A Distributed Algorithm for a Back-bone Planar Subnetwork Selection in a MANET to Improve the Efficiency of Face Routing Algorithm

Chunhua Zhang

A Thesis

in

the Department of Computer Science

Presented in Partial Fulfillment of the Requirements
for Degree of Master of Computer Science at
Concordia University
Montreal, Quebec, Canada

August 2005

Your file Votre référence
ISBN: 0-494-10300-0
Our file Notre référence
ISBN: 0-494-10300-0

ABSTRACT

A Distributed Algorithm for a Back-bone Planar Subnetwork Selection
in a MANET
to Improve the Efficiency of Face Routing Algorithm

Chunhua Zhang

This thesis proposes a new method of using face routing [2] in an ad hoc, position-based, network that aims to lower the hop count of routes. In our method, we first establish a back-bone subnetwork of a given mobile ad hoc network (MANET). This back-bone subnetwork includes only some nodes of the MANET, but any node of the MANET is at most one hop away from a node in the subnetwork. Furthermore, the back-bone subnetwork is connected whenever the original MANET forms a connected network. The algorithm for the back-bone subnetwork extraction is fully distributed. In our proposal, the routing in a MANET is done by first sending the message to a neighboring node of the back-bone subnetwork, then the routing continues in the back-bone subnetwork using the face routing on the Gabriel Graph of the back-bone subnetwork until a neighbor of the destination need is reached. An empirical evaluation of the new approach is done.

Key Words: mobile ad-hoc wireless networks (MANETs), face routing algorithm, planar, relative neighbor graph (RNG), Gabriel Graph (GG), multihop

ACKNOWLEDEMENTS

The author would like to thank Professor Jaroslav Opatrny for supervising the new approach, reading and commenting on all versions of this thesis.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

A mobile ad hoc network (MANET) consists of a system of wireless mobile nodes, often called hosts. Each host can communicate directly only with other hosts within its transmission range. Typically, not all hosts are within transmission range of each other, and the hosts should freely and dynamically self-organize into a network, allowing people and devices to communicate without any preexisting communication infrastructure. The mobility implies that the physical communication connectivity in such a network may keep changing. When an ad hoc wireless network is used, a routing of messages between hosts that are not within transmission range of each other needs to be considered. Routing protocols are to find a path to be followed by data packets from a source node to a destination node. Communication between two hosts in the ad hoc wireless network is generally achieved by multi-hop routing along a route where intermediate nodes forward the packets. Because of the highly dynamic topology, absence of established infrastructure for centralized administration, bandwidth-constrained wireless links, and energy-constrained nodes, routing protocols in traditional wired networks cannot be directly applied in ad hoc wireless networks [18]. A variety of routing protocols for ad hoc wireless networks has been proposed. The protocols depend on what information is available to the hosts. When the location of the destination host is not available, routing protocols are mostly based on *flooding* [29, 30], where a host sends the message to all its neighbors. Routing algorithms based on flooding guarantee the delivery of the message; however, it uses a lot of the available bandwidth. When each host knows its geometrical position and the position of the destination host is known, routing can be done without the use of flooding [2, 4, 5, 9]. An important technique for discovering routes between any two nodes in a position-based ad hoc wireless network is the face routing [2]. A face routing algorithm succeeds in discovering a route from a source node to a target node in a network using only local knowledge of the network, providing that the underlying network corresponds to a planar graph. If an ad-hoc network consists of hosts that have the same transmission range, there are several algorithms that allow an extraction of a planar subnetwork from such an ad hoc network. The two best known algorithms are the Relative Neighborhood Graph (RNG) [11, 5] and Gabriel Graph (GG) [11, 5] algorithms. But both of these algorithms suffer from the *multiple hop effect:* the elimination of crossing is done by elimination of longer links in the network and the routes in the extracted planar subnetwork contain many short hops. It has been pointed out that the use of short hops should be avoided [28].

In this thesis we propose a method by which, given a location based ad-hoc network G consisting of hosts with identical transmission ranges, we select a subset of hosts G' such that:

- G' forms a connected network,
- any host of G is within at most one hop from a node in G',
- planar subnetwork extracted from G does not contain many short hops.

The algorithm for selection of G' is distributed and uses only local information of the network.

The routing from node u to node v in this network is done by first sending a message at most one hop to a neighbor in G', routing the message in the planar subnetwork extracted from G to a neighbor of v in G', followed by at most one hop to v.

This thesis is organized as follows: Section 2 presents the relevant literature reviews on networks, graphs, and routing. In Section 3 we give the details of our approach to extracting a geometric planar subgraph. Section 4 contains empirical studies of our approach. Section 5 gives the conclusions and possible extensions of our work.

2. REVIEW OF RELATED RESULTS

2.1 Networks and network layer in OSI

A communication network, in its simplest form, is a set of equipment and facilities that provides a service: the transfer of information between users located at various geographical points. The most familiar example of a communication network is the telephone network, which provides telephone service; other examples of networks include computer networks, television broadcast networks, cellular networks and the internet. Different services differ in the details of how and in what form information is transferred. The details of the service influence the design of the network. Despite of any transfer mode, the network must be designed to have enough flexibility to provide support for current services and to accommodate future services. Communication functions in a network can be grouped into related and manageable sets, according to the following issues related to the network [33]:

1. The transport across a network of data from a process in one machine to the process at another machine.
2. The routing and forwarding of packets across multiple hops in a network. In this thesis, we focus on the routing which is to find a path from the source to the destination.
3. The transfer of a frame of data from one physical interface to another.

The sets that the communication functions are grouped into are called layers. Layers are essential elements for routing in a network. The term network architecture refers to a set of protocols that specify how every layer is to function. The Open System Interconnection (OSI) [15] is the commonly used system in the network architecture. It specifies how the overall communication process can be organized into functions that are carried in seven layers. The most important current network architecture is TCP/IP.

OSI is composed of 7 layers. From the bottom to the top, they are physical layer, data link layer, network layer, transport layer, session layer, presentation layer and application layer. In general each layer adds a header, possible a trailer, to the block of information it accepts from the layer above. At the destination each layer reads its corresponding header to determine what action to take and it eventually passes the block of information to the layer above after removing the header and trailer.

When the block of information reaches the network layer, this layer provides the transfer of data in the form of packets across a communication network. A key aspect of the packet transfer service is the routing of the packets from the source machine to the destination machine, typically traversing a number of transmission links and network nodes where routing is carried. Routing protocol is used to select paths across a network. The nodes in the network must work together to perform the routing effectively. This function makes the network layer the most complex in the reference model.

When two machines are connected to the same network, a single address space and routing procedure are used. However, when the two machines are connected to different networks, the transfer of data must traverse two or more networks that possible differ in their internal routing and addressing scheme. Internetworking protocols are necessary to route the data between gateways/routers that connect the intermediate networks. The internetworking protocols must also deal with differences in addressing and differences in the size of the packets that are handled within each network.

2.2 Classification of networks

Networks can be classified into wired and wireless networks. Wireless networks can be classified into Infrastructure Dependent Wireless Networks and Mobile Ad hoc Wireless Networks [31]. We will describe basic concepts about them respectively in the following paragraphs. However, the main focus is on mobile ad hoc wireless networks. Figure 2.1 shows the classification of networks.

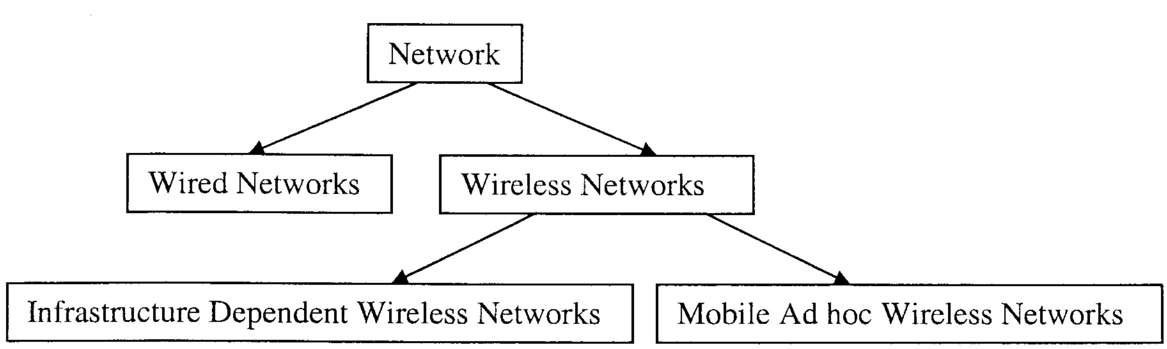


Figure 2.1 Classification of networks

2.2.1 Wired versus Wireless Networks

Based on the use of wired connection, networks can be classified into two major categories. One is *wired networks*, the other is *wireless network*.

Wired networks, also called Ethernet networks, can be classified as local area network (LAN), metropolitan area network (MAN) and wide area network (WAN), depending on the size of the network. A wired network is simply a collection of two or more machines linked by Ethernet cables, such as twisted pair, coaxial copper cable, optic fiber and so on. Their popularity stems from the speed at which data can be transferred over an Ethernet connection. Ethernet is the fastest wired network protocol, with connection speeds of 10 megabits per second (Mbps) to 100 Mbps or higher. Wired networks can also be used as part of other wired and wireless networks.

Wireless network uses electromagnetic radio waves for exchange of information, instead of wired cable. The primary objective of most wireless communication in existence today is to support voice communication. As shown in Figure 2.2, the analog voice signal is converted into digital signal by sampling of the analog signal, quantization, and binary encoding [31]. Next, modulation techniques are to perform the conversion from digital signal to electromagnetic. During the transmission of electromagnetic waves, Multiple Access Techniques and Error Control must be considered. In this thesis, we mainly focus on wireless networks.

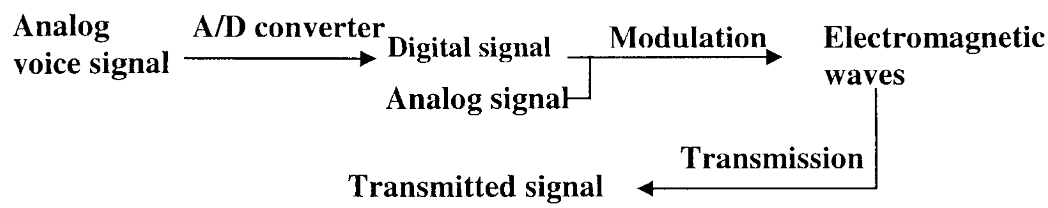


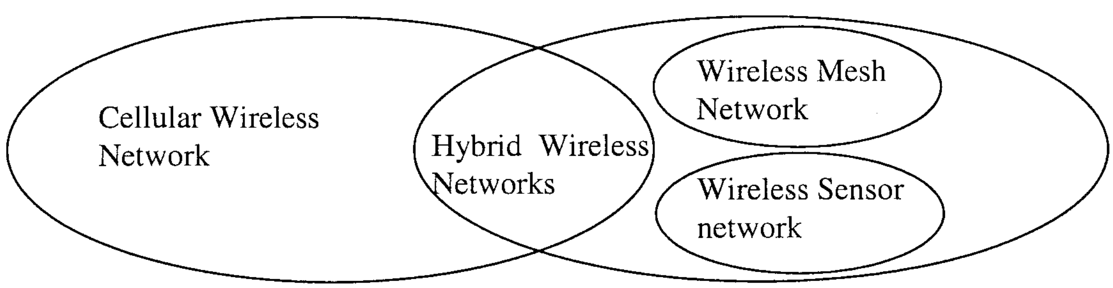
Figure 2.2 Wireless network Communication Procedure

2.2.2 Classification of Wireless Networks

Wireless networking enables two or more computers to communicate using standard network protocols without network cables. Since their emergence in the1970s, *wireless networks* have become increasingly popular in the computing industry. In the past decade, wireless networks have enabled true mobility. There are two versions of mobile wireless networks based on the use of infrastructure.

The first one is an *infrastructure dependent wireless network* [31]. It typically contains a wired backbone with the last hop being wireless. So it is also called *single-hop wireless network.* The *Cellular Phone Syste*m is an example of an infrastructure dependent network [17].

The second one is a *Mobile ad hoc wireless network (MANET).* It utilizes multi-hop radio relaying and is capable of operating without supports of any fixed infrastructure. It is entirely wireless. So it is also called *infrastructureless networks* [31]. The specific examples of MANETs are *Wireless mesh networks* and *wireless sensor networks* [31]. Wireless mesh networks are multi-hop systems in which devices assist each other in transmitting packets through the network, especially in adverse conditions. You can drop these ad hoc networks into place with minimal preparation, and they provide a reliable, flexible system that can be extended to thousands of devices [19]. Wireless sensor networks are formed when a set of small sensor devices that deployed in an ad hoc fashion cooperate for sensing a physical phenomenon [20]. The activity of sensing can be periodic or sporadic. Figure 2.3 shows a representation of different wireless networks.



Infrastructure Dependent Wireless Networks (Single-hop Wireless Networks)

Mobile Ad hoc wireless Network (Multi-hop Wireless Networks)

Figure 2.3: Infrastructure and Ad Hoc Wireless Network [31]

Ad hoc mobile wireless networks consist of wireless hosts that utilize multi-hop radio relaying and communicate with each other in the absence of fixed infrastructure. On contrast to cellular wireless network, a MANET doesn't have any central coordinator or base station. Due to considerations such as radio power limitation, power consumption and channel utilization, a mobile host may not be able to communicate directly with other hosts in a single-hop fashion. In this case, the packets sent by the source host are relayed by several intermediate hosts before reaching the destination host. A multi-hop scenario occurs. So it is also called *multi-hop wireless network.*

In MANETs, a central coordinator is absent, and routing and resource management are done in a distributed manner. All hosts cooperate to enable communication among themselves. This requires each node to be more intelligent so that it can function both as a network host for transmitting and receiving data and as a network router for routing packets from other hosts. Hence, the mobile hosts in MANETs are more complex than their counterparts in a cellular wireless networks.

In this thesis, we simply introduce Infrastructure Dependent Wireless Networks and focus on MANETs.

2.3 Graph terminology

Networks and many real-world situations can conveniently be described by means of a diagram consisting of a set of points together with lines joining certain pairs of these points. For example, the points could represent people and lines join pairs of friends; or points might be communication centers and lines represent communication lines. This type of situation is mathematically abstracted into a *graph.* We follow the terminology from [10].

A ***graph*** G is an ordered triple consisting of a nonempty set $V(G)$ of vertices, a set $E(G)$ of *edges* which is disjoint from $V(G)$, and an incidence function ψ_G that associates with each edge of G an unordered pair (not necessarily distinct) of vertices of G. If e is an edge and u and v are vertices such that $\psi_G(e) = uv$, then e is said to ***join*** u and v; the vertices u and v are called the ***ends*** of e.

A ***walk*** in a graph is a finite non-null sequence $W = v_0 e_1 v_1 e_2 v_2 \ldots e_k v_k$, whose terms are alternately vertices and edges, such that, for $1 \leq i \leq k$, the ends of e_i are v_{i-1} and v_i. If the vertices $v_0, v_1, \ldots, v_k$ are distinct, W is called a ***path***.

A graph H is a ***subgraph*** of G if $V(H) \subseteq V(G)$, $E(H) \subseteq E(G)$, and ψ_H is the restriction of ψ_G to E(H). $H \subseteq G$ represents that H is a subgraph of G. If H is a subgraph of G, G is a ***supergraph*** of H.

A graph can be represented in a plane by drawing a point for each vertex and drawing a line between two vertices jointed by an edge. In general, positions of vertices in this representation are arbitrary.

To represent networks, we often use a ***geometric graph***, instead of a graph. In a *geometric graph*, each vertex has its coordinates and we place the vertices in the place according to their coordinates. The edges are represented by straight line segments. A position based ad hoc wireless network can be represented by a geometric graph G = (V, E) where V represents the hosts in the network. E is a set of edges and an edge connects any pair of nodes that can communicate directly.

Unit disk graph *U(S)* is a geometric graph that contains a vertex for each element of *S*. An edge (*u*, *v*) is present in *U(S)* if and only if *dist(u, v)* is less than or equal with a fixed unit, where *dist(u, v)* denotes the Euclidean distance between *u* and *v* [2].

Network radio hardware is traditionally viewed as having a nominal open-space transmission range R. For example, R is 250 meters for 900 MHz DSSS WaveLAN. An ad hoc networks in which all nodes have the same transmission range can be represented by a unit graph *U = (V, E')*, a geometric graph that contains a vertex for each element of *V*. An edge *(u, v)* is present in *G* if and only if *dist(u, v)* <= transmission range *R*, where *dist(u, v)* denotes the Euclidean distance between *u* and *v*. Unit disk graphs are a reasonable mathematical abstraction of wireless networks in which all nodes have equal broadcast ranges. We will refer to the elements of V alternately as hosts, nodes or vertices.

Those geometric graphs whose edges intersect only at their ends are called ***geometrical planar graph***.

Note that when calculating a path length, we take the cost of any edge as 1 regardless of actual distance, since in wireless communication, we assume the delivery cost between two nodes at distance at most R is 1. So the cost of any edge is 1 and we refer to a path length as the *hop-count*.

2.4 Routing in an ad hoc wireless network

When an ad hoc wireless system is designed and deployed, many issues should be considered, such as medium access scheme, routing, multicasting, transport layer protocol, pricing scheme, quality of service provisioning, self-organization, security, energy management, addressing and service discovery, scalability, and deployment consideration [13].

Among these issues, the routing problem is essential to MANETs. Many *routing protocols* have been proposed for MANETs [2, 4, 5, 6, 7, 8, 9]. They are more complicated compared with single-hop wireless network. The responsibilities of a routing protocol are to:

- Exchange the route information
- Find a feasible path to a destination based on hop count, power or cost metrics.
- Gather information about the path breaks.
- Mend the broken path.
- Utilize minimum bandwidth.

Finding a feasible path to a destination is referred as a *routing algorithm.* Routing algorithms can be classified into several categories based on different metrics.

Based on whether the positions for the hosts in an ad hoc wireless network are used, routing algorithms are classified into *topology-based routing algorithm* and *position-based routing algorithm* [16]. We are going to introduce them respectively in Section 2.4.1 and 2.4.2. Since MANETs change their topology frequently and without prior notice, routing algorithms in such networks are more complicated and challenging than others. In this thesis, we briefly review the topology-based routing algorithms and focus more on the position-based routing algorithm.

2.4.1 Topology-based routing algorithms

Topology-based routing algorithms use the information about the links that exist in the network to perform packet forwarding. They can be further divided into *proactive*, *reactive* and *hybrid* algorithms [18].

Proactive routing algorithms maintain routing information about the available paths in the network even if these paths are not currently used. The main drawback of these algorithms is that the maintenance of paths may occupy a significant part of the available bandwidth if the topology of the network changes frequently. Distance-vector routing

(DSDV) [21] and link-state routing (e.g., OLSR) are examples of proactive routing algorithms [22].

Reactive routing algorithms maintain only the routes that are currently in use. This reduces the burden on the network when only a small subset of all available routes is in use at any time. However, these algorithms have three limitations: 1) since routes are only maintained while in use, it is required to perform a route discovery before package can be exchanged between communication peers. This leads to a delay for the first packet that is to be transmitted. 2) even though route maintenance for reactive routing algorithms is restricted to the routes currently in use, it may still generate a significant amount of network traffic when the topology of the network changes frequently. 3) packages during route to the destination are likely to be lost if the route to the destination changes. Dynamic souce routing (DSR) [23], TORA [24] and ad hoc on-demand distance vector (AODV) [25] are examples of reactive routing algorithms.

Hybrid routing algorithms combine local proactive routing and global reactive routing in order to achieve a higher level of efficiency and scalability. However, even a combination of both strategies still needs to maintain at least those network paths that are currently in use, limiting the amount of topological changes that can be tolerated within a given amount of time. Zone routing protocol (ZRP) [26], core extraction distributed ad hoc routing (CEDAR) [18] and zone-based hierarchical link state (ZHLS) [18] are examples of hybrid routing algorithms.

2.4.2 Position-based routing algorithms

Position-based routing algorithms eliminate some of the limitations of topology-based routing by using physical position of the participating nodes. There are two methods to get the positions. Firstly, the distance between neighboring nodes can be estimated on the basis of incoming signal strengths or the time delays in direct communications, so relative coordinates of neighboring nodes can be obtained by exchanging such information between neighbors [32]. Alternatively, the location of nodes may be available directly by communicating with a satellite (for outdoor networks), using GPS (Global Positioning System), if nodes are equipped with a small low power GPS receiver [32]. The position-based approach in routing becomes practical due to the rapidly developing software and hardware solutions for determining absolute or relative positions of nodes in indoors/outdoor ad hoc networks.

We distinguish seven main classes of existing position-based routing algorithms [17]:

1) Basic distance, progress, and direction based methods
2) Partial flooding and multi-path based path strategies
3) Depth first search based routing with guaranteed delivery
4) Nearly stateless routing with guaranteed delivery
5) Hierarchical routing
6) Assisted routing
7) Power and cost aware routing

Each class includes kinds of position-based routing algorithms. Here we do not introduce algorithms for each class, but describe the fourth class, the nearly stateless routing with guaranteed delivery. This class includes *Face routing algorithm* [2], *Greedy-Face-Greedy* (GFG) *routing algorithm* [32] and *Greedy Perimeter Stateless Routing* (GPSR) *algorithm* [5].

Face routing algorithm works well in a planar graph. It applies routing along the faces of the graph (e.g. by using the right hand rule) that intersect the line between the source and the destination. In the next section we describe it in detail.

2.5 Face Routing Algorithm

Face routing algorithm belongs to position-based routing algorithms. As same as position-based routing algorithms, we consider face routing algorithms which has following properties: Hosts know nothing about the network except their own location, the locations of the hosts to which they can communicate directly, and the exact geographical position of the destination.

In this section, we describe two kinds of *face routing algorithms* in a planar graph [2]. The first algorithm is called FACE-1. The second algorithm, called FACE-2, is a modification of this algorithm that performs better in practice.

A connected planar geometric graph G partitions the plane into *faces* that are bounded by polygonals made up of edges of G. Given a vertex v on a face f, the boundary of f can be traversed in the counterclockwise (clockwise if f is the outer face) direction using the well-known *right hand rule* which states that it is possible to traverse a maze by keeping your right hand on the wall while walking forward. Treating this face traversal technique as a subroutine, the algorithm for routing a packet from s to d is given. The operation of algorithm FACE-1 is illustrated in figure 2.4.

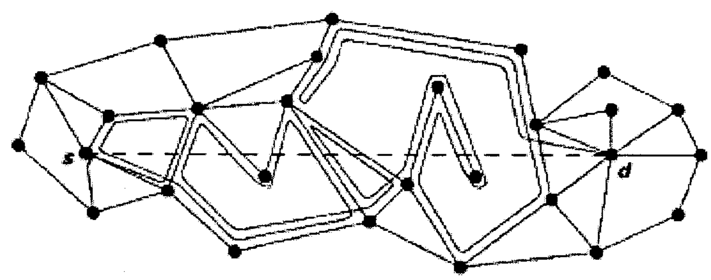


Figure 2.4 Routing from s to d using FACE-1 [2]

Algorithm: FACE-1 [2]

// Draw a line from source node *s* to destination node *d*. This line intersects some faces, and these faces will be checked one by one.

1: $p \leftarrow s$

2: **repeat**

3: let f be the face of G such that p is on the boundary of f and the boundary intersects the line (p, d)

4: **for** each edge (u, v) of f

5: **if** (u, v) intersects (p, d) in a point p' and $dist(p', d) < dist(p, d)$

$p \leftarrow p'$

end if

end for

Traverse f until reaching the edge (u, v) containing p

until $p = d$

Algorithm FACE-1 traverses completely each face of the graph intersected by the source to target line segment. In principle, this is not necessary, the traversal of a face can stop when an intersection point closer to the target is reached. This is the idea behind the algorithm FACE-2.

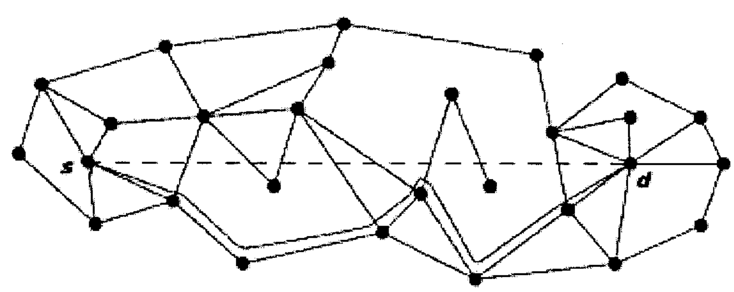


Figure 2.5 Routing from s to d using FACE-2

Algorithm: FACE-2 [2]

// Draw a line from source node s to destination node d. This line intersects some faces, and these faces will be checked one by one.

p ← s

repeat

3: let f be the face of G such that p *is* on the boundary of f and the boundary intersects line (p, d)

4: traverse f until reaching an edge (u, v) that intersects (p, d) at some point $p' \neq p$

5: $p \leftarrow p'$

6: **until** $p = d$

The operation of FACE-2 is illustrated in figure 2.5. Clearly this algorithm also terminates in a finite number of steps, since the distance to t is decreasing during each round. However, in pathological cases it may visit $\Omega(n^2)$ edges of G [2]. This can occur, for example, when G is a snakelike path from s to t that crosses the segment (s, t) many times (see Figure 2.6).

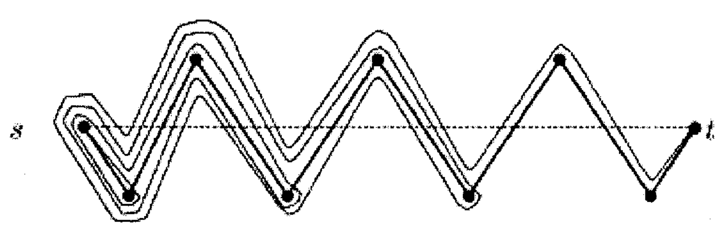


Figure 2.6 A bad input using FACE-2

Theorem [2]: Let G be a planar connected graph and s and d be two nodes of G. Then the algorithms FACE-1 and FACE-2 find a path in G from s to d.

GFG algorithm combines greedy and FACE algorithms [32]. Greedy algorithm is applied as long as possible, until delivery or a failure. In case of failure, the algorithm switches to FACE algorithm until a node closer to destination than last failure node is found, at which point greedy algorithm is applied again.

GPSR algorithm combines GFG and IEEE 802.11 medium access control scheme [32]. It uses geography to achieve small per-node routing state, small routing protocol message complexity, and extremely robust packet delivery on densely deployed wireless networks [5].

2.6 Planarisation of a Unit Disk Graph

As stated in the theorem of the preceding section, the face routing succeeds provided that the given network is planar. In this section we describe algorithms that can be used to extract a planar subgraph from a given unit disk graph. Two well-known graph planarization are the ***Relative Neighborhood Graph*** **(RNG)** [11] and ***Gabriel Graph*** **(GG)** [12]; ***RNG test*** and ***Gabriel test*** [11, 12] are two alternative algorithms to extract a planar subgraph from a given unit disk graph using local information only. Using a local test, both of algorithms yield a planar graph by removing some edges from a unit disk graph.

2.6.1 RNG graph [11, 5]

The RNG test is defined as follows:

Given a geometric graph G, an edge (u, v) is kept between vertices u and v for RNG if the distance between them, $d(u,v)$, is less than or equal to the distance between every other vertex w, and whichever of u and v is farther from w. In equational form:

$$\forall w \neq u,v : d(u,v) \leq \max\ [d(u,w), d(v,w)]$$

Figure 2.7 depicts the rule for constructing the RNG. The shaded region, the *lune* between u and v, must be empty of any *witness* node w. The boundary of the lune is the intersection of the circles about u and v of radius $d(u, v)$.

When we begin with a connected unit graph and remove edges which are not part of the RNG, note that we cannot disconnect the graph. Edge (u,v) is only eliminated from the graph when there exists a w within the lune of u and v, romove edge (u, v).

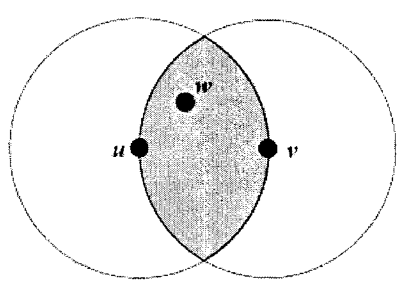


Figure 2.7: the RNG graph.
For edge (u,v) to be included, the shadow lune must contain no witness w.

Thus, eliminating an edge requires an alternate path though a witness. Each connected components in a network will not be disconnected by removing edges not in the RNG.

2.6.2 GG graph [12, 5]

The GG is defined as follows:

Given a geometric graph *G*, an edge *(u,v)* exists between vertices *u* and *v* in GG if no other vertex *w* is present within the circle whose diameter is $\overline{uv}$. In equation form:

$$\forall w \neq u,v\text{: } d^2(u,v) < [d^2(u,w) + d^2(v,w)]$$

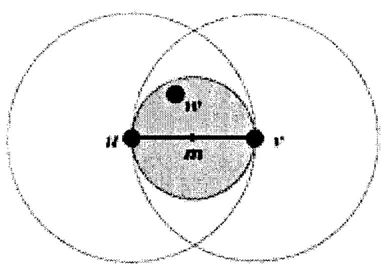


Figure 2.8: the GG graph.
For edge *(u,v)* to be included, the shadow circle must contain no witness w.

Figure 2.8 depicts the rule for constructing the RNG. Eliminating edges in the GG cannot disconnect a connected unit graph, for the same reason as was the case for the RNG.

Theorem [11, 12]: If G is a connected unit disk graph, then the subgraph G´ of G extracted by RNG test or the subgraph *G´´* extracted by the Gabriel test is a planar connected spanning subgraph of G.

Compared with the RNG test, Gabriel test merely shrinks the "test region" and creates a planar subgraph that keeps some of links that would be eliminated by the RNG test. So RNG is s subset of the GG. This is the reason why we just consider GG instead of both of RNG and GG in this thesis.

We should note that both of RNG and GG suffer from the multiple hop effect because the elimination of crossing in the geometric graph is done by elimination of longer links. The Morelia test proposed in [1] keeps slightly larger number of edges, than GG, but it is not substantially different. There has been some routing algorithm proposed with the aim of reducing the hop count. We should mention the Internal Node and Short Based Routing [8] which proposed to use a subset of nodes for routing. Similarly, some clustering schemes were proposed [8, 27]. They differ substantially from the scheme proposed in the thesis. Our new approach in Section 3 is meant to result in a routing algorithm that needs fewer hops. Hop-count is used as the metric.

3. BACK-BONE SUBNETWORK

3.1 Introduction

For multihop wireless networks, routing over many short hops (short-hop routing) may reduce energy consumption and has low interference [28]. But it stems from an oversimplified analysis that neglects some issues [28]. It has been pointed out that using short hops should be avoided [28]. In [28], twelve reasons are given why short-hop is not as beneficial as it seems to be: 1) Interference, 2) End-to end reliability, 3) Capacity and channel coding, 4) Total energy consumption, 5) Path efficiency in random network, 6) Sleep Modi or cooperation, 7) Routing overhead & route maintenance, 8) Route longevity in mobile environments, 9) Traffic accumulation and energy balancing, 10) Variance in hop length, 11) Bounded attenuation, 12) Multicast advantage. So routing over a smaller number of longer hops (long-hop routing) is a very competitive strategy.

Our approach is to select a planar subgraph that we call the ***back-bone network*** which can lead to a routing using fewer hops. More specifically, given a geometric unit disk graph G = (V, E), we extract out of it a subnetwork G' = (V', E') such that the subnetwork is connected, planar, and every node of network in V-V' is one hop away from a vertex in V'. The subnetwork G' should not contain many short hops.

The selection of the subnetwork V' is essential to get fewer hops. The process is described in detail in Section 3.2.

First we state our assumptions on the network.

- Each host has the same transmission range ***R*** along which a package can be deliveried by a host.
- Each host knows: 1) its geometric coordinates, 2) the geometric coordinates of the hosts to which it can communicate directly, 3) the geometric coordinates of the destination.
- For the purpose of the creation of the back-bone subnetwork, each host is given value ***S***, as shown in Figure 3.1. The possible values of S are discussed later.

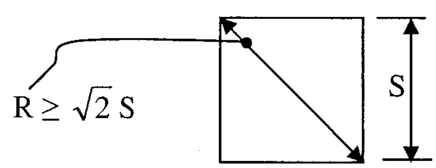


Figure 3.1 A square in a grid

3.2 The Construction of the Back-bone Network

Our approach is to extract a new planar graph through the four following steps, as shown in Figure 3.2. The last two steps are the same as those in the previous approach. The two first steps are essential to extract a planar graph with fewer hop count. We will address the first two steps in detail.

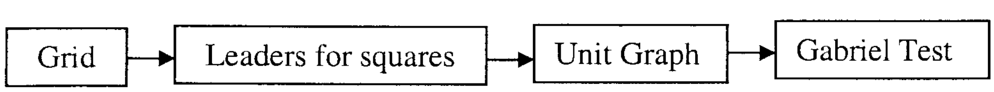


Figure 3.2 A new approach to extract a planar graph

We imagine that the plane is partitioned into squares whose side is of size S. We assume that the corners of squares are in positions $[iS,jS]$ for some integers i and j. For simplicity we identify each square whose corners are $[iS,jS]$, $[(i+1)S,jS]$, $[iS,(j+1)S]$ and $[(i+1)S,(j+1)S]$ as the square $[i][j]$, as shown in Figure 3.3. A node n belongs to square $[i][j]$ if ($i \le x < i+S$ and $j \le y < j+S$).

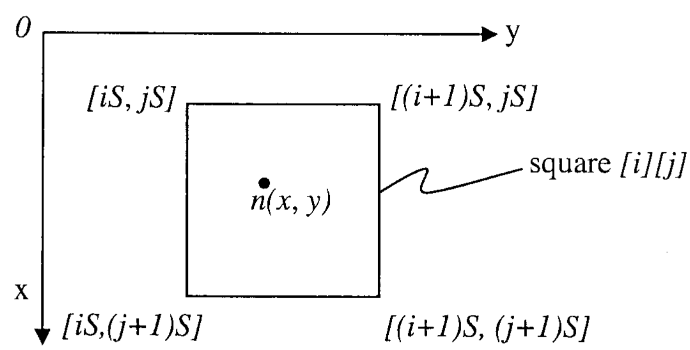


Figure 3.3 Identification of a square[i][j]

We will assume that $\sqrt{2}$ S $\le$ R $\le$ 2R. The reason for requiring $\sqrt{2}$ S $\le$ R is to allow a node in a square to be able to communicate in one hop with all nodes in the square. The main reasons for R$\le$ 2S is to be able to construct a connected back-bone network using at most neighboring squares at distance one or two. If $R > 2S$ we would need to consider neighboring squares at further distance which would result in a more complicated back-bone node selection algorithm, and it would deal with smaller squares which most likely would create a back-bone subnetwork containing a larger number of nodes and thus shorter edges. An example for partition of a MANET is shown in Figure 3.4.

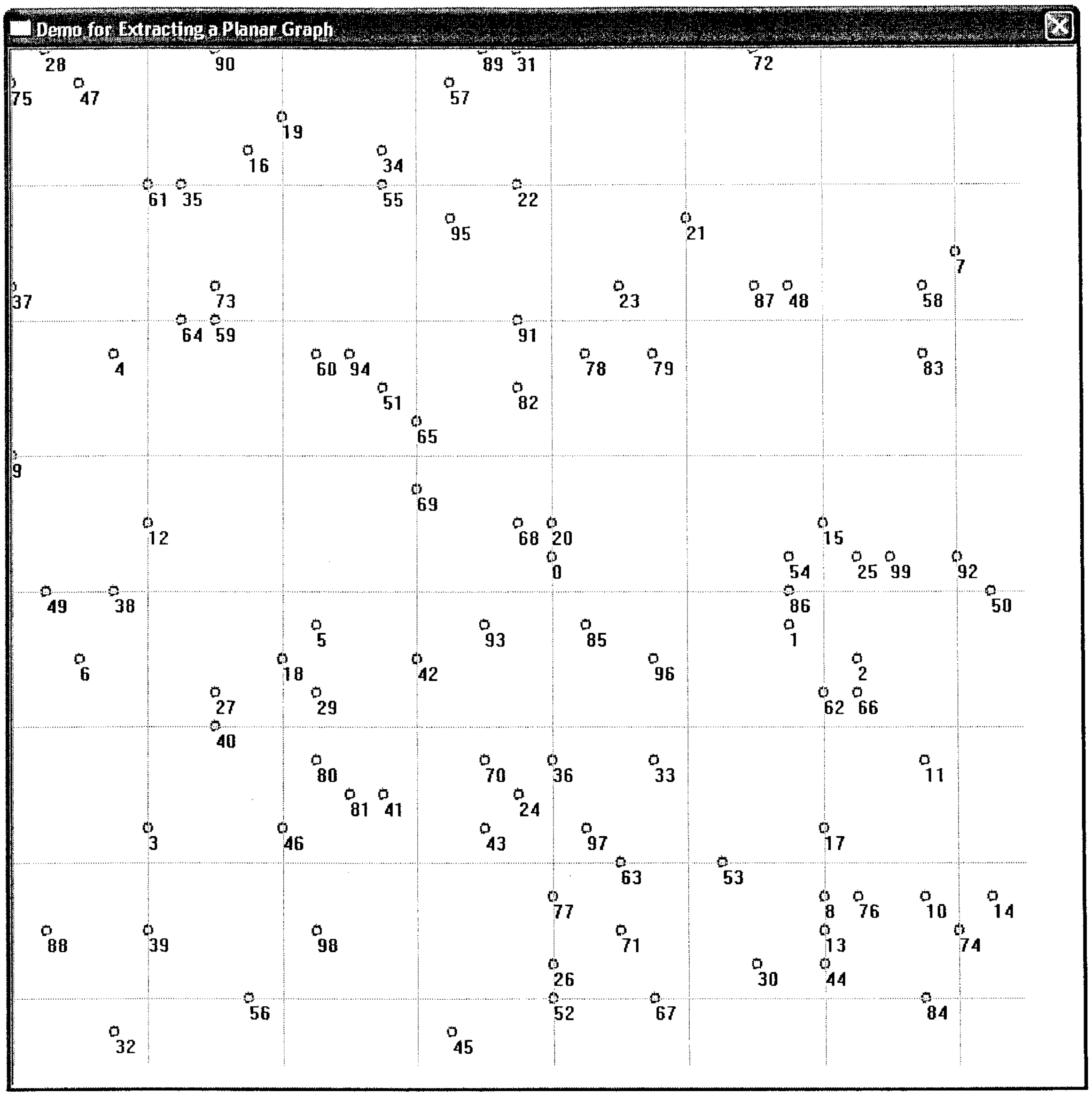


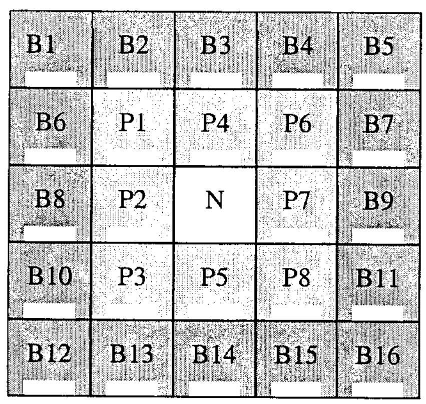
Figure 3.4 The partition of a MANET

3.2.1 Leaders' Selection Process

We regard the nodes in a square as a group. Note that a node cannot belong to two different squares and, due to the upper bound on the size of *S*, all nodes inside one group are within the communication range of each other. The algorithm for selection of the nodes for the back-bone network aims to select the nodes such that

- there are fewer "short" edges in the back-bone network,
- the back-bone network preserves the connectivity of the original network,
- and the selection can be done in a distributed manner with each node using only the knowledge of its own neighborhood.

The back-bone nodes consist of primary, secondary and tertiary leaders.



(a)

(i-2, j-2)	(i-1, j-2)	(i, j-2)	(i+1, j-2)	(i+2, j-2)
(i-2, j-1)	(i-1, j-1)	(i, j-1)	(i+1, j-1)	(i+2, j-1)
(i-2, j)	(i-1, j)	(i, j)	(i+1, j)	(i+2, j)
(i-2, j+1)	(i-1, j+1)	(i, j+1)	(i+1, j+1)	(i+2, j+1)
(i-2, j+2)	(i-1, j+2)	(i, j+2)	(i+1, j+2)	(i+2, j+2)

(b)

Figure 3.5 2-neighbor squares.
"□" means *neighbor* of the square *N*.
"□" means *2-distance neighbors* of the square *N*.

Due to the restriction on the size of *S,* a node in the square denoted on the picture as *N* can potentially communicate with nodes in the squares shown in the Figure 3.5 (a). Squares *P1, P2, …, P8*, which circle square *N*, are called ***neighbors*** of *N*. Squares *B1, B2, …, B16*, are called ***2-distance neighbors*** of *N*. Labels given in Figure 3.17 (b) are used to describe a exact position of a 2-distance neighbor. The algorithm for selection of the nodes for the back-bone subnetwork needs to consider the situation in both neighbors and 2-distance neighbors.

3.2.1.1 Selection of *Primary Leaders*

In each square, the *primary leader* (PL) is the node that is closest to the middle of a square. If more than one node has this property, then the node with smallest X coordinate among them is selected. If there is a tie, the node with the smallest Y coordinate is selected.

Clearly, as long as nodes exist in a square, a unique primary leader for the square is selected and it can be done in a distributed manner. No central control and no communication between groups is needed, and every node can determine the primary leader on its own.

If we just use primary leaders for the back-bone subnetwork, it should reduce the hop count. But at the same time, a problem may occur: the subnetwork of primary leaders may be disconnected. As shown in Figure 3.6, because the primary leader in square[i][j] cannot communicate with the PL in square[i+1][j], no node in square[i][j] can communicate with any node in square[i+1][j]. But in fact, some nodes such as *s* can communicate with the primary leader in square[i+1][j]. This problem will be resolved by adding secondary leaders to the backbone network.

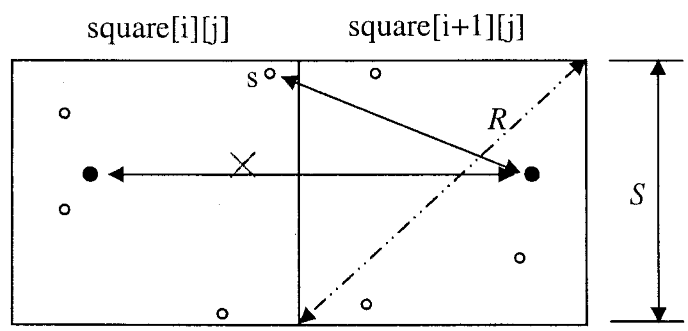


Figure 3.6 Communication between primary leaders.
"● " is a primary leader.

3.2.1.2 Selection of Secondary Leaders for Neighbors

Primary leaders (PLs) ensure that any node in the network can communicate with a node in the back-bone subnetwork. Secondary leaders ensure that when there are nodes in two neighboring squares that can communicate with each other, there will be a connection between the leaders in these squares in the backbone network. This may be needed if the distance between PLs in neighboring squares is bigger than the transmission range *R*.

If a PL in a square cannot communicate with another PL in its neighboring square, the square leader selects a *secondary leader* (SL) as an intermediate node, by which the PL can communicate with the neighboring PL.

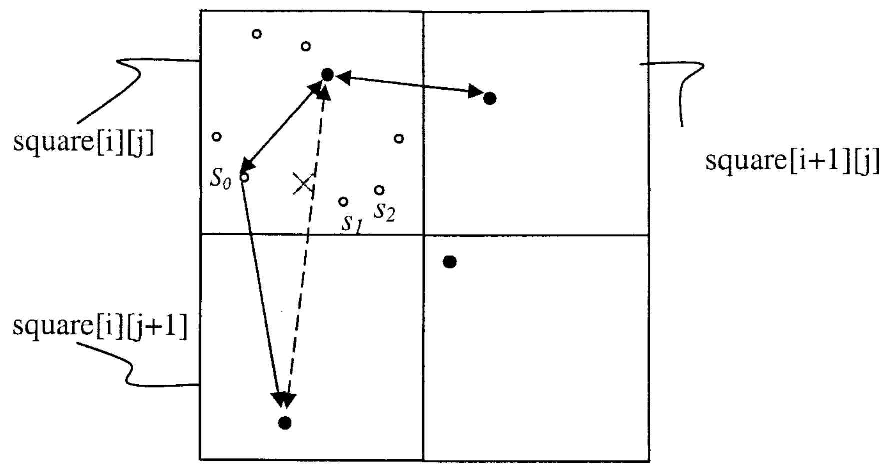


Figure 3.7 Selection of the secondary leaders
"●" means a primary leader for a square

As shown in Figure 3.7, square[i][j] neighbors on square[i][j+1] and square[i+1][j]. The PL in square[i][j] can communicate with the PL in square[i+1][j]. In contrast, it cannot communicate with the PL in square[i][j+1] because the Euclidean distance between the two PLs is more than transmission range *R*. So the PL in square[i][j] asks each node of its own square whether they can communicate with the PL in its neighboring square[i][j+1]. If such a node exists, square[i][j] designates the node, such as s_0 which satisfies this condition, as the *second leader* (SL) in square[i][j] for its neighbor square[i][j+1]. If there are several nodes eligible for the SL, it selects the one with lowest subscript number. As shown in Figure 3.7, although all nodes s_0, s_1, s_2 satisfy the condition, node s_0 is designated as SL because it has the lowest subscript number. Thus there is only one SL in a square for one of its neighbors.

Every square is circled by up to eight nonempty neighbors, as shown in Figure 3.8 So a square may have eight SLs at most, which happens when the PL in this square can not communicate with any PL in its eight neighbors. Note that SLs in a square for this square's neighbors may be the same or different. Figure 3.8 illustrates this situation.

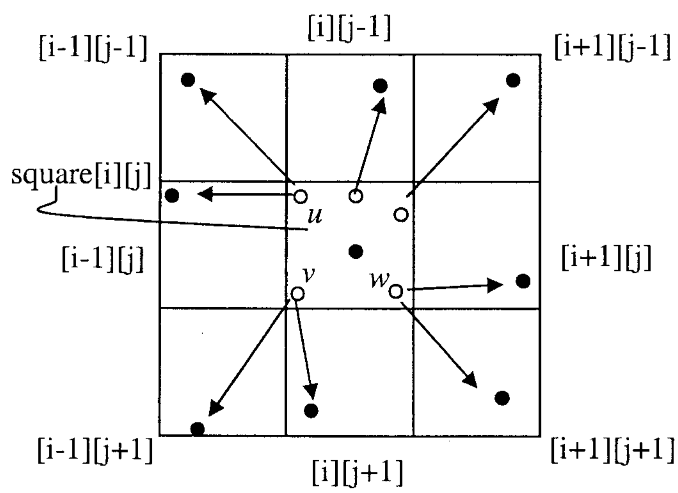


Figure 3.8 Selection of the secondary leaders
"•" means a primary leader
"o" means a secondary leader

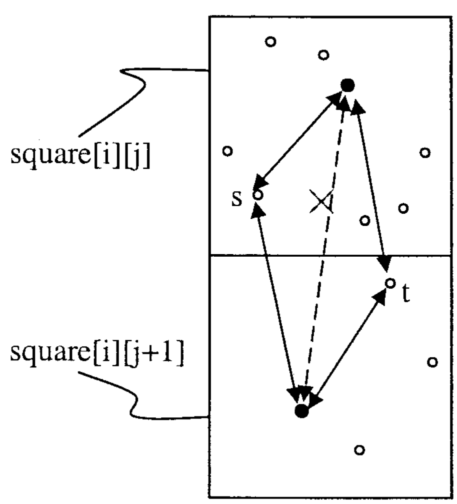


Figure 3.9 Different square has different secondary leader

Note that the selection of SLs is done in each square independent of the selection in the square's neighbors. As shown in Figure 3.9, two PLs can not communicate with each other, so SLs are necessary. The square[i][j] select the node s as the secondary leader for the neighboring square[i][j+1] whereas the square[i][j+1] select the node t as the secondary leader for the neighboring square[i][j].

The geometric subnetwork consisting of primary and secondary leaders is not necessarily a connected subnetwork of the original network. As shown in Figure 3.10, if the primary leader in square[i][j] can not communicate with the PL in square[i+1][j] and square[i][j] cannot find a second leader for square[i+1][j], any nodes in square[i][j] cannot communicate with any one in square[i+1][j]. But in fact, some nodes such as t_1 in square[i][j] can communicate with some nodes such as t_2 in square[i+1][j]. To resolve this disconnection, we also select the tertiary leaders as described in the next section.

3.2.1.3 Selection of the Tertiary Leaders (TLs) for Neighbors

If a PL in a square cannot communicate with any nodes in a neighbor and no node in the square can communicate with a PL in the neighbor, then the PL asks nodes in the square whether they can communicate with any nodes in the neighbor. If a nodes t_1 in the square can communicate with a node t_1' in the neighbor, then the PL designates nodes t_1 as a tertiary leader and asks node t_1 to inform t_1' to be a tertiary leader. We can view the pair $\{t_1, t_1'\}$ together as a *tertiary leader* (TL). If several such pairs exist, the node with lowest subscript in the square is selected. Thus the selection of tertiary leaders can be done in a distributed manner in each square and it only requires communications at distance at most two. As shown in Figure 3.10, the PL in square[i][j] cannot communicate with the PL in square[i][j+1]; moreover, the node t_1 in square[i][j], which is the closest node to the PL in square[i][j+1], cannot communicate with the PL in square[i][j+1]. The pair $\{t_1, t_1'\}$ is selected as the TL, which are the intermediate nodes by which the PL can communicate with another PL in square[i][j+1]. Although $\{t_2, t_2'\}$ or $\{t_3, t_3'\}$ satisfies the condition to be TL too, $\{t_1, t_1'\}$ has priority because t_1 has the lowest subscript number.

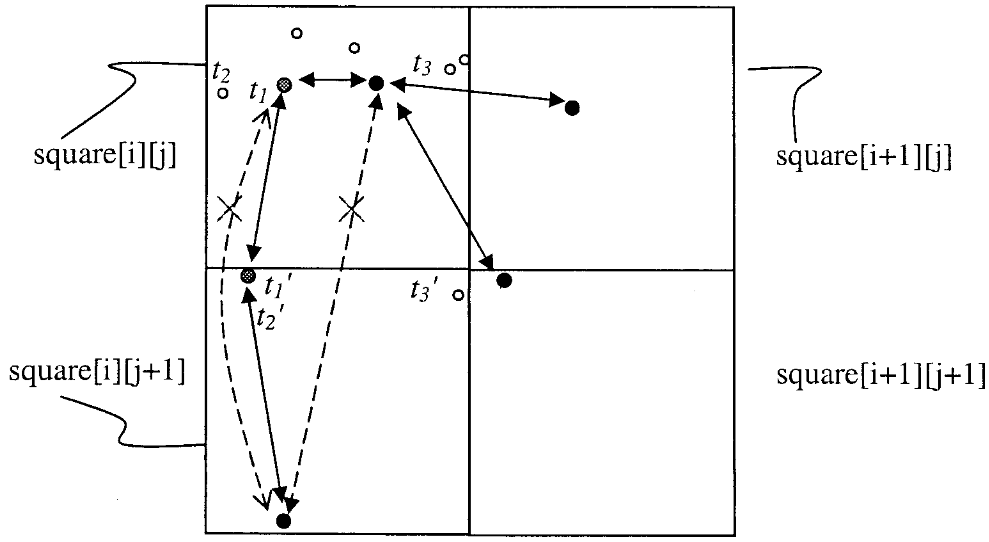


Figure 3.10 Tertiary Leader. " ⊗ " means the *tertiary leader*

A square may have a TL for a neighbor, so a square may have eight TLs for its neighbors at most, which is same as the secondary leader.

Note that a square should have different TLs for its different neighbors. As shown in Figure 3.11, all of the nodes in square[i][j], including the PL, *a* and *b*, cannot communicate with the PL in square[i-1][j-1], so square[i][j] selects node *b* and *c* as the

TL for its neighbor square[i-1][j-1]. Because of the same reason, square[i][j] selects *a* and *d* as the TL for square[i-1][j+1]. Node *c* and *d* belong to square[i][j]'s different neighbors. That is, among a TL composed of a pair of nodes, there must be a node which comes from a neighbor. So a square has different TLs for its different neighbors.

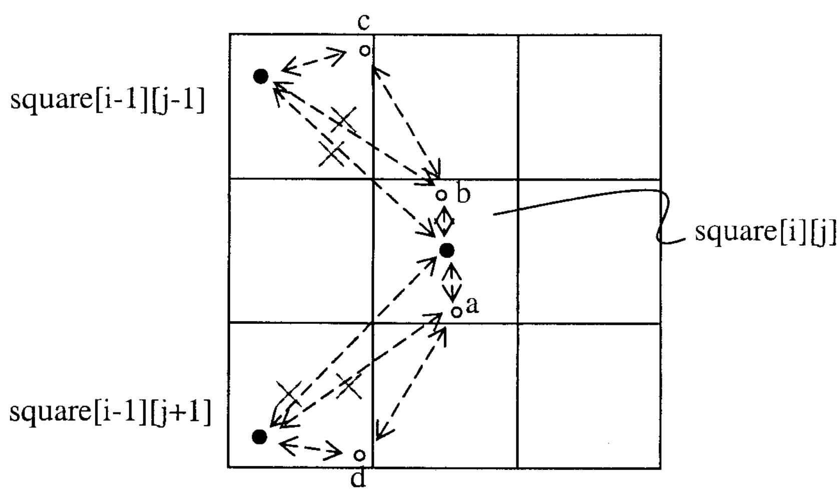


Figure 3.11 A square has different *tertiary leaders* for its different neighbors.

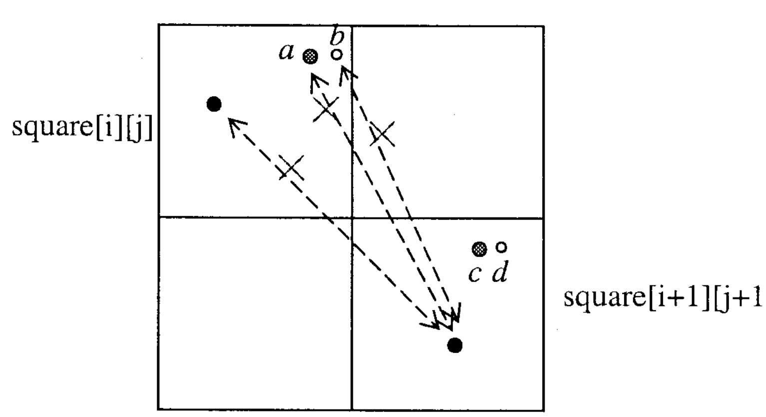


Figure 3.12 Different squares have same or different *tertiary leaders*.
" ● " means the *tertiary leader*

Note that different square may have same or different TLs, which is not same as the SLs.

As shown in Figure 3.12, all of the nodes, including the PL, nodes *a* and *b*, in square[i][j] cannot communicate with the primary leader in square[i+1][j+1], so {*a*, *c*} are selected as the TL in square[i][j] for square[i+1][j+1]. Because of the same reason, square[i+1][j+1] needs a TL. If square[i+1][j+1] selects {*c*, *a*} as the TL for its neighbor square[i][j], the two squares have the same TLs; if {*d*, *b*}, {*d*, *a*} or {*c*, *b*}are selected as the TL, the two squares have different TLs.

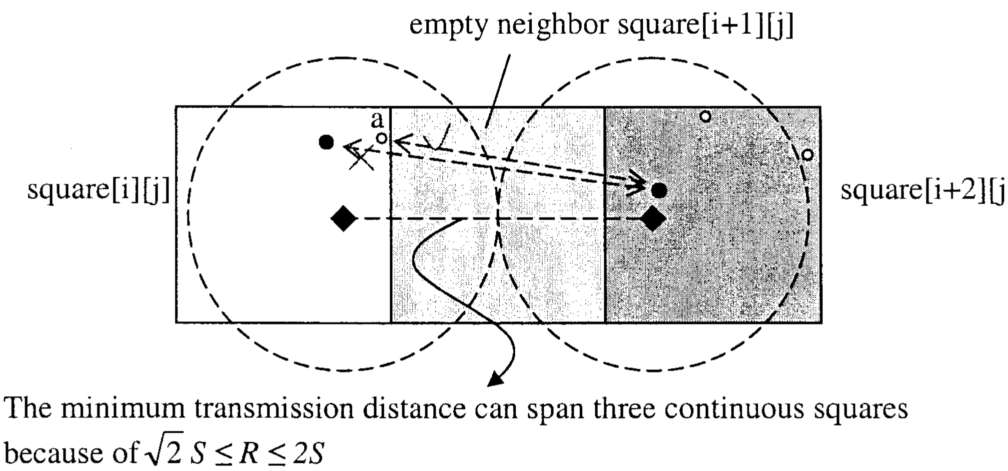


Figure 3.13 Communication among nodes which span three squares.

Because of $\sqrt{2}\,S \le R \le 2S$, the minimum transmission distance between two nodes can span three continuous squares, as shown in Figure 3.13. For example, node *a* in square[i][j] can communicate with a node in square[i+2][j] because the distance between them is less than *R*. If we ignore the transmission between a square and its 2-distance neighbors, a disconnected subnetwork might result. We give an example to explain the problem. As shown in Figure 3.13, use PLs, SLs, and TLs to construct a geometric graph. The PL in square[i][j] cannot communicate with the PL in square[i+2][j] because the distance between them is bigger than *R*. At the same time, because the neighbor square[i+1][j] is empty, a SL or TL for the neighbor is not available. The PL in square[i][j] find nothing as intermediate nodes to another PL in square[i+2][j]. As a result, no nodes in square[i][j] can communicate with those in square[i+2][j]. But in fact, node *a* can communicate with the PL in square[i+2][j]. We thus design the SLs and TLs for 2-distance neighbors as the resolution to the problem. As in the example, this is needed when some of the neighboring squares are empty or there is no node in some squares it can communicate with. In next section we address this issue in detail.

3.2.1.4 Selection of Secondary Leaders (SLs) for 2-distance neighbors (2-DNs)

When the PL in a square detects that there is a neighbor such that it cannot communicate with any node in the square and the PL in the square cannot communicate with another PL in a 2-DN next to a seemingly empty neighbor, the PL asks each node of its own square whether they can communicate with a PL of the 2-DN. If such a node exists, the square designates a node which satisfies this condition as the *secondary leader* (SL) for the 2-DN. If there are several such nodes it selects the one with lowest subscript number from itself. Thus there is only one second leader in a square for a 2-DN.

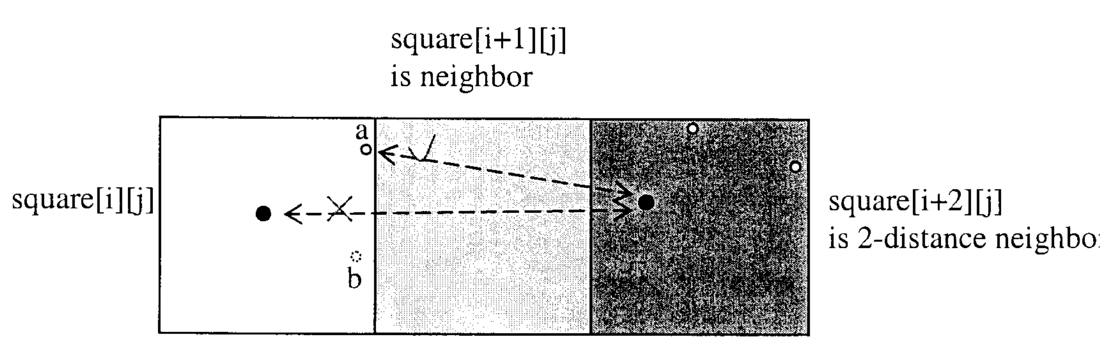


Figure 3.14 A secondary leader for a 2-distance neighbor

As shown in Figure 3.14, square[i][j] has an empty neighbor square[i+1][j]; the PL in square[i][j] cannot communicate with another PL in its 2-DN square[i+2][j] next to the empty neighbor. So square[i][j] looks for a SL. Distance between node *a* in square[i][j] and the PL in square[i+2][j] is less than *R*, node *a* is selected as the SL in square[i][j] for square[i+2][j].

3.2.1.5 Selection of Tertiary Leaders (TLs) for 2-distance neighbors (2-DNs)

When a square has an empty neighbor and no node in the square can communicate with a PL in a 2-DN, then the square asks whether any node in the square can communicate with any one in the 2DN. If yes, the pair of nodes is selected as a *tertiary leader* (TL). *The tertiary leader* is an intermediate node, by which the PL in the square can communicate with another PL in its 2-DN next to the empty neighbor.

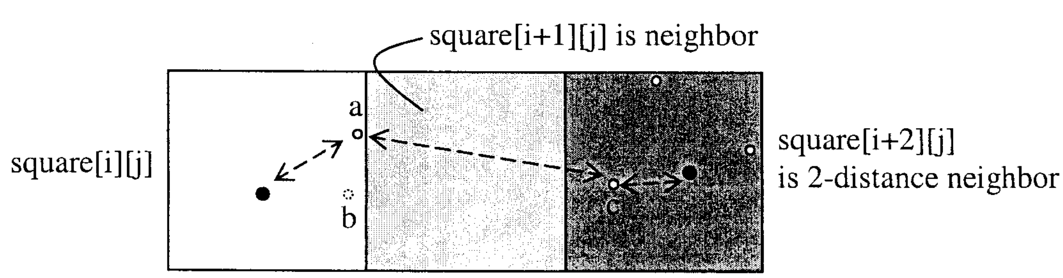


Figure 3.15 A Tertiary leader for a 2-distance neighbor

As shown in Figure 3.15, square[i][j] has an empty neighbor square[i+1][j]. the PL in square[i][j] can not communicate with the PL in 2-DN square[i+2][j], and the square[i][j] cannot find a SL for its 2-DN square[i+2][j]. That means no node in square[i][j] cannot communicate with the PL in its 2-DN square[i+2][j] next to the empty neighbor. At this time, a TL is necessary. The square[i][j] tries to find a node a in square[i][j] and another node c in its 2-DN square[i+2][j], and the distance between nodes a and c is less than R. The pair of nodes $\{a, c\}$ are a TL in square[i][j] for 2-DN square[i+2][j] .

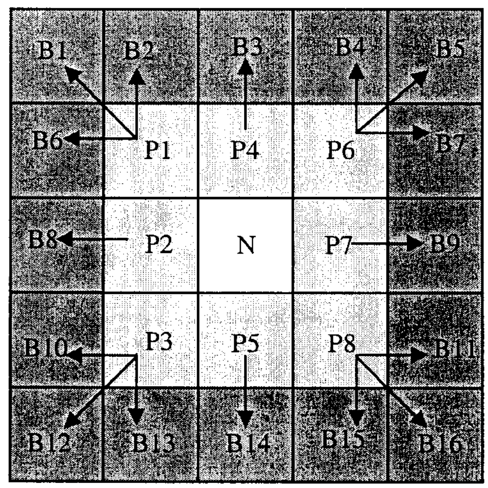


Figure 3.16 2-distance neighbors which are next to an empty neighbor

A square has eight neighbors and sixteen 2-DNs. We illustrate when one of its neighbors is empty, for which 2-DNs next to the empty neighbor a square considers to ask SLs or TLs. As shown in Figure 3.16, node N has eight neighbors. When P1 is empty, node N ask a SL or TL for 2-DN B1, B2, and B6 respectively; in the same way, empty P3 results in a possible SL or TL for B10, B12, B13 respectively; empty P6 results in a possible SL or TL for B4, B5, B7 respectively; empty P8 results in a possible SL or TL for B11, B15, B16 respectively. When P2 is empty, node N ask a possible SL or TL for 2-DN B8; in the same way, empty P4 results in a possible SL or TL for B3; empty P5 results in a possible SL or TL for B14; empty P7 results in a possible SL or TL for B9. Any node except the

PL in a square can be a SL or TL for a 2-DN as long as the distance between the node and the PL in the 2-DN is less than *R*.

Section 3.2.1 explains how a square to select a PL, possible SLs and TLs. Here we give all kinds of leaders shown in Table 3.1 for the MANET shown in Figure 3.4. The illustration is shown in Figure 3.17.

Table 3.1: Leaders for a MANET

PL:	47 16 34 57 + 72 + + 37 73 55 95 23 87 58 7 4 59 94 82 79 + 83 + 9 12 + 68 20 54 99 92 6 27 29 93 96 1 2 50 + 40 81 70 97 + 11 + 88 39 98 + 71 30 76 14 32 56 + 45 67 + 84 +
SLs & TLs: (* is for 2-DNs)	*tertiary leader from square[2][2] to square[2][4]: 51,5 *tertiary leader from square[2][4] to square[2][2]: 5,51 *tertiary leader from square[2][5] to square[4][6]: 41,26 *tertiary leader from square[2][5] to square[4][7]: 41,52 *secondary leader from square[3][0] to square[5][0]: 31 *tertiary leader from square[3][2] to square[2][4]: 65,5 *secondary leader from square[3][5] to square[3][7]: 24 secondary leader from square[4][5] to square[5][4]: 33 *secondary leader from square[4][5] to square[6][5]: 33

	*secondary leader from square[4][6] to square[2][6]: 26 *tertiary leader from square[4][7] to square[2][5]: 52,41 *secondary leader from square[4][7] to square[3][5]: 52 *secondary leader from square[4][7] to square[2][6]: 52 *tertiary leader from square[5][1] to square[3][0]: 21,31 *secondary leader from square[5][1] to square[5][3]: 48 tertiary leader from square[5][4] to square[4][5]: 86,33 *secondary leader from square[5][6] to square[5][4]: 53 *secondary leader from square[6][3] to square[5][1]: 15 *secondary leader from square[6][3] to square[4][2]: 15 *tertiary leader from square[6][4] to square[4][5]: 62,33 *tertiary leader from square[6][4] to square[4][6]: 62,63 *tertiary leader from square[6][4] to square[5][6]: 62,53 *secondary leader from square[6][4] to square[7][6]: 62 *secondary leader from square[6][5] to square[4][5]: 17 *secondary leader from square[6][6] to square[4][5]: 8

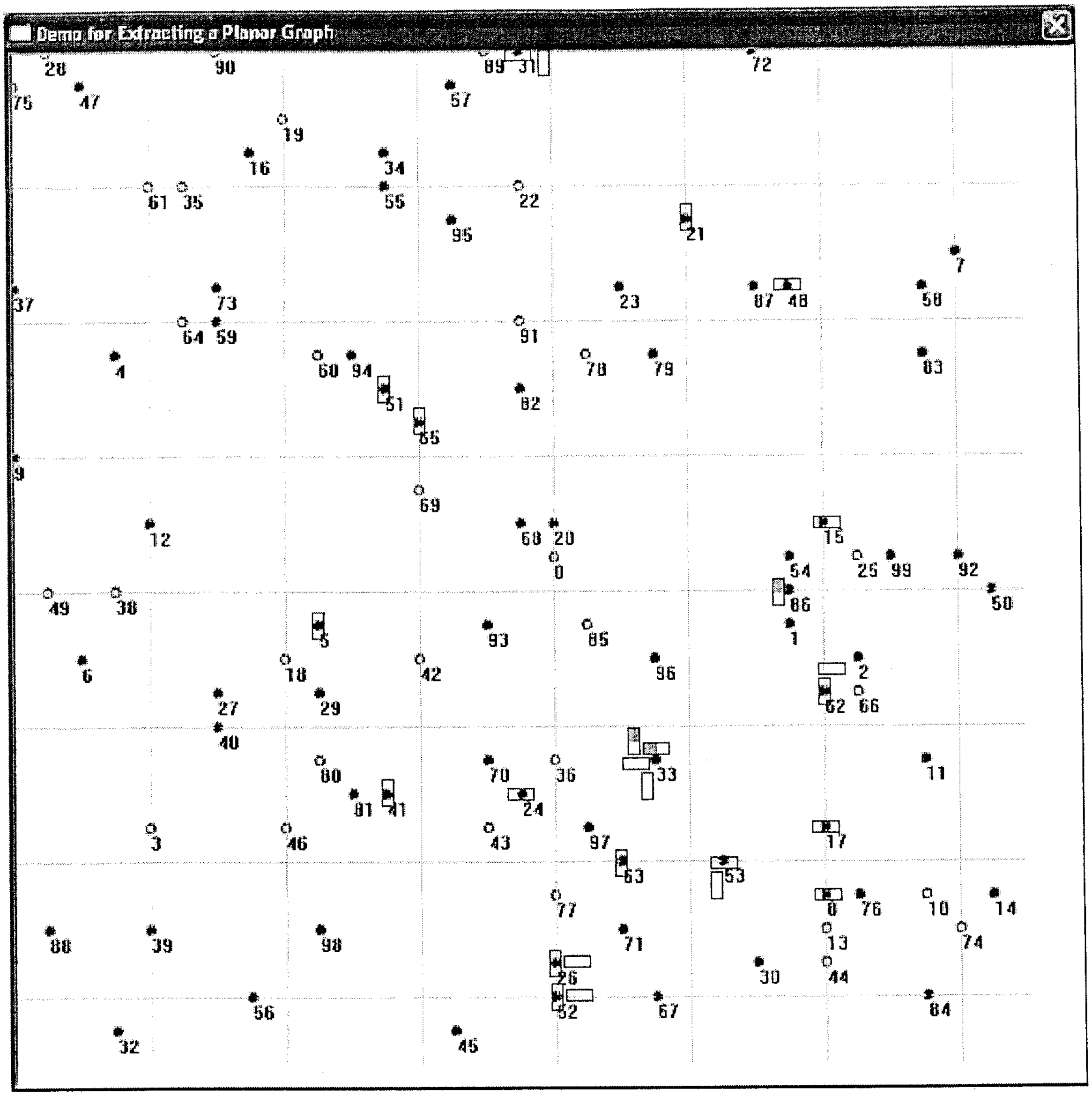


Figure 3.17 Leaders for a MANET

Blue points without the following marks are primary leaders

A secondary leader for a neighbor

A tertiary leader for a neighbor

A secondary leader for a 2-distance neighbor

A tertiary leader for a 2-distance neighbor

3.2.2 Back-bone Network and Routing in G

The nodes set V' of the back-bone network consists of the primary, secondary and tertiary leaders as selected from V in a distributed manner in all squares in the previous section. The back-bone network is then obtained by

1. considering nodes in V' as a geometrical network with transmission radius R, and
2. applying the Gabriel test to the edges to extract a planar subgraph $G'=(V',E')$.

The routing from node s in V to a node d in V is done as follows:

- if s and v are within the transmission range R then they communicate directly
- if s and v are not within the transmission range R then
    - i. if v is in V' then perform the face routing to d.
    - ii. if v is not in V' then send a message to the primary leader t in the square containing v. Perform the face routing in V' from t to d.

Clearly, any node of V is either in V' or is at distance at most R from a primary leader. Since G' is planar, the routing succeeds as long as the back-bone subnetwork G' is connected. The connectivity of G' is shown in the next theorem.

Theorem: If G is a unit disk graph network, then the subgraph G' consisting of V' (the set of leaders) is a **connected** subgraph of G such that any node in G is at distance at most R from a node in G'.

Proof. Clearly, any nonempty square contains a leader. Since $R \geq \sqrt{2}\,S$, any node in a square is within a distance at most R from all nodes in the square. Thus, any node of G is at most one hop from a node in *V'*. To show the connectivity of *G'*, it is sufficient to show that for any edge *(u,v)* of *G* the geometric graph induced by *V'* contains a path between *l(u)* and *l(v),* where *l(x)* denotes the primary leader of the square containing *x.* We prove it by considering several cases.

Case 1: If *u* and *v* are contained in the same square, then *l(u) =l(v),* as shown in Figure 3.18.. Since $R \geq \sqrt{2}\,S$, u and v can communicate directly with each other.

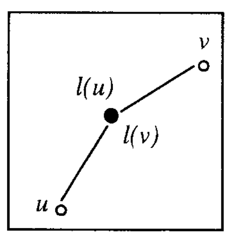
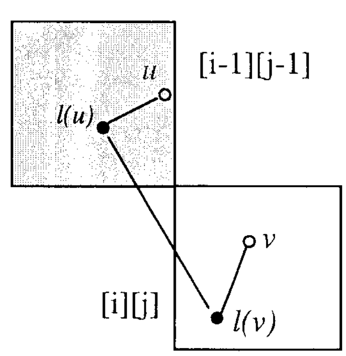
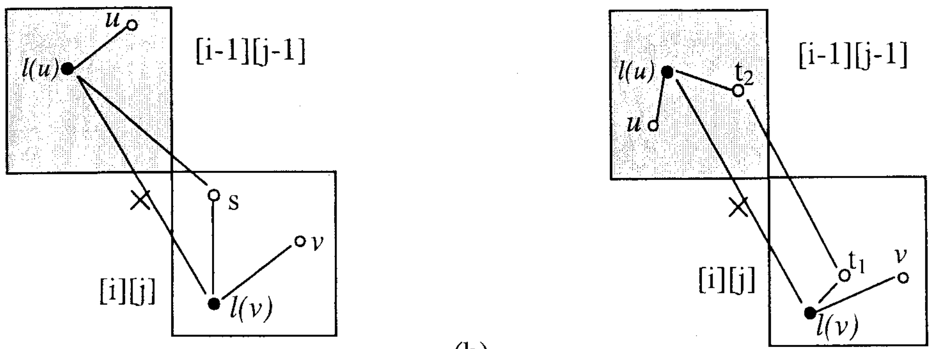


Figure 3.18 Case 1

Case 2: *u* and *v* are contained in neighboring squares. If *dist(l(u),l(v))≤R* then there is an edge between *l(u)* and *l(v),* as shown in Figure 3.19 (a). If *dist(l(u),l(v))>R* then, since there is an edge between the two squares, secondary leaders or tertiary leaders are selected in the squares which implies that there is a path in between *l(u)* and *l(v)* in the geometric subnetwork created by nodes in *V'*, as shown in Figure 3.19(b).



(a)



(b)

Figure 3.19 Case 2

Case 3: *u* and *v* are not contained in neighboring squares. Clearly, *u* and *v* are contained in 2-distance neighboring squares. If the squares between them are seen as empty by at least one of the squares then secondary leaders or tertiary leaders are selected in the two squares which create a path between *l(u)* and *l(v)*, as shown in Figure 3.20. If some square between then is seen as nonempty, then there is a path between *l(u)* and *l(v)* through the leaders of this nonempty square. □

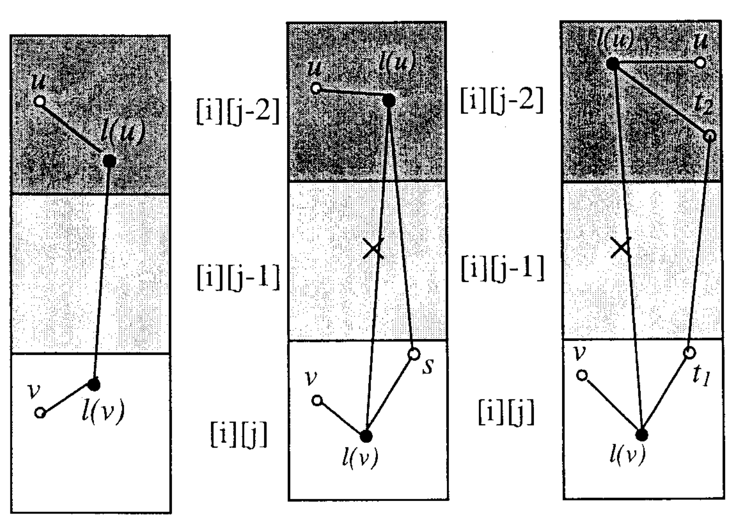


Figure 3.20 Case 3

To test whether our new approach of extracting a geometric planar subgraph which needs fewer hops in routing, we performed simulations that are discussed in the next section.

4. EMPIRICAL RESULTS

In this section we give results of our experiments with the back-bone subnetwork. The purpose of these experiments is twofold:

1. Measure how the value of *S*, the size of the square used for back-bone subnetwork extraction, influences the hop-count in routing using the back-bone planar subnetwork.
2. Compare the hop-count in the planar subnetwork *G* extracted by the Gabriel Test directly from the original network with the hop-count for the routing done using the back-bone planar subnetwork *G'*.

Since the number of nodes selected for the back-bone subnetwork is influenced by the density of the nodes in each square, we measure the hop-count for networks with different densities.

We have debated how to measure the hop-count in *G* and *G'* to get a fair comparison, since the two networks are quite different. If we use a face routing algorithm, the question is since there are quite many different variations and implementations of a face routing, different implementations in substantially different networks *G* and *G'* may skew the results. We finally opted for using the *Dijkstra's shortest path algorithm* [33] to calculate the number of hops, or hop count, between every two nodes of the two networks. Since path of any route found by a face routing algorithm are shortest path, this should give a general indication of the path-length improvement.

4.1 Framework of Experiments.

We implemented a computer program to generate networks at random. Platform used for the implementation is windowXP, the coding was done in C++ and Graphic User Interface (GUI) is developed by MFC. The program allows to select different values of the number of nodes, transmission radius *R* and square size *S*. Each of these parameters has an impact on the back-bone network.

For our experiments we generate nodes of the network in an area such as 30x30 or 40x40. For each node, the geometric coordinates are generated with uniform distribution. The number of nodes used is between 50 and 450 in increments of 50. For each set of nodes and a given values of *R* and *S* we construct the networks *G* and *G'* . The following measurements are performed for *G* and *G'*:

- Measure the number of hops between any two nodes in the networks.
- Calculate the spectrum of distances, which we define as the set ratios of the number of pairs of nodes which have same distance and the total number of pairs of nodes.
- Measure the *average hop distance*. Here we just consider distances between any pair of nodes that are connected.

4.2 Example of data obtained by simulation.

Consider the nodes shown in Figure 3.4. The following results are obtained

Network *G*:

Construct a network *G* shown in Figure 4.1, using the nodes shown in Figure 3.4.

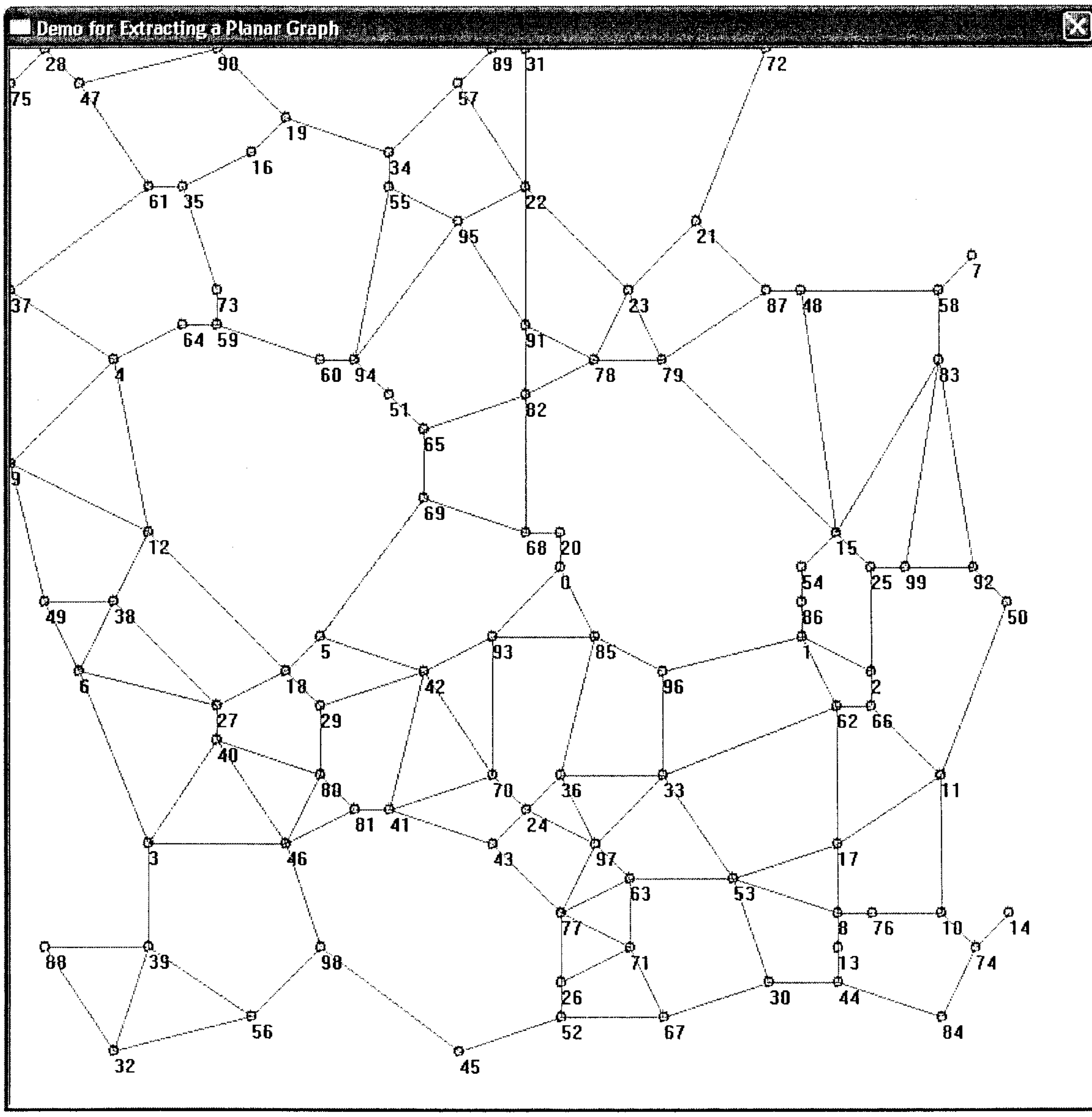


Figure 4.1 Example of network G

From 0 to 1, shortest path is 1 in cost. Path is: 0 1*

From 0 to 2, shortest path is 4 in cost. Path is: 0 85 96 1 2

From 0 to 3, shortest path is 6 in cost. Path is: 0 93 42 29 80 40 3

......

From 97 to 98, shortest path is 5 in cost. Path is: 97 77 26 52 45 98

From 97 to 99, shortest path is 6 in cost. Path is: 97 33 62 1 2 25 99

From 98 to 99, shortest path is 10 in cost. Path is: 98 45 52 67 30 53 17 11 50 92 99

The spectrum of hop distances (The largest distance is 16):

d=1	850 Paths	0.171717	d=10	368 Paths	0.0743434
d=2	20 Paths	0.0040404	d=11	276 Paths	0.0557576
d=3	140 Paths	0.0282828	d=12	169 Paths	0.0341414
d=4	371 Paths	0.0749495	d=13	82 Paths	0.0165657
d=5	511 Paths	0.103232	d=14	37 Paths	0.00747475
d=6	569 Paths	0.114949	d=15	9 Paths	0.00181818
d=7	563 Paths	0.113737	d=16	1 Paths	0.00020202
d=8	526 Paths	0.106263	d=infinite	0 Paths	0
d=9	458 Paths	0.0925253			

The total distance is 31513

Number of paths (do not include the infinite paths): 4950

Task1.3: The Average hop distance: 31513/ 4950 = 6.36626

- **Network *G'*:**

Construct a network G' shown in Figure 4.2, using the leaders shown in Figure 3.7. Note those leaders are selected from the nodes shown in Figure 3.4.

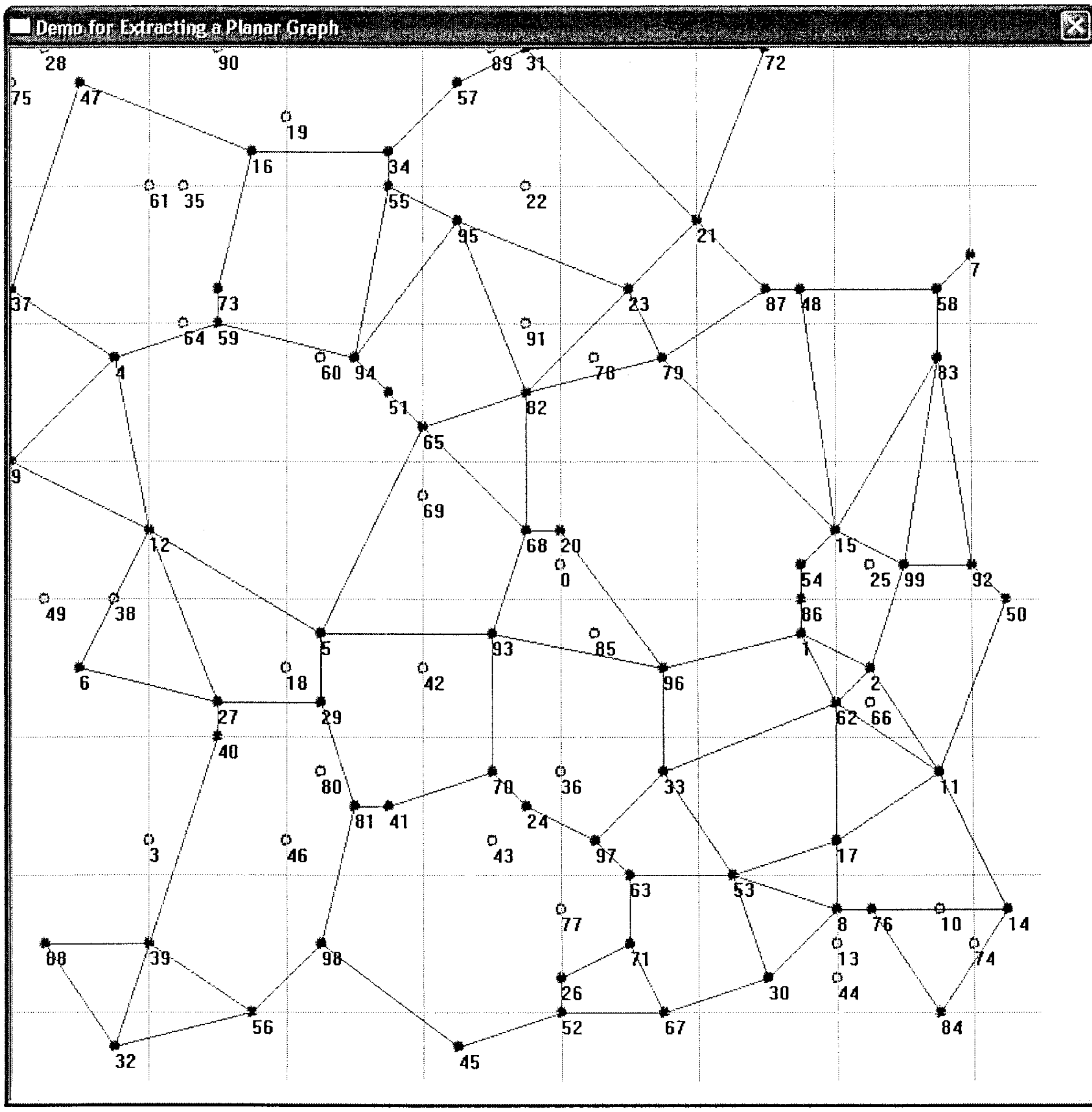


Figure 4.2 Example of network G'

From 0 to 1, shortest path is 1 in cost. Path is: 0 1*

From 0 to 2, shortest path is 4 in cost. Path is: 0 20 96 1 2

From 0 to 3, shortest path is 8 in cost. Path is: 0 20 68 65 5 12 27 40 3

....

From 97 to 98, shortest path is 5 in cost. Path is: 97 24 70 41 81 98

From 97 to 99, shortest path is 4 in cost. Path is: 97 33 62 2 99

From 98 to 99, shortest path is 8 in cost. Path is: 98 81 29 5 65 82 79 15 99

Spectrum of distance (The largest Distance is 13):

d=1	850 Paths	0.171717	d=8	606 Paths	0.122424
d=2	23 Paths	0.00464646	d=9	449 Paths	0.0907071
d=3	144 Paths	0.0290909	d=10	274 Paths	0.0553535
d=4	383 Paths	0.0773737	d=11	157 Paths	0.0317172
d=5	591 Paths	0.119394	d=12	82 Paths	0.0165657
d=6	702 Paths	0.141818	d=13	16 Paths	0.00323232
d=7	673 Paths	0.13596	d=infinite	0 Paths	0

The Total Distance: 29286

Number of paths(do not include the infinite paths): 4950

Task2.7: The Average hop distance(totalDist/numPath): 29286/ 4950 = 5.91636

4.3 Empirical Results

As mentioned above, the number of nodes, the transmission range *R*, and the square size *S* have each an impact on the average hop distance. We measure experimentally influences of these parameters on the average hop distance of both, *G* and *G'* networks. We generate a random set of nodes and obtain an average hop distance for the networks *G* and *G'*. In the following section, we will give the empirical results in detail.

4.3.1 Changed the number of nodes

To compare the original and the new approach, we obtain the average hop distance resulting from the original approach and from the new approach via *fifty iterations.* A new parameter *average hop distance ratio*, defined by the following formula, is used to analyze the empirical values.

$$\text{Ratio} = \frac{\text{Average hop distance resulting from the original approach}}{\text{Average hop distance resulting from the new approach}}$$

We design the test cases as follows:

Case 1: assume R = 6 and S = 4. Change the number of nodes and get the empirical values resulting from the original approach and from the new approach, as shown in Table 4.1. To compare two kinds of average hop distance, we illustrate the results, as shown in Figure 4.3 (a).

Case 2: Assume R = 8, and S = 4. Change the number of nodes; the empirical values are shown in Table 4.2; the illustration is shown in figure 4.3(b).

Table 4.1 The average hop distance when the number of nodes is changed

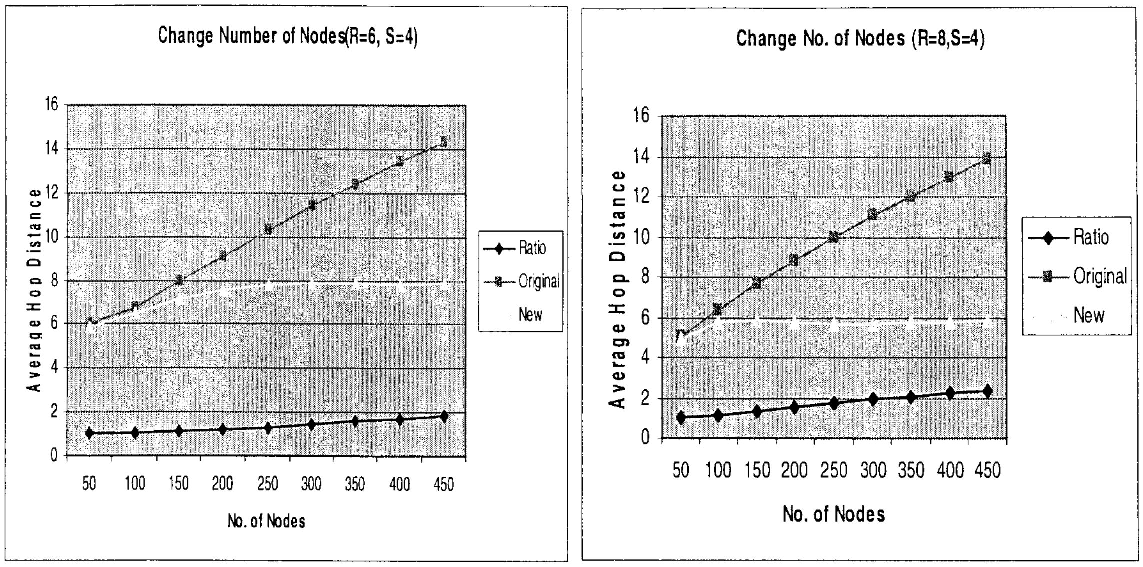
($R = 6$, $S = 4$, $coordinate_{max} = 30$, squares = 10x10)

# of Nodes	50		100		150	
Approach	**original**	**new**	**original**	**new**	**original**	**new**
Average Dist	5.99832	5.8778	6.79009	6.48469	7.95756	7.18715
Ratio	1.02050427		1.047095544		1.107192698	
# of Nodes	200		250		300	
Approach	**original**	**new**	**original**	**new**	**original**	**new**
Average Dist	9.12338	7.53045	10.3043	7.77856	11.4019	7.86547
Ratio	1.211531847		1.32470534		1.449614581	
# of Nodes	350		400		450	
Approach	**original**	**new**	**original**	**new**	**original**	**new**
Average Dist	12.3939	7.86974	13.4211	7.82824	14.299	7.84655
Ratio	1.574880492		1.71444667		1.822329559	

Table 4.2 The average hop distance when the number of nodes is changed

($R = 8$, $S = 4$, $coordinate_{max} = 30$, squares = 10x10)

# of Nodes	50		100		150	
Approach	**original**	**new**	**original**	**new**	**original**	**new**
Average Dist	5.05673	4.89595	6.35293	5.69607	7.74182	5.8106
Ratio	1.032839		1.115318		1.332362	
# of Nodes	200		250		300	
Approach	**original**	**new**	**original**	**new**	**original**	**new**
Average Dist	8.79033	5.74416	9.97869	5.65827	11.095	5.67274
Ratio	1.530307		1.763558		1.955845	
# of Nodes	350		400		450	
Approach	**original**	**new**	**original**	**new**	**original**	**new**
Average Dist	11.9786	5.69827	12.944	5.78071	13.8779	5.8025
Ratio	2.102147		2.239171		2.39171	



(a) R=6, S=4 (b) R=8, S=4

Figure 4.3 Average distance when the number of nodes is changed

Observe the curves in Figure 4.1. When the number of nodes rises, the average hop distances for *G* and *G'* ascend as well. A greater number of hops is needed to deliver messages. Obviously, when the number of nodes reaches 250, the average hop distance based on *G'* converges to a stable value. The back-bone subnetwork works well when nodes are distributed in high density. Note that the average hop distance ratio between *G* and *G'* is also up. Especially when the number of nodes is up to 450, the average hop distance ratio between *G* and *G'* is close to 2. That means that the new approach just needs half the number of hops to finish communication between every two nodes. The use of the back-bone subnetwork obviously reduces the hop count, and the new approach really results in a new subgraph, which needs a fewer number of nodes for a MANET when the number of nodes is changed.

4.3.2 Changed the transmission range

To compare the original and new approach, we will keep the square size S and the number of nodes to be fixed for each test case when R is changed. Repeat for several set of nodes and do a statistical evaluation. The test cases are designed as follows:

Case1: Assume that S is 4.0 and number of nodes are *250*. R is defined by $\sqrt{2}\,S \leq R \leq 2S$. When S is equal with 4.0, R is defined by $5.66 \leq R \leq 8$.The reason why we assume that the number of nodes is equal with or more than 250 for test cases is that the use of back-bone subnetworks reaches an obvious improvement when the number of nodes is up to 250, according to the empirical results in section 4.3.1. When R is changed, empirical values are shown in Table 4.3. The illustration is shown Figure 4.4 (a).

Case 2: Assume that S is 4.0 and the number of nodes is *300*. Change R, which is defined by $5.66 \leq R \leq 8$; the empirical values are shown in Table 4.4; the illustration is shown in Figure 4.4 (b).

Case 3: Assume that S is 4.0 and the number of nodes is *350*. Change R, which is defined by $5.66 \leq R \leq 8$; the empirical values are shown in Table 4.5; the illustration is shown in Figure 4.4 (c).

Case 4: Assume that S is 4.0 and the number of nodes is *400*. Change R, which is defined by $5.66 \leq R \leq 8$; the empirical values are shown in Table 4.6; the illustration is shown in Figure 4.4 (d).

Each test case must obey the test procedure:

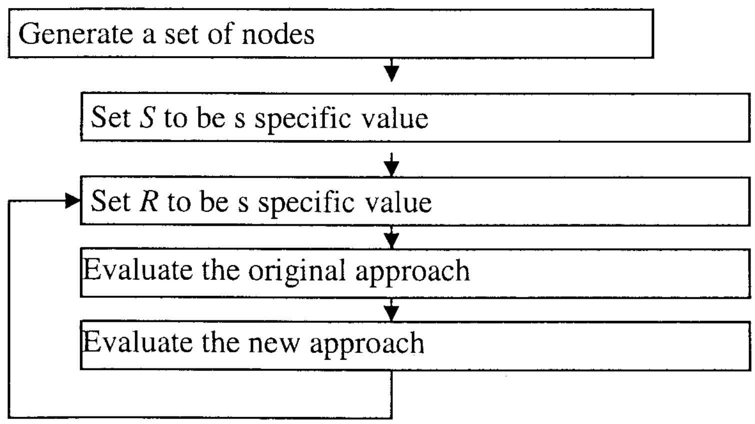


Table 4.3 The average hop distance when the transmission range R is changed

(number of nodes=**250**, S=4.0 , $5.66 \leq R \leq 8$, $coordinate_{max}$=30, squares=10x10)

Trans Range $5.66 \leq R \leq 8$	5.8		6.0		6.2	
Approach	original	new	original	new	original	new
Average Dist	10.0249	8.58651	9.98814	7.76061	9.96251	7.73684
Ratio	1.167517		1.28703		1.287672	

Trans Range	6.4		6.6		6.8	
Approach	original	new	original	new	original	new
Average Dist	9.93555	7.64671	9.90959	6.60013	9.88402	6.38994
Ratio	1.299323		1.501423		1.54681	

Trans Range	7.0		7.2		7.4	
Approach	original	new	original	new	original	new
Average Dist	9.86885	6.37947	9.82496	6.12594	9.76508	5.55476
Ratio	1.54697		1.603829		1.757966	

Trans Range	7.6		7.8		8.0	
Approach	original	new	original	new	original	new
Average Dist	9.76508	5.55476	9.73719	5.53748	9.69051	5.42538
Ratio	1.757966		1.758415		1.786144	

Table 4.4 The average hop distance when the transmission range R is changed

(number of nodes=**300**, S=4.0 , $5.66 \leq R \leq 8$, coordinate$_{max}$=30, squares=10x10)

Trans Range $5.66 \leq R \leq 8$	5.8		6.0		6.2	
Approach	original	new	original	new	original	new
Average Dist	11.3608	8.25333	11.322	7.91483	11.2956	7.99402
Ratio	1.376511		1.430479		1.413006	
Trans Range	**6.4**		**6.6**		**6.8**	
Approach	**original**	**new**	**original**	**new**	**original**	**new**
Average Dist	11.2676	8.29206	11.2401	6.19478	11.2102	6.17188
Ratio	1.358842		1.814447		1.816335	
Trans Range	**7.0**		**7.2**		**7.4**	
Approach	**original**	**new**	**original**	**new**	**original**	**new**
Average Dist	11.1945	6.1623	11.1499	5.94796	11.0884	5.59396
Ratio	1.816611		1.874575		1.982209	
Trans Range	**7.6**		**7.8**		**8.0**	
Approach	**original**	**new**	**original**	**new**	**original**	**new**
Average Dist	11.0884	5.59396	11.0571	5.57639	11.0116	5.55177
Ratio	1.982209		1.982842		1.98344	

Table 4.5 The average hop distance when the transmission range R is changed

(number of nodes=**350**, S=4.0 , $5.66 \leq R \leq 8$, coordinate$_{max}$=30, squares=10x10)

Trans Range $5.66 \leq R \leq 8$	5.8		6.0		6.2	
Approach	original	new	original	new	original	new
Average Dist	12.1676	8.27081	12.1229	7.95193	12.0931	7.92792
Ratio	1.47115		1.524523		1.525381	
Trans Range	**6.4**		**6.6**		**6.8**	
Approach	**original**	**new**	**original**	**new**	**original**	**new**
Average Dist	12.0636	8.01657	12.0337	6.61986	12.0022	6.3191

Ratio(p/n)	1.504833		1.817818		1.899353	
Trans Range	**7.0**		**7.2**		**7.4**	
Approach	**original**	**new**	**original**	**new**	**original**	**new**
Average Dist	11.9857	6.30918	11.9366	6.10248	11.87	5.66959
Ratio	1.899724		1.956024		2.093626	
Trans Range	**7.6**		**7.8**		**8.0**	
Approach	**original**	**new**	**original**	**new**	**original**	**new**
Average Dist	11.87	5.66959	11.8355	5.65036	11.7834	5.53076
Ratio	2.093626		2.094645		2.130521	

Table 4.6 The average hop distance when the transmission range *R* is changed

(Number of nodes=**400**, S=4.0 , $5.66 \leq R \leq 8$, coordinate$_{max}$=30, quares=10x10)

Trans Range $5.66 \leq R \leq 8$	**5.8**		**6.0**		**6.2**	
Approach	**original**	**new**	**original**	**new**	**original**	**new**
Average Dist	13.0789	8.80573	13.031	7.88867	12.998	7.86474
Ratio	1.485272		1.651863		1.652693	
Trans Range	**6.4**		**6.6**		**6.8**	
Approach	**original**	**new**	**original**	**new**	**original**	**new**
Average Dist	12.9632	7.85783	12.9297	6.51628	12.8953	6.43244
Ratio	1.649718		1.984215		2.004729	
Trans Range	**7.0**		**7.2**		**7.4**	
Approach	**original**	**new**	**original**	**new**	**original**	**new**
Average Dist	12.8759	6.42284	12.8222	6.39471	12.7502	6.12371
Ratio	2.004705		2.005126		2.082104	
Trans Range	**7.6**		**7.8**		**8.0**	
Approach	**original**	**new**	**original**	**new**	**original**	**new**
Average Dist	12.7502	6.12371	12.7129	6.10439	12.6549	5.86138
Ratio	2.082104		2.082583		2.159031	

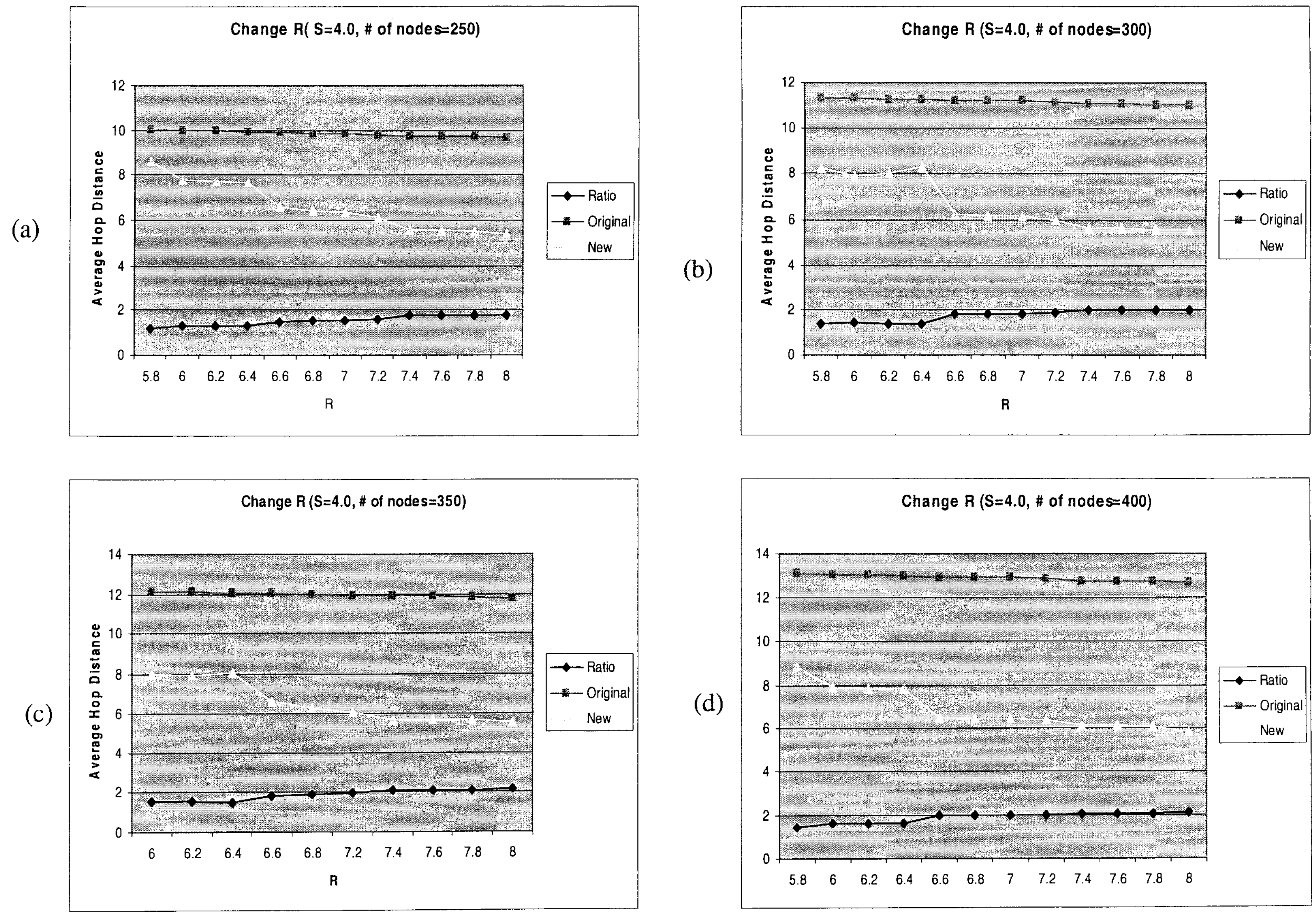


Figure 4.4 Average hop distance when R is changed

Observe the curves in Figure 4.4. The number of nodes and square size S are fixed. When the transmission range R rises, the average hop distances for *G* are a little bit lower while the average hop distance for *G'* descends a lot. The average hop distance ratio between G and G' keeps up. When the R is equal with 8.0, the ratio is close to 2. That means that the use of the subnetwork obviously reduces the number of hops. The new approach is better for routing in MANETs.

4.3.3 Changed the square size

In this experiment, we will keep the number of nodes and the transmission range *R* to be fixed for each test case. Assume that the transmission ranges R is 16 and the maximum coordinate is 40. According to $R/2 \leq S \leq R/\sqrt{2}$, when *R* is16, S is defined by $8 \leq S \leq$ 11.31 Repeat for several sets of nodes and do a statistical evaluation. The test cases are designed as follows

Case 1: Assume that the number of nodes is 250. When R is changed, the empirical values are shown in Table 4.7 and the illustration is shown Figure 4.5 (a).

Case 2: Assume that the number of nodes is 300. When R is changed, the empirical values are shown in Table 4.8 and the illustration is shown Figure 4.5 (b).

Case 3: Assume that the number of nodes is 350. When R is changed, the empirical values are shown in Table 4.9 and the illustration is shown Figure 4.5 (c).

Case 4: Assume that the number of nodes is 400. When R is changed, the empirical values are shown in Table 4.10 and the illustration is shown Figure 4.5 (d).

Case 5: Assume that the number of nodes is 450. When R is changed, the empirical values are shown in Table 4.11 and the illustration is shown Figure 4.5 (e).

Table 4.7 Average hop distance When S is Changed.

(R=16.0, **Number of nodes=250** and $Coordinate_{max}$=40)

Square Size $8 \leq S \leq 11.31$	**8**		**8.4**		**8.8**	
Approach	**original**	**new**	**original**	**new**	**original**	**new**
Average Dist	8.08257	3.73626	8.08257	3.71454	8.08257	3.81629
Ratio	2.1632783		2.175928		2.117913	
Square Size	**9.2**		**9.6**		**10.0**	
Approach	**original**	**new**	**original**	**new**	**original**	**new**
Average Dist	8.08257	3.74406	8.08257	3.56745	8.08257	3.5974
Ratio	2.158771		2.265644		2.246781	
Square Size	**10.4**		**10.8**		**11.2**	
Approach	**original**	**new**	**original**	**new**	**original**	**new**
Average Dist	8.08257	3.80533	8.08257	3.98024	8.08257	3.97462
Ratio	2.1240129		2.030674		2.033545	

Table 4.8 Average hop distance When S is Changed.

(R=16.0, **Number of nodes=300** and $Coordinate_{max}$=40)

Square Size $8 \le S \le 11.31$	8		8.4		8.8	
Approach	original	new	original	new	original	new
Average Dist	9.57093	3.89135	9.57093	3.92497	9.57093	3.77904
Ratio	2.4595397		2.438472		2.532635	
Square Size	9.2		9.6		10.0	
Approach	original	new	original	new	original	new
Average Dist	9.57093	3.65144	9.57093	3.69483	9.57093	4.08542
Ratio	2.621139		2.590357		2.342704	
Square Size	10.4		10.8		11.2	
Approach	original	new	original	new	original	new
Average Dist	9.57093	4.14372	9.57093	4.64491	9.57093	4.6031
Ratio	2.3097434		2.06052		2.079236	

Table 4.9 Average Distance When S is Changed.

(R=16.0, **Number of nodes=350** and $Coordination_{max}$=40)

Square Size $8 \le S \le 11.31$	8		8.4		8.8	
Approach	original	new	original	new	original	new
Average Dist	10.2716	3.83049	10.2716	3.73703	10.2716	3.51679
Ratio	2.6815368		2.7486		2.920732	
Square Size	9.2		9.6		10.0	
Approach	original	new	original	new	original	new
Average Dist	10.2716	3.59979	10.2716	3.47018	10.2716	3.56031
Ratio	2.853389		2.959962		2.88503	
Square Size	10.4		10.8		11.2	
Approach	original	new	original	new	original	new
Average Dist	10.2716	4.12262	10.2716	3.91009	10.2716	4.0122
Ratio	2.4915224		2.626947		2.560092	

Table 4.10 Average hop distance When S is Changed.

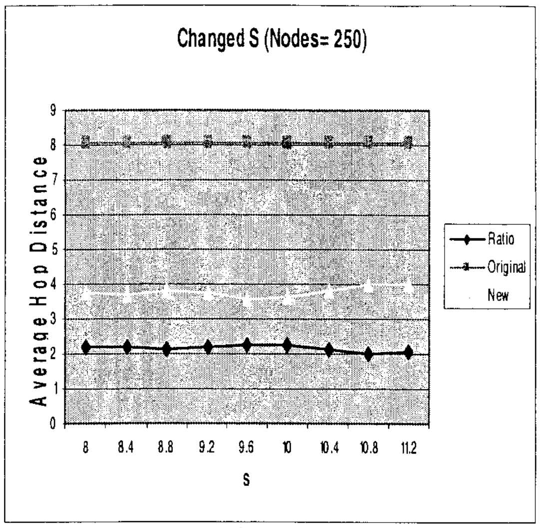
(R=16.0, **Number of nodes=400** and $Coordinate_{max}$=40)

Trans Range $8 \leq S \leq 11.31$	8		8.4		8.8	
Approach	original	new	original	new	original	new
Average Dist	11.1174	3.95972	11.1174	3.91056	11.1174	3.94729
Ratio	2.8076228		2.842918		2.816464	
Trans Range	9.2		9.6		10.0	
Approach	original	new	original	new	original	new
Average Dist	11.1174	3.76024	11.1174	3.84273	11.1174	3.83213
Ratio	2.956567		2.893099		2.901102	
Trans Range	10.4		10.8		11.2	
Approach	original	new	original	new	original	new
Average Dist	11.1174	4.07951	11.1174	4.06229	11.1174	4.54612
Ratio	2.7251802		2.736732		2.44547	

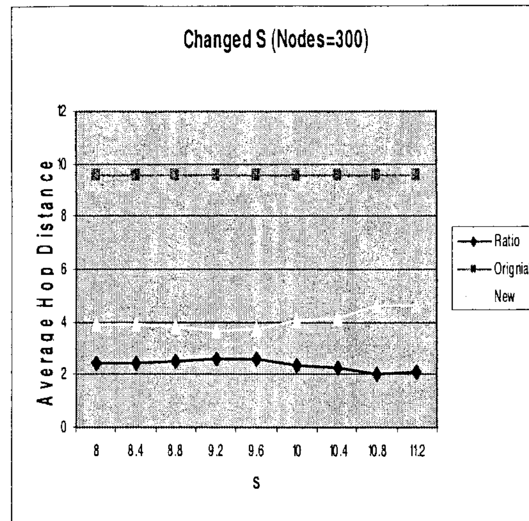
Table 4.11 Average hop distance When S is Changed.

(R=16.0, **Number of nodes=450** and $Coordinate_{max}$=40)

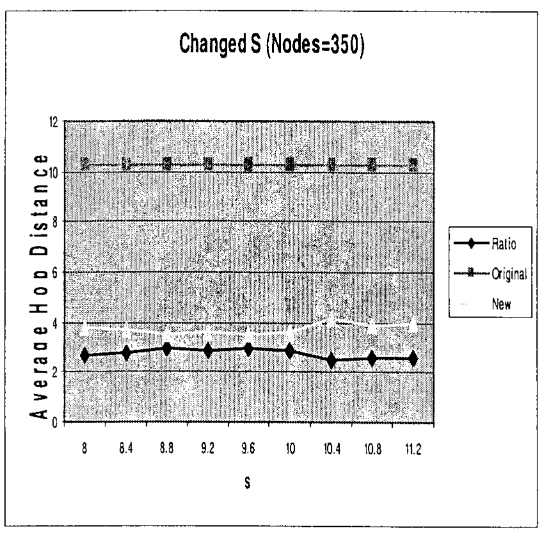
Trans Range $8 \leq S \leq 11.31$	8		8.4		8.8	
Approach	original	new	original	new	original	new
Average Dist	11.5913	3.96787	11.5913	3.78637	11.5913	3.57654
Ratio	2.9212903		3.061323		3.240926	
Trans Range	9.2		9.6		10.0	
Approach	original	new	original	new	original	new
Average Dist	11.5913	3.46689	11.5913	3.37123	11.5913	3.31938
Ratio	3.343429		3.4383		3.492008	
Trans Range	10.4		10.8		11.2	
Approach	original	new	original	new	original	new
Average Dist	11.5913	3.81694	11.5913	3.53858	11.5913	4.08658
Ratio	3.0368044		3.275693		2.83643	



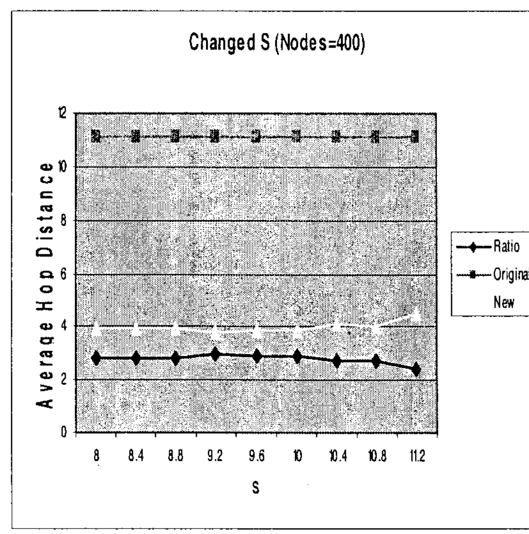
(a)



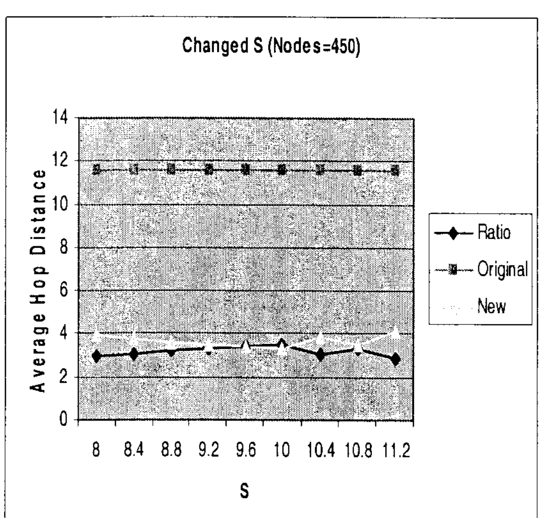
(b)



(c)



(d)



(e)

Figure 4.5 Average hop distance when the square size *S* is changed

Observe the curves in Figure 4.5. For each illustration, the number of nodes is fixed. When the transmission range *S* is up, the average hop distance for *G* has no change while the average hop distance for *G'* ascends. In contrast, the average hop distance ratio between *G* and *G'* descends. All of the ratios between *G* and *G'* are more than 2. This means the new approach just need less than half of hop count of the original approach. The use of subnetworks obviously reduces the number of hops. The new approach really results a new algorithm which need a fewer number of nodes for a MANET.

4.3.4 Summary of Experiments

According to the above empirical values and illustrations, we can draw a conclusion as follows:

1. The use of the subnetwork obviously reduces the average number of hops when the density of nodes in a MANET is high.
2. Obviously, the average hop distance is correlative with the ratio between the *R* and *S*. To better observe the relationship between the average hop distance and ratio R/S, we draw two more illustrations, as shown in Figure 4.6 and 4.7. From the Figure 4.6, when R/S rises, the average hop distance ratio ascends. But we should note that when R is changed, a network is changed. From Figure 4.7, we see that for high-density nodes *R/S* should between 1.6 ~ 1.8 to obtain a high average hop distance ratio. Note that a network has no change when S is changed. The reason for the results seems to be the overall number of leaders in the backbone network: when *S* is as large as possible, we have the least number of primary

leaders, but more of the secondary and tertiary leaders. When S is $R/2$, then we have more primary leaders, but few of other leaders. Thus the optimum of the average hop distance ratio is reached between the two extremes.

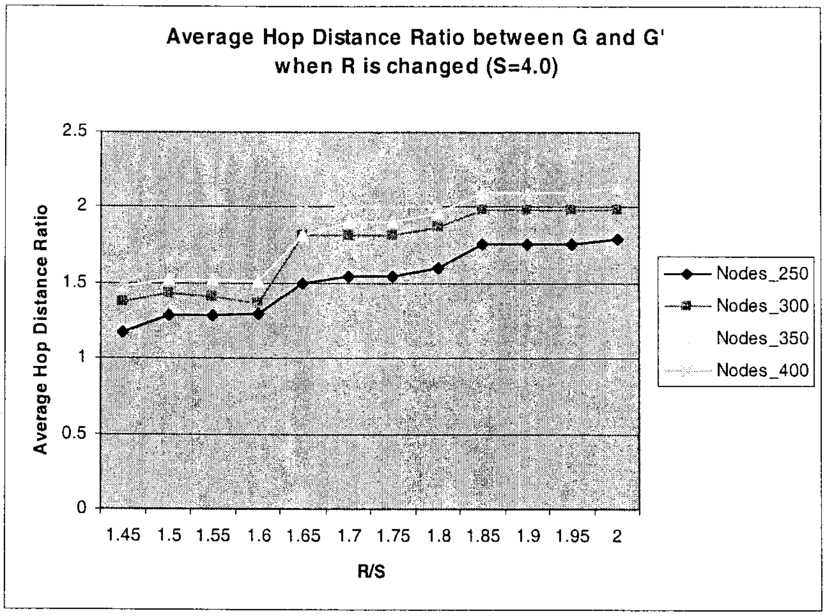


Figure 4.6 Average hop distance Ratio between G and G' when R is changed

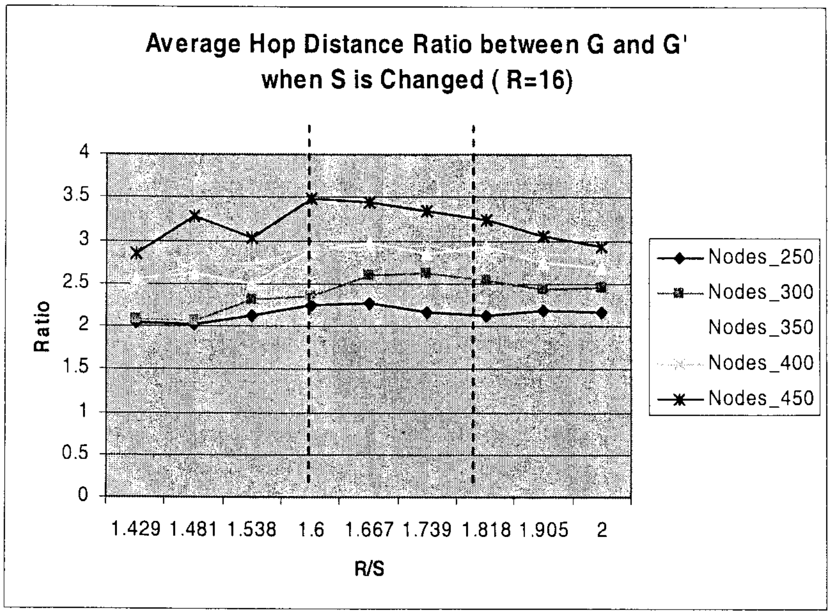


Figure 4.7 Average hop distance Ratio between G and G' when S is changed

5. CONCLUSIONS

We have proposed in this thesis an alternative way for using phase routing in a MANET: a back-bone planar subnetwork of the given MANET is extracted. The back-bone network is constructed using a partition of the plane into squares of size *S*. This back-bone network is used for routing between nodes. Based on the simulations as reported in the previous sections, this routing should give routes with a fewer number of hops on average and the use of a back-bone subnetwork is more advantageous if the density of hosts in a MANET is sufficiently high. Furthermore, the results suggest that one should use a limited square size, which is defined by $1.6 \leq R/S \leq 1.8$. Clearly, one could consider an extraction of a back-bone network by partitioning the plane into 'tiles' other than squares, e.g., hexagons, triangles, etc. What would be the best shape of a tile for partitioning should be investigated in future work.

One possible drawback of having a back-bone network is the possible higher transmission load of the hosts in the back-bone. This could lead to higher power consumption in these hosts and a possible slowdown in routing. Notice that the selection of leaders in the squares into which the plane is partitioned depends on the exact position of the squares. In this thesis, we assumed for simplicity that the the corners of the squares are in coordinates $[iS, jS]$ for some integers *i.j*. However, if we shift these positions to, say $[iS+a, jS+b]$ for some real numbers $0<a,b<S$, we obtain a different partitioning and most likely a different back-bone network. Our algorithm for choosing such a back-bone subnetwork applies as well, all we would need is to have the knowledge of *a* and *b*

available in the hosts. Thus, to alleviate a possible higher power consumption and a possible slowdown in routing in one fixed back-bone network, a MANET could select several back-bone subnetworks for routing based on a fixed set of pairs *[a.b]*. Thus we hope that the approach proposed in this thesis could be of practical use.

6. LIST OF REFERENCES

[1] Paul Boone, Edgar Chavez, Lev Gleitzky, Evangelous Kranakis, Jaroslav Opatrny, Gelasio Salazar, Jorge Urrutia, *Morelia Test: Improving the Efficiency of the Gabriel Test and Face Routing in Ad Hoc Networks*, LNCS 3104, pp.23-24, 2004

[2] Prosenjit Bose, Pat Morin, Ivan Stojmenovic, Jorge Urrutia, *Routing with Guaranteed Delivery in Ad Hoc Wireless*, Wireless Networks, pp. 609–616, 2001

[3] S. Basagni, I. Chlamtac, V.R. Syrotiuk and B.A. Woodward, *A distance routing effect algorithm for mobility (DREAM),* in: ACM/IEEEInternational Conference on Mobile Computing and Networking (Mobicom'98), pp. 76–84, 1998

[4] Wan-Hwa Liao, Yu-Chee Tseng, Jang-Ping Sheu, *GRID: A Fully Location-Aware Routing Protocol for Mobile Ad Hoc* Networks, Telecommunication Systems, v.18, no.1-3, pp.37-60, 2001.

[5] Brad Karp, H. T. Kung, *GPSR: Greedy Perimeter Stateless Routing for Wireless Network*, Mobile Computing and Networking, pp. 243-254, 2000.

[6] Vincent D. Park and M. Scott Corson, *A highly adaptive distributed routing algorithm for mobile wireless networks*, Proceedings of the Conference on Computer Communications (IEEE Infocom) (Kobe, Japan), pp.1405–1413, Apr. 1997.

[7] Perkins, C. *Ad hoc on demand distance vector (AODV) routing*, Internet-Draft, draft-ietf-manet-aodv-04.txt, Oct.1999.

[8] Susanta Datta, Ivan Stojmenovic, Jie Wu, *Internal node and shortcut based routing with guaranteed delivery in wireless networks*, Cluster Computing, v.5, pp.169—178, 2002.

[9] Silvia Giordano, Ivan Stojmenovic, Ljubica Blazevic, *Position Based Routing Algorithm for Ad Hoc Networks: a Taxonomy*, jan.09, 2001.

Current date: Apr.17, 2005, <http://citeseer.ist.psu.edu/giordano01position.html>

[10] J.A. Bondy and U.S.R. Murty, *Graph Theory with Applications* (Elsevier North-Holland, 1976).

[11] G. Toussaint, *The relative neighbourhood graph of a finite planar set*, Pattern Recognition, 12(4):261--268, 1980

[12] K. Gabriel and R. Sokal, *A new statistical approach to geographic variation analysis*, Systematic Zoology, 18:259--278, 1969.

[13] E. Chavez, S. Dobrev, E. Kranakis, J. Opatrny, L. Stacho and J. Urrutia, *Route Discovery with Constant Memory in Oriented Planar Geometric Network,* Oriented Planar Geometric Networks, 2004.

[14] Peter H. Dana. "Global Positioning System Overview." Revised: 05/01/2000. Feb.14, 2005 < http://www.colorado.edu/geography/gcraft/notes/gps/gps_f.html>

[15] Alberto Leon-Garcia, *Communication Networks* (2nd edition)

[16] Martin Mauve, Jorg Widmer, Hannes Hartenstein, *A Survey on Position-based Routing in Mobile Ad-Hoc Networks*, IEEE Network Magazine, 15(6):30—39, November, 2001.

[17] Xiuzhen Cheng, Xiao Huang, Ding-Zhu Du, *Ad hoc wireless networking*, Foreword. Kluewer Academic Publishers, 2004.

[18] C.Siva Ram Murthy, B.S. Manoj, *Ad Hoc Wireless Networks Architectures and Protocols*, Bernard M. Goodwin Publisher, pages 299-364, 2004.

[19] Sensors Online, "Wireless Mesh Networks". Current date: April, 18,2005. <http://www.sensorsmag.com/articles/0203/38/main.shtml>

[20] Paul J.M. Havinga, "**sensor networks** for monitoring", the current date: April, 18,2005, < http://europa.eu.int/information_society/istevent/2004/cf/document.cfm?doc_id=1234>

[21] C. Perkins and P. Bhagmat, *Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers.* Computer Communication Review, Pages 234-244, October 1994.

[22] P. Jacquet, P. Muhlethaler, A. Qayyum, A. Laouiti, L.Viennot, and T. Clausen, *Optimized Link State Routing Protocol.* Internet Draft, draft-ietf-manet-olsr-04.txt. work in progress, September 2001.

[23] D. Johnson and D. Maltz. *Mobile Computing*, chapter 5- Dynamic Source Routing, pages 153-181. Kluwer Academic Publishers, 1996.

[24] V.Park and M. Corson. *A highly adaptive distributed routing algorithm for mobile wireless networks.* In Proc. of INFOCOM'97, 1997.

[25] C. Perkins and E. Royer, *Ad-hoc on-demand distance vector routing*. In Proc. of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, pages 90-100, February 1999.

[26] Z. Haas and M. Pearlman. *The performance of query control schemes for the zone routing protocol.* ACM/IEEE Transactions on Networking, 9(4): 427-438, August 2001.

[27] Jie Gao, Leonidas J. Guibas, John Hershberger, Li Zhang, An Zhu, *Geometric Spanner for Routing in Mobile Networks*, Proc. ACM MobiHoc01, pp. 45-55, 2001.

[28] Martin Haenggi, *Twelve Reasons not to Route over Many Short Hops, IEEE Vehicular Technology Conference (VTC'04 Fall)*, (Los Angeles, CA), Sept. 2004

[29] Young-Bae Ko, Nitin H. Vaidya, *Geocasting in Mobile Ad Hoc Networks: Location-Based Multicast Algorithms*, Second IEEE Workshop on Mobile Computer Systems and Applications, 1998.

[30] Steven Deering, Deborah Estrin, Van Jacobson, Puneet Sharma and, et al. *Protocol Independent Multicast-Sparse Mode (PIM-SM) : Motivation and Architecture*, Internet Draft, Internet Engineering Task Force, draft-ietf-idmr-pim-arch-05.txt, Aug 1998.

[31] C.Siva Ram Murthy, B.S. Manoj, *Ad Hoc Wireless Networks Architectures and Protocols*, pages 191-223. Bernard M. Goodwin Publisher, 2004.

[32] Xiuzhen Cheng, Xiao Huang, Ding-Zhu Du, *Ad hoc wireless networking*, pages 103 – 135. Kluewer Academic Publishers, 2004.

[33] Alberto Leon-Garcia, *Communication Networks* (2^{nd} edition), Higher Education, pages 490 – 561.

7. ABBREVIATION

MANETs	Mobile Ad-hoc Wireless Netwroks
PL	Primary Leader
SL	Secondary Leader
TL	Tertiary Leader
2-DN	2-Distance Neighbor