement rule fits in GG and RNG.	19
Figure 3.2	Definition and an example of STG.	20
Figure 3.3	Nodes A and B , and $R(AB)$, $ \overline{AB} $, $ \overline{AB} _X$, $ \overline{AB} _Y$	21
Figure 3.4	$\overline{AB} \in E$ implies $R(AB)$ can transform into an “empty square” passing through A and B with side length d	22
Figure 3.5	$STG(V, E)$ is a planar graph.	23
Figure 3.6	That $\overline{AB} \notin E$ implies $ \overline{CD} _X \leq \overline{AB} _Y$, by Lemma 2.	25
Figure 3.7	Definitions of S_{ADB} , $rR(AB)$, $rB(AB)$, S_{ACB} , $lR(AB)$, $lB(AB)$	26
Figure 3.8	Two diagonals of $S_{ADB}(\overline{AB} _Y)$ divide the $rR(AB)$ into four or three regions	26
Figure 3.9	Edges on the path $p(AB)$ when C appears in region 3 or region 1	28
Figure 3.10	C always appears in the region 1 or 2.	29
Figure 3.11	C always appears in the region 1 or 2 until event (I.I) or event (I.II) happens at some step of the loop process.	31
Figure 3.12	Use u_1 to stand for either u' or u'' , thus $u_1 > \frac{u_0}{1-u_0}$ holds.	33
Figure 3.13	Follow a top-down order to prove the existence of the path $p(AB)$ while C recursively appears in the region 4.	34
Figure 3.14	Emanating nodes $A_{2^1}^1$ and $B_{2^1}^1$ surely enter in or stop before the next recursion depth 2.	36
Figure 3.15	The maximum depth n is bounded by a constant $\lceil \frac{h}{w} - 2 \rceil$	38
Figure 3.16	Inequation (3.1) holds for any pair of nodes that are on a leaf of the recursion tree.	39
Figure 3.17	Inequation (3.1) holds for any pair of nodes in the depth m	40

Figure 3.18	If $R(AB)$ is not empty, $p(A \sim B)$ is divided into many pieces to consider, and each resulting piece satisfies CASE 1.	43
Figure 3.19	The STG does not obey the short-edge replacement rule. . . .	44
Figure 3.20	A comparison of communication range and transmission range between UDG network model and USG network model.	45
Figure 4.1	The data structure used by current node v to store a broadcast message from u	51
Figure 4.2	Each wireless node A uses 4 tables, Q_1, Q_2, Q_3, Q_4 , to represent its 4 Quadrants, I, II, III, VI	52
Figure 5.1	$GG(V)$, $RNG(V)$ and $LSTG(V)$ on the same set V of nodes.	56
Figure 5.2	A screen shot from the simulation kit.	63
Figure 5.3	Edge sets of USG and LSTG vary with the transmission range r	64
Figure 5.4	A number n of nodes in a plane of size $a \times b$ ($n = 500, a = b = 500$ meters).	67
Figure 5.5	A number n of nodes in a plane of size $a \times b$ ($n = 1000, a = b = 500$ meters).	68
Figure 5.6	A comparison of the average length of all-pairs shortest-paths among LSTG, GG and RNG ($r \geq 40$).	70
Figure 5.7	Comparisons of the practical spanning ratios of LSTG, GG and RNG ($r \geq 40$).	72

List of Symbols

$STG(V, E)$	a square test graph on a set V of nodes, having a set E of edges, see Figure 3.2, page 20
$LSTG(V)$	a localized square test graph on a set V of nodes, constructed locally, see an example in Figure 5.1(c), page 49
$USG(V)$	a unit-square graph on a set V of nodes, a new model for wireless ad hoc networks proposed in this thesis, see Figure 3.20, page 45
$RNG(V) / GG(V)$	two proximity graphs on a set V of nodes, i.e. relative neighborhood graph / Gabriel graph, see Figure 2.2, page 9
$UDG(V)$	a unit-disk graph on a set V of nodes, a model of wireless ad hoc networks universally used prior to this thesis, see Figure 2.1, page 7
$\overline{AB} / \overline{AB} $	an edge (or a line segment) in the Euclidean plan between two nodes, A and B / the length of $\overline{AB}$, see Figure 3.3, page 21
$ \overline{AB} _X / \overline{AB} _Y$	the projection to X -Axis of $ \overline{AB} $ / the projection to Y -Axis of $ \overline{AB} $, see Figure 3.3(a), page 21
$\max(\overline{AB} _X, \overline{AB} _Y)$	the larger between $ \overline{AB} _X$ and $ \overline{AB} _Y$, see Figure 3.3, page 21
$\min(\overline{AB} _X, \overline{AB} _Y)$	the smaller between $ \overline{AB} _X$ and $ \overline{AB} _Y$, see Figure 3.3, page 22
$A_{2m}^t[0], \dots, A_{2m}^t[\kappa_A]$	notations used in the proof of the Subsubcase 1.3.1 of Theorem 2 to denote a sequence of κ_A nodes, all of which together are treated as one node A_{2m}^t (called “emanating node”) of the t th pair of nodes, in the recursion depth of m , where $2^0 \leq t \leq 2^m$, see Figure 3.14, page 36

$A_{2^n}^t, B_{2^n}^t, C_{2^n}^t, D_{2^n}^t$	notations used in the proof of the Subcase 1.3 of Theorem 2 to denote four current nodes “A”, “B”, “C” and “D”. They all derive from the t th pair of nodes in the recursion depth of n , where $2^0 \leq t \leq 2^n$, see Figure 3.13, page 34
A_n, B_n, C_n, D_n	notations used in the proof of the Subcase 1.2 of Theorem 2 to denote four current nodes “A”, “B”, “C” and “D” at the n th step of the “loop process”, see Figure 3.10, page 28
C_V	the coverage of a set V of nodes, equaling the union of communication ranges (in USG model) of all nodes in V , page 59
$lB(AB) / lB(AB) $	the partial boundaries of $S_{ACB}(\overline{AB} _Y)$ clockwise from A to B / the length of $lB(AB)$, see Figure 3.7, page 25
$p(A \sim B) / p(A \sim B) $	a path of $STG(V, E)$ between nodes A and B , where $R(AB)$ contains at least one node of V / the length of $p(A \sim B)$, page 42
$p(AB) / p(AB) $	a path of $STG(V, E)$ between nodes A and B , where $R(AB)$ contains no nodes of V / the length of $p(AB)$, page 24
$R(AB)$	the closed rectangular area using nodes A and B as two diagonal corner points, see Figure 3.3(b), page 21
$rB(AB) / rB(AB) $	the partial boundaries of $S_{ADB}(\overline{AB} _Y)$ counterclockwise from A to B / the length of $rB(AB)$, see Figure 3.7, page 25
$rR(AB) / lR(AB)$	the part of $S_{ADB}(\overline{AB} _Y)$ right to $R(AB)$, a closed rectangular area or a line segment / the part of $S_{ACB}(\overline{AB} _Y)$ left to $R(AB)$, see Figure 3.7, page 25
$S_{AB}^{[A]}$	the unique closed square (not necessarily empty) having node A at one of its corner points and passing through A and B , with four sides of length $\max(\overline{AB} _X, \overline{AB} _Y)$, page 46

$S_{ADB}(\left|\overline{AB}\right|_Y)$

the closed square passing through three nodes, A , D and B , and with four sides of length $\left|\overline{AB}\right|_Y$, see Figure 3.7(a), page 25

u_m

“thickness factor”, used in the proof of the **Subcase 1.3** of Theorem 2 to define a property of three current nodes “ A ”, “ B ” and “ D ” (or “ A ”, “ B ” and “ D ”), in the recursion depth of m , refer to Formula (3.2) and see Figure 3.13, page 32

Abstract

In recent years, several localized routing protocols guarantee packets' delivery when the underlying network topology is a planar graph. The relative neighborhood graph (RNG), Gabriel graph (GG) or Delaunay triangulation (Del) is used for such a planar structure. But, the spanning ratio of RNG or GG cannot be bounded by any constant. So, researchers proposed improved Delaunay triangulations in combination with structures such as Yao Graph, MST, etc, because the distances traveled by packets can be shorter than a constant times of the minimums. But, such Delaunay triangulations are impossible to construct in any localized manner.

In this thesis, we propose a planner proximity graph—square test graph (STG)—that has a stretch factor $\sqrt{20}$ to the complete Euclidean graph. The STG looks like the “Gabriel graph for the L_1 metric” and adds one edge at a time, but it is defined purely in the Euclidean plane. Similar research was conducted by Chew [4] who proved that the L_1 -Delaunay triangulation has a stretch factor of $\sqrt{10}$ over the complete Euclidean graph. A wireless ad hoc network is usually modeled as the unit-disk graph (UDG), in which an edge exists only if its Euclidean distance is at most one. But, we model nodes' communication ranges as a set of equal-sized unit-squares inscribed in every unit-disk, thus the network is accordingly modeled as a unit-square graph (USG) rather than a UDG for the first time. Then, we present a localized expected $\Theta(\ln^2(n))$ -time algorithm to construct $STG \cap USG$. The resulting graph is called localized square test graph (LSTG), and it is a planar $\sqrt{20}$ -spanner of the USG. By our algorithm, each node needs only to know its entire 1-hop neighbors in USG for its edge computation.

Our experimental results confirm the existence of the minimum transmission range such that the induced network topologies, USG and LSTG, are connected almost surely. And, the edges of LSTG incident on every node are almost unchanged with the increase in the nodes' transmission ranges. Our experimental results also show the traveled distance of the packets could be significantly less when efficient routing algorithms are applied on LSTG, rather than applied on GG or RNG.

Acknowledgements

I want to thank all of you who have given your time, assistance and patience so generously. My deepest appreciation goes to Professor Evangelos Kranakis for the opportunity to make my ideas a reality, and for your consistent trust and support as I floundered my way through this process. Your timely guidance has always steered me in the right direction during my research, yet you gave me enough freedom to explore my creative side.

Thanks to the Graduate Director Dr. Jean-Pierre Corriveau. When I first came to Carleton University with trouble, you took the time to see me and lent me a helpful hand warmly.

Thanks you Claire Ryan. Once I need an advice, you are always there. You know what might be important to me and you tell me so.

Thanks to my families for your unending support, willingness to accept and eagerness to love. All your consistent encouragement, constructive criticisms and laughter are in here somewhere. Special thanks to my wife. You are the best reason I have to strive for a dream.

Chapter 1

Introduction

This is the Golden Age of the Internet, but no one knows how long it will last and what the next generation will bring. With its continued evolution, the wireless network will become more important, and the technology solutions used to support it will become more critical. This will motivate a host of research and development projects and activities in industry and academia alike. Someone said the earth used to be very large, but the Internet has made it smaller. And with the availability everywhere of the wireless network connectivity, the earth will become even smaller.

Within the past decade, people have seen great advances in network infrastructures, the growing availability of wireless applications, and the emergence of omnipresent wireless devices such as portable or handheld computers, PDAs (Personal Digital Assistants), and cell phones. All these are getting more powerful in their capabilities. Travelers with portable computers can surf the Internet from airports, railway stations, cafes, and other public locations; tourists can use GPS (Global Position System) terminals installed inside rental cars to view driving maps and locate tourist attractions; mobile users can rely on their cellular phones to check e-mail and browse the Internet; files or other types of information can be exchanged by connecting portable computers via wireless LANs (Local Area Networks) while attending conferences; and at home, a family can synchronize data and transfer files between portable devices and desktop computers.

Wireless devices are playing an ever-increasing role in our lives. Along with numerous mature techniques migrating from labs to the industry, mobile devices are getting smaller, cheaper, more convenient, and more powerful. They also run more applications and network services. From the explosive growth of the mobile computing equipment market seen today, we can expect the total number of mobile Internet users to exceed that of fixed-line Internet users in the not too distant future.

Network connections and corresponding data transmissions, which serve as the

base of the applications and services running on mobile devices, are in the highest demand. Currently, the main method of wireless connections is fixed-infrastructure-based service providers or private networks. For example, two cell phones set up a connection through BS (Base Station) and MSC (Mobile Services Switching Center) in cellular networks; laptops connect to the Internet via wireless access points. Although this manner provides a great way for mobile devices to get network services, this infrastructure is costly and time consuming to install. Furthermore there are cases where the user-required infrastructure is not available, cannot be installed, or cannot be installed in time in a given geographic area. Therefore, other alternatives are required to provide the needed connectivity and network services .

For these reasons, combined with the current rapid progress in science and in electronic engineering technology and standardization, researchers have recently proposed new alternative ways to set up an ad hoc mobile network that is both flexible and powerful, where mobile devices within the transmission range will connect to each other through automatic configuration. It is called an “ad hoc network” because each node is willing to forward data for other, and so the determination of which nodes forward data is made dynamically based on the network connectivity. This characteristic distinguishes it from the older network technologies because the latter has to designate some nodes, usually routers, switches, hubs, and firewall devices, to perform the task of forwarding data. In the mobile ad hoc networks with specific applications or services deployed, not only can mobile nodes communicate with each other, but they also may receive Internet services through an Internet gateway node, effectively extending both network and Internet services to areas without any infrastructures. Ad hoc networking is widely expected to form the essential piece of the 4G network architecture.

1.1 Motivation

Although impressive research and development results have been documented in a large number of papers, many open issues remain to clear the path for the successful ad hoc network deployment and commercialization.

In view of the intrinsic complexity of communication networks, researchers usually separate the network design problem from the management and control of the

network. This is a convenient separation and can help simplify these two phases. However, these two phases are not unrelated because decisions made at the network-design phase may strongly affect the network management and control phase. For example, a network designer should take into account hierarchical routing methods if he designs a backbone-like network topology instead of a flat one.

Compared to traditional wire-based networks, wireless ad hoc networks have many intrinsic characteristics and unavoidable limitations that should be taken into designers' consideration. For example, wireless nodes are often moving during the communication; they are often self-powered with integrated batteries and often have limited memories. Naturally, the network topology designed (or adopted) is expected to be more suitable for deploying an efficient routing scheme to save energy and memory consumption than the traditional wire-based networks. Here, both topology control and geographic routing usually assume the idealized unit-disk graph (UDG) network model: by a proper scaling, we assume all wireless nodes in the network have the same maximum transmission range equal to one unit; any two of them have a link if and only if their Euclidean distance is at most one; consequently, all wireless nodes form a unit-disk graph (UDG) at any moment.

Generally, the goals of network topology control and management are maintaining networks' connectivity, reducing energy consumption at nodes, optimizing networks' lifetime and throughput, and making power-efficient routing design possible. These goals motivate researchers to find a subgraph of the unit-disk graph. It should not only have nice features such as a linear number of edges, a low stretch factor, and bounded node degree, but also support attractive routing schemes such as localized routing with guaranteed performance.

1.2 Statement of the Problem

Not every connected subgraph of the unit-disk graph has as enough qualified features to be adopted in network design. In fact, measures have often been used to evaluate them. For example, one of these requires the shortest path connecting any two nodes in the subgraph is not longer than a constant time of the shortest path connecting them in the original unit-disk graph. Such a subgraph is called a spanner of UDG, and the constant is called the stretch factor describing one aspect of path quality of

this subgraph.

According to their characteristics, the routing protocols used in wireless ad hoc networks may be categorized as table-driven protocols or demand-driven protocols. Unlike typical wire-based networks with fixed network topologies, wireless ad hoc networks have ever-dynamic network topologies due to nodes' mobility. This could cause wireless network topologies even more dynamic in that each wireless node can potentially adjust its transmission range to control its set of neighbors [18]. Route discovery in both of the two types of protocols is potentially expensive, thus reducing networks' response time. In addition, under the circumstances where wireless nodes move around freely and have limited power and memory, maintaining explicit routes at every node would be costly and potentially impossible.

The sparse (or even planar) graph is chosen to make routing design easier, since route discovery in a sparse graph is easier than that in a dense graph. A planar graph is also a sparse graph since it has only a linear number of edges. It is also preferred the network topology be constructed by an efficient localized algorithm. This will lead to a significant reduction in the network overhead since messages for the purpose of network construction and maintenance never need to propagate throughout the network. Therefore, localized constructions of wireless networks are scalable. Here, a distributed algorithm constructing a graph G is called a localized algorithm if every node u can exactly determine all edges of G incident on u based only on the information from all nodes within a constant number of hops of u (plus a constant number of additional nodes' information if necessary.)

However, recent practices have demonstrated it is far from easy to design a planar spanner of UDG that can be constructed in a localized manner. In this thesis, we define a novel network topology named localized square test graph (LSTG), which is a planar $\sqrt{20}$ -spanner of the unit-square graph (USG). We also present a localized algorithm to construct the LSTG with an overall expected running time of $\Theta(\ln^2(n))$.

1.3 Organization of Thesis

Chapter 2 reviews the previously used geometric structures and formalizes the preferred features of these structures. Chapter 3 describes the main theoretical treatments of this thesis, including proposing a new planar proximity graph—STG, modeling original wireless networks as another structure—USG. Chapter 4 presents a parallel algorithm using an overall expected running time of only $\Theta(\ln^2(n))$ to construct a connected LSTG, a planar $\sqrt{20}$ -spanner of USG, as network topology (see pseudocode in Appendix). Chapter 5 shows the existence of the minimum transmission range both theoretically and experimentally, and also demonstrates that LSTG, as network topology, has a better path quality than GG and RNG. Finally, Chapter 6 summarizes the work and suggests future work.

Chapter 2

Background and State of the Art

2.1 Introduction

Mobile ad hoc networks are infrastructureless networks since they do not require the same infrastructures as base stations do for their operations. Each wireless node has a transmission range. Two wireless nodes can directly communicate if they are within each other's transmission range. Otherwise, they communicate through multihop wireless links by using intermediate nodes to relay messages. Thus each node in the wireless network acts as both a terminal of data packets and a router, forwarding data packets for other nodes. In recent years, topology control for wireless ad hoc networks has drawn considerable attention. An excellent survey was conducted by Rajaraman in [29]. Roughly speaking, topology control methods try to construct and maintain a structure that can be used for efficient routing. Li et al. made a good survey of these efficient routing protocols in [20]. Different structures with different properties have been proposed in the literature for a long time. Several geometrical structures were studied by both computational geometry scientists and network designers. Some of them will be reviewed in this chapter.

2.2 Unit-disk Graph (UDG)

In a wireless ad hoc network, we assume each node in the network has a unique identifier (ID), all nodes are location aware through GPS or other devices, and all nodes have the same maximum transmission range. By scaling, nodes' maximum transmission ranges are standardized to one unit, thus nodes' communication ranges are accordingly modeled as a set of unit-disks centered at every node (see Figure 2.1(a)). In a manner of broadcast, each node notifies its position to nodes within its communication range, thus forming 1-hop wireless links. Consequently, all nodes together define a dynamic network structure. We observe this structure at any given

Figure 2.1: Definition and an example of the UDG.

moment. Clearly, it can be modeled as a unit-disk graph (UDG), which is defined by the following.

Definition 1. *Given a set V of nodes distributed in a two-dimensional plane, a unit-disk graph $UDG(V)$ is a graph with node set V such that any two nodes of V have an edge in the $UDG(V)$ if and only if their Euclidean distance is at most one (see Figure 2.1(b)).*

The UDG is often used to model mobile wireless ad hoc networks and other types of wireless networks such as cellular networks or sensor networks. A UDG is always assumed connected.

Selecting the UDG as the final underlying network topology is not a good choice because an UDG is usually dense in edge and volatile in structure. A high density and frequent moving of nodes could make it more complex and more volatile. Route discovery in such a UDG alike network will use tremendous messages and energy. Since each wireless node carries limited batteries, power conservation is an important issue. A wireless node with a low battery can not last long. Nodes' power failure could in turn result in bad network performance in terms of routing efficiency and overall

maintenance overhead, and it could even disconnect the entire network. Particularly, each wireless node has only limited bandwidth to use. All these conflicts impose challenges upon designing an effective and efficient network routing protocol. To avoid these intrinsic deficiencies in UDG, a method to optimize power consumption at nodes and the lifetimes of networks is topology control. Topology control is a technique choosing and maintaining a suitable network structure that can be used for efficient routing, thus improving overall networking performance. It can be done by selecting a set of the available edges in UDG to form a "simplifying UDG" spanning the entire nodes. Recently, several proximity graphs have been used to simplify the UDG. Earlier topology control algorithms are often based on one or a combination of several of them. Next, some of these proximity graphs will be reviewed.

2.3 Proximity Graphs

Let V be the set of nodes in a two dimensional plane. Proximity graphs (also known as the neighborhood graphs) of V define neighbor relationships between (or among) nodes. Generally, any two nodes have an edge in a proximity graph if they are close in some sense. A more formal definition of proximity graphs is given in [10].

Edges of a proximity graph are usually defined by influence areas. Given any two nodes A and B of V , in some way they delimit an open (or closed) area of the two dimensional plane, called their influence area and denoted by $IA_{(A,B)}$. Nodes A and B have an edge $\overline{AB}$ of this proximity (neighborhood) graph if and only if no nodes of V are contained in $IA_{(A,B)}$. Specific methods used for the delimitation of influence areas make possible many kinds of proximity graphs. A survey conducted by Jaromczyk and Toussaint [10] discussed many of these in detail, such as relative neighborhood graph (RNG), Gabriel graph (GG), Yao graph. Different proximity graphs with the same node set V can be compared in terms of many measures such as edge density, average length of all-pairs shortest-paths, maximum and average (and deviation) of the all-pairs shortest-path length ratios of the proximity graph to the complete graph on V .

(a) Influence Area of A and B in the case of RNG

(b) Influence Area of A and B in the case of GG

(c) Influence Area of A , B and C in the case of Del

(d) Definition of the Yao graph

Figure 2.2: Definitions of RNG, GG, Del and Yao graph.

2.3.1 Relative Neighborhood Graph (RNG) and Gabriel Graph (GG)

The relative neighborhood graph is a geometric concept proposed by Toussaint [31]. Derived from the definition of proximity graphs in section 2.3, we define the relative neighborhood graph using the following (see Figure 2.2(a)):

Definition 2. Let A and B be two arbitrary nodes of V , and $D(A)$ and $D(B)$ denote two open disks centered at A and B respectively with the same radius of their Euclidean distance $|\overline{AB}|$. Nodes A and B have an edge $\overline{AB}$ of the relative neighborhood graph if and only if no nodes of V are contained in the lune—the common part of $D(A)$ and $D(B)$. The relative neighborhood graph on V is denoted by $RNG(V)$.

The Gabriel graph is another proximity graph proposed by Gabriel and Sokal [7], and was first introduced into the scope of wireless network topology control by Rodoplu and Meng [30]. They described a distributed protocol to construct a general version of the Gabriel graph. In the same way as RNG, the Gabriel graph is defined as follows (see Figure 2.2(b)):

Definition 3. *For any pair of nodes A and B of V , they have an edge $\overline{AB}$ of the Gabriel graph if and only if no nodes of V are contained in the open disk with a diameter of their Euclidean distance $|\overline{AB}|$. The Gabriel graph on V is denoted by $GG(V)$.*

2.3.2 Delaunay Triangulation (Del)

The Delaunay triangulation, a famous proximity graph, was first proposed by Boris Delaunay in 1934. It has been widely studied in both mathematics and computational geometry since then. Different from RNG and GG, the Delaunay triangulation's method for the delimitation of influence areas involves three nodes instead of two. Supposing no four nodes of V are co-circular, then we define the Delaunay triangulation as follows (see Figure 2.2(c)):

Definition 4. *Let A , B and C are three arbitrary nodes of V , and $\triangle ABC$ denotes the triangle formed by them. $\triangle ABC$ is a Delaunay triangle of the Delaunay triangulation if and only if no nodes of V are contained in the circumcircle of $\triangle ABC$. The Delaunay triangles on V is denoted by $Del(V)$.*

2.3.3 Yao Graph

The Yao graph is proposed by Andrew C. Yao in [33] to construct Minimum Spanning Tree (MST) of a set of points in high dimensions efficiently. According to Li et al. in [20], we formalize the definition of the Yao graph with an integer parameter $k \geq 6$ as follow (see Figure 2.2(d)).

Definition 5. *At each node $A \in V$, any k equally separated rays originated at A define k equal cones. In each cone, choose the node B having shortest Euclidean distance to A , if there is any (or if there are many, choose arbitrarily or by the smallest ID), and add a directed edge $\overrightarrow{AB}$. The resulting directed graph is called the Yao graph, denoted*

by $\overrightarrow{YG}_k(V)$. If we add the reverse directed edge from A to B , we obtain the Reverse Yao Graph, denoted by $\overleftarrow{YG}_k(V)$. If we ignore the direction of edges, we obtain the Undirected Yao Graph, denoted by $YG_k(V)$.

A structure named the Θ -graph [25] is similar in construction to the Yao graph. The Θ -graph chooses the edge in each cone with the shortest projection on the axis of the cone instead of the shortest edge chosen by the Yao graph, where the axis of each cone is usually the angular bisector of the cone. Both the traditional Yao graph and the Θ -graph are called the Yao structure when used in the topology control of wireless networks.

Because the Yao structure is not necessarily a planar graph, it sometimes combines with other structures to increase its sparseness. For example, Li et al. proposed [22] to apply the Yao structure on top of the Gabriel graph structure, obtaining a resulting graph denoted by $\overrightarrow{YGG}_k(V)$. They also applied the Gabriel graph structure on top of the Yao structure to obtain another resulting graph denoted by $\overleftarrow{GYG}_k(V)$. These two structures are all connected and sparser than both the Yao structure and the Gabriel graph. In [32], Li et al. further applied the Yao structure on the Local Delaunay graph (LDel), obtaining a new planar structure that can be constructed by nodes locally.

In summation, the following relationships among the above proximity graphs have been proven in [9, 22].

- ◇ $RNG(V) \subseteq GG(V) \subseteq Del(V)$
- ◇ $RNG(V) \subseteq YG_k(V)$, where $k \geq 6$

In addition, the minimum spanning tree on V , $MST(V) \subseteq RNG(V)$.

2.4 Use of Proximity Graphs to Simplify the Unit Disk Graph

Topology control algorithms strive to remove nonessential links from the network in order to force nodes to use several essential hops instead. This approach uses a smaller amount of communication interfaces and energy. However, if too many edges or a wrong selection of edges is removed, the hops in the paths could become unacceptably long and the network may even disconnect. In case of these occurrences, the final network topology, a subgraph of the UDG resulting from using some proximity graph

to simplify UDG's edge set, is commonly expected to have some of the following desirable features.

2.4.1 Sparseness, Planarizations and Spanner

Routing is the foremost issue in the scope of wireless network studies. A sparse network topology (i.e. with linear edges $O(n)$) enables numerous routing algorithms (e.g. routing based on the shortest path) to run more efficiently. It also makes the network maintenance and management easier for wireless nodes than those before the topology reduction. A planar graph is also a sparse graph because its edges range from $n - 1$ to $2n - 1$. Some location-based routing algorithms require the underlying network topologies to be planar geometric graphs, e.g. compass routing and random compass routing (RCR) in [15], greedy face routing in [3], greedy perimeter stateless routing in [11] and the right-hand rule etc. Therefore, a planar subgraph of UDG is desirable.

Location-based routing algorithms route data packets by using the location information of wireless nodes. They include greedy routing and face routing. Greedy routing routes data efficiently by using short hop paths, but cannot guarantee messages' delivery. Another eminent face routing algorithm proposed by Kranakis et al. in [15] was originally called compass routing. Face routing routes data along the graph's faces. Some improvements to the face routing algorithm have been proposed. For example, the adaptive face routing algorithm [16] requires face routing strictly within a size-adaptive ellipse to achieve asymptotically optimal paths. The face routing is usually combined with greedy routing to achieve transmission efficiency and guaranteed message delivery, e.g. greedy face routing in [3], greedy other adaptive face routing [17] etc. The existing face routing algorithms mainly apply to three types of planar network structures using the UDG network model, which are the Gabriel graph, the relative neighborhood graph, and the Delaunay triangulation. However there is a drawback of the face routing while applied to GG or RNG: although message delivery is theoretically guaranteed, the route discovered may be many times longer than the shortest path in the UDG network model. We also notice that both face routing (or compass routing) and greedy routing, while applied to the Delaunay

triangulation, guarantee the message delivery [15, 3] because the Delaunay triangulation, different from RNG and GG, is a spanner of UDG. Here, the definition of spanner is formalized as follows.

Definition 6. *A subgraph $G'(V)$ a t -spanner of a graph $G(V)$ if and only if, for any two arbitrary nodes A and B in V , the Euclidean length of the shortest path between A and B in $G'(V)$ is not longer than t times of that in $G(V)$. The constant t is called the length stretch factor (or stretch factor) of $G'(V)$ to $G(V)$.*

The length stretch factor is a measure partially presenting the path quality of the spanner relative to its supergraph, because it indicates the worst ratio of the shortest-path lengths in subgraph $G'(V)$ to those in $G(V)$. Clearly, a planar spanner of UDG is more desirable.

In this thesis, we propose a novel proximity graph named Square Test Graph (STG). A STG is a $\sqrt{20}$ -spanner of the complete Euclidean graph.

2.4.2 Efficient Localized Construction and Bounded Degree

The ideal wireless ad hoc network is an infrastructureless system. Due to nodes' mobility, it is almost impossible for each wireless node to know in time the exact positions of other nodes excepting those within its transmission range. It is also time consuming for nodes to know even those within constant hops away, not to mention the energy consumed for the entire network. Therefore, we prefer an underlying network topology constructed by nodes in a localized manner, i.e. all wireless nodes determine their incident links in the final network structure based only on the information from their local area. Network topologies constructed in this manner are realistically significant because messages, for the purpose of network construction and maintenance, never need to propagate more than a constant number of hops away from their originator nodes. This will lead to a significant reduction in network overhead. Bose, Morin, Stojmenovic and Urrutia first defined what is a localized algorithm in [3]. In this thesis, we apply STG on the top of the unit-square graph instead of unit-disk graph to obtain a structure, Localized Square Test Graph (LSTG), which can be easily constructed by nodes through acquiring the positions of their 1-hop neighbors.

As well, a small degree for each wireless node will reduce the MAC-level (Media

Access Control) contention and interference, and also may help mitigate the well-known hidden and exposed terminal problems. However, in this thesis, the issue of bounded node degree for LSTG is still open.

2.4.3 Key Areas of Related Work

Following is a summary of the previous work done in topology control of wireless ad hoc networks. Among numerous proximity graphs applied to simplifying the UDG, the Gabriel graph and the relative neighborhood graph are the two pervasively used representatives due to easy construction in a localized manner. Unfortunately, it is known [1, 6] that the spanning ratios of GG and RNG are not bounded by any positive constant. For example, as posited in [1], for a set of n wireless nodes uniformly and randomly distributing in a square, the spanning ratio of the Gabriel graph on them is at least $\Omega(\sqrt{\log n / \log \log n})$. Under this circumstance, no matter what efficient routing algorithms are deployed on the two topologies, the spanning ratio achieved is almost sure to be $\Omega(\sqrt{\log n / \log \log n})$. Note that the practical spanning ratio achieved by a specific routing schema should be larger than the theoretical one. Nonetheless, a structure with a small spanning ratio is necessary since it provides the potentiality to some routing schemas of performing well—[24].

The Delaunay triangulation has been shown [4, 5, 12, 13] a planar t -spanner of the complete Euclidean graph. However, it is almost impossible to construct and maintain a Delaunay triangulation in wireless communication environment because many edges belonging to the Delaunay triangulation may not belong to the UDG network model or massive communications are required. Previously, some approaches are proposed to approximate the Delaunay triangulation locally. On the top of a UDG network model, Li et al. [19] and Gao et al. [8] extended the definition of the Delaunay triangulation to a new structure—unit Delaunay triangulation: given a set of nodes V , the unit Delaunay triangulation, denoted by $UDel(V)$, is the graph obtained by removing all edges of the Delaunay triangulation $Del(V)$ longer than one unit. They proved [19, 8] that $UDel(V)$ is a t -spanner of $UDG(V)$. In [8], Gao et al. proposed a structure—restricted Delaunay graph (RDG)—and showed it has good spanning ratio properties. They also provided a distributed algorithm to describe how to maintain a RDG locally so that, at the end of the algorithm, the topology finally obtained is

a planar supergraph of $UDel(V)$. According to their algorithm, each node A need only acquire the position of its 1-hop neighbors $N_1(A)$ and compute the Delaunay triangulation $Del(N_1(A))$ on $N_1(A)$, including A itself. However, the communication cost could be as large as $\Theta(n^2)$, and the computation cost could be as large as $\Theta(n^3)$. Previously, a network structure similar to the RDG was used by Bose et al. [3] and Karp et al. [11], called a localized Delaunay graph $LDel(1)(V)$, in which each node A only acquires the position of its 1-hop neighbors $N_1(A)$. Even though the spanning ratios of the structures derived from Delaunay triangulation could be bounded by a constant, communication and computation costs of constructing and maintaining such structures are usually unacceptable high.

Some researchers [22] tried to use Yao graph (also called Θ -graph) in the topology control of wireless ad hoc networks, so derived topologies could have both bounded node degrees and bounded spanning ratios to the UDG [13]. However, the Yao graph [22] is not necessarily planar.

Recently, combining two structures, i.e. the minimum spanning tree (MST) and the localized Delaunay graph, Bose et al. [2] firstly proposed a centralized $O(n \log n)$ -time algorithm to compute a planar spanner with bounded node degree. Then, Li and Wang improved the value of the “bounded node degree” by giving another centralized method in [23]. Then, combining the localized Delaunay graph with the Yao structure (or Θ -graph), Li and Wang [32] were the first to propose a centralized algorithm that can construct a planar t -spanner of UDG with a bounded node degree in a localized manner.

In this thesis, we provide a new heuristic to the topology control of wireless ad hoc networks. We use a unit-square graph (USG) to model wireless ad hoc networks. In contrast with the UDG network model previously used, the USG network model enables the square test graph (STG) to simplify itself into a localized square test graph (LSTG), i.e. $STG \cap USG$. This LSTG is a planar $\sqrt{20}$ -spanner of the USG network model, in which message delivery is therefore guaranteed whereas that is not guaranteed in GG or RNG. In contrast with Delaunay triangulation (Del), this LSTG can be efficiently constructed in a localized manner. Furthermore, the localized algorithm presented in thesis, for LSTG’s construction, uses only limited computation and communication: the overall computation cost is as good as $\Theta(\ln^2(n))$; the

total communication cost is $\Theta(n \log n)$ bits, assuming every node's identifier can be represented by $\log n$ bits. This communication cost is the optimum to construct any structure in a distributed manner, because each of n nodes has to send at least one message to its neighbors to notify its existence in any protocol and this implies the communication cost is at least $\Theta(n \log n)$ bits for any protocol [21].

2.4.4 Del and GG in the L_1 Metric Plane

The stretch factor for the Delaunay triangulation under the Euclidean metric was studied by [5, 12, 13]. In an effort to improve this stretch factor different metrics were introduced by Chew [4], including the L_1 metric as well as metrics first used by Minkowski in 1911.

The Delaunay triangulation is usually defined or constructed by two means: 1) as the dual is the “Voronoi diagram”; 2) a “triangle” exists if their circumscribed “circle” has no other nodes in its interior. It is commonly believed that these two equivalent features hold for Delaunay triangulations under various metrics. For example, for the Euclidean metric, we get the Euclidean Delaunay triangulation; for the L_1 metric, we obtain a triangle when the L_1 -circle (a square tipped at 45°) contain no other nodes. In [4], Chew first used the L_1 Voronoi diagram to construct the L_1 Delaunay triangulation, and then used L_1 -circles to bound the stretch factor of this L_1 -Delaunay. But, it was not proven that the L_1 -Delaunay is straight-line planar graph. According to Chew, the L_1 Voronoi diagram divides the plane into regions whose boundaries are almost all polylines. Therefore, it is also reasonable that the L_1 -Delaunay is defined as a “planar” graph in terms of its edges that are all polylines or curves instead of straight lines. In fact, the L_1 plane uses a definition of “planarization” different from the Euclidean plane. For example, two pair of nodes, (v_0, v_1) and (v_2, v_3) , have their relative positions in the L_1 metric plane, as shown in the left part of Figure 2.3(a) or Figure 2.3(b). They could be two edges of a straight-line planar graph (see Figure 2.3(a)). However, if their relative positions are measured by the L_1 metric and are then restored to the Euclidean plane for people to consider, the two straight lines connecting them respectively are viewed as two intersecting edges. Consequently, they cannot appear in any “planar” graph simultaneously. Figure 2.3(b) illustrates an opposite example, in which $\overline{v_0v_1}$ and $\overline{v_2v_3}$ cannot simultaneously appear in any

Figure 2.3: The L_1 plane and the Euclidean plane define the “planar” graph in different ways.

straight-line planar graph, but they could be two edges of some “planar” graph.

The Delaunay triangulation is a global $O(n \log n)$ algorithm; for this reason the Gabriel graph (GG) has been defined: an edge exists if and only if the circle with the diameter connecting these two points contains no other point. In this thesis, we define the STG or “Gabriel graph for the L_1 metric” (by adding one edge at a time). Now GG is a proper subset of Del. The resulting graph should have fewer edges than the L_1 -Delaunay the same way that the Euclidean-GG has less than the Euclidean-Delaunay. In this thesis we prove that the resulting graph is a pure straight line planar graph. As well, it thus provides a localized method to construct the L_1 -Delaunay that was previously unknown.

2.5 Summary

This chapter reviews the model of wireless mobile ad hoc networks, and some proximity graphs as well. We also discuss some desirable features for a proximity graph if it is considered as the underlying network topology. As summarized in section 2.4.3, many important issues need better solutions even though great achievement has been made by researcher’s constant endeavors. In Chapter 3, we propose a new proximity graph. It has almost all the features as listed in section 2.4 except for the “bounded node degree.”

Chapter 3

Theoretical Treatment

3.1 Introduction

The studies of proximity graphs first arose from computational geometry and then spread to the field of the wireless network. According to Jaromczyk and Toussaint [10], proximity graphs are graphs in which nodes close to each other by “some definition of closeness” are connected. Therefore, in terms of computational geometry, proximity graphs initially are graphs proximate to complete Euclidean graphs. The reason they drew the researchers’ attention in the wireless ad hoc network domain is that “some definition of closeness” can also define a graph proximate to the unit-disk graph.

In this Chapter, we track the footprints that proximity graphs left from the computational geometry to the wireless ad hoc network domain. However, before we invent a proximity graph purely in the computational geometry domain, we need to know what enables proximity graphs to traverse the gap between the computational geometry and the topology control of wireless mobile ad hoc networks, or what enables the proximity graph to simplify the edge sets of both complete Euclidean graphs and unit-disk graphs.

3.2 Short-edge Replacement Rule in the Reduction of UDG

We observe the procedure of simplifying a complete Euclidean graph into two representative planar proximity graphs, GG and RNG. Some edges appear “nonessential” and hence are removed because they can be replaced by those more “essential.” Consequently, the proximity graph, GG and RNG, consist of the most essential edges that are finally reserved. We call this rule “Short-edge Replacement Rule” because the replacing (or more essential) edges are always shorter than the edges replaced (or nonessential edges). For example, we consider a Gabriel graph (see Figure 3.1(a)).

Figure 3.1: The short-edge replacement rule fits in GG and RNG.

The reason for $\overline{AB} \notin GG$ is that $\overline{AC} \in GG$, $\overline{CB} \in GG$ and $\overline{AC}$ and $\overline{BC}$ can replace the edge $\overline{AB}$ to bridge nodes A and B . Similar cases happen to a relative neighborhood graph. For example (see Figure 3.1(b)), $\overline{AB} \notin RNG$ because $\overline{AC} \in RNG$, $\overline{CB} \in RNG$ and the edge $\overline{AC}$ and the edge $\overline{BC}$ replace the edge $\overline{AB}$. Trivially, in the above two examples, $\overline{AC}$ and $\overline{CB}$ are shorter, and thus more essential, than $\overline{AB}$.

If somehow we have a planar subgraph of the complete Euclidean graph satisfying the short-edge replacement rule, then this subgraph can also be used to simplify the UDG, because the entire edge replacements happen exclusively among the edges of UDG. In other words, if a long edge of UDG is removed, then both this long edge and the shorter edges causing the removal of this long edge are edges of UDG because all of them are not longer than one unit. This fact guarantees the connectivity of that subgraph under the assumption that a UDG is always connected. We observe another proximity graph, Delaunay triangulation (Del). Similar to GG and RNG, the Del with respect to the complete Euclidean graph also follows the short-edge replacement rule, except for the more complex edge replacements.

Suppose we invent a method identifying a proximity graph proximate to the complete Euclidean graph. According to the above observation, when we use this proximity graph to simplify a UDG, the resulting UDG's subgraph must be connected if this proximity graph obeys the short-edge replacement rule.

Figure 3.2: Definition and an example of STG.

3.3 A New Proximity Graph—the Square Test Graph

This section presents a new proximity graph, a planar $\sqrt{20}$ -spanner of the complete Euclidean graph.

Given a set V of nodes in the Euclidean plane with the established X -Axis and Y -Axis such that the line segment linking any two nodes of V is parallel to neither the X -Axis nor the Y -Axis. We define the Square Test Graph $STG(V, E)$ as follows:

Definition 7. For two arbitrary nodes A and B of V , $\overline{AB}$ is an edge of E if and only if there exists at least one closed square such that (1) its boundaries are parallel to either X -Axis or Y -Axis and simultaneously pass through A and B , and (2) it contains no nodes of V (see Figure 3.2). Such a closed square is called an “empty square”.

Note: The boundaries of those squares and rectangles discussed in this thesis are parallel to either X -Axis or Y -Axis at any time (see Figure 3.2 and Figure 3.3).

For two arbitrary nodes A and B of V , suppose the lengths of line segment $|\overline{AB}|$ projections to X -Axis and to Y -Axis are $|\overline{AB}|_X$ and $|\overline{AB}|_Y$ respectively (see Figure

Figure 3.3: Nodes A and B , and $R(AB)$, $|\overline{AB}|$, $|\overline{AB}|_X$, $|\overline{AB}|_Y$.

3.3(a)). Then we denote by $R(AB)$ the closed area covered by the rectangle (or square), which passes through A and B and has two sides of length $|\overline{AB}|_X$ and two sides of length $|\overline{AB}|_Y$ (see Figure 3.3(b)). Then we have:

Lemma 1. *If $\overline{AB} \in E$, then no nodes of V are contained in $R(AB)$. Or, if any one node of V is contained in $R(AB)$, then $\overline{AB} \notin E$.*

Proof. All of the closed squares passing through A and B contain $R(AB)$. If $R(AB)$ contains some nodes of V , then those closed squares must contain these nodes too. Consequently, we cannot find any one “empty square” passing through A and B . By the Definition 7 of $STG(V, E)$, $\overline{AB} \notin E$. ■

Lemma 2. *The edge $\overline{AB} \in E$ if and only if there exists at least an “empty square” with side length $\max(|\overline{AB}|_X, |\overline{AB}|_Y)$ that passes through A and B , where $\max(|\overline{AB}|_X, |\overline{AB}|_Y)$ represents the larger of $|\overline{AB}|_X$ and $|\overline{AB}|_Y$. Such a square has the smallest side length among all of the squares passing through A and B , so it is called the “minimum empty square” corresponding to $\overline{AB} \in E$.*

Proof. [Sufficiency]. If there exists an “empty square” passing through A and B and has a side length $\max(|\overline{AB}|_X, |\overline{AB}|_Y)$, by the Definition 7 of $STG(V, E)$, then $\overline{AB} \in E$.

[Necessity]. If $\overline{AB} \in E$, then according to the Definition 7 of $STG(V, E)$, there exists at least one “empty square” passing through A and B . We assume its side

Figure 3.4: $\overline{AB} \in E$ implies $R(AB)$ can transform into an “empty square” passing through A and B with side length d .

length d . According to Lemma 1, $R(AB)$ is an “empty rectangle” and it is contained in the “empty square” using d as its side length. Hence, $d \geq \max(|\overline{AB}|_X, |\overline{AB}|_Y) \geq \min(|\overline{AB}|_X, |\overline{AB}|_Y)$, where $\min(|\overline{AB}|_X, |\overline{AB}|_Y)$ represents the smaller of $|\overline{AB}|_X$ and $|\overline{AB}|_Y$. Considering the edge $\overline{AB}$ (or the line segment between A and B) is a diagonal of $R(AB)$, if the two shorter sides of $R(AB)$, i.e. $\min(|\overline{AB}|_X, |\overline{AB}|_Y)$, equidirectionally and equidistantly extend along X -axis or Y -axis until arriving at the length $\max(|\overline{AB}|_X, |\overline{AB}|_Y)$, then the transformed $R'(AB)$ has actually become a square with a side length $\max(|\overline{AB}|_X, |\overline{AB}|_Y)$ and still passes A and B (see Figure 3.4). Since $d \geq \max(|\overline{AB}|_X, |\overline{AB}|_Y)$, we can keep $R'(AB)$ inside of the initial “empty square” constantly during its transformation. Finally, the transformed $R'(AB)$ is also an “empty square” passing through A and B and having a side length $\max(|\overline{AB}|_X, |\overline{AB}|_Y)$. ■

Theorem 1. $STG(V, E)$ is a Euclidean planar graph.

Proof. Assume, for the purpose of contradiction, that $STG(V, E)$ is not a planar graph, then at least two edges in E , say $\overline{AB}$ and $\overline{CD}$, intersect at a point in the plane, say X , where no nodes of V are located. X is obviously a point in $R(AB)$ because X is on $\overline{AB}$. For the same reason, X is also a point in $R(CD)$. Thus, the rectangular regions $R(AB)$ and $R(CD)$ must have a common part, at least commonly containing the point X . In view of symmetry, there are only three possible cases for the common part of $R(AB)$ and $R(CD)$. In the following section they are discussed one by one.

Figure 3.5: $STG(V, E)$ is a planar graph.

CASE 1. Two or more corner points of rectangle $R(CD)$ are contained in rectangle $R(AB)$.

Thus, either C or D must be contained in $R(AB)$ because C, D are two corner points and $\overline{CD}$ is a diagonal of $R(CD)$. By Lemma 1, $\overline{AB} \notin E$ that contradicts our assumption $\overline{AB} \in E$.

CASE 2. One corner point, say M , of the rectangle $R(CD)$ is contained in rectangle $R(AB)$.

According to Lemma 1, our assumption $\overline{AB} \in E$ means neither C nor D locates at the corner point M or M 's diagonal point because $\overline{CD}$ is a diagonal of $R(CD)$. Consequently, C and D must locate at the rest two corner points of $R(CD)$ adjacent to M . Due to symmetry, when a corner point M of $R(CD)$ is contained in $R(AB)$, a corner point of $R(AB)$, say N , is also contained in $R(CD)$. Similar to the positions nodes C and D are at, nodes A and B occupy two corner points of $R(AB)$ adjacent to N (see Figure 3.5(a)).

Suppose the straight line passing through A and B divides the whole plane into two half-planes. According to our assumption that $\overline{AB}$ intersects $\overline{CD}$ at the point X , node C must lie in a half-plane different with node D . Based on following reasons: $\diamond$

M is a point in $R(AB)$, whereas both C and D are not in $R(AB)$, $\diamond \overline{MC}$ and $\overline{MD}$ consist of a horizontal line-segment and a vertical one, $R(CD)$ thus must contain a node either A or B . By Lemma 1, $\overline{CD} \notin E$, contradicting our assumption $\overline{CD} \in E$.

CASE 3. None of the 4 corner points of $R(CD)$ is contained in $R(AB)$.

It has been proved that $R(CD)$ and $R(AB)$ share a common part, therefore they must form a cross shape. Next, we consider the closed rectangle passing through nodes A, B, C and D . It has two sides of length $\max(|\overline{AB}|_X, |\overline{CD}|_X)$, and two sides of length $\max(|\overline{AB}|_Y, |\overline{CD}|_Y)$ (see Figure 3.5(b)). W.l.o.g., we assume $\max(|\overline{AB}|_X, |\overline{CD}|_X) \geq \max(|\overline{AB}|_Y, |\overline{CD}|_Y)$. Then, all the closed squares passing through the two nodes (assuming C and D) that lie on the two vertical sides of that closed rectangle must contain at least one node (A, B or both) lying on a horizontal side of that closed rectangle (see Figure 3.5(b)). According to the Definition 7 of $STG(V, E)$, $\overline{CD} \notin E$ that contradicts our assumption $\overline{CD} \in E$.

To sum up, two arbitrary edges of E can only intersect at some node of V . Therefore, graph $STG(V, E)$ is a Euclidean planar graph. ■

Theorem 2. $STG(V, E)$ is a connected graph and a $\sqrt{20}$ -spanner of $K(V)$, where $K(V)$ is a complete Euclidean graph on V .

Proof. For any two nodes of V , say A and B , we need to prove: (1) the existence of at least one path between A and B , and (2) at least one of these paths is not longer than $\sqrt{20}$ times the straight-line distance between A and B ($|\overline{AB}|$). If $\overline{AB} \in E$, the edge $\overline{AB}$ trivially satisfies conditions (1) and (2). So, we only discuss those paths between A and B when $\overline{AB} \notin E$ hereafter.

Following Case 1, we prove the existence of two paths between A and B whose lengths are no longer than $\sqrt{10}$ times $|\overline{AB}|$, when no nodes of V lie in $R(AB)$. Later in Case 2, we show even if some nodes (or a node) of V lie in $R(AB)$, the condition (1) and the condition (2) still hold for that pair of nodes, A and B .

CASE 1. Assume $R(AB)$ is an “empty rectangle”, then the paths between A and B are denoted by $p(AB)$.

Figure 3.6: That $\overline{AB} \notin E$ implies $|\overline{CD}|_X \leq |\overline{AB}|_Y$, by Lemma 2.

According to Lemma 2, our assumptions $R(AB)$ is empty and $\overline{AB} \notin E$ imply $|\overline{AB}|_X \neq |\overline{AB}|_Y$. W.l.o.g., we assume $|\overline{AB}|_Y > |\overline{AB}|_X$, then $\max(|\overline{AB}|_X, |\overline{AB}|_Y) = |\overline{AB}|_Y$.

Consider two nodes, C and D , such that $\diamond$ they locate outside of two perpendicular sides of $R(AB)$ respectively, $\diamond$ each of them is the node closest to the corresponding vertical side of $R(AB)$ at the respective side, $\diamond$ their projections to the Y -Axis are all between those of A 's and B 's. According to Lemma 2, $\overline{AB} \notin E$ implies $|\overline{CD}|_X \leq |\overline{AB}|_Y$ (see Figure 3.6).

Let $S_{ADB}(|\overline{AB}|_Y)$ denote the closed square passing through nodes A , D and B , and with side length $|\overline{AB}|_Y$ (see Figure 3.7(a)). Thus, $S_{ACB}(|\overline{AB}|_Y)$ similarly denotes the closed square passing through nodes A , C and B , and with side length $|\overline{AB}|_Y$ (see Figure 3.7(d)). Let $rB(AB)$ denote the partial boundaries of $S_{ADB}(|\overline{AB}|_Y)$ counterclockwise from A to B (see Figure 3.7(b)), then $lB(AB)$ symmetrically denotes the partial boundaries of $S_{ACB}(|\overline{AB}|_Y)$ clockwise from A to B (see Figure 3.7(e)).

It has been proven that $|\overline{CD}|_X \leq |\overline{AB}|_Y$, thus the part of $S_{ADB}(|\overline{AB}|_Y)$ right to $R(AB)$ must exist and is denoted by $rR(AB)$ (see Figure 3.7(c)). Here, $rR(AB)$ represents a closed rectangle if $|\overline{CD}|_X < |\overline{AB}|_Y$, yet $rR(AB)$ represents a line segment if $|\overline{CD}|_X = |\overline{AB}|_Y$. In a similar way, we let $lR(AB)$ denote the part of $S_{ACB}(|\overline{AB}|_Y)$ left to $R(AB)$ (see Figure 3.7(f)).

Next, the divide-and-conquer method is used to prove the existence of two $p(AB)$ s, whose intermediate nodes lie in (or on) $lR(AB)$ and $rR(AB)$ respectively, and their lengths ($|p(AB)|$) are shorter than their corresponding $|lB(AB)|$ or $|rB(AB)|$. W.l.o.g.,

Figure 3.7: Definitions of S_{ADB} , $rR(AB)$, $rB(AB)$, S_{ACB} , $lR(AB)$, $lB(AB)$.

Figure 3.8: Two diagonals of $S_{ADB}(\overline{AB}|_Y)$ divide the $rR(AB)$ into four or three regions

we only consider the path $p(AB)$ in $rR(AB)$, thus we only need to prove the Inequation (3.1). If (it should be) so, we Let β be the angle formed by $\overline{AB}$ and X -Axis, then we have $|rB(AB)| \leq (3 \sin \beta - \cos \beta) \times |\overline{AB}| \leq \sqrt{10}|\overline{AB}|$ (and $|lB(AB)| \leq \sqrt{10}|\overline{AB}|$). Finally, the inequation $|p(AB)| \leq \sqrt{10}|\overline{AB}|$ always holds for **Case 1**.

$$|p(AB)| < |rB(AB)| \quad (3.1)$$

Suppose two diagonals of $S_{ADB}(|\overline{AB}|_Y)$ divide the $rR(AB)$ into four (possible three) regions as shown in Figure 3.8, or they turn $rR(AB)$ into the only region 3 if $rR(AB)$ is a line segment. Considering the symmetry between region 1 and region 2, the path $p(AB)$ is discussed in Subcases 1.1, 1.2 and 1.3 based on the region where node C may appear. For all of these subcases, we will show that Inequation (3.1) always holds as follows.

Subcase 1.1. Node C appears in region 3 of $rR(AB)$ (see Figure 3.9(a)).

We compare the projection (to Y -Axis) of the line segment $\overline{AC}$ with the projection (to X -Axis) of the line segment $\overline{CD}$, then obviously $|\overline{CC}_1| \leq |\overline{CC}_3| \leq |\overline{AB}|_Y$. If we extend upward two shorter sides of $R(C_1C_3)$ until they are as long as $\overline{CC}_3$, then the new generated square $S_{CAD}(|\overline{CC}_3|)$ is a part of the closed rectangle passing through nodes A, B, C and D . This closed rectangle contains only four nodes that all lie at its boundary, therefore $S_{CAD}(|\overline{CC}_3|)$ can only contain nodes at its boundary accordingly. The following two facts guarantee an “empty square” passing through A and C can be found in $S_{CAD}(|\overline{CC}_3|)$: $\diamond$ neither A nor C locates at a corner point of $S_{CAD}(|\overline{CC}_3|)$, $\diamond$ A and C lie on two sides of $S_{CAD}(|\overline{CC}_3|)$ that are adjacent to each other. According to the Definition 7 of $STG(V, E)$, $\overline{AC} \in E$. Similarly, $|\overline{CC}_2| \leq |\overline{CC}_3| \leq |\overline{AB}|_Y$ implies $\overline{CB} \in E$. Putting together the edges $\overline{AC}$ and $\overline{CB}$ makes the path $p(AB)$.

Now, we compare the summed projections (to X -Axis) of the edges consisting of the path $p(AB)$, i.e. $\overline{AC}$ and $\overline{CB}$, with the summed projections (to X -Axis) of all horizontal line segments of $rB(AB)$. Simultaneously, the summed projections (to Y -Axis) of the edges consisting of the path $p(AB)$ are compared with the summed projections (to Y -Axis) of the vertical line segment of $rB(AB)$. The Inequation (3.1) trivially holds.

Subcase 1.2. Node C appears in region 1 or region 2 (see Figure 3.9(b)) of $rR(AB)$.

Figure 3.9: Edges on the path $p(AB)$ when C appears in region 3 or region 1

Due to symmetry, the appearances of node C in region 1 or in region 2 are the same case. W.l.o.g., suppose C appears in region 1. For the same reason as discussed in Subcase 1.1, $|\overline{CC_2}| \leq |\overline{CC_3}| \leq |\overline{AB}|_Y$ implies $\overline{CB} \in E$ (see Figure 3.9(b)).

It is noted $R(AC)$ is an closed empty rectangle just like $R(AB)$ is, thus our problem changes from determining the path $p(AB)$ into determining its subpath $p(AC)$. We also notice that, for any pair of nodes lying on $p(AB)$, every time their corresponding “ C ” appears in region 1 or region 2, the undetermined path between them will decompose into a determined edge (e.g. $\overline{CB}$) and a undetermined subpath (e.g. $p(AC)$). Therefore, determining the path $p(AB)$ can reasonably be treated as a “loop process” (see Figure 3.10).

[At the 0th step of the loop process, see Figure 3.10(a)].

We consider the path $p(AB)$, and denote nodes A, B, C and D as A_0, B_0, C_0 and D_0 .

[At the 1st step of the loop process, see Figure 3.10(b)].

Figure 3.10: C always appears in the region 1 or 2.

We consider the subpath $p(AC)$, and denote nodes A_0 and C_0 as A_1 and B_1 . Choose a node, which is the closest to the left vertical side of $R(A_1B_1)$ among those in $lR(A_1B_1)$, and denote it as D_1 . Another node, which is the closest to the right vertical side of $R(A_1B_1)$ among those in $rR(A_1B_1)$, is chosen and denoted as C_1 . Thereafter, if we keep choosing C_m within $rR(A_mB_m)$ and add it to $p(AB)$ as a intermediate node, our problem will change from determining a path between A_0 and B_0 , into determining a subpath between A_1 and B_1 , then into determining a subpath between A_2 and B_2 (see Figure 3.10(c)), and finally into determining a subpath between A_n and B_n , where $n > m \geq 0$, unless the following events occur:

- (I.I). C_m appears outside of $rR(A_mB_m)$, which implies $|\overline{C_mD_m}|_X > |\overline{A_mB_m}|_Y$,
- (I.II). C_m appears in region 3 of $rR(A_mB_m)$ or
- (I.III). C_m appears in region 4 of $rR(A_mB_m)$.

Subsubcase 1.2.1. Suppose events (I.I), (I.II) and (I.III) don't happen until the number in the loop process arrives at its possible maximum n_1 , where we cannot find any node of V that appears in $rR(A_{n_1}B_{n_1})$ and n_1 is trivially no larger than the number of nodes in $rR(AB)$. Put another way, C_m always appears in region 1 or region 2 of $rR(A_mB_m)$ if $m \leq n_1 - 1$ (see Figure 3.11(a)).

At the n_1 th step of the loop process, we obviously have $|\overline{C_{n_1}D_{n_1}}|_X > |\overline{AB}|_Y > |\overline{A_{n_1}B_{n_1}}|_Y$, and this implies $\overline{A_{n_1}B_{n_1}} \in E$ (by Lemma 2). Therefore, a path between A and B including the edge $\overline{A_{n_1}B_{n_1}}$ is determined. What we actually have is the event

(I.I) happening between A_{n_1} and B_{n_1} . This is because we show (in Subsubcase 1.2.2) that the Inequation (3.1) holds when event (I.I) happens at the n th step of the loop process ($n < n_1$) and this Subsubcase 1.2.1 is totally covered by the latter analysis. We will not do the same proof here.

Subsubcase 1.2.2. During the loop process to determine $p(AB)$, suppose an event (I.I) happens between A_n and B_n at the n th ($n < n_1$) step, before the loop number arrives at its possible maximum n_1 . In other words, C_n appears outside of $rR(A_nB_n)$, then the loop number arrives at its actual maximum n (see Figure 3.11(b)).

At the n th step of the loop process, similar to Subsubcase 1.2.1 we $\overline{A_nB_n} \in E$. Therefore, a path between A and B including the edge $\overline{A_nB_n}$ is determined here.

According to the method and order we choose “ C ” and “ D ” at each step, node D_n must be located left to D_{n-1} , where $1 \leq n \leq n_1$. In addition, the sides of square $S_{A_nD_nB_n}(\overline{A_nB_n}|_Y)$ are shorter than those of $S_{A_{n-1}D_{n-1}B_{n-1}}(\overline{A_{n-1}B_{n-1}}|_Y)$. Thus, $rB(A_nB_n)$ is enclosed in a shape formed by $rB(A_{n-1}B_{n-1})$ and “line segment” $\overline{A_{n-1}B_{n-1}}$ (refer to Figure 3.10(a), Figure 3.10(b) and Figure 3.10(c)). Now, we compare the summed projections (to X -Axis) of the edges consisting of the path $p(AB)$ with the summed projections (to X -Axis) of $rB(AB)$. Simultaneously, we compare the summed projections (to Y -Axis) of the edges that consist of the path $p(AB)$ with the summed projections (to Y -Axis) of $rB(AB)$. The Inequation (3.1) trivially holds.

Subsubcase 1.2.3. During the loop process to determine $p(AB)$, suppose an event (I.II) happens between A_n and B_n at the n th ($n < n_1$) step of the loop process— C_n appears in region 3 of $rR(A_nB_n)$ (see Figure 3.11(c)).

According to our analysis in Subcase 1.1, $\overline{A_nC_n} \in E$ and $\overline{C_nB_n} \in E$. Thus, a path $p(AB)$ between A and B including $\overline{A_nC_n}$ and $\overline{C_nB_n}$ is determined.

Note $rB(A_nB_n)$ is also enclosed by a shape formed by $rB(A_{n-1}B_{n-1})$ and “line segment” $\overline{A_{n-1}B_{n-1}}$ in this case. Therefore, we still use the “projection method” (to X -Axis and to Y -Axis respectively), then the Inequation (3.1) still holds.

(a) Event (I.I) happens between A_{n_1} and B_{n_1} at the n_1 th step of the loop process $\Rightarrow \overline{A_{n_1}B_{n_1}} \in E \Rightarrow$ a path between A and B including edge $\overline{A_{n_1}B_{n_1}}$ is determined

(b) Event (I.I) happens between A_n and B_n at the n th step of the loop process $\Rightarrow \overline{A_nB_n} \in E \Rightarrow$ a path between A and B including edge $\overline{A_nB_n}$ is determined

(c) Event (I.II) happens between A_n and B_n at the n th step of the loop process $\Rightarrow \overline{A_nC_n} \in E, \overline{C_nB_n} \in E \Rightarrow$ a path between A and B including edge $\overline{A_nC_n}$ and edge $\overline{C_nB_n}$ is determined

Figure 3.11: C always appears in the region 1 or 2 until event (I.I) or event (I.II) happens at some step of the loop process.

Subsubcase 1.2.4. During the loop process that determines $p(AB)$, suppose an event (I.III) happens between A_n and B_n at the n th ($n < n_1$) step of the loop process— C_n appears in region 4 of $rR(A_nB_n)$ (see Figure 3.11(b)).

For this Subsubcase 1.2.4, the proof that path $p(A_nB_n)$ exists and Inequation (3.1) holds is covered by the following Subcase 1.3.

Subcase 1.3. Region 4 of $rR(AB)$ exists and node C appears in region 4 of $rR(AB)$ (see Figure 3.12).

In order to determine the path $p(AB)$, we first let $|\overline{AB}|_Y = h$ and $|\overline{DB}|_X = w$, and then introduce a “thickness factor” u for nodes A and B , where $u = w/h$ (see Figure 3.12(a)). Rectangles $R(AC)$ and $R(CB)$ are “empty” just like $R(AB)$ is. Therefore the path the path $p(AB)$ to be determined accordingly decomposes into its two undetermined subpaths, i.e. $p(AC)$ and $p(CB)$. To determine $p(AC)$ in a similar way, we also let $|\overline{AC}|_Y = h'$ and $|\overline{D'C}|_X = w'$, then the thickness factor of nodes A and C is accordingly $u' = w'/h'$ (see Figure 3.12(b)). Even if the node D' , a node closest to the right side of $R(AC)$ among those in $lR(AC)$, could be exactly the node D or another node in $lR(AC)$. It always holds $u' = \frac{w'}{h'} > \frac{w}{h-w} = \frac{w/h}{1-w/h} = \frac{u}{1-u}$ no matter which condition node D' is in. For the same reason, the thickness factor u'' for nodes C and B also satisfies $u'' > \frac{u}{1-u}$ (see Figure 3.12(c)). If u_1 stands for either u' or u'' , and u_0 stands for u , then Formula (3.2) always holds even if u' is not necessarily equal to u'' .

$$u_1 > \frac{u_0}{1 - u_0} \quad (3.2)$$

We notice, for a pair of nodes lying on $p(AB)$, every time their corresponding “ C ” appears in region 4, the undetermined path between them will decompose into its two undetermined subpaths, and an isolated node “ C ”. In this Subcase 1.3, such an isolated node “ C ” is defined as a “inflexed node” (see Figure 3.13), and for convenience two end nodes A and B are also defined as inflexed nodes. Using this assumption, determining the path $p(AB)$ can reasonably be treated as a “recursive process”, and this “recursive process” will form a corresponding “recursion tree”. We introduce some notations in Subcase 1.3 to handle this “recursive process” that are different from those used in the “loop process” in Subcase 1.2. In the following part, we first use the divide-and-conquer method and follow a “top-down” order to prove

(a) Definitions of “thickness factor”, u , and “inflexed node”, C , for A and B where $u = \frac{w}{h}$

(b) The “thickness factor”, u' , for A and C such that $u' = \frac{w'}{h'} > \frac{u}{1-u}$

(c) The “thickness factor”, u'' , for C and B such that $u'' = \frac{w''}{h''} > \frac{w}{h-w} = \frac{w/h}{1-w/h} = \frac{u}{1-u}$

Figure 3.12: Use u_1 to stand for either u' or u'' , thus $u_1 > \frac{u_0}{1-u_0}$ holds.

for $(A_{20}^1, B_{20}^1) : u_0 < 0.5$

(a) (Top-down in the recursion depth 0) Thickness factor u_0 for (A_{20}^1, B_{20}^1) such that $u_0 < 0.5$

for $(A_{21}^1, B_{21}^1), (A_{21}^2, B_{21}^2) : u_1 < 0.5$

(b) (Top-down in the recursion depth 1) Thickness factor u_1 for (A_{21}^1, B_{21}^1) or (A_{21}^2, B_{21}^2) such that $u_1 < 0.5$

Figure 3.13: Follow a top-down order to prove the existence of the path $p(AB)$ while C recursively appears in the region 4.

the existence of the path $p(AB)$, and then follow a “bottom-up” order to prove the Inequation (3.1) holds for this Subcase 1.3.

Subsubcase 1.3.1. Use the divide-and-conquer method and follow a “top-down” order to prove the existence of the path $p(AB)$.

[In the recursion depth 0, see Figure 3.13(a)].

We consider the path $p(AB)$, and denote A, B and their thickness factor as A_{20}^1, B_{20}^1 and u_0 , and denote C and D as C_{20}^1 and D_{20}^1 respectively.

[In the recursion depth 1, see Figure 3.13(b)].

We first consider the subpath $p(AC)$, and denote nodes A, C as A_{21}^1, B_{21}^1 respectively. Choose a node, closest to the left vertical side of $R(A_{21}^1 B_{21}^1)$ among those in $lR(A_{21}^1 B_{21}^1)$, and denote it as D_{21}^1 . Choose another node, closest to the right vertical side of $R(A_{21}^1 B_{21}^1)$ among those in $rR(A_{21}^1 B_{21}^1)$, and denote it as C_{21}^1 . Symmetrically, we then consider $p(CB)$, and let C and B be A_{21}^2, B_{21}^2 . Similarly, we can find C_{21}^2 and D_{21}^2 . If u_1 stands for either the thickness factor for nodes A_{21}^1 and B_{21}^1 or the

thickness factor for nodes A_{21}^2 and B_{21}^2 , it follows that Formula (3.2) holds.

To clearly understand the recursion depth and notations related to it, we observe the positions where C_{21}^1 and C_{21}^2 could appear as follows.

Suppose C_{21}^2 appears in region 4 of $rR(A_{21}^2 B_{21}^2)$ as shown in Figure 3.13(b). Then the pair of nodes, denoted by A_{21}^2 and B_{21}^2 in the depth 1, clearly enter the next depth 2. At this point and afterwards, we always follow an “A-to-B” order to consider a sequence of undetermined subpaths of $p(AB)$ in every recursion depth and number their corresponding end-node pairs in an ascending manner. Thus A_{21}^2 and B_{21}^2 are re-denoted by A_{22}^3 , B_{22}^4 , where the base 2 of their subscripts indicates the recursion depth they are currently in, and their superscripts 3 and 4 indicate their sequence numbers among all end-node pairs re-denoted in the same depth.

Suppose C_{21}^2 appears outside of $rR(A_{21}^2 B_{21}^2)$, which implies $|\overline{C_{21}^2 D_{21}^2}|_X > |\overline{A_{21}^2 B_{21}^2}|_Y$. By Lemma 2, $\overline{A_{21}^2 B_{21}^2} \in E$, then $p(A_{21}^2 B_{21}^2)$ consists of the only edge $\overline{A_{21}^2 B_{21}^2}$ that is determined. Or, suppose C_{21}^2 appears in region 3 of $rR(A_{21}^2 B_{21}^2)$. According to Subcase 1.1, $p(A_{21}^2 B_{21}^2)$ consists of the two edges, $\overline{A_{21}^2 C_{21}^2}$ and $\overline{C_{21}^2 B_{21}^2}$, that are determined. In both of these examples, the pair of nodes, denoted by A_{21}^2 and B_{21}^2 in the depth 1, cannot clearly enter and thus be re-denoted in the next depth 2. Their absence from all pairs of end nodes to be re-denoted in the next depth 2 leads to a sequence of discontinuous superscripts for those node-pairs.

Suppose C_{21}^1 appears in region 2 (or region 1) of $rR(A_{21}^1 B_{21}^1)$ as shown in Figure 3.13(b). According to Subcase 1.2, $\overline{A_{21}^1 C_{21}^1} \in E$, we compare the path $p(A_{21}^1 B_{21}^1)$ that was entirely undetermined before C_{21}^1 is found and its subpaths $p(C_{21}^1 B_{21}^1)$ that are entirely undetermined. Based on following reasons, $\diamond C_{21}^1$ is not in the region 4 of $rR(A_{21}^1 B_{21}^1)$, $\diamond$ there is still one undetermined subpath between A_{21}^1 and B_{21}^1 instead of two, and $\diamond$ the thickness factor for two end nodes of $p(C_{21}^1 B_{21}^1)$ is larger than for two end nodes of $p(A_{21}^1 B_{21}^1)$ (u_1) and hence is larger than u_0 too, node-pair (C_{21}^1, B_{21}^1) can be reasonably viewed as in the same depth 1 as node-pair (A_{21}^1, B_{21}^1) . Therefore, in terms of the recursion depth 1, C_{21}^1 is an “equivalent-node” of A_{21}^1 . Hereafter, every newfound C_{21}^1 will be added to either of two series of equivalent-nodes respectively of A_{21}^1 and B_{21}^1 , until the latest-found pair of equivalent-nodes of A_{21}^1 and B_{21}^1 surely enter in or stop before the next depth. From the original node A_{21}^1 to its last-added equivalent-node inclusive, a series of equivalent-nodes of A_{21}^1 are

Figure 3.14: Emanating nodes A_{21}^1 and B_{21}^1 surely enter in or stop before the next recursion depth 2.

denoted by $A_{21}^1[0], A_{21}^1[1], \dots, A_{21}^1[\kappa_A]$, with κ_A standing for the number of entire equivalent-nodes of A_{21}^1 (see Figure 3.14(a)). $B_{21}^1[0], B_{21}^1[1], \dots, B_{21}^1[\kappa_B]$ are similar. For consistency of notation and concepts in the rest, part of this Subcase 1.3, A_{21}^1 is redefined to denote an “emanating node” that is either the original node A_{21}^1 or the isolated edge (or path) emanating from the original node A_{21}^1 and passing through its entire equivalent-nodes one by one, and so is B_{21}^1 . However, and with no special explanation given, a “emanating node” A_{21}^1 (or B_{21}^1) is assumed to be a node, i.e. $A_{21}^1[0] = A_{21}^1[\kappa_A]$ (or $B_{21}^1[0] = B_{21}^1[\kappa_B]$). Trivially, emanating nodes A_{21}^1, B_{21}^1 still stay in the depth 1.

The above process happening in the depths 0 and 1—some emanating-node-pairs have their respective inflexed nodes, and thus enter and are re-denoted (with their newfound inflexed nodes) in the next depth—may repeat in every recursion depth

until, in some depth n , no more inflexed nodes can be found for all emanating-node-pairs entering in this depth. Such a n (if exists) is the maximum depth of the recursion. Those emanating-node-pairs, for which no more inflexed nodes can be found, are on the leaves of the recursion tree. According to our analysis of the node-pairs in the depth 1, all emanating-node-pairs on the leaves of the recursion tree could scatter in different depths n' , where either of these two events must happen on each of them (see Figure 3.14(b) and Figure 3.14(c)):

- (II.I). $C_{2^{n'}}^t$ ($1 \leq n' \leq n$, $2^0 \leq t \leq 2^{n'}$) appears in region 3 of $rR(A_{2^{n'}}^t[\kappa_A]B_{2^{n'}}^t[\kappa_B])$,
- (II.II). $C_{2^{n'}}^t$ appears outside of $rR(A_{2^{n'}}^t[\kappa_A]B_{2^{n'}}^t[\kappa_B])$.

Recall it has been proven in Subcase 1.1 and Subcase 1.2 that such subpaths as $\overline{A_{2^{n'}}^t[0]B_{2^{n'}}^t[0]}$ are easily determined, then put them together with those isolated edges or paths defined as emanating nodes in every recursion depth to make the path $p(AB)$.

Until now, to prove a path's existence $p(AB)$, only the recursive process has a bounded maximum depth n needs to be proven. We consider any pair of nodes entering in the recursion depth m , $A_{2^m}^t$ and $B_{2^m}^t$ ($1 \leq m \leq n$, $2^0 \leq t \leq 2^m$). According to Formula (3.2), their thickness factor u_m such that $u_m > \frac{u_{m-1}}{1-u_{m-1}}$. By applying simple mathematical knowledge to this "recursion formula", we have $u_m > \frac{u_0}{1-mu_0} = \frac{w/h}{1-m(w/h)}$. Let $f(x) = \frac{w/h}{1-x(w/h)}$, then the function $f(x)$ is obviously monotonically increasing in x because $\frac{df(x)}{dx} = (\frac{w/h}{1-x(w/h)})^2 > 0$. Therefore, a sequence of thickness factors, $\{u_m\}$, also monotonically increase along with the recursion depth m .

On the one hand, if a node $C_{2^m}^t$ is inflexed, then the thickness factor u_m of its corresponding two nodes, $A_{2^m}^t$ and $B_{2^m}^t$, must be less than 0.5; otherwise $C_{2^m}^t$ cannot appear in the region 4 of $rR(A_{2^m}^tB_{2^m}^t)$ (see Figure 3.15). For example, initially C ($C_{2^0}^1$) is an inflexed node, thus the thickness factor for the two nodes A ($A_{2^0}^1$) and B ($B_{2^0}^1$) is such that $u_0 = w/h < 0.5$. On the other hand, the thickness factor sequence $\{u_m\}$ monotonically increases along with the recursion depth m . Suppose none of the events (II.I) and (II.II) happens between every pair of nodes $A_{2^m}^t$ and $B_{2^m}^t$ until, in some recursion depth, the thickness factor u_m increases up to a value no less than 0.5. When this happens, every pair of nodes $A_{2^m}^t$ and $B_{2^m}^t$ cannot have an inflexed node and are thus in the maximum recursion depth n (i.e. $m = n$), and n also achieves its upper bound.

Let $f(x) = 0.5$, then $x = \frac{h}{w} - 2$. Suppose all pair of nodes $A_{2^m}^t$ and $B_{2^m}^t$ enter

Figure 3.15: The maximum depth n is bounded by a constant $\lceil \frac{h}{w} - 2 \rceil$

in the depth $m = \lceil \frac{h}{w} - 2 \rceil$, then the thickness factor for any pair of nodes in this depth are such that $u_m = u_{\lceil \frac{h}{w}-2 \rceil} \geq f(\lceil \frac{h}{w} - 2 \rceil) \geq 0.5$. Clearly, n must satisfy $n \leq \lceil \frac{h}{w} - 2 \rceil$, therefore the recursion is thereby proved to have the maximum depth n that is bounded by a constant $\lceil \frac{h}{w} - 2 \rceil$.

Subsubcase 1.3.2. Follow a “bottom-up” order (from maximum recursion depth upward till 0) to joint in pairs all the subpaths in every recursion depth; meanwhile prove by “mathematical induction” that Inequation (3.1), $|p(AB)| < |rB(AB)|$, holds throughout the entire “subpath-jointing” procedure, and finally also holds for the SUBcase 1.3.

[Basic]. For any pair of nodes, denoted by $A_{2^{n'}}^t$ and $B_{2^{n'}}^t$, in the depth n' (n is the maximum recursion depth, $1 \leq n' \leq n$, $2^0 \leq t \leq 2^{n'}$), the Inequation $|p(A_{2^{n'}}^t[0]B_{2^{n'}}^t[0])| < |rB(A_{2^{n'}}^t[0]B_{2^{n'}}^t[0])|$ holds if they are on a leaf of the recursion tree (see Figure 3.16).

That emanating nodes $A_{2^{n'}}^t$ and $B_{2^{n'}}^t$ cannot enter and thus be re-denoted in the next depth implies either event (II.I) or event (II.II) must happen between $A_{2^{n'}}^t[\kappa_A]$ and $B_{2^{n'}}^t[\kappa_B]$. It has been proven in Subcase 1.1 and Subcase 1.2 that the

Figure 3.16: Inequation (3.1) holds for any pair of nodes that are on a leaf of the recursion tree.

inequation $|p(A_{2n'}^t[0]B_{2n'}^t[0])| < |rB(A_{2n'}^t[0]B_{2n'}^t[0])|$ holds for both events. $\square$

[Inductive Step]. Suppose any pair of nodes, denoted by $A_{2^{m-1}}^t$ and $B_{2^{m-1}}^t$ in the depth $m-1$ (n is the maximum recursion depth, $1 \leq m \leq n$, $2^0 \leq t \leq 2^{m-1}$), also enter in the next recursion depth m . Then, the resulting two pairs of nodes are denoted by $(A_{2^m}^{2t-1}, B_{2^m}^{2t-1})$, $(A_{2^m}^{2t}, B_{2^m}^{2t})$ in the depth m such that $A_{2^m}^{2t-1} = A_{2^{m-1}}^t[\kappa_A]$, $B_{2^m}^{2t-1} = A_{2^m}^{2t}$ and $B_{2^m}^{2t} = B_{2^{m-1}}^t[\kappa_B]$. If Inequations (3.3) and (3.4) simultaneously hold for their respective $(A_{2^m}^{2t-1}, B_{2^m}^{2t-1})$ and $(A_{2^m}^{2t}, B_{2^m}^{2t})$, then Inequation (3.5) also holds for $A_{2^{m-1}}^t[0]$ and $B_{2^{m-1}}^t[0]$ (see Figure 3.17).

$$|p(A_{2^m}^{2t-1}B_{2^m}^{2t-1})| < |rB(A_{2^m}^{2t-1}B_{2^m}^{2t-1})| \quad (3.3)$$

$$|p(A_{2^m}^{2t}B_{2^m}^{2t})| < |rB(A_{2^m}^{2t}B_{2^m}^{2t})| \quad (3.4)$$

$$|p(A_{2^{m-1}}^t[0]B_{2^{m-1}}^t[0])| < |rB(A_{2^{m-1}}^t[0]B_{2^{m-1}}^t[0])| \quad (3.5)$$

Figure 3.17: Inequation (3.1) holds for any pair of nodes in the depth m .

To prove Inequation (3.5), first we will prove that Inequation (3.6) holds for the node-pair $(A_{2^{m-1}}^t[\kappa_A], B_{2^{m-1}}^t[\kappa_B])$ (see Figure 3.17(a)), where

- ◇ $C_{2^{m-1}}^t$, which is chosen for $A_{2^{m-1}}^t[\kappa_A]$ and $B_{2^{m-1}}^t[\kappa_B]$, is a inflexed node,
- ◇ $A_{2^{m-1}}^t[\kappa_A], B_{2^{m-1}}^t[\kappa_B]$ are two nodes actually entering into the next depth m , and
- ◇ $A_{2^{m-1}}^t[\kappa_A] = A_{2^m}^{2t-1}, C_{2^{m-1}}^t = B_{2^m}^{2t-1} = B_{2^m}^{2t}$, and $B_{2^{m-1}}^t[\kappa_B] = B_{2^m}^{2t}$.

$$|p(A_{2^{m-1}}^t[\kappa_A]B_{2^{m-1}}^t[\kappa_B])| < |rB(A_{2^{m-1}}^t[\kappa_A]B_{2^{m-1}}^t[\kappa_B])| \quad (3.6)$$

We observe the positions of three “D”s corresponding to the three node-pairs, $(A_{2^{m-1}}^t[\kappa_A], B_{2^{m-1}}^t[\kappa_B])$, $(A_{2^{m-1}}^{2t-1}, B_{2^{m-1}}^{2t-1})$ and $(A_{2^m}^{2t}, B_{2^m}^{2t})$. According to the method and order in which they are chosen, nodes $D_{2^m}^{2t-1}$ and $D_{2^m}^{2t}$ must lie left to node $D_{2^{m-1}}^t$. Now, we consider the worst condition, the projections (to X -Axis) of nodes $D_{2^m}^{2t-1}$

and $D_{2^m}^{2t}$ are extremely close to the projection (to X -Axis) of node $D_{2^{m-1}}^t$, which implies $|rB(A_{2^m}^{2t-1}B_{2^m}^{2t-1})|$ and $|rB(A_{2^m}^{2t}B_{2^m}^{2t})|$ are close to their respective maximums. Thus, $D_{2^m}^{2t-1}$, $D_{2^m}^{2t}$ and $D_{2^{m-1}}^t$ have almost the same X -coordinate. Referring to the Figure 3.17(a), we have:

$$\left. \begin{aligned} & \left| \overline{P_6P_7} \right| < \left| \overline{P_5P_7} \right| = \left| \overline{P_3P_7} \right| \\ & \left| \overline{P_9P_{11}} \right| = \left| \overline{P_8P_{11}} \right| \\ & \left| \overline{P_4P_{12}} \right| = \left| \overline{P_9P_{12}} \right| = \left| \overline{P_9P_{11}} \right| + \left| \overline{P_{11}P_{12}} \right| \\ & \Rightarrow \left| \overline{P_6P_7} \right| < \left| \overline{P_{11}P_{12}} \right| \\ & \Rightarrow \left| \overline{P_{10}P_{11}} \right| + \left| \overline{P_6P_7} \right| < \left| \overline{P_{10}P_{11}} \right| + \left| \overline{P_{11}P_{12}} \right| = \left| \overline{P_{10}P_{12}} \right|. \end{aligned} \right\} \Rightarrow$$

It follows that Inequation (3.7) holds, and in the same way Inequation (3.8) does the same.

$$\left| \overline{P_{10}P_{11}} \right| + \left| \overline{P_6P_7} \right| < \left| \overline{P_{10}P_{12}} \right| \quad (3.7)$$

$$\left| \overline{P_2P_3} \right| + \left| \overline{P_6P_8} \right| < \left| \overline{P_2P_4} \right| \quad (3.8)$$

Substitute Inequations (3.3), (3.4), (3.7) and (3.8) into $|p(A_{2^{m-1}}^t[\kappa_A]B_{2^{m-1}}^t[\kappa_B])|$ as follows:

$$\begin{aligned} & |p(A_{2^{m-1}}^t[\kappa_A]B_{2^{m-1}}^t[\kappa_B])| \\ &= |p(A_{2^m}^{2t-1}B_{2^m}^{2t-1})| + |p(A_{2^m}^{2t}B_{2^m}^{2t})| \\ &\leq |rB(A_{2^m}^{2t-1}B_{2^m}^{2t-1})| + |rB(A_{2^m}^{2t}B_{2^m}^{2t})| \quad \text{Inequations (3.3), (3.4) substituted} \\ &= (\left| \overline{P_{10}P_{11}} \right| + \left| \overline{P_8P_{11}} \right| + \left| \overline{P_6P_8} \right|) + (\left| \overline{P_6P_7} \right| + \left| \overline{P_3P_7} \right| + \left| \overline{P_2P_3} \right|) \\ &= (\left| \overline{P_{10}P_{11}} \right| + \left| \overline{P_6P_7} \right|) + (\left| \overline{P_2P_3} \right| + \left| \overline{P_6P_8} \right|) + (\left| \overline{P_8P_{11}} \right| + \left| \overline{P_3P_7} \right|) \\ &< \left| \overline{P_{10}P_{12}} \right| + \left| \overline{P_2P_4} \right| + \left| \overline{P_4P_{12}} \right| \quad \text{Inequations (3.7), (3.8) substituted} \\ &= |rB(A_{2^{m-1}}^t[\kappa_A]B_{2^{m-1}}^t[\kappa_B])|. \end{aligned}$$

Thus, Inequation (3.6) has been proven to hold.

Let the projections (to X -Axis) of nodes $A_{2^{m-1}}^t[\kappa_A]$ and $B_{2^{m-1}}^t[\kappa_B]$ be P_A and P_B respectively (see Figure 3.17(b)). We notice that emanating node $A_{2^{m-1}}^t$ could denote (a) a node $A_{2^{m-1}}^t[0]$, where $A_{2^{m-1}}^t[0] = A_{2^{m-1}}^t[\kappa_A]$, or (b) a path $p(A_{2^{m-1}}^t[0]A_{2^{m-1}}^t[\kappa_A])$ passing through nodes $A_{2^{m-1}}^t[0], A_{2^{m-1}}^t[1], \dots, A_{2^{m-1}}^t[\kappa_A]$. If $A_{2^{m-1}}^t$ is in above case (b), then we still use the “projection method” (to X -Axis and to Y -Axis respectively) to consider the path $p(A_{2^{m-1}}^t[0]A_{2^{m-1}}^t[\kappa_A])$. Clearly, following the Inequation (3.9) always holds even if $A_{2^{m-1}}^t$ is in above case (a) (i.e. $0 \leq 0$).

A similar relationship, shown as following Inequation (3.10), also holds for the emanating node $B_{2^{m-1}}^t$.

$$\left| p(A_{2^{m-1}}^t[0]A_{2^{m-1}}^t[\kappa_A]) \right| \leq \left| \overline{P_A A_{2^{m-1}}^t[0]} \right| + \left| \overline{P_A A_{2^{m-1}}^t[\kappa_A]} \right| \quad (3.9)$$

$$\left| p(B_{2^{m-1}}^t[0]B_{2^{m-1}}^t[\kappa_B]) \right| \leq \left| \overline{P_B B_{2^{m-1}}^t[0]} \right| + \left| \overline{P_B B_{2^{m-1}}^t[\kappa_B]} \right| \quad (3.10)$$

Substitute Inequations (3.6), (3.9) and (3.10) into $\left| p(A_{2^{m-1}}^t[0]B_{2^{m-1}}^t[0]) \right|$ as follows:

$$\begin{aligned} & \left| p(A_{2^{m-1}}^t[0]B_{2^{m-1}}^t[0]) \right| \\ = & \left| p(A_{2^{m-1}}^t[0]A_{2^{m-1}}^t[\kappa_A]) \right| + \left| p(A_{2^{m-1}}^t[\kappa_A]B_{2^{m-1}}^t[\kappa_B]) \right| + \left| p(B_{2^{m-1}}^t[0]B_{2^{m-1}}^t[\kappa_B]) \right| \\ < & (\left| \overline{P_A A_{2^{m-1}}^t[0]} \right| + \left| \overline{P_A A_{2^{m-1}}^t[\kappa_A]} \right|) + \left| rB(A_{2^{m-1}}^t[\kappa_A]B_{2^{m-1}}^t[\kappa_B]) \right| + (\left| \overline{P_B B_{2^{m-1}}^t[0]} \right| \\ & + \left| \overline{P_B B_{2^{m-1}}^t[\kappa_B]} \right|) \quad \text{Inequations (3.9), (3.6), (3.10) substituted} \\ < & \left| rB(A_{2^{m-1}}^t[0]B_{2^{m-1}}^t[0]) \right|. \end{aligned}$$

Thus, Inequation (3.5) is proved to hold. $\square$

Now, we follow a “bottom-up” order (from the maximum recursion depth n upward until 0) to joint in pairs together the subpaths in each depth. Based on the proof by “mathematical induction” as above, Inequation (3.1) holds throughout the entire subpath-jointing procedure, thus for this Subcase 1.3.

CASE 2. Assume $R(AB)$ is not “empty”, then the paths between A and B are denoted by $p(A \sim B)$.

According to Lemma 1, that some nodes (or a node) of V are contained in $R(AB)$ implies $\overline{AB} \notin E$. Therefore $p(A \sim B)$, if in existence, must contain other nodes of V . Our idea to prove the existence of $p(A \sim B)$ is to choose nodes from $R(AB)$ in a “A-to-B” order as the intermediate nodes. And we simultaneously guarantee any two adjacent intermediate nodes are connected. Such intermediate nodes are defined as “junction nodes” (see Figure 3.18). For convenience, two end-nodes A and B are also defined in the same way.

The following **algorithm** is used to create an ordered set Q of all junction nodes from A to B , and node O represents the current node.

1. $Q \leftarrow \{A\}$ // initialization
2. $O \leftarrow A$
3. $i \leftarrow 1$ // define a counter

Figure 3.18: If $R(AB)$ is not empty, $p(A \sim B)$ is divided into many pieces to consider, and each resulting piece satisfies **CASE 1**.

4. choose the node O_i , which is the closest, among those in $R(OB)$, to the longer of the two sides of $R(OB)$ that passes through node O .
5. $Q \leftarrow Q \cup \{O_i\}$
6. **if** $O_i \neq B$
 then $O \leftarrow O_i$
 go back to 4th step
 else return Q

Choose any two adjacent junction nodes from Q , say O_i and O_{i+1} . According to the manner created Q in above algorithm, $R(O_i O_{i+1})$ is obviously an “empty rectangle”. Based on the proof in **CASE 1**, O_i and O_{i+1} must be connected, and the path between them, $p(O_i O_{i+1})$, such that $|p(O_i O_{i+1})| < |rB(O_i O_{i+1})| \leq \sqrt{10} |O_i O_{i+1}|$. Putting all such paths as $p(O_i O_{i+1})$ together makes the path $p(A \sim B)$.

To prove $|p(A \sim B)|$ is bounded, an ordered set of line segments needs to be considered connecting all pairs of adjacent junction nodes of Q in an “A-to-B” manner. This ordered set of line segments is denoted by $p(A-B)$ (see Figure 3.18).

Figure 3.19: The STG does not obey the short-edge replacement rule.

Let β be the angle formed by line segment $\overline{AB}$ and X -Axis, then we have $p(A-B) < |\overline{AB}| \sin \beta + |\overline{AB}| \cos \beta \leq \sqrt{2} |\overline{AB}|$. Based on the proof in **CASE 1**, $|p(A \sim B)| \leq \sqrt{10} |p(A-B)| \leq \sqrt{10} (\sqrt{2} |\overline{AB}|) = \sqrt{20} |\overline{AB}|$, i.e. Inequation $|p(A \sim B)| \leq \sqrt{20} |\overline{AB}|$ is proved to hold.

With the above proof in both **CASE 1** and **CASE 2**, a square test graph $STG(V, E)$ is connected and a $\sqrt{20}$ -spanner of $K(V)$, where $K(V)$ is a complete Euclidean graph on V . ■

3.4 Use Unit-square Graph as the Original Network Model

Now, we have a new proximity graph, square test graph (STG). Next, we want to apply STG to constructing underlying topologies for wireless ad hoc networks. The first-choice heuristic is to consider the graph $UDG \cap STG$. Unfortunately, we notice the STG does not follow the short-edge replacement rule. In fact, a $UDG \cap STG$, if used, is not necessarily connected, and neither a $\sqrt{20}$ -spanner of the UDG. The following example illustrates this.

Given a set V of wireless nodes distributing in the Euclidean plane with the established X -Axis and Y -Axis, two nodes A and B of V have an edge $|\overline{AB}|$ of $UDG(V)$ (i.e. $|\overline{AB}| \leq 1$), and $|\overline{AB}|_Y > |\overline{AB}|_X$ (see Figure 3.19). Other two wireless nodes C and D in this plane such that: $\diamond |\overline{CD}|_X < |\overline{AB}|_Y$, $\diamond |\overline{AC}| > 1$, $|\overline{AD}| > 1$, $\diamond$ node C (or D) is located inside the square that passing through nodes A , B and D (or C , correspondingly). Suppose node A is aware of the positions of C and D in some manner. According to the Definition 7 of STG, node A breaks the edge to

Figure 3.20: A comparison of communication range and transmission range between UDG network model and USG network model.

B because $\overline{AB} \notin STG(V)$. However, A and D (or C) do not have an edge of the $UDG(V)$ and they may not connect to each other in the $UDG(V)$. Consequently, the resulting network structure $UDG(V) \cap STG(V)$ is not necessarily connected. As another way of understanding, node A considers the short edge $\overline{AB}$ of $UDG(V)$ less essential than other two edges $\overline{AC}$ and $\overline{CB}$ (or $\overline{AD}$ and $\overline{DB}$), which include a long edge $\overline{AC}$ (or $\overline{AD}$) not belonging to $UDG(V)$. Clearly, the square test graph does not follow the short-edge replacement rule although this rule is required for a proximity graph if it is used to simplify the unit-disk graph.

To make the square test graph applicable to wireless ad hoc networks, we will introduce a new network model, named unit-square graph (USG). The definition follows.

Definition 8. Given a set V of nodes in the Euclidean plane with established X -Axis and Y -Axis, a unit-square graph $USG(V, E)$ is a graph defined on V such that any two nodes of V have a edge in $USG(V, E)$ if and only if their Euclidean distances along both X -Axis (horizontal projection) and Y -Axis (vertical projection) are at most $\frac{\sqrt{2}}{2}$ unit. On the same set V of nodes, the $USG(V, E)$ is a subgraph of the $UDG(V, E)$.

In this thesis, we reuse the term “communication range” for the USG network model to denote the “unit-square” centered at each of the wireless nodes and having

four sides of length $2 \times \frac{\sqrt{2}}{2}$ unit. We also reuse the term “transmission range” to denote the distance from each node to any boundary of the unit-square, which equals $\frac{\sqrt{2}}{2}$ unit. A comparison of communication range and transmission range between the general UDG network model and the USG network model is illustrated in Figure 3.20. The communication range of a wireless node is reasonably modeled as a unit-disk in realistic environment. In this thesis, it is also reasonably modeled as a unit-square because we can let each wireless node simply ignore the nodes appearing in its unit-disk but not in its unit-square.

Next, we follow the approach that defines STG to simplify a USG, and the following Theorem 3 guarantees the entire edge replacements happen totally among edges of the USG. Based on a common assumption that USG is always connected (derived from the similar assumption for UDG), therefore the resulting graph $USG \cap STG$ is guaranteed to be connected.

Theorem 3. *Given a set V of nodes in the Euclidean plane with established X -Axis and Y -Axis, if two arbitrary nodes A and B of V have an edge of the $USG(V)$, then they must have at least a path in $STG(V)$ consisting of the edges all belonging to $USG(V)$.*

Proof. In this proof, we still use $R(AB)$ to denote the closed rectangle using A and B as its two diagonal corner points, and use $\max(|\overline{AB}|_X, |\overline{AB}|_Y)$ to denote the larger of $|\overline{AB}|_X$ and $|\overline{AB}|_Y$ (refer to Figure 3.3(b)). A unique closed square (not necessarily empty) has node A at one of its corner points, and passes through A and B . We denote this closed square as $S_{AB}^{[A]}$, which clearly has four sides of length $\max(|\overline{AB}|_X, |\overline{AB}|_Y)$.

For the two arbitrary nodes A and B of V , we have proven, by the two cases of $R(AB)$ in Theorem 2, they must have at least a path in $STG(V)$. Next we show, for both of the two cases of $R(AB)$, the paths found in the proof of Theorem 2 consist of the edges all belonging to $USG(V)$ if $\overline{AB} \in USG(V)$ (i.e. $\max(|\overline{AB}|_X, |\overline{AB}|_Y) \leq \frac{\sqrt{2}}{2}$).

CASE 1. Assume $R(AB)$ is an “empty rectangle”.

According to the CASE 1 of Theorem 2, A and B have at least a path $p(AB)$ in $STG(V)$ consisting of all intermediate nodes contained in $S_{AB}^{[A]}$. We use $P_1, P_2, \dots, P_n$

to denote the n intermediate nodes from A to B inclusively, where $P_1 = A$ and $P_n = B$. We notice that every edge $\overline{P_i P_{i+1}}$ ($1 \leq i \leq n-1$) is contained in the square $S_{AB}^{[A]}$, and simultaneously inscribed in the square $S_{P_i P_{i+1}}^{[P_i]}$. Thus, the side length of every $S_{P_i P_{i+1}}^{[P_i]}$ cannot be larger than the side length of $S_{AB}^{[A]}$, i.e. $\max(|\overline{P_i P_{i+1}}|_X, |\overline{P_i P_{i+1}}|_Y) \leq \max(|\overline{AB}|_X, |\overline{AB}|_Y) \leq \frac{\sqrt{2}}{2}$. In addition, every square $S_{P_i P_{i+1}}^{[P_i]}$ uses node P_i as one of its corner points. By the Definition 8 of USG, every edge $\overline{P_i P_{i+1}} \in USG(V)$.

CASE 2. Assume $R(AB)$ is not “empty”.

Similar to the CASE 2 of Theorem 2, we first choose those “junction nodes” on the path $p(A \sim B)$ in $STG(V)$. We use $O_1, O_2, \dots, O_n$ to denote the n junction nodes from A to B inclusively, where $O_1 = A$ and $O_n = B$. Every line segment $\overline{O_i O_{i+1}}$ ($1 \leq i \leq n-1$) is contained in the square $S_{AB}^{[A]}$, and simultaneously inscribed in the square $S_{O_i O_{i+1}}^{[O_i]}$. Thus the side length of every $S_{O_i O_{i+1}}^{[O_i]}$ cannot be larger than that of $S_{AB}^{[A]}$, i.e. $\max(|\overline{O_i O_{i+1}}|_X, |\overline{O_i O_{i+1}}|_Y) \leq \max(|\overline{AB}|_X, |\overline{AB}|_Y) \leq \frac{\sqrt{2}}{2}$. This implies $\overline{O_i O_{i+1}} \in USG(V)$. In addition, by Theorem 2, any two adjacent junction nodes O_i and O_{i+1} have the “empty” $R(O_i O_{i+1})$. According to the CASE 1 of this proof, it follows that every subpath $p(O_i O_{i+1})$ of $p(A \sim B)$ consists of element edges all belonging to $USG(V)$. ■

3.5 Summary

This chapter proposes a novel proximity graph, Square Test Graph (STG). In section 3.3, we prove STG has at least three nice features—planarization, connectivity and a bounded path spanning ratio. In section 3.2, we observe other proximity graphs such as the Gabriel graph (GG), the relative neighborhood graph (RNG) and the Delaunay triangulation (Del). We find, if their approaches are used to simplifying the underlying topologies of wireless ad hoc networks, all of these pervasively studied structures satisfy the “short-edge replacement rule”, whereas STG does not. So, in section 3.4, we propose to use a new shape “unit-square” to model the communication ranges of wireless nodes so that STG is realistically applicable to the topology control of wireless ad hoc networks. The usage of “unit-square” results in a change to the network model, i.e. a wireless ad hoc network is hereafter modeled as a unit-square graph (USG) in this thesis, instead of the generally used unit-disk graph (UDG).

Chapter 4 will present a parallel algorithm applying STG on the top of USG. As a result, a new network structure $\text{STG} \cap \text{USG}$ is obtained in our simulation. Due to limited computation and communication costs, this algorithm is practically well-performed and robust.

Chapter 4

Description of Algorithmic Designs

4.1 Introduction

A proximity graph G can be constructed by nodes locally [21], if each node u can determine the edges of G incident on u by using only the location information of its entire k -local nodes—the nodes within k -hops from u . By section 3.4, $USG \cap STG$ is such a graph that can be constructed locally, where $k = 1$. Now, we let a wireless ad hoc network use the unit-square graph (USG) as its network model. Therefore, a $USG \cap STG$ alike network structure can be easily constructed by wireless nodes in a localized manner.

In this chapter, we present a distributed and localized algorithm. It follows the approach that defines STG to dynamically break those “nonessential” wireless links corresponding to those “nonessential” edges of the USG network model. As a result, a dynamical global network structure is maintained constantly. If we ignore the communication and computation time of nodes, all “essential” wireless links reserved at any moment is in correspondence with those edges in both USG and STG. Or, the underlying network topology forms a $USG \cap STG$ at any moment. And, the $USG \cap STG$ is called the localized square test graph (LSTG) hereafter.

We have known that a $LSTG(V)$ with the node set V is a planar $\sqrt{20}$ -spanner of $USG(V)$. In Chapter 5, we implement this algorithm in the simulation to verify the hypothesis, i.e. in a realistic environment, a $LSTG(V)$ has not only a better stretch factor to $USG(V)$ than $\sqrt{20}$ but also a better path quality than $GG(V)$ and $RNG(V)$.

4.2 Discussion of Design Issues

Assume each u of a set V of wireless nodes has a unique identifier and always knows its location in a mobile environment. Let $N_k(u)$ be the set of k -local nodes of u . A

k -local node v of u is also called a k -neighbor of u , i.e. $v \in N_k(u)$.

On the one hand, u is aware of the positions and IDs of $N_1(u)$ after one broadcast by each of $N_1(u)$; however, u can also know the positions and IDs of other nodes in $N_k(u)$ through the data packets retransmitted by $N_1(u)$. Still, when u receives a retransmitted data packet issued by a k -hop distant ($k > 1$) neighbor v , the information included is not necessary accurate due to the frequent movement of v and the time delay caused by multiple retransmissions of this data packet. Especially with an increase in k 's value, the incremental errors in v 's position could be considerably large. In addition, a larger k to u implies more k -neighbors whose positions u needs to acquire in time, and hence more position-related data packets to be or being retransmitted at any moment. Consequently, too much bandwidth expending on network maintenance could impact the overhead and adaptability of routing protocols since bandwidth is a scarce and precious resource in wireless networks. To avoid this overuse of bandwidth, it is significant to restrict the constant k to a small integer, such as 1 or 2.

The constant k in the following algorithm is 1. As a result, the overhead expended by nodes for maintaining their incident wireless links is considerably low; this includes only periodically broadcasting their positions and identifiers, and doing limited computation on the broadcast messages received. Since no data packet retransmissions are needed, all wireless nodes are under better support to update in time the links incident on themselves, even if they may be extremely-mobile. This makes the LSTG constructed by the following algorithm robust in practice and well performed in terms of the bandwidth utilization ratio.

4.3 A Distributed and Localized Algorithm for the Construction of LSTG

Given a set V of wireless nodes, this distributed algorithm includes two parts, which run parallel at each wireless node u of V .

PARALLEL—ONE (see Appendix A) runs once every short-time interval t . This part verifies the existence of u 's entire 1-neighbors $N_1(u)$, based on the USG network model. If one (or some) of $N_1(u)$ loses contact with u , then this part updates the edges of $LSTG(V)$ currently incident on u .

PARALLEL—TWO (see Appendix B) runs once u receives a broadcast message from

	<i>ID</i>	<i>Position</i>	<i>Time Stamp</i>	<i>Link Status</i>	
$\{v\}$:	<i>id</i>	<i>posX,posY</i>	<i>time</i>	<i>hasLink</i>	$\dots$

Figure 4.1: The data structure used by current node v to store a broadcast message from u .

some node. If the broadcast message received is issued by an unknown 1-neighbor (in USG model) of u or by a known 1-neighbor (in USG model) of u whose location has changed, then this part also updates the edges of $LSTG(V)$ currently incident on u .

Two parts of the algorithm commonly need to compute the edges of $LSTG(V)$ currently incident on u , therefore we separate the pseudocode fulfilling this task as a function, `computeAllLinks` (see Appendix C).

4.3.1 Two Main Data Structures

Every wireless node v periodically—in a short-time interval t —broadcasts a message, including ID, position etc. When current node u receives this, it stores this message into a data structure as shown in Figure 4.1, denoted by $\{v\}$. The “Time Stamp” field stores the time when v sends this message. The “Link Status” is a field to be filled in by current node u once it finishes its wireless link computations, which represents if u has a wireless link to v (or, if $\overline{uv}$ is a edge of $LSTG(V)$ incident on u). Initially the “Link Status” field is always set to *False*.

By section 3.4, each node determines the edges of LSTG incident on itself based on the positions of its entire 1-local nodes in USG. In following algorithm, current wireless node u computes if it has a wireless link to its known 1-neighbor v as follows. First, u checks if the rectangle $R(uv)$ is empty. If $R(uv)$ is empty, then u tries to find a “bigger empty rectangle” passing u , v and such other two nodes as “ C ”, “ D ” in the **Case 1** of Theorem 2 (refer to Figure 3.6). When one of such two nodes as “ C ” and “ D ” is absent, a boundary of its communication range is used as a side of this “bigger empty rectangle”, where u ’ communication range is a square area centered at u in correspondence with the unit-square used in USG model. Finally, u checks if the “bigger empty rectangle” is big enough to hold an “empty square” passing u , v . If it is, then current node u adds a wireless link to v . For example (see Figure 4.2(a)), where

(a) Each wireless node A divides its comm. range into 4 Quadrants—I, II, III and VI.

Q1 (data structure for Quadrant I)

No.	saved message from neighbors	horizontal next node	vertical next node
1	{A}	2	5
2	{B}	3	4
3	{C}	5	/
4	{D}	/	3
5	{E}	4	2

→ : $A \rightarrow B \rightarrow C \rightarrow E \rightarrow D$ ↑ : $A \rightarrow E \rightarrow B \rightarrow D \rightarrow C$

(b) Table Q_1 used to record A 's entire known 1-neighbors in Quadrant I

Q3 (data structure for Quadrant III)

No.	saved message from neighbors	horizontal next node	vertical next node
1	{A}	2	4
2	{L}	4	/
3	{K}	5	6
4	{H}	3	5
5	{I}	6	3
6	{J}	/	2

← : $A \rightarrow L \rightarrow H \rightarrow K \rightarrow I \rightarrow J$ ↓ : $A \rightarrow H \rightarrow I \rightarrow K \rightarrow J \rightarrow L$

(c) Table Q_3 used to record A 's entire known 1-neighbors in Quadrant III

Figure 4.2: Each wireless node A uses 4 tables, Q_1 , Q_2 , Q_3 , Q_4 , to represent its 4 Quadrants, I, II, III, VI.

current node A finds $R(AC)$ contains a node B , and thus does not add a wireless link to C . For another 1-neighbor E , A finds the “bigger empty rectangle” passing A , E , B and M and is big enough to hold a square passing A and E . Therefore, A adds a wireless link to E . For the 1-neighbor B , the “bigger empty rectangle” can also be found, whose three sides pass A , B , E respectively and another side overlaps with the left boundary of the square centered at A . Obviously, A has a wireless link to B .

Figure 4.2 illustrates a data structure of kernel importance for the edge computation by each node. Here, each wireless node A divides its communication range (the square area) into four quadrants, I , II , III and VI (see Figure 4.2(a)). It uses four tables, Q_1 , Q_2 , Q_3 and Q_4 , to represent these four quadrants (see Figure 4.2(b) and Figure 4.2(c)). All of its currently known 1-neighbors (in USG model) are in one-to-one correspondence with the entire items of the four tables. For the convenience of finding such two nodes as “ C ”, “ D ” for any 1-neighbor, A indexes all 1-neighbors recorded in every table Q_i ($1 \leq i \leq 4$) along both X -Axis and Y -Axis based on 1-neighbors’ geometrical positions. The resulting two series of indexes are stored in the “horizontal next node” field and the “vertical next node” field of every Q_i (see Figure 4.2(b) and Figure 4.2(c)). For example, wireless node A knows the geometrical positions of nodes B , C , E and D . Since they are all in its Quadrant I and $A \rightarrow B \rightarrow C \rightarrow E \rightarrow D$ is the order from current node A along the positive X -Axis, A fills out those “horizontal next node” fields in Q_1 as 2, 3, 5, / and 4 to record this order. At the same time, A fills out those “vertical next node” fields in Q_1 as 5, 4, /, 3 and 2 to record the order $A \rightarrow E \rightarrow B \rightarrow D \rightarrow C$ from current node A along the positive Y -Axis.

Considering the parallel computation in the algorithm is performed on the same data set—the four tables, which includes adding items to tables for newfound nodes, eliminating items from tables for those wireless nodes losing contact and determining links to 1-neighbors, therefore a synchronization lock is placed on the four tables (see Appendixes A and B).

4.3.2 Explanations of Functions Used in Pseudocode

Several functions are used in the pseudocode. We explain them as follows:

checkQuadrant($\{v\}$): Check if the originator node of the message $\{v\}$ is located

in the communication range (the square area). If it is, then return the quadrant it is in; otherwise return -1 .

insert($\{v\}$, Q_i), delete($\{v\}$, Q_i): Create and then store the message $\{v\}$ in a new item of table Q_i (Deletes from table Q_i the item that stores the message $\{v\}$). Then, re-index the fields of “horizontal next node” and “vertical next node” in Q_i .

hasNextFailureNode(Q_i), getNextFailureNode(Q_i): For some reason, e.g. power failure, antenna disability, moving out of the disk area—the real communication range of current node, or moving out of the square area but still in the disk area of current node, a previously known 1-neighbor (in USG model) of current node may lose contact with (or be ignored by) the current node. However its outdated message is still reserved in some table. In order to keep the messages reserved in the four tables up-to-date, current node needs to periodically eliminate those outdated messages. The judgment of outdated messages is based on the “Time Stamp” field and current time. If their difference is larger than a threshold (e.g. $2 \times t$), then this message is outdated, and then it is eliminated from the corresponding table.

4.4 Running Time of the Algorithm

The running time involves the time cost of PARALLEL—ONE, PARALLEL—TWO and the function computeAllLinks, all of which are obviously related to the expected number $\bar{N}$ of nodes within every unit-square.

For PARALLEL—ONE (see Appendix A), the running time of step 1 is $\Theta(1)$; the expected running time of steps 4–7 is $\Theta(\bar{N})$.

For PARALLEL—TWO (see Appendix B), the running time of steps 1–2 is $\Theta(1)$; the expected running time of steps 5–7 is $\Theta(\bar{N})$; the expected running time of step 8 is $\Theta(\bar{N})$.

For the function computeAllLinks (see Appendix C), steps 1–2 imply the block (steps 1–32) will be executed $\bar{N}$ times. In each execution of the block (steps 1–32), steps 3–6 take $\Theta(1)$ operations, thus the overall expected running time of steps 3–6 is $\Theta(\bar{N})$; the sub-block (steps 7–12) will be executed $\bar{N}$ times and steps 8–12 takes $\Theta(1)$ operations, thus the overall expected running time of steps 7–12 is $\Theta(\bar{N} \times \bar{N} \times 1) = \Theta(\bar{N}^2)$; similar to steps 3–6, the overall expected running time of step 13 is also $\Theta(\bar{N})$; similar to steps 7–12, the overall expected running time of steps

14–19 is trivially $\Theta(\overline{N}^2)$; steps 20–32 take $\Theta(1)$ operations, thus the overall expected running time of steps 20–32 is $\Theta(\overline{N})$. Therefore, the overall expected running time of the Function `computeAllLinks` is $\Theta(\overline{N}^2)$, and the overall expected running times of `PARALLEL–ONE` and `PARALLEL–TWO` are also $\Theta(\overline{N}^2)$.

In Chapter 5, we will discuss a set of n wireless nodes that randomly and uniformly distribute in a rectangular area of size $a \times b$ of the Euclidean plane. All nodes use the same transmission range r (in USG model). By Conclusion 2 to be discussed in section 5.3.2, $\overline{N} = \frac{4nr^2}{a \times b}$. To guarantee the connectivity of the LSTG constructed by this algorithm, we also conclude (Conclusion 3 in section 5.3.2) that at least $r^2 = \frac{a \times b \times (\ln n + \ln s_1)}{4n}$, where s_1 is a large constant. Thus, $\overline{N} = \ln n + \ln s_1$.

Finally, the overall expected running times of `PARALLEL–ONE` and `PARALLEL–TWO` are all $\Theta(\ln^2(n))$, then the expected running time of this parallel algorithm is also $\Theta(\ln^2(n))$. Furthermore, if an improvement is made in the function `computeAllLinks`, an expected running time as good as $\Theta(\ln n)$ is the norm that can be expected.

4.5 Summary

This chapter presents a localized algorithm constructing a localized square test graph (LSTG) with an expected running time $\Theta(\ln^2(n))$, where n is the number of wireless nodes. This algorithm's execution will enable each wireless node to instantly determine the edges of LSTG incident on itself. As a result, a LSTG alike dynamical global network structure is maintained constantly. In Chapter 5, we implement this algorithm to construct a LSTG in the simulation. We prove by experiment the LSTG has a better path quality than the GG and the RNG in practice.

Chapter 5

Simulation and Presentation of Experimental Results

5.1 Introduction

In Chapter 4, we presented a distributed algorithm that can construct a localized square test graph $LSTG(V)$ on a set V of wireless nodes. According to Chapter 3, the $LSTG(V)$ is a $\sqrt{20}$ -spanner of the $USG(V)$. In contrast, the spanning ratio of $GG(V)$ or $RNG(V)$ to $UDG(V)$ cannot be bounded by any constant [1, 6]. However, it is not strong enough to guarantee a better path quality of LSTG than that of GG or RNG only depending on that LSTG's spanning ratio has a smaller theoretical upper-bound. The path quality of GG or RNG acquired in practice could be better than that of LSTG. In addition, the stretch factor of LSTG is not achieved to the same supergraph as GG's and RNG's. So, in this Chapter, we simulate the behaviors of a set V of wireless nodes, and compare $LSTG(V)$ with $GG(V)$ and $RNG(V)$ in terms of various spanning ratios (see Figure 5.1).

The spanning ratio is a relationship between a proximity graph and its connected supergraph. For example $STG(V)$ vs $K(V)$ (complete Euclidean graph), $LSTG(V)$ vs $USG(V)$, $GG(V)$ or $RNG(V)$ vs $UDG(V)$ etc. Proximity graphs are supposed

Figure 5.1: $GG(V)$, $RNG(V)$ and $LSTG(V)$ on the same set V of nodes.

to maintain their supergraphs' connectivity. In order to obtain connected proximity graphs (GG, RNG and LSTG) so that we can compare their spanning ratios, we first need to guarantee they all have connected supergraphs (UDG and USG). However, neither a UDG alike network nor a USG alike network is necessarily connected in practice. In our simulation, we use the two network models on the same set V of wireless nodes. By Definition 8 of USG, a $USG(V)$ is a subgraph of the $UDG(V)$. So, to obtain connected $UDG(V)$, $GG(V)$, $RNG(V)$ and $LSTG(V)$ in simulation, we first need a connected $USG(V)$ at any time.

5.2 About the Connectivity of the Unit-square Graph

We assume wireless nodes have the transmission range r if they form a USG alike network. This means their USG network model uses the scale ratio $\frac{\sqrt{2}}{2}$ unit : r (see Figure 3.20). A connected $USG(V)$ depends on many factors such as the node density, the value of r , the distribution manner of nodes etc. Among them, the first and foremost is the value of r because the $USG(V)$ will always be connected as long as a large enough r is chosen. However, if an overlarge r is chosen, each node may spend more than necessary energy on communication where the energy consuming at each node is proportional to r^2 . As a result, fast energy consuming could result in a disconnected network or nodes' malfunction, e.g. failing to monitor certain parts of their communication ranges. To save power consumption without losing network connectivity, we need to find the "minimum transmission range" universally adopted by all wireless nodes such that the induced network topology USG is connected all the time. However, wireless nodes move around freely, and the distance among them changes all the time. It is impossible to have a unanimous minimum transmission range for nodes to guarantee the connectivity for all instances of the network configuration. Fortunately, by some statistical methodology in section 5.3, we can find a minimum transmission range for nodes, with which all nodes form a connected USG at any time almost surely (with high probability of almost one). Later, our experimental results also prove the existence of such minimum transmission ranges.

In addition, we notice when r is close to or greater than the minimum transmission range, the edge set of LSTG rarely increases with the increase in r , unlike the situation that r is much lower than the minimum transmission range. In other words, the

edge set of LSTG relative to r tends to be stable if r is greater than the minimum transmission range. As a result, the shortest paths in LSTG between all pairs of nodes also tend to be stable. So, a further increased r will not lead to a improved spanning ratio of LSTG to USG.

5.3 Stochastic Geometry

In a realistic wireless ad hoc network, and in spite of the free movement of nodes, if we take snapshots of all nodes of V once every short-time interval during a given time period, then we have a set of τ unit-square graphs, i.e. $USG_1(V)$, $USG_2(V)$, $USG_3(V)$, $\dots$, $USG_\tau(V)$. We say a graph is connected with a probability Pr , actually there are two explanations for this Pr . One is there exists average $\tau \times Pr$ connected graphs among this set of τ graphs. Another explanation is the biggest node clique in any $USG_i(V)$ ($1 \leq i \leq \tau$) has average $n \times Pr$ nodes, where n is the number of nodes of V . In the rest part of this section 5.3, we take any $USG_i(V)$ as an example to analyze its connectivity, and the probability Pr clearly belongs to the second case.

5.3.1 Key-Node Set

For simplicity, we assume the set V of n nodes “randomly” and “uniformly” distributes in a rectangular area of size $a \times b$ of the Euclidean plane, and all nodes have the same transmission range r (in the case of USG). Our question is under what circumstance n , a , b and r are, the $USG(V)$ is connected almost surely (with a high probability of almost one). Prior to finding the solution, we transform this question into the famous “Coupon Collector’s Problem” [14, 26]. Based on the analysis of the “Coupon Collector’s Problem”, the solution to this question will be found.

One of the terms to be used frequently in the following part is the “coverage” of a set V of nodes, denoted by C_V . C_V stands for the union of communication ranges (in the case of USG) of the entire nodes of V . Coverage is not same as connectivity, and (unless required) C_V does not necessarily cover the whole rectangular area of size $a \times b$. Another two terms are “key-node” and “key-node set”. By some means, we can pick up a subset K of V such that the coverage of K is equal to the coverage of V , i.e. $C_K = C_V$. We call such a node set K a “key-node set” of V , and call the nodes in K “key-nodes”. Obviously, there are numerous eligible key-node sets in V .

Using different methods to choose those key-nodes, we may get different key-node sets with different numbers of key-nodes. However, among those key-node sets of V , one (if there is any) can ensure the connectivity of $USG(V)$.

Theorem 4. *$USG(V)$ is connected if and only if there is at least one key-node set K of V such that $USG(K)$ is connected.*

Proof. [Sufficiency]. By the definition of key-node set, each node u in V but not in K must be in the communication range of some key-node v , then u must have a edge linking to the key-node v . Consequently, the entire nodes of V are connected to each other.

[Necessity]. By the definition of key-node set, it is obvious that V itself is a key-node set of V . Let $K = V$, then $USG(K) = USG(V)$ is connected. ■

Next, we provide a method of picking every key-node from V and adding them to K one by one. When we are adding a newfound key-node to K , we simultaneously update K 's coverage C_K . Initially, let K only include some node of V , denoted by k_1 . At this time, C_K equals the communication range (of size $2r \times 2r$) of k_1 , and the area of C_K is $4r^2$. Henceforth, once a node is found in the communication range of k_i such that its joining in K enlarges K 's coverage C_K , then we pick this node as the next key-node, denoted by k_{i+1} . We repeat this process to get $k_2, k_3, \dots$, until a key-node $k_m (m \leq n)$ is found whose joining in K makes C_K as large as C_V .

Clearly, if an appropriate transmission range r is chosen, then, by above method, the next key-node k_{i+1} ($2 \leq i+1 \leq m$) can always be found by k_i at every i th step until k_m . By Theorem 4, $USG(V)$ is thus connected. Put another way, to ensure the connectivity of $USG(V)$, the transmission range r should be chosen to make the communication range of every current key-node k_i large enough to hold sufficient 1-neighbors so the next key-node k_{i+1} can be found among them almost surely (with a high probability of almost one). Trivially, such a minimum r is the “minimum transmission range”.

5.3.2 Coupon Collector's Problem Model

This makes our problem the “Coupon Collector's Problem”. Assume every time the collector buys a product (bubble gum or Cracker Jack, for example) he receives a

coupon (a baseball card or a toy, for example) which is equally likely to be any one of n types. The “Coupon Collector’s Problem” tells us this person will collect all n types of coupons with a probability $e^{-e^{-c}}$ in the expected times, $n \ln n + c \times n$, of purchases, where n goes infinity and c is a positive constant. I.e.

$$\lim_{n \rightarrow \infty} Pr(X \leq n \ln n + c \times n) = e^{-e^{-c}}, \quad (5.1)$$

where X denotes purchase times or denotes the number of coupons this person received before he collects all n types of coupons. It follows that, once this person collects a new coupon, he will find another new coupon with a probability $e^{-e^{-c}}$ from an expected number $(\ln n + c)$ of coupons that he receives later, where n goes on to infinity.

Recall, according to the method of picking every key-node, its joining in K must be capable of enlarging C_K . If the communication range of each key-node is compared to a different type of coupon, then the process that the communication ranges of all of the key-nodes in K tile the whole C_V is equivalent to the process that the coupon collector collects all n types of coupons. Therefore, Conclusion 1 is as follows:

Conclusion 1. *During the process of picking every key-node of K one by one, if every current node k_i has an expected number $\ln n + c$ of 1-neighbors in its communication range, then the next key-node k_{i+1} can be found with a probability $e^{-e^{-c}}$.*

On the other hand, the nodes of V randomly and uniformly distribute in a rectangular area of size $a \times b$ of the Euclidean plane. Thus, $\diamond$ the probability of any node appearing at a specific position is almost zero, $\diamond$ the number of nodes appearing in a specific region should be proportional to the area of this region. Trivially, the probability that a given region R contains exactly m nodes obeys a “Poisson distribution”, i.e. $Pr = \frac{\lambda^m}{m!} \times e^{-\lambda}$, where $\lambda = \frac{n \times |R|}{a \times b}$ denotes the expected number of nodes in the region R . Then we have Conclusion 2 as follows. By adding together Conclusion 1 and Conclusion 2, we can make the following Conclusion 3 hold.

Conclusion 2. *The communication range (of size $2r \times 2r$) of each node contains an expected number $\frac{4nr^2}{a \times b}$ of 1-neighbors.*

Conclusion 3. *If $\frac{4nr^2}{a \times b} \geq \ln n + c$, then the communication ranges of all of the key-nodes in K will tile the whole C_V with a probability at least $e^{-e^{-c}}$. Therefore, $USG(K)$*

and thus $USG(V)$ are connected with a probability at least $e^{-e^{-c}}$. I.e.,

$$Pr[USG(V) \text{ is connected} \mid r \geq \sqrt{\frac{a \times b \times (\ln n + c)}{4n}}] \geq e^{-e^{-c}} \quad (n \rightarrow \infty).$$

Let $e^c = s_1$, then we have Formula (5.2) as follows:

$$Pr[USG(V) \text{ is connected} \mid r \geq \sqrt{\frac{a \times b \times (\ln n + \ln s_1)}{4n}}] \geq e^{-\frac{1}{s_1}} \approx 1 - \frac{1}{s_1} \quad (n \rightarrow \infty) \quad (5.2)$$

Note that the number of key-nodes of K , assume m , cannot be larger than n (the number of nodes of V). The implication here is that Formula (5.2) can guarantee a connected $USG(K)$ and thus a connected $USG(V)$ in the worst case, i.e. $K = V$. This is similar to that the coupon collector need only to collect m ($m \leq n$) types of coupons. But, Formula 5.1 can ensure the coupon collector will collect more than m types of coupons.

5.4 Discussion of the Parameter Space

We simulate a wireless mobile ad hoc network consisting of a set V of n nodes, and study various network structures formed by nodes of V . For the convenience of discussion, we divide them into two groups. One group including $USG(V)$ and $LSTG(V)$ is called “ $USG(V)$ -group”; another including $UDG(V)$, $GG(V)$ and $RNG(V)$ is called “ $UDG(V)$ -group”. Due to their different definitions of the transmission range (refer to Figure 3.20), a conversion of the transmission range is made from $UDG(V)$ -group to $USG(V)$ -group for consistency’s sake, i.e. that nodes have the transmission range r in any network topology of $USG(V)$ -group implies that they have a transmission range $\sqrt{2} \times r$ in those of $UDG(V)$ -group.

According to [27, 28], we have following Formula (5.3) holding for the $UDG(V)$ -group, which is similar to Formula (5.2) holding for the $USG(V)$ -group.

$$Pr[UDG(V) \text{ is connected} \mid r \geq \sqrt{\frac{a \times b \times (\ln n + \ln s_2)}{2\pi n}}] \geq e^{-\frac{1}{s_2}} \approx 1 - \frac{1}{s_2} \quad (n \rightarrow \infty) \quad (5.3)$$

To ensure all connected network topologies in the $UDG(V)$ -group (i.e. $Pr[UDG(V) \text{ is connected}] \approx 1$), we only need the parameter s_2 in Formula (5.3) to be a constant large enough to make $\frac{1}{s_2}$ close to zero. Then, we compare Formula (5.2) with Formula

(5.3). For convenience sake, we rewrite them as follows:

$$Pr[USG(V) \text{ is connected}] \geq 1 - \frac{1}{s_1} \quad \text{where } s_1 = \frac{e^{4 \times \frac{nr^2}{ab}}}{n}$$

$$Pr[UDG(V) \text{ is connected}] \geq 1 - \frac{1}{s_2} \quad \text{where } s_2 = \frac{e^{2\pi \times \frac{nr^2}{ab}}}{n}$$

Clearly, $Pr[UDG(V) \text{ is connected}] > Pr[USG(V) \text{ is connected}]$ because $s_2 > s_1$. This implies, if a chosen r is large enough to make the $USG(V)$ -group almost surely connected, then the $UDG(V)$ -group is connected even more surely. This is consistent with the Definition 8 of USG , i.e. $USG(V) \subseteq UDG(V)$.

In addition, we consider a large probability Pr close to 1. As mentioned in section 5.3, $n \times Pr$ indicates the average number of nodes in the biggest node-clique. If there are several biggest node-cliques of equal size, according to the rules by which a key-node set is generated (in section 5.3.1), $n \times Pr$ indicates the average number of nodes in any one of them. However, this case will never happen because the large Pr (≈ 1) can guarantee a unique biggest node-clique. It follows that $n \times (1 - Pr)$ is the average number of nodes disconnected to the unique biggest node-clique.

5.5 Analysis of Experimental Results

We conducted all experiments with a simulation kit developed in Java language. This kit uses four inputs to generate a node configuration, which are “Width of rectangular area”, “Height of rectangular area”, “Number of nodes” and “Transmission range of nodes” (see a screen shot in Figure 5.5). Once nodes are randomly and uniformly settled in the rectangular area, the kit is ready to compute the two groups of graphs on nodes (i.e. the USG -group and the UDG -group). It computes each edge of the $LSTG$ by using the “local empty square test”. An example and the pseudocode of this local test are provided in section 4.3.1 and in Appendix C respectively. It computes edges of the GG by using the “local empty disk test”, and it computes edges of the RNG by using the “local empty lune test”. If two groups of graphs have been constructed and they are all connected, this kit will use Dijkstra’s algorithm to solve shortest path problems. For example, Figure 5.5 illustrates the shortest path between the node (ID=10) and the node (ID=28) in the $LSTG$ (or, $STG \cap USG$).

Figure 5.2: A screen shot from the simulation kit.

For each of the two groups of graphs, this kit outputs the number of edges, the number of nodes disconnected to the biggest node-clique (if disconnected), and all-pairs shortest-paths (if connected). In addition, this kit performs related computations to generate further outputs. For example, it computes the shortest paths between the node (ID=10) and the node (ID=28) in both LSTG and USG, and then computes the ratio of shortest-paths' lengths.

In the following experiments, we always chose a rectangular area of size 500×500 (meters²) as simulation inputs. But, other two simulation inputs, "Number of nodes" and "Transmission range of nodes", depended on the specific circumstances.

5.5.1 Minimum Transmission Range vs. Stability of Edges of LSTG

At first, we need to prove by experiment our analysis in section 5.2—"There exists a minimum transmission range. If the chosen transmission range r is greater than the minimum transmission range, the edge set of LSTG rarely increases with the increase in r . The edge set of LSTG relative to r tends to be stable."

In our experiments, we first choose a total of $n = 500$ nodes and let r be various lengths from 20 meters to 120 meters. For every input length of r , we count the

(a) A number n of nodes in a plane of size $a \times b$ ($n = 500$, $a = b = 500$ meters).

(b) A number n of nodes in a plane of size $a \times b$ ($n = 750$, $a = b = 500$ meters).

Figure 5.3: Edge sets of USG and LSTG vary with the transmission range r .

numbers of edges in both USG and LSTG. Figure 5.3(a) illustrates the experimental results where all statistics are computed over 50 different node configurations. For example, the experimental results show that USG and LSTG have an average of 2359 edges and 1205 edges respectively over 50 runs at $r = 50$ meters. The experimental results also show that the minimum transmission range exists about between 50 meters and 60 meters. Then, we observe the two curves in Figure 5.3(a). For USG, the number of its edges increases rapidly with the increase in r ; for LSTG, conversely, the number of its edges rarely changes with the increase in r if $r \geq 60$ meters. It follows that “the edge set of LSTG relative to r tends to be stable”.

We also choose $n = 750$ nodes to carry out the same experiments. Figure 5.3(b) is an illustration of the results. According to the experimental results, the minimum transmission range in this case is about between 40 meters and 45 meters.

Now, our experimental results have confirmed the stability of LSTG’s edge set has a relationship with the minimum transmission range r . Then, we need to find out what makes this relationship hold. For this question, we assume that a set V of wireless nodes use the USG network model and choose the LSTG as their underlying network topology. The $LSTG(V)$ will divide the Euclidean plane into numerous finite faces and an infinite face as it is a planar graph. We define a boundary of the infinite face as a closed path consisting of the edges in both the infinite face and some finite face. Then, the infinite face has only one boundary in that $LSTG(V)$ is a connected graph (or, all nodes form only one clique). For an arbitrary node A of V , there are three cases.

(1) A lies neither in the infinite face nor on its boundary. Then, A ’s entire adjacent faces must have a closed outer boundary, which consists of the partial boundaries of A ’s adjacent faces and contains in its interior at least a node A . We denote all nodes excepting A lying on or in this closed outer boundary as $\aleph(A)$. This $\aleph(A)$ trivially is a superset of $N(A)$, a node set consisting of A ’s entire 1-hop neighbors in $LSTG(V)$. While wireless nodes’ transmission ranges r increase in length, some new edges may join in $LSTG(V)$ and the resulting graph is therefore a planar supergraph of $LSTG(V)$. On the one hand, these new added edges could cause some nodes in $\aleph(A)$ to join in $N(A)$; on the other hand, these new added edges could cause A ’s smaller adjacent faces, as a result of which some nodes originally belonging to $\aleph(A)$

will never belong to $\aleph(A)$. If r keeps increasing until $N(A) = \aleph(A)$, then a further increased r will not lead to more edges of $LSTG(V)$ incident on A .

(2) A lies in the infinite face. Then, we consider all nodes lying in the infinite face or on its boundary excepting A . From them, we pick up the nodes to form a set $\aleph(A)$ such that, for each node B in $\aleph(A)$, the edge $\overline{AB}$ (if added) will not break $LSTG(V)$'s planarization. Clearly, $\aleph(A)$ consists of the nodes that have or potentially have edges of $LSTG(V)$ connecting to A if wireless nodes' transmission ranges r keep increasing. Let d be the distance between A and the furthestmost node in $\aleph(A)$. If r keeps increasing until $r \geq d$, then a further increased r will not lead to more edges of $LSTG(V)$ incident on A .

(3) A lies on the boundary of the infinite face. Since this case is a combination of the above two cases, putting together the above discussion explains why the edges of $LSTG(V)$ incident on A rarely increase with the increase in r for this condition.

5.5.2 Correctness of Formula (5.2) in Simulation

Looking at it one way, the experimental results in section 5.5.1 have shown the existence of the minimum transmission range for any given input of n nodes. On the other hand, according to section 5.4, such a the minimum transmission range can be calculated by the Formula (5.2) (in section 5.3.2). Given an input of n nodes and a target probability Pr close to one, we need to prove by experiment the transmission range calculated by the Formula (5.2) is the practical minimum transmission range such that two group of network topologies in simulation, including USG, LSTG, UDG, GG and RNG, are connected almost surely or with a practical probability no lower than that given target probability Pr .

In this section, a given target probability of all network structures being connected is named a "theoretical probability", yet a practical probability of all structures being connected in the simulation is named an "experimental probability". Every experimental probability was obtained over a large amount of different node configurations, and it equals one minus the average disconnected nodes in simulation divided by the total number n of nodes. However, we notice that the same "theoretical probability" will not lead to the same average disconnected nodes if different simulation inputs of n are used. In fact, in comparison with requiring all structures to be connected with

transmission range (r)	theoretical/experimental probability	theoretical/experimental disconnected nodes
35.57849	0.980 / 0.998533	10 / 0.733333
36.77597	0.990 / 0.998667	5 / 0.666667
38.30155	0.996 / 0.9994	2 / 0.300000
39.41639	0.998 / 0.999533	1 / 0.233333
40.50056	0.999 / 0.999867	0.5 / 0.066667

Table 5.1: A number n of nodes in a plane of size $a \times b$ ($n = 500$, $a = b = 500$ meters).

Figure 5.4: A number n of nodes in a plane of size $a \times b$ ($n = 500$, $a = b = 500$ meters).

transmission range (r)	theoretical/experimental probability	theoretical/experimental disconnected nodes
26.82458	0.9900 / 0.999633	10 / 0.366667
27.62027	0.9950 / 0.999433	5 / 0.566667
28.63822	0.9980 / 0.999867	2 / 0.133333
29.38485	0.9990 / 0.999900	1 / 0.100000
30.11297	0.9995 / 0.999867	0.5 / 0.133333

Table 5.2: A number n of nodes in a plane of size $a \times b$ ($n = 1000$, $a = b = 500$ meters).

Figure 5.5: A number n of nodes in a plane of size $a \times b$ ($n = 1000$, $a = b = 500$ meters).

the “theoretical probability”, it is more realistically significant to require the average disconnected nodes less than a target number. As posited in 5.4, this target number divided by n is actually a “theoretical probability”, so it is named “theoretical disconnected nodes”. Correspondingly, those practical average disconnected nodes we counted out in the simulation are named “experimental disconnected nodes”. Thus, the correctness of Formula (5.2) will be proven if an “experimental probability” is always higher than its corresponding “theoretical probability”, or alternatively those “experimental disconnected nodes” are always less than their corresponding “theoretical disconnected nodes”.

Experiments for this purpose are performed on $n = 500$ nodes and $n = 1000$ nodes respectively (see Table 5.1 and Table 5.2). To obtain those minimum transmission ranges as the simulation inputs of r , we first let the amounts of “theoretical disconnected nodes” be 10, 5, 2, 1 and 0.5 respectively, then we have their respective “theoretical probabilities”. Substituting these “theoretical probabilities” into Formula (5.2), we have those values of s_1 and then those desired simulation inputs of r . For every input of r , we count out the “experimental disconnected nodes” (over 50 different node configurations) and calculate the corresponding “experimental probability”.

For example, we let “theoretical disconnected nodes” be 5 nodes in the two cases of $n = 500$ nodes and $n = 1000$ nodes respectively. The two transmission ranges are calculated as $r = 36.77597$ meters ($n = 500$) and $r = 27.62027$ meters ($n = 1000$). By experimental results: when $r = 36.77597$ meters and $n = 500$ nodes, the number of “experimental disconnected nodes” is 0.666667 (Table 5.1); when $r = 27.62027$ meters and $n = 1000$ nodes, the number of “experimental disconnected nodes” is 0.566667 (Table 5.2). Both of them are much less than 5 nodes, the “theoretical disconnected nodes”. Accordingly, the “experimental probabilities” (0.998667 and 0.999433) are larger than their respective “theoretical probabilities” (0.990 and 0.9950).

Curves derived from Table 5.1 and 5.2 are shown in Figure 5.4 and Figure 5.5. Obviously, for each r adopted in experiments, the “experimental probability” is always higher than the corresponding “theoretical probability”. This proves the correctness of the Formula (5.2) in determining the “minimum transmission range” for any given set of nodes.

Figure 5.6: A comparison of the average length of all-pairs shortest-paths among LSTG, GG and RNG ($r \geq 40$).

5.5.3 Comparisons of Spanning Ratios

At this point, there is only one thing left to prove by experiment: on the same set of wireless nodes and as connected underlying network topologies, the LSTG always has a better path quality than the GG and the RNG in practice.

Experiments for this purpose are still performed on $n = 500$ nodes, and all statistics are also obtained over 50 different node configurations. For those simulations input of r , we choose a series of different lengths such that each of them ensures a high “theoretical probability” for all network structures. By section 5.4 and the last two sections 5.5.1 and 5.5.2, such inputs of r can safely, in a both theoretical and experimental sense, be calculated by Formula (5.2). Since we expect all $n = 500$ nodes to form always only one clique, we choose $r = 40$ meters as the minimum in that the resulting “theoretical disconnected nodes” are less than one node at any time (refer to Table 5.1).

For every simulation input of r , we first compare the “average length of all-pairs shortest-paths” of LSTG with those average lengths of GG’s and RNG’s (see Figure 5.6). Three curves illustrate that a LSTG always has a smaller average length than GG and RNG. In addition, this average length for LSTG will further decrease with the increase in r , but those for GG and RNG will not. When $r = 50$, this average length for LSTG reaches an almost-minimum, which is considerably lower than those for GG and RNG. This leads to the following conclusion.

Conclusion 4. *Suppose a set V of wireless nodes simultaneously have three types*

length of r ($r \geq 40$)	LSTG-USG all-pairs shortest-path length ratios		GG-UDG all-pairs shortest-path length ratios		RNG-UDG all-pairs shortest-path length ratios	
	average	deviation	average	deviation	average	deviation
40	1.03357455	6.88E-04	1.10370737	0.00240729	1.25064028	0.01226478
45	1.04046088	7.98E-04	1.11008895	0.00270531	1.25994933	0.01305603
50	1.04520734	8.57E-04	1.11509312	0.00280014	1.26616068	0.01324269
55	1.05116721	9.33E-04	1.11810107	0.00284937	1.26957829	0.01335166
60	1.05493941	9.80E-04	1.12037041	0.00294129	1.27212111	0.01340295
65	1.05750744	0.00100734	1.12155261	0.00297773	1.27344478	0.01341171
70	1.05964891	0.00103057	1.12237719	0.00300156	1.27437333	0.01343074

Table 5.3: Comparisons of the averages and deviations of the all-pairs shortest-path length ratios of GG-to-UDG, GG-to-UDG and RNG-to-UDG ($r \geq 40$).

of underlying network topologies, $LSTG(V)$, $GG(V)$ and $RNG(V)$. Overall, the all-pairs shortest-paths in $LSTG(V)$ are much shorter than the all-pairs shortest-paths in $GG(V)$ and $RNG(V)$.

For every simulation input of r , the ratios of all-pairs shortest-paths' lengths in the LSTG to all-pairs shortest-paths' lengths in the USG are also computed. The average and deviation of these ratios are named "average" and "deviation" of "LSTG-USG all-pairs shortest-path length ratios", as arranged respectively in the 2nd and 3rd columns of Table 5.3. For the sake of comparison, those "averages" and "deviations" for "GG-UDG all-pairs shortest-path length ratios" and "RNG-UDG all-pairs shortest-path length ratios" are computed and settled in the 4th, 5th, 6th and 7th columns of Table 5.3. Table 5.3 illustrates that the "average" of "LSTG-USG all-pairs shortest-path length ratios" is always lower than those average ratios of GG-to-UDG and RNG-to-UDG. This further confirms the Conclusion 4. In addition, three "deviation" columns demonstrate the lengths of all-pairs shortest-paths in LSTG are closer to respective presumable lengths. Here, for any pair of nodes, the presumable length of their shortest path in the LSTG is their shortest-path's length in the USG times the corresponding "average" ratio in the 2nd column of Table 5.3. Therefore, we have following Conclusion 5.

Conclusion 5. Suppose a set V of wireless nodes simultaneously have three types of underlying network topologies, $LSTG(V)$, $GG(V)$ and $RNG(V)$. Any pair of nodes

Figure 5.7: Comparisons of the practical spanning ratios of LSTG, GG and RNG ($r \geq 40$).

of V have the shortest-path(s) in $LSTG(V)$ with a more presumable length than those they have in $GG(V)$ and $RNG(V)$.

For every simulation input of r , we finally compare the “practical spanning ratio” of LSTG (to USG) with those of GG (to UDG) and RNG (to UDG) (see Figure 5.7). The “practical spanning ratio” of LSTG (to USG) indicates the worst case of “LSTG-USG all-pairs shortest-path length ratios”, therefore it is calculated as the maximum of the latter. In a similar way, the “practical spanning ratios” of GG and RNG are calculated. By the three curves in Figure 5.7, we then have following Conclusion 6.

Conclusion 6. *If a set V of wireless nodes simultaneously have three types of underlying network topologies, $LSTG(V)$, $GG(V)$ and $RNG(V)$, then the practical spanning ratio of $LSTG(V)$ -to- $USG(V)$ is always both much lower than $\sqrt{20}$ and lower a lot than the practical spanning ratios of $GG(V)$ -to- $UDG(V)$ and $RNG(V)$ -to- $UDG(V)$.*

Based on all comparisons, we conclude that the Localized Square Test Graph (LSTG) practically performs better than the Gabriel graph (GG) and the relative neighborhood graph (RNG) when it is chosen by a mobile wireless ad hoc network as its final underlying network topology.

5.6 Summary

In this chapter, we present various experimental proofs for a better path quality of LSTG than that of GG and RNG.

Before the experiments were conducted, we first analyzed the conditions and parameters of our experiments. Since we expect all connected network topologies in experiments (see section 5.2), we introduce the “Coupon Collector’s Problem” model (in section 5.3) to determine the “minimum transmission range” r for any given set of nodes. Section 5.4 shows that such a minimum transmission range can ensure both a connected USG (or LSTG) and a connected UDG (GG or RNG).

All experimental results leading to our analysis and conclusions are presented in section 5.5. On experimental-results basis, we first show the existence of “minimum transmission range” r . Then, we prove by experiment that the correctness of Formula (5.2) in determining the “minimum transmission range” r for any given set of nodes. Finally, we choose a series of different lengths as the simulation inputs of r , and we require the minimum r to ensure the all connected network structures in experiments. Then, comparisons of various ratios prove a connected LSTG has a better path quality than GG and RNG in many senses.

Chapter 6

Summary, Conclusions, Contributions and Future Work

6.1 Summary of work and Conclusions

In this thesis, we propose a new network model for wireless mobile ad hoc networks—unit-square graph (USG), and we also define a new proximity graph—Square Test Graph (STG). We prove that a Localized Square Test Graph (LSTG) is a planar $\sqrt{20}$ -spanner of USG, by showing that the STG is a $\sqrt{20}$ -spanner of the complete Euclidean graph and LSTG and USG have the same relationship. We also provide a distributed and localized algorithm for the construction of LSTG. Our experimental results show: on the one hand, LSTG achieves its connectivity later than GG or RNG with the increase in nodes' transmission range; on the other hand, once LSTG is connected almost surely, then it has a better path quality than GG and RNG.

Now, we consider a set of wireless nodes that randomly and uniformly distribute in a rectangular area of size $a \times b$. We model them as a USG and a UDG at the same time, and also expect that the two network models have the same high probability Pr of being connected. By Formula (5.2), we have the minimum transmission range of nodes in the USG network model— $r_{(USG)}$. By Formula (5.3), we have the minimum transmission range of nodes in the UDG network model— $r_{(UDG)}$. We notice that $r_{(USG)}$ is very close to $r_{(UDG)}$ if the value of $\frac{a \times b}{n}$ is small enough. This implies that, if the node density ($\frac{n}{a \times b}$) is large enough, then the USG network model and the UDG network model have almost the same high probability of being connected. On the top of the USG network model, the underlying network topology LSTG accordingly has almost the same high probability of being connected as those network topologies based on the UDG network model. As a result, we prefer to choose LSTG as the underlying topology of those wireless ad hoc networks in which wireless nodes “densely” distribute.

6.2 Contributions

In this thesis, a new proximity graph was defined, square test graph (STG), and a new wireless network model, unit-square graph (USG). Both of these were never known previously.

This thesis also provides a new heuristic for the construction of the underlying network topology of wireless ad hoc networks. Here, the USG is chosen as the network model so the approach defining STG is applicable to simplifying the edges of USG.

In addition, this thesis presents a distributed and localized algorithm using the approach defining STG and only the messages from all 1-neighbors of each nodes to simplify the edges of USG. The resulting underlying network topology is called localized square test graph (LSTG).

6.3 Future Work

By Definition 8 of USG, a “unit-square” is always smaller than a “unit-disk” (refer to Figure 3.20), which leads to the conclusion each wireless node has to ignore on purpose some of its 1-neighbors (in UDG network model), which in turn leads to that the connectivity of LSTG is not as good as GG or RNG when wireless nodes sparsely distribute.

We would like to further investigate if we can use another shape, such as the hexagon or the octagon, to model wireless nodes’ communication ranges so that the entire wireless nodes will define a new network model for wireless ad hoc network, such as “unit-hexagon graph” or “unit-octagon graph”. We expect the area of this new shape to be closer to the “unit-disk” than that of “unit-square”. We also expect to find some new proximity graph, such as “Hexagon Test Graph” or “Octagon Test Graph”, whose approach can be used to simplifying the edges of the new network model and simultaneously maintain the network connectivity.

Appendix A

Part 1 of the parallel algorithm

Once every short-time interval t , start a thread as follows.

PARALLEL—ONE []

1. broadcast ($\{u\}$) *//broadcast peripherad the message about its own information*
2. **synchronized** (Q_1, Q_2, Q_3, Q_4)
3. **beginning** of the synchronized block:
4. for each table Q_i *//eliminate out-dated message of neighbor nodes*
5. while (hasNextFailureNode (Q_i)*)
6. $\{v\} \leftarrow \text{getNextFailureNode} (Q_i)^\dagger$
7. delete ($\{v\}, Q_i$) ‡
8. computeAllLinks § *//re-compute the links (edges) incident on itself*
9. **end** of the synchronized block

*The detail of the function hasNextFailureNode() is in Section 4.3.2

† The detail of the function getNextFailureNode() is in Section 4.3.2

‡ The detail of the function delete() is in Section 4.3.2

§ The detail of the function computeAllLinks is in Appendix C

Appendix B

Part 2 of the parallel algorithm

Once receive a broadcast message from a node v , start a new thread as follows.

PARALLEL-TWO [$\{v\}$]

1. $m \leftarrow \text{checkQuadrant}(\{v\})^*$ *//check if node v is in the **unit-square** of u*
2. if ($m = -1$) *Return*
3. **synchronized** (Q_1, Q_2, Q_3, Q_4)
4. **beginning** of the synchronized block:
5. for each table Q_i *//eliminate the outdated message that is previously from node v*
6. for each table item j in Q_i
7. if ($Q_i[j].\text{"savedmessagefromneighbors"} = \{v\}$) {delete ($Q_i[j], Q_i$)}
8. $\text{insert}(\{v\}, Q_m)^\dagger$ *//insert v into the corresponding table based on v 's location to u*
9. $\text{computeAllLinks}^\ddagger$ *//re-compute the links (edges) incident on itself*
10. **end** of the synchronized block

*The detail of the function $\text{checkQuadrant}()$ is in Section 4.3.2

† The detail of the function $\text{insert}()$ is in Section 4.3.2

‡ The detail of the function computeAllLinks is in Appendix C

Appendix C

FUNCTION COMPUTE ALL LINKS []

```
1.   for each table  $Q_i$ 
2.       for each table item  $j$  in  $Q_i$     /* where  $j$  begins from 2 because the node recorded in
                                           the table item  $Q_i[1]$  is node  $u$  itself, see Figure 4.2*/
                                           //check if there is at least one neighbor node in the rectangle  $R(uv)$ 
3.       {endPoint}  $\leftarrow Q_i[j].\text{"savedmessagefromneighbors"}$ 
4.       {interimNode}  $\leftarrow \{u\}.\text{"horizontalnextnode"}$ 
5.       continue  $\leftarrow \text{true}$ 
6.       hasLink  $\leftarrow \text{true}$     //initialize  $u$  and  $v$  to be connected
7.       while (continue && ({interimNode}  $\neq$  {endPoint}))
8.           upperboundaryY  $\leftarrow \max(\{u\}.\text{posY}, \{\text{endPoint}\}.\text{posY})$ 
9.           lowerboundaryY  $\leftarrow \min(\{u\}.\text{posY}, \{\text{endPoint}\}.\text{posY})$ 
10.          if ({interimNode}.\text{posY} > lowboundY &&
                {interimNode}.\text{posY} < upboundY)
11.              hasLink  $\leftarrow \text{false}$ 
12.              continue  $\leftarrow \text{false}$ 
13.          {interimNode}  $\leftarrow u.\text{"verticalnextnode"}$ 
14.          while (continue && {interimNode}  $\neq$  endPoint)
15.              upperboundaryX  $\leftarrow \max(\{u\}.\text{posX}, \{\text{endPoint}\}.\text{posX})$ 
```

```

16.         lowerboundaryX  $\leftarrow \min(\{u\}.posX, \{endPoint\}.posX)$ 
17.         if ( $\{interimNode\}.posX > lowboundX$  &&
             $\{interimNode\}.posX < upboundX$ )
18.             hasLink  $\leftarrow \textbf{false}$ 
19.             continue  $\leftarrow \textbf{false}$ 

        /* check if there is a "bigger empty rectangle" between u and v that is
           big enough to put the test square in */
20.         if ( $abs(\{u\}.posY - \{endPoint\}.posY) \geq abs(\{u\}.posX - \{endPoint\}.posX)$ )
21.              $\{leftPoint\} \leftarrow \maxX(\{u\}, \{endPoint\})$ 
22.              $\{rightPoint\} \leftarrow \minX(\{u\}, \{endPoint\})$ 
23.             widthOfGap  $\leftarrow abs(\{leftPoint\}."horizontalnextnode"$ 
                 $.posX - \{rightPoint\}."horizontalnextnode".posX)$ 
24.             if ( $widthOfGap < abs(\{u\}.posY - \{endpoint\}.posY)$ )
25.                  $\{hasLink \leftarrow \textbf{false}\}$ 
26.             else
27.                  $\{upPoint\} \leftarrow \maxY(\{u\}, \{endPoint\})$ 
28.                  $\{downPoint\} \leftarrow \minY(\{u\}, \{endPoint\})$ 
29.                 heightOfGap  $\leftarrow abs(\{upPoint\}."verticalnextnode"$ 
                     $.posY - \{downPoint\}."verticalnextnode".posY)$ 
30.                 if ( $heightOfGap < abs(\{u\}.posX - \{endpoint\}.posX)$ )
31.                      $\{hasLink \leftarrow \textbf{false}\}$ 
32.                  $\{endPoint\}."link status" \leftarrow hasLink$ 

```

Bibliography

- [1] P. Bose, L. Devroye, W. Evans, and D. Kirkpatrick. On the spanning ratio of gabriel graphs and beta-skeletons. *SIAM J. Discret. Math.*, 20(2):412–427, 2006.
- [2] P. Bose, J. Gudmundsson, and M. Smid. Constructing plane spanners of bounded degree and low weight. In *ESA '02: Proceedings of the 10th Annual European Symposium on Algorithms*, pages 234–246, London, UK, 2002. Springer-Verlag.
- [3] P. Bose, P. Morin, I. Stojmenović, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. *Wirel. Netw.*, 7(6):609–616, 2001.
- [4] L. P. Chew. There are planar graphs almost as good as the complete graph. *Journal Of Computer And System Sciences.*, 39:205–219, 1989.
- [5] D. P. Dobkin, S. J. Friedman, and K. J. Supowit. Delaunay graphs are almost as good as complete graphs. *Discrete Comput. Geom.*, 5(4):399–407, 1990.
- [6] D. Eppstein. Beta-skeletons have unbounded dilation. *Comput. Geom. Theory Appl.*, 23(1):43–52, 2002.
- [7] K. R. Gabriel and R. R. Sokal. A new statistical approach to geographic variation analysis. *Systematic Zoology*, 18(3):259–278, 1969.
- [8] J. Gao, L. J. Guibas, J. Hersberger, L. Zhang, and A. Zhu. Geometric spanner for routing in mobile networks. In *MobiHoc '01: Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, pages 45–55, New York, NY, USA, 2001. ACM Press.
- [9] J. E. Goodman and J. O'Rourke. *Handbook of Discrete and Computational Geometry*. CRC Press, New York, 1997.
- [10] J. W. Jaromczyk and G. T. Toussaint. Relative neighborhood graphs and their relatives. *Proceedings of the IEEE*, 80(9):1502–1517, 1992.
- [11] 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, New York, NY, USA, 2000. ACM Press.
- [12] J. M. Keil and C. A. Gutwin. The delauney triangulation closely approximates the complete euclidean graph. In *WADS '89: Proceedings of the Workshop on Algorithms and Data Structures*, pages 47–56, London, UK, 1989. Springer-Verlag.
- [13] J. M. Keil and C. A. Gutwin. Classes of graphs which approximate the complete euclidean graph. *Discrete and Computational Geometry*, 7(1):13–28, 1992.

- [14] E. Kranakis, D. Krizanc, and E. Williams. Directional versus omnidirectional antennas for energy consumption and k -connectivity of networks of sensors. In *OPODIS*, pages 357–368, 2004.
- [15] E. Kranakis, H. Singh, and J. Urrutia. Compass routing on geometric networks. In *Proc. 11 th Canadian Conference on Computational Geometry*, pages 51–54, Vancouver, August 1999.
- [16] F. Kuhn, R. Wattenhofer, and A. Zollinger. Asymptotically optimal geometric mobile ad-hoc routing. In *DIALM '02: Proceedings of the 6th international workshop on Discrete algorithms and methods for mobile computing and communications*, pages 24–33, New York, NY, USA, 2002. ACM Press.
- [17] 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 & computing*, pages 267–278, New York, NY, USA, 2003. ACM Press.
- [18] L. Li, J. Y. Halpern, P. Bahl, Y.-M. Wang, and R. Wattenhofer. Analysis of a cone-based distributed topology control algorithm for wireless multi-hop networks. In *PODC '01: Proceedings of the twentieth annual ACM symposium on Principles of distributed computing*, pages 264–273, New York, NY, USA, 2001. ACM Press.
- [19] X. Li, G. Calinescu, and P. Wan. Distributed construction of planar spanner and routing for ad hoc wireless networks. *Proc. 21st IEEE INFOCOM*, 3:1268–1277, 2002.
- [20] X.-Y. Li. *Mobile Ad Hoc Networking*, chapter Topology Control in Wireless Ad Hoc Networks, pages 175–203. Wiley-IEEE Press, 2004.
- [21] X.-Y. Li, G. Calinescu, P.-J. Wan, and Y. Wang. Localized delaunay triangulation with application in ad hoc wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 14(10):1035–1047, 2003.
- [22] X.-Y. Li, P.-J. Wan, Y. Wang, and O. Frieder. Sparse power efficient topology for wireless networks. In *HICSS '02: Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS'02)-Volume 9*, page 296.2, Washington, DC, USA, 2002. IEEE Computer Society.
- [23] X.-Y. Li and Y. Wang. Efficient construction of low weight bounded degree planar spanner. In *COCOON*, pages 374–384, 2003.
- [24] X.-Y. Li and Y. Wang. Efficient localized routing for wireless ad hoc networks. In *ICDCSW '03: Proceedings of the 23rd International Conference on Distributed Computing Systems*, page 691, Washington, DC, USA, 2003. IEEE Computer Society.

- [25] T. Lukovszki. *New Results on Geometric Spanners and Their Applications*. PhD thesis, University of Paderborn, 1999.
- [26] S. Martinez. Some bounds on the coupon collector problem. *Random Struct. Algorithms*, 25(2):208–226, 2004.
- [27] M. D. Penrose. The longest edge of the random minimal spanning tree. *The Annals of Applied Probability*, 7(2):340–361, 1997.
- [28] M. D. Penrose. On k -connectivity for a geometric random graph. *Random Struct. Algorithms*, 15(2):145–164, 1999.
- [29] R. Rajaraman. Topology control and routing in ad hoc networks: a survey. *SIGACT News*, 33(2):60–73, 2002.
- [30] V. Rodoplu and T. H. Meng. Minimum energy mobile wireless networks. *IEEE J. Selected Areas in Communications*, 17(8):1333–1344, 1999.
- [31] G. T. Toussaint. The relative neighborhood graph of a finite planar set. *Pattern Recognition*, 12(4):261–268, 1980.
- [32] Y. Wang and X.-Y. Li. Localized construction of bounded degree and planar spanner for wireless ad hoc networks. In *DIALM-POMC '03: Proceedings of the 2003 joint workshop on Foundations of mobile computing*, pages 59–68, New York, NY, USA, 2003. ACM Press.
- [33] A. C. Yao. On constructing minimum spanning trees in k -dimensional spaces and related problems. *SIAM Journal on Computing*, 11(4):721–736, 1982.