12-2008

Energy efficient geographic routing for wireless sensor networks.

Gang Zhao
University of Louisville

ENERGY EFFICIENT GEOGRAPHIC ROUTING FOR WIRELESS SENSOR NETWORKS

By

Gang Zhao
B.S. 2002, M.S. 2005, Shanghai JiaoTong University

A Dissertation
Submitted to the Faculty of the
Graduate School of the University of Louisville
in Partial Fulfillment of the Requirements
for the Degree of

Doctor of Philosophy

Department of Electrical and Computer Engineering
University of Louisville
Louisville, Kentucky

December 2008

Energy Efficient Geographic Routing for Wireless Sensor Networks

By

Gang Zhao
B.S. 2002, M.S. 2005, Shanghai JiaoTong University

A Dissertation Approved On

6/30/2008
Date

by the following Dissertation Committee:

Xiangqian Liu, Committee Chair

Anup Kumar

Min-Te (Peter) Sun

Cindy K. Harnett

Tamer Inanc

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my advisor, Dr. Xiangqian Liu, for his guidance, support, and encouragement during the course of my PhD. I am grateful for his patience in leading me into a brand new area of wireless communications. What I have benefited from him are not only his contributions to the problems in this dissertation, but also his diligence, insights, and dedication to the intellectual and personal growth of his graduate students. I can never forget the hours that we spent together on our research projects. Working with him has been a true privilege.

I would like to thank all my dissertation committee members for all the fruitful discussions we had on this work.

My special thanks go to Dr. Anup Kumar, Dr. Xiaoli Ma and Dr. Min-Te (Peter) Sun, from whom I have received persistent encouragement, insightful advice and assistance throughout my graduate studies.

I also would like to thank Tom Carroll, Lisa Bell, Susan Cunningham, Nazanin Casebier and Almedina Pepic for their kind assistance.

Thanks to my wife Xiao Wang and all my friends (Jingli Li, Jun Liu, Yanfeng Hou, Dongqing Chen, Bin Xie, Weizhong Zhang, etc.) in University of Louisville for their tremendous help and encouragement.

Above all, I would like to thank my parents for bringing me up.

ABSTRACT

ENERGY EFFICIENT GEOGRAPHIC ROUTING FOR WIRELESS SENSOR NETWORKS

Gang Zhao

September 30, 2008

A wireless sensor network consists of a large number of low-power nodes equipped with wireless radio. For two nodes not in mutual transmission range, message exchanges need to be relayed through a series of intermediate nodes, which is a process known as multi-hop routing. The design of efficient routing protocols for dynamic network topologies is a crucial for scalable sensor networks. Geographic routing is a recently developed technique that uses locally available position information of nodes to make packet forwarding decisions. This dissertation develops a framework for energy efficient geographic routing. This framework includes a path pruning strategy by exploiting the channel listening capability, an anchor-based routing protocol using anchors to act as relay nodes between source and destination, a geographic multicast algorithm clustering destinations that can share the same next hop, and a lifetime-aware routing algorithm to prolong the lifetime of wireless sensor networks by considering four important factors: PRR (Packet Reception Rate), forwarding history, progress and remaining energy. This dissertation discusses the system design, theoretic analysis, simulation and testbed implementation involved in the aforementioned framework. It is shown that the proposed design significantly improves the routing efficiency in sensor networks over existing geographic routing protocols. The routing methods developed in this dissertation are also applicable to other location-based wireless networks.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Wireless sensor networks (WSNs) have been used in many areas such as environment monitoring, health care, crisis management, intrusion detection and tracking. A wireless sensor network usually consists of a large number of nodes. Each node has limited processing capability. It may have different types of memory such as program, data and flash memories and is usually equipped with a RF (Radio Frequency) transceiver, a power source and various sensors and actuators. The nodes often self-organize after being deployed in an ad-hoc fashion.

Due to the small scale and limited communication capability of sensor nodes, a packet sent from one node to another one usually has to go through multiple hops, which makes routing a critical service required for wireless sensor networks. Different from wired networks, wireless sensor networks have no fixed infrastructure. Routing protocols used in the wired networks can not be used or perform inefficiently here. In sensor networks, routing includes *unicast, multicast, anycast* and *broadcast*. *Unicast* is to send a packet from a source to a destination. *Multicast* is the delivery of information to a group of destinations simultaneously. *Anycast* is a network routing scheme whereby data is routed to the "nearest" or "best" destination as viewed by the routing topology. *Broadcast* is to send a packet to all other nodes in the network. Broadcast is a special case of multicast.

Traditionally, the routing protocols for wireless sensor networks can be classified into three types: proactive, reactive, and hybrid routing protocols. In proactive protocols, each node actively maintains a routing table to route the packet. In reactive protocols [1, 2], instead of keeping a routing table, a node may flood the network to search for a route to the destination when it has a packet to send. In

TABLE 1.1

Skeleton of Geographic Routing Protocol

A node n holding packet m runs the following
1) **if** n finds some *better neighbors* **then** /* forwarding mode */
n forwards m to the best neighbor
2) **else** /* detouring mode */
n employs a different strategy to find a possible *better* node

hybrid protocols [3], a node maintains a routing table for nodes within a few hops away and queries the network if its routing table can not provide the information of the next hop. In recent years, a different type of routing protocols [4–9] has been proposed that utilizes the available location information at each node to route the packet. These protocols share two similar assumptions. First, each node in the network knows the geographic locations of itself and its neighbors. This could be realized at the time of network deployment if nodes are immobile or through a location service [10, 11] and the exchanges of beacons between neighbors. Second, the location of the destination is known at the time when the packet is generated. Such a scenario is reasonable if the destination is a particular sink or in case of the geocasting [12]. In these protocols, if a node holding the packet finds some *better neighbors* within its own proximity, the node *forwards* the packet to the best one. This is referred to as the forwarding mode herein. When a local minimum is reached (i.e., no better neighbor can be found), each of these protocols falls back to a different mode to recover the packet by finding a *detour* to leave the local minimum and then move toward the destination. This is referred to as the detouring mode. The general skeleton of geographic routing protocols is shown in Table 1.1.

One of the major advantages of the geographic routing protocols over traditional ones is that the node in the network does not need to maintain a large routing table. This saves the communication and storage overhead associated with

the routing table maintenance. Additionally, if a route is discovered by only using the greedy forwarding mode, the route is known to be sub-optimal [7]. This provides a performance bound for geographic routing protocols that incorporate greedy forwarding strategy. However, greedy forwarding alone has a low delivery rate even in connected networks. Without a routing table, different geographic routing protocols have different ways to find detours with various costs on energy and overhead. The performance of the detouring strategies of each protocols has not been fully examined. In [6], flooding is used to find the detour. While the detour found in this case is optimal, the flooding itself is too expensive. When a packet reaches a local minimum, most geographic routing protocols [4,5,8,9] first reduce the original network topology to a planar graph by dropping some edges and then explore the boundaries of a certain set of faces. However, planarization and face traversal commonly result in an excessive number of hops. This dissertation is concerned with studying energy efficient geographic routing protocols.

1.1 Literature Review

There are two key issues in geographic routing: How to define the better neighbors and what strategy should be used to find a detour? Different geographic routing protocols propose different means to address these two issues.

Assume that the node currently holding the packet is denoted as n and the destination as d. For a pair of nodes a and b, the distance between them is denoted as $dist(a, b)$. In [5–9], a better neighbor a for node n is defined as one such that $dist(a, d) < dist(n, d)$. In [13], a better neighbor a is defined as the one with smaller angle span from $\overrightarrow{nd}$ to $\overrightarrow{na}$. In [14], a better neighbor a is the one whose projection on $\overline{nd}$ yields the most advancement toward d. In [15], a better neighbor is the one whose cell in Voronoi diagram intersects $\overline{nd}$. In [16], an analytical model is given to show that to achieve more reliable packet delivery, the criteria of the better neighbor in geographic routing protocols should base on the product of the expected reception power and the forwarding distance. Note that, if only the forwarding

mode of the skeleton is involved in the routing process, it has been shown in [7] that any definition of the better neighbor in [5–9,14,15] leads to a sub-optimal path.

When there is no better neighbor, several non-flooding recovery strategies were proposed for the detouring mode [4,5,8,9]. These strategies employ a similar two-step process:

- They first reduce the network topology to a planar graph distributively. After the reduction, the topology contains no cross edges. The remaining edges divide the two-dimensional space into faces.
- Each strategy picks a certain set of faces in the resulting planar graph. The boundaries of these faces are then explored until the destination is reached or the packet is switched back to the greedy forwarding mode.

For a given network topology, several distributed algorithms [17–20] are available to planarize a network topology. In these algorithms, each node autonomously eliminates its connections (i.e., edges) to its neighbors from the consideration of the routing based on the locations of the neighbors so that the network topology contains no cross edges. In the Relative Neighborhood Graph (RNG) [19], a node u eliminates a link to a neighbor v if there exists at least one node in the intersection of radio coverages of u and v. In the Gabriel Graph (GG) [17], a node u eliminates a link to a neighbor v if there exists at least one node in the circle with diameter $\overline{uv}$. The Planar Spanner in [18] and the Morelia test in [20] employ more complicated algorithms to compute the planar graph so that a smaller number of edges are deleted from the original topology. Note that, the edge elimination process in graph planarization has two potentially negative impacts to the routing. First, the graph may become disconnected. Fortunately, the simulation results in [5, 18] have shown that the possibility of disconnecting a random graph during planarization is small no matter which planarization algorithm is used. Second, some edges that could be used as the shortcut to the destination may be eliminated in the planarization process. This impact, however, has not been fully studied.

After the network topology is planarized, the resulting graph is composed of a set of faces. Each non-flooding based geographic routing protocol picks a subset of faces and explores the boundaries of the faces to find a detour to the destination. The GPSR (Greedy Perimeter Stateless Routing) [5] algorithm and the GOAFR$^+$ (Greedy Other Adaptive Face Routing plus) family [7–9] are among the most popular geographic routing protocols. For instance, in the perimeter mode of GPSR, the packet is forwarded successively on closer faces of the planar graph (i.e., faces toward the perimeter of all faces) until it reaches the destination or a node closer to the destination than the previous local minimum, where the packet is switched back to the greedy forwarding mode. GOAFR$^+$ uses a dynamically adjustable bound in its face exploration and falls back to the greedy forwarding mode if one has visited (up to a constant factor) more nodes on the face boundary closer to the destination than nodes not closer to the destination.

1.2 Main Contributions

This dissertation develops a framework for energy efficient geographic routing. This framework includes a path pruning strategy, an anchor-based routing protocol, a geographic multicast algorithm and a lifetime-aware geographic routing algorithm. The proposed methods are studied systematically and their effectiveness is demonstrated by extensive computer simulations.

First, an efficient path pruning strategy is proposed to reduce the excessive number of hops caused by the detouring mode of geographic routing protocols. The path pruning algorithm finds routing shortcuts by exploiting the channel listening capability of wireless nodes, and is able to reduce a large number of hops with the help of a little state information passively maintained by a subset of nodes on the route. We also discuss algorithm properties and provide simulation results demonstrating the effectiveness of the proposed algorithm in shortening the routing path and improving delivery rate when it is applied to existing geographic routing protocols.

Second, an anchor-based geographic routing protocol is proposed, where anchors are set as relay nodes. A packet is routed from the source to the destination through a sequence of anchor nodes. The anchor list is obtained based on the projection distance of nodes in detouring mode with respect to the virtual line linking the source and destination. For existing anchor-based schemes, once an anchor list is obtained, the path from the source to the destination usually does not change unless the network topology changes, which may lead to the quick depletion of the energy for some nodes. To better distribute energy consumption among nodes in the network and thus prolong the network lifetime, a random shift is introduced to the location of anchors to obtain virtual anchors for each packet sent. Simulation results show that the projection distance based algorithm outperforms existing anchor-based algorithms with shorter paths and fewer anchors in random network topology. It is also demonstrated that the lifetime-improving strategy with virtual anchors is effective in increasing the number of packets delivered in the lifetime of sensor networks.

Third, a location-based multicast algorithm is proposed, namely the Destination Clustering Geographic Multicast (DCGM) for wireless sensor networks. Geographic routing is efficient in providing scalable unicast routing in resource-constrained wireless sensor networks. However, its applications in multicast routing remain largely unexplored. The idea of DCGM is to cluster destinations that can share the same next hop, and then iteratively select the next hop as the neighbor with the maximum number of destinations. The complexity of DCGM is proven to be $\mathcal{O}(n\ell)$, where n is the number of neighbors of the current node and ℓ is the number of destinations associated with the current node. Simulation results show that DCGM achieves better performance than existing geographic multicast routing algorithms in terms of average number transmissions, with much lower computation complexity. To further reduce the number of transmissions, clustering strategy is applied to GMR and DCGM. It improves the performance of GMR and DCGM by dividing the destinations into many clusters and sending the packet

first to the closest destination in each cluster, which then sends the packet to other nodes in the cluster. Simulation results show that the strategy can reduce the number of transmissions up to 35% percent.

Finally, a realistic link layer model is applied to wireless sensor networks and a new geographic routing algorithm is proposed to prolong the lifetime of wireless sensor networks. Maximizing the lifetime of wireless sensor networks under constrained resources is an interesting problem that has gained increasing attention. However, how to prolong the lifetime of wireless sensor networks with geographic routing remains largely unexplored. An ideal link layer model is assumed in many methods in improving the lifetime of wireless sensor networks. This dissertation introduces a realistic link layer model to the framework and defines a function consisting of four important factors: PRR (Packet Reception Rate), forwarding history, progress and remaining energy. With this function, various characteristics of each neighbor can be evaluated and a packet is forwarded to the optimal neighbor. Simulation results show that the algorithm can usually double the lifetime of wireless sensor networks compared with existing approaches.

1.3 Assumptions

Represent a sensor network as an undirected graph $G(V, E)$, where V is the set of vertices and E is the set of edges. Assume that the network is two dimensional (every node in V is embedded in the plane) and all nodes are represented by vertices of the graph. For simplicity, assume that all nodes have the same maximum communication range. An edge exists between two nodes $X, Y \in V$ if and only if they are within each other's communication range.

Assume in this work that each node in the network knows the geographic locations of itself and its neighbors. This could be realized at the time of network deployment if nodes are immobile or through a location service [10, 11] and the exchanges of beacons between neighbors.

Moreover, assume all links are bidirectional. The widely used IEEE 802.11

wireless network MAC [21] sends link-level acknowledgements for all unicast packets, so that all links in an 802.11 network must be bidirectional.

Finally, the location of the destination is known at the time when the packet is generated. Thus, assume a location registration and lookup service that maps node addresses or node identities to locations [22]. Queries to this system use the same geographic routing system as data packets. The querier geographically addresses its request to a location server. The scope of this work is limited to geographic routing.

1.4 Outline

The rest of the dissertation is organized as follows:

Chapter 2 describes the path pruning algorithm that exploits the channel listening capability of wireless nodes. Chapter 3 presents a novel method to select anchors in anchor-based routing. Chapter 4 discusses a new multicast routing algorithm with low complexity. A lifetime-aware geographic routing algorithm is discussed in Chapter 5. Chapter 6 summarizes the dissertation results and draws the conclusion.

CHAPTER 2

A PATH PRUNING STRATEGY

In geographic routing, while a route discovered by only using the greedy forwarding mode is known to be sub-optimal [7], detouring mode usually results in an excessive number of hops. If each time a source only generates one packet for a particular destination, a lengthy route can be tolerated because no states need to be maintained. However, in many network applications, such as multimedia communications, ssh sessions, and file transfers, it is frequent for a source to generate multiple packets for the same destination. If the same lengthy route is used repeatedly for all the packets, the energy consumption can be quite high. Some alternatives have been developed to address the issue of lengthy paths of stateless routing [23–26], by either using some global state information about the network topology such as anchors [23] and topology-based routing regions [24], both of which are assumed to be known *a priori*, or by using waypoints [25] that are fed back to the source by the destination at the expense of added communication overhead. On average, those methods obtain paths that are shorter than that of the stateless geographic routing. However, none of them can guarantee improvement for every route, in other words, in some situations, it is possible that longer routes are obtained by these methods than using stateless geographic routing.

By carefully examining face routing, two possible reasons are found to explain why the discovered route to be lengthy. First, these strategies planarize the network topology by having each node distributively drop some edges. The edges dropped may contain shortcuts between source and destination. Second, to avoid repeated loop, the face traversal used to find the detour after the planarization intentionally skip some edges. This again may overlook some shortcuts to the des-

tination. If the shortcuts are utilized, the route may be shortened.

The proposed path pruning algorithm finds routing shortcuts by exploiting the channel listening capability of wireless nodes, and reduces the number of hops to obtain a path beyond the planar graph. With the help of a little passive state information maintained at a subset of nodes along the path, the path pruning algorithm can effectively identify shortcuts beyond the planar graph. Notice that a similar channel listening mechanism has been used [27], but it is for dealing with node mobility in proactive routing protocols. Although the term *shortcut* has been adopted in [28], the proposed idea is completely different in the following two aspects: (1) In [28], each node needs to *actively* maintain the location information of nodes within two hops in order to identify the shortcuts; (2) The shortcut computation in [28] is still based on the planar graph. As a result, when a longer loop occurs in the detouring mode, unlike the proposed approach, the algorithm proposed in [28] is unable to identify any shortcut that reduces the path length beyond two hops.

Simulation results show that in average the proposed path pruning algorithm can reduce as much as 80% of hops on the routes obtained by existing geographic protocols such as GPSR [5] and GOAFR$^+$ [9] in a critical network density region.

The rest of this chapter is organized as follows. The path pruning algorithm is described and its properties are discussed in Section 2.1, and extensive simulation results are presented and analyzed in Section 2.2. Section 2.3 concludes the chapter.

2.1 The Path Pruning Algorithm

This section is started with a simple routing example to illustrate the proposed path pruning strategy. Then the pseudo code of the path pruning algorithm is provided. In addition, it is shown that the algorithm possesses a number of desired properties.

In wireless networks, each node actively listens to the channel for any packet possibly destined for it. It is natural to assume that a node is able to identify a packet that was previously forwarded by itself and the sender of the current hop of the packet from the header of the packet. For example, after node n_i forwards a packet to its neighbor n_j, if later n_i hears that the same packet is forwarded by another neighbor n_k, it can immediately tell that the link from n_i to n_k is a shortcut, which bypasses at least node n_j. This simple strategy can be implemented as long as we allow each node to keep a little passive state information for a period of time to identify the packet it previously forwarded.

This path pruning strategy can be applied to the detouring mode of any geographic routing protocols. However, it is not necessary to apply this strategy to the greedy forwarding mode. The reason will be explained later. The strategy is applied to prune the path found by a geographic routing protocol when the first packet is routed. Therefore when the first packet is delivered, a pruned path is also obtained, and subsequent packets can be forwarded using the pruned path.

2.1.1 A Routing Example

In this example, consider a simple network with 16 nodes, and apply the proposed path pruning algorithm to improve GPSR path. The network topology is shown in Figure 2.1 (a). Node A and node K are the source and destination, respectively. Two nodes are neighbors to each other if a line is drawn between them. The arc centered at K has a radius equal to the distance between A and K and the arc around A denotes the transmission range of node A. The intersection (gray area) is the void region. Figure 2.1 (b) is the network topology after planarization based on RNG. Some crossing edges have been removed after planarization (dash lines).

According to GPSR, when node A has a packet destined for node K, the packet enters perimeter mode since A has no neighbor closer to K than itself. Using right-hand rule on the planar graph (Figure 2.1 (b)), this packet will traverse

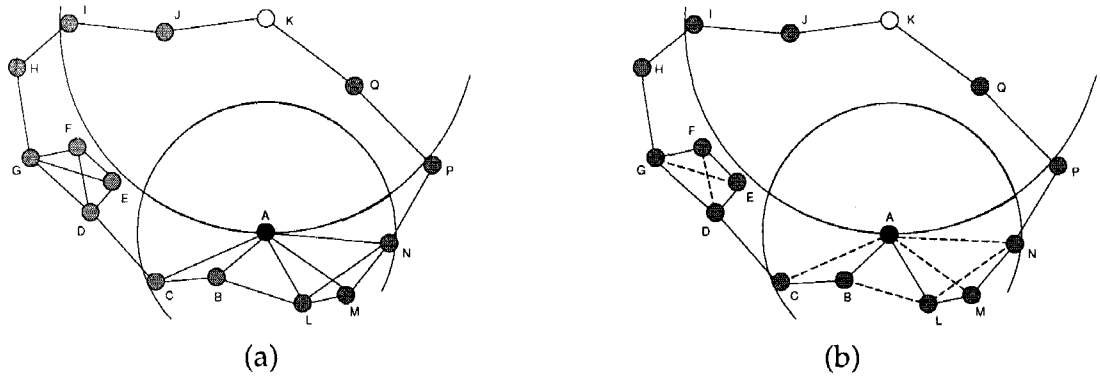


Figure 2.1. (a) Example network topology: A is the source node and K is the destination node; (b) Planarized network topology. Dash lines are the removed edges, and the remaining edges divide the space into faces.

through B, C, D, E, F, G, and H. When the packet reaches node H, its neighbor I is at a geographic location closer to K than A, where the packet last entered perimeter mode, so the packet is switched to greedy forwarding mode at node H. The packet then reaches the destination K via I and J. The path discovered by the GPSR protocol is shown as solid lines in Figure 2.2 (a).

The path pruning algorithm can be applied to the GPSR protocol. When the first packet is forwarded using the GPSR protocol, a node in perimeter mode listens to the channel and identifies possible shortcuts after it forwards the packet. As a result, node A will hear the same packet transmitted from node C, so it sets $A.next$ to C. Node B will set $B.next$ to C, but this passive state information will soon be dropped as no subsequent packet will be forwarded to node B from node A. Similarly, node D will first set $D.next$ to E, then F, and finally G as it hears the sequential transmissions. Therefore, starting from node A, the subsequent packets will traverse through nodes C, D, G, H, I, and J before it reaches the destination node K. This shortened path found by the path pruning algorithm is shown in Figure 2.2 (b). From this example, it is observed that there are three types of edges in the pruned route. One is the edges in the original GPSR route; e.g., edges $\overline{GH}$, $\overline{IJ}$, and $\overline{JK}$. The second type is the edges which is in the planar graph, but is not on the GPSR route; e.g. edge $\overline{DG}$. The last type is the edges which have been

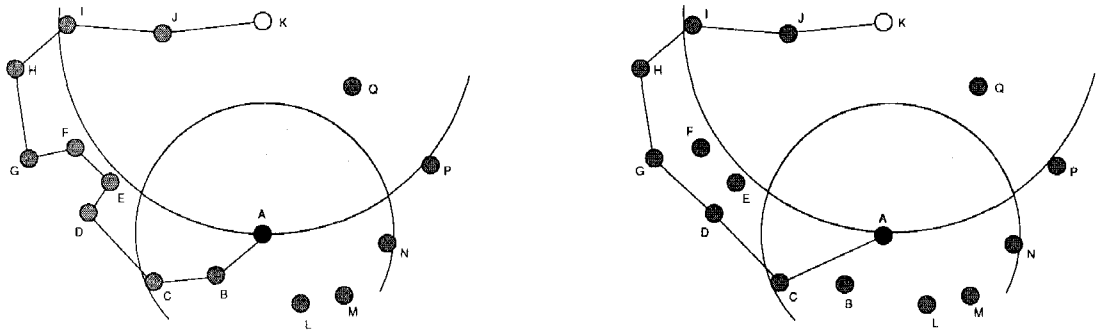


Figure 2.2. (a) The GPSR route; (b) The GPSR route with path pruning.

removed in planarization; e.g., edge $\overline{AC}$. Counting the number of hops, it is clear that the path pruning algorithm improves GPSR routing.

2.1.2 The Path Pruning Algorithm

The path pruning algorithm developed from this strategy is shown in Table 2.1. In this algorithm, nodes run the steps of a given geographic routing algorithm to find the route for the first packet. At the same time, the route is also pruned. Packet m is identified by $m.id =< n_s, n_d, seq >$, where n_s is the source node, n_d is the destination node, and seq is the packet sequence number. Let $m.hop$ denote the hop count of packet m with an initial value zero. Its value increases by one as the packet goes through one hop. For the first packet (i.e., $m.seq = 1$), if it reaches node n_i and it is in detouring mode, node n_i keeps a state $n_i.hop$ to track the number of hops that the packet has traversed before reaching n_i, as shown in Table 2.1. A node n_i participating the detouring mode also records the next hop which the packet is forwarded to for a given connection (i.e., $n_i.next$). The recorded $n_i.next$ and $n_i.hop$ are associated with a timer. Node n_i clears $n_i.next$ and $n_i.hop$ if it does not receive the next packet for this connection for a period of time. Before transmitting packet m, node n_i increases $m.hop$ in the header of m by one. After n_i delivers the packet, it listens to the channel. If n_i hears a neighbor n_j transmitting the same packet with $m.hop > n_i.hop + 1$, it recognizes a shortcut from n_i to n_j

TABLE 2.1

The Path Pruning Algorithm

During the delivery of packet m from source n_s to destination n_d, a node n_i runs the steps:

if m is the first packet, i.e., $m.seq = 1$, then
  1) **if** n_i holds packet m and $n_i \neq n_d$, **then**
    if n_i is in the detouring mode
      $n_i.hop = m.hop + 1$
      $m.hop = n_i.hop$
      n_i decides the next hop n_{i+1} using detouring rules and delivers m to n_{i+1}
      $n_i.next = n_{i+1}$
    else
      $m.hop = m.hop + 1$
      n_i decides the next hop using greedy forwarding and delivers m
  2) **if** n_i hears a neighbor $n_j \in nbr(n_i)$ transmitting the same m with $m.hop > n_i.hop + 1$ and the receiver is not n_i itself, **then** $n_i.next = n_j$
else
  if $n_i.next = n_j$, n_i delivers m to n_j
  else n_i forwards m using greedy forwarding rules

that saves $m.hop - n_i.hop - 1$ hops, and then n_i sets its next hop $n_i.next = n_j$. This state will be used as a pointer for the delivery of subsequent packets. If packet m is not the first one ($m.seq \neq 1$) and n_i is holding the packet, n_i uses the state $n_i.next$ to deliver the packet, or if there is no $n_i.next$, it delivers the packet using greedy forwarding rules.

2.1.3 Properties of the Path Pruning Algorithm

In the following, name the nodes that participate packet delivery in detouring mode as nodes in detouring mode and similarly for nodes in forwarding mode. Note that for simplicity we consider the node that transits from detouring mode to

forwarding mode as a node in detouring mode (the end of detouring mode), and the node that transits from forwarding mode to detouring mode also as a node in detouring mode (the first of detouring mode). The properties of the path pruning algorithm are summarized as follows. Without loss of generality, GPSR is chosen as an example of geographic routing protocols in our proofs.

Property 1 *Suppose that a better neighbor is defined as a neighbor closer to the destination. The path pruning algorithm does not change the path if only forwarding mode is involved in routing.*

Proof: Suppose that the node set of the GPSR route is $\mathcal{N}_{gpsr}$. The node set of the pruned route is $\mathcal{N}_{pr}$. For the algorithm to prune the path (remove at least one node), there must exist at least two nodes in $\mathcal{N}_{gpsr}$ that have a common neighbor in $\mathcal{N}_{gpsr}$. It will be proven by contradiction that this situation does not exist among the nodes if only forwarding mode is involved in routing. Assume n_1, n_2 and n_3 are three nodes on the route in forwarding mode. Suppose that both n_2 and n_3 are the neighbors of node n_1 and they appear in order of n_1, n_2 and n_3. Then, $dist(n_2, n_d) > dist(n_3, n_d)$ because of the greedy forwarding rule. However, since n_2 is n_1's neighbor and it is in $\mathcal{N}_{gpsr}$, that means $dist(n_2, n_d) < dist(n_3, n_d)$ because of the selection of n_2 as the next node of n_1 in greedy mode. This contradiction proves that the path pruning algorithm does not change the path in greedy forwarding mode route.

Property 1 shows that it is not necessary to apply the path pruning algorithm if the routing only involves forwarding mode. Note that Property 1 holds true for the greedy forwarding criterion in [5]. However, it may not hold for other forwarding criteria. For example, the path pruning algorithm can alter the forwarding path when the smallest angle criterion [13] is adopted for forwarding mode. Furthermore, if both forwarding and detouring modes are involved in the route discovery, it will be shown later by simulations that the path pruning algorithm still only needs to be applied to nodes in detouring mode.

Property 2 *The node set of the pruned route is a subset of the one from the original route.*

Proof: Suppose the packet is in forwarding mode. According to Property 1, if a node n_a is in forwarding mode and $n_a \in \mathcal{N}_{pr}$, then $n_a \in \mathcal{N}_{gpsr}$.

Suppose the packet is in detouring mode. According to Table 2.1, if a node $n_a \in \mathcal{N}_{gpsr}$, then its next state $n_a.next$ also belongs to $\mathcal{N}_{gpsr}$ because it is delivering the packet. Thus, if a node belongs to $\mathcal{N}_{pr}$, it also belongs to $\mathcal{N}_{gpsr}$; i.e.,

$$\mathcal{N}_{pr} \subseteq \mathcal{N}_{gpsr}. \tag{2.1}$$

If there exists one pruning step in the routing path, then there exists at least one node $n_c \in \mathcal{N}_{gpsr}$ but $n_c \notin \mathcal{N}_{pr}$. Thus, the number of hops for the pruned path is equal to or less than the number of hops in GPSR, i.e., $|\mathcal{N}_{pr}| \leq |\mathcal{N}_{gpsr}|$ where $|\cdot|$ denotes the cardinality of a set.

Property 3 *In a pruned path, if two nodes are in the same mode but not consecutive on the route, then they can not be neighbors.*

Proof: In detouring mode, suppose that n_i and n_j are not consecutive on the pruned route but they are neighbors, and n_j appears later on the route than n_i. Then during the routing of the first packet, $n_i.next$ will be set to n_j because $n_j.hop$ apparently is greater than $n_i.hop+1$. Therefore, n_j becomes consecutive with n_i on the pruned route.

The proof for the forwarding mode case is similar to that for Property 1.

Property 4 *The path pruning algorithm converges when the first packet reaches the destination.*

Proof: In the path pruning algorithm, the first packet is routed based on a geographic routing protocol and the nodes on the route are listening and building their states ($n.next$ and $n.hop$) as the first packet travels. Therefore, after the first packet reaches the destination, all nodes in detouring mode have already built their states and the whole route has been pruned. All subsequent packets can

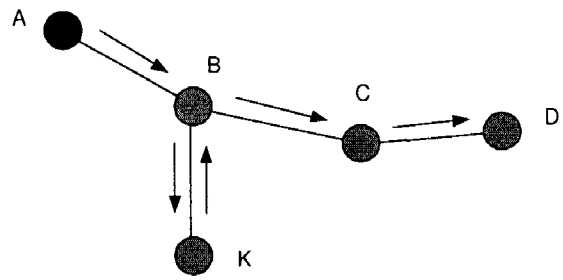


Figure 2.3. Network topology with a loop in GPSR routing.

follow the pruned route with the built-in states. Therefore, the pruning processing is converged after the first packet is delivered.

Property 5 *A pruned path is loop free.*

Proof: Although the detouring mode of geographic routing protocols does not repeat loops, it may still create loops in the routing path. An example of a GPSR route involving a loop is shown in Figure 2.3. The packet is delivered from node A to node D in detouring mode. Following the right-hand rule, the routing path found by GPSR is A-B-K-B-C-D. The path pruning algorithm will remove the loop by listening to the neighbors. The pruned path in this case is A-B-C-D, which is loop free.

In general, there may exist one or more loops starting at a node. Suppose that n_1 is a starting node for several loops, and thus n_1 is a node in detouring mode. Because we assume the network is connected, after detouring these loops, the packet will be delivered to a neighbor node n_j of n_1 to leave the loops. In this case, $n_1.next$ will be set to n_j to bypass those loops in the pruned path. If there is no loop in the route of the GPSR route $\mathcal{N}_{gpsr}$, then without changing the order of the nodes, the pruned route with node set $\mathcal{N}_{pr} \subseteq \mathcal{N}_{gpsr}$ also has no loop.

Corollary 1 *A pruned path based on GOAFR$^+$ is also loop free.*

In Property 5 , it is proven that if the original route is found by GPSR, then there is no loop for the pruned path. Here we briefly sketch the proof that if the original route is found by GOAFR$^+$, there is still no loop in the pruned path.

First, based on the design of GOAFR$^+$ for both forwarding and detouring modes, it can be shown that there is no loop that includes both modes or more than one face from the detouring mode. This is because in forwarding mode, the packet is delivered toward the destination. In detouring mode, the faces selected are also in the sequence toward the destination. If there is a loop which includes both modes or more than one face, then there must exist at least one face or one forwarding step that moves away from the destination. Second, the loops caused by face probing can also be removed by active listening, because these loops occur within one face. Therefore, although GOAFR$^+$ may change direction during the exploration of a possible route, our path pruning algorithm still guarantees the removal of all loops.

Property 6 *The path pruning algorithm has the following modifications with respect to the geographic routing protocol it applies to:*

M1) *Insertion of the number of hops in the header of the first packet;*

M2) *Maintenance of three states for nodes in the detouring mode.*

Proof: From Table 2.1, it is not difficult to observe that to implement the path pruning algorithm, the number of hops should be inserted in the first packet's header ($m.hop$) and three states ($n.hop$, $n.next$, and $m.id$) for the nodes in detouring mode are maintained, where $m.id$ is used to identify the packet when nodes listen to the channel.

2.1.4 Improving Delivery Rate

The path pruning algorithm is also effective in improving the delivery rate when the packets are subject to the constraint of time-to-live (TTL). TTL is widely adopted in ad hoc wireless routing as well as wired network routing [29]. A preset TTL value is initialized when a packet is generated (see, e.g., [29,30] for optimizing TTL for ad hoc network routing and flooding). Every time a packet is relayed, its

TTL value decreases. This continues until TTL reaches zero or when the packet reaches the destination. TTL is particularly important to wireless networks due to unreliable wireless links and node mobility. Without TTL, a packet destined for unreachable nodes may stay in the network for an excessive period of time, thus reducing the efficiency of the network.

When a node finds that the TTL of a packet reaches zero before the packet arrives at its destination, the node discards the packet without forwarding it. In this case, the initial TTL has to be increased when the packet is retransmitted, otherwise the retransmitted packet will be dropped at the same node using the same geographic routing protocol as long as the network topology remains unchanged, since each packet is routed independently in geographic routing. Note that increasing TTL is not an optimal solution because it reduces energy efficiency. There are many reasons a packet may be dropped, such as node mobility and unreliable wireless links. In such cases, increasing TTL may not help the retransmitted packet reach the destination. It only permits the retransmitted packet to stay in the network longer. Using our pruning algorithm, if a packet is dropped due to TTL, a retransmission can travel further than the previous packet through the pruned path, and thus the probability that the packet is delivered to the destination can be increased without increasing TTL.

An example is given to illustrate how the algorithm can improve delivery rate under the constraint of TTL. The network topology is illustrated in Figure 2.4. In Figure 2.4, both dashed and solid edges represent wireless links between nodes. The solid edges are the links remained after the graph planarization process; while the dashed edges are those deleted in the process. Assume that a packet destined for node J is generated at node A.

With GPSR, a packet will go through the following path:

$$\langle A, C, E, F, E, D, C, B, A, B, C, D, G, H, J \rangle, \tag{2.2}$$

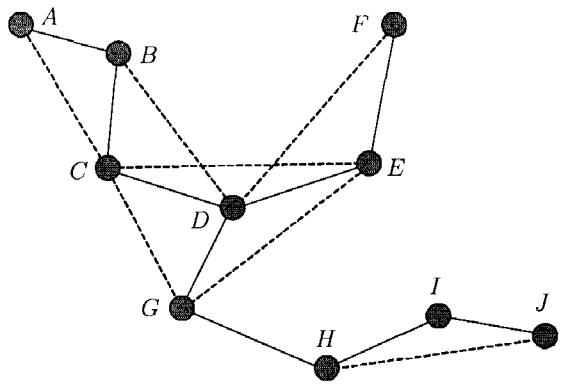


Figure 2.4. An example of network topology. The source node is A and the destination node is J. Both dashed and solid edges denote links, and solid edges also denote links in the planar graph.

where $\langle A, C, E\rangle$ and $\langle H, J\rangle$ are obtained by the greedy forwarding mode, and

$$\langle E, F, E, D, C, B, A, B, C, D, G, H\rangle \tag{2.3}$$

is obtained by the perimeter mode using the right hand rule. Suppose the maximum number of hops allowed, or TTL, is initialized as 12 at node A when the packet is generated. The packet will be dropped at node G since 12-hop limit has been reached. By applying the path pruning algorithm, a shortened path can be obtained as the packet propagates along the GPSR route. For example, after node C transmits the packet, it listens to the channel, and updates its next hop information when it hears that the same packet is transmitted by its neighboring nodes at a later time. Therefore after a series of updates, node C will set its next hop as node G, and the pruned path is obtained as $\langle A, C, G\rangle$, where node G is the point where the packet is dropped under the constraint of TTL=12. Now the retransmitted packet can travel along the pruned path to node G in two hops and then resume the routing using the GPSR protocol, thus reaching the destination node through the route $\langle A, C, G, H, J\rangle$, which results in a successful delivery as the number of hops is smaller than the TTL.

In the case that a node discards a packet due to TTL, it is necessary for the node to record whether the packet is ended in greedy mode or detouring mode,

and if in detouring mode, also to record the node that forwards the packet to it and the node where the packet last entered the detouring mode, so that the pruned path does not repeat the path found by GPSR. Use the same example to illustrate this point. As shown in Figure 2.4, consider the transmission of a packet from node A to node J with a TTL value of 10. The packet will be dropped at node C with the GPSR route $\langle A, C, E, F, E, D, C, B, A, B, C\rangle$. When the packet is retransmitted one more time, the packet will go directly from node A to node C (i.e., the pruned path $\langle A, C\rangle$). At this point, however, without additional information, the packet will be routed again as the GPSR route $\langle A, C, E, F, E, D, C, B, A, B, C\rangle$, and the packet will be dropped again at node C. To avoid this situation, the following information need to be recorded at node C: i) the mode the packet was in when last dropped (i.e., detouring mode in this example); and if it is detouring mode, ii) the last node (node B in this example) that delivered the packet to node C; and iii) the node where the packet last entered the detouring mode. In the retransmission, after going through the pruned path $\langle A, C\rangle$, node C will deliver the packet following the right hand rule with respect to edge $\overline{BC}$, so the packet will be delivered from node C to node D. The information where the packet last entered the detouring mode is needed for determining when the packet can be switched back to the greedy mode as in GPSR.

2.2 Simulation Results

In this section, the simulation results are presented to demonstrate the proposed path pruning algorithm. Apply the path pruning algorithm to two geographic routing protocols: GPSR and GOAFR$^+$. Both GPSR and GOAFR$^+$ combine a greedy forwarding mode and a detouring mode (perimeter mode in GPSR, and face routing mode in GOAFR$^+$). Assume that the routing algorithms execute faster compared to possible network mobility, and thus node mobility is not simulated.

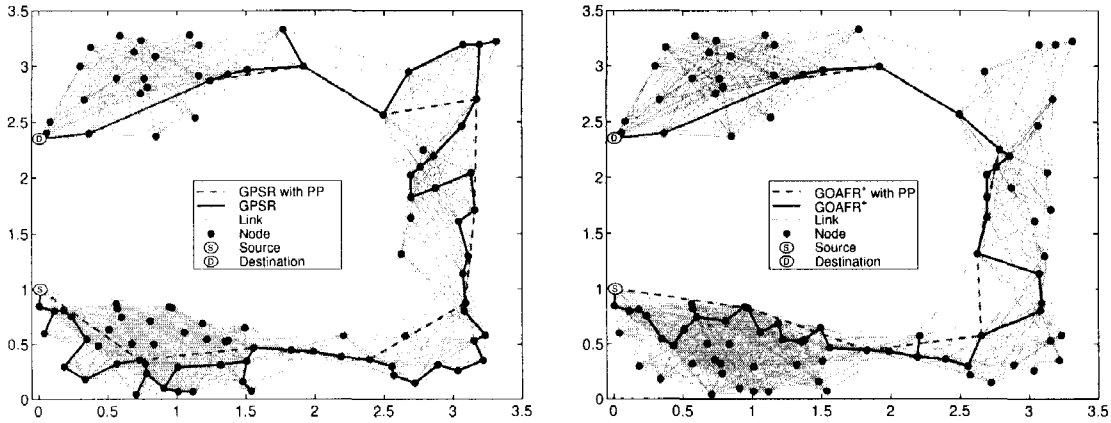


Figure 2.5. (a) GPSR and GPSR with path pruning (PP) for a C-shape network topology. GPSR: 68 hops; GPSR with PP: 11 hops. (b) GOAFR$^+$ and GOAFR$^+$ with PP for the same network topology. GOAFR$^+$: 47 hops; GOAFR$^+$ with PP: 10 hops.

2.2.1 A Routing Example with Void Region

The path pruning algorithm is tested using the ns-2 [31] environment and a routing example is reported here. The performance metric for the routing algorithms is the number of hops of a routing path. The transmission radius of each node is fixed at one unit distance. In this example, a network topology with 100 nodes covering a square field of side length 3.5 units is simulated. The topology is approximately a C-shape with void region in the center as shown in Figure 2.5 (a). The source node is located close to the lower left corner; while the destination node is located close to the up left corner. The first packet is delivered using the GPSR protocol. At the same time, the path pruning algorithm is applied, so that when the first packet is delivered, a pruned path (denoted as GPSR with PP in Figure 2.5, where "PP" stands for "Path Pruning") is also obtained based on the GPSR route. Subsequent packets can be delivered using the pruned path. As shown in Figure 2.5 (a), the GPSR route has 68 hops, while the route found after path pruning has only 11 hops. For the same topology, our simulation shows that the route found by GOAFR$^+$ has 47 hops and the route found by applying the path pruning algorithm to GOAFR$^+$ has only 10 hops, which is shown in Figure 2.5 (b).

In general, when the source and the destination of packets fall in different sides of a void region, it is likely that GPSR or GOAFR$^+$ enters one or multiple times of detouring mode, and the path pruning algorithm is efficient in shortening the routing path. Note that in the worst case, the pruned path may be the same as the corresponding GPSR or GOAFR$^+$ routes (i.e., no shortcut can be found). In the following, we evaluate the improvement on the routing performance of the proposed algorithm in the average case.

2.2.2 Average Performance

To evaluate the average performance improvement of the proposed path pruning algorithm, four routing algorithms: GPSR, GOAFR$^+$, GPSR with path pruning, and GOAFR$^+$ with path pruning are compared. The simulation configuration is similar to that in [8, 9]. The communication range of each node is fixed at one unit distance. The network topology is generated by randomly and uniformly placing nodes on a square field of side length 20 units and by randomly choosing a source-destination pair. The network density changes as the number of nodes in the square field is changed. In the simulation, the number of nodes generated in the square field ranges from 100 to 1900, which corresponds to network densities ranging from 0.79 to 14.9 nodes per unit disk of area π. For a given network density, 2000 realizations of network graphs (N_i, n_{s_i}, n_{d_i}) are generated.

The performance of an algorithm A on a network is defined as [8]

$$P_A(i) = \frac{h_A(N_i, n_{s_i}, n_{d_i})}{h_D(N_i, n_{s_i}, n_{d_i})}, \tag{2.4}$$

where $h_A(N_i, n_{s_i}, n_{d_i})$ is the number of hops of the route obtained by routing algorithm A on network N_i with source node n_{s_i} and destination node n_{d_i}, and $h_D(N_i, n_{s_i}, n_{d_i})$ is the number of hops of the shortest path between n_{s_i} and n_{d_i} on network N_i founded by the Dijkstra algorithm [32]. The average performance of algorithm A is given by

$$\overline{P_A} = \frac{1}{K} \sum_{i=1}^{K} P_A(i), \tag{2.5}$$

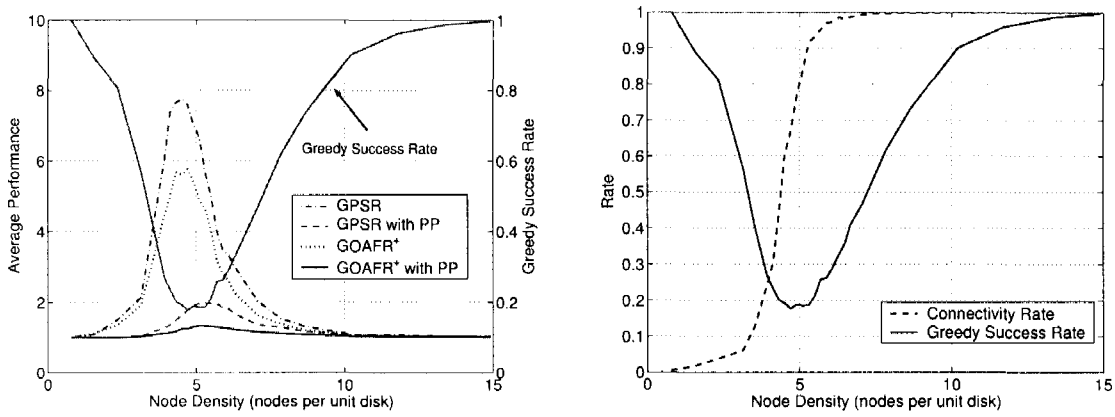
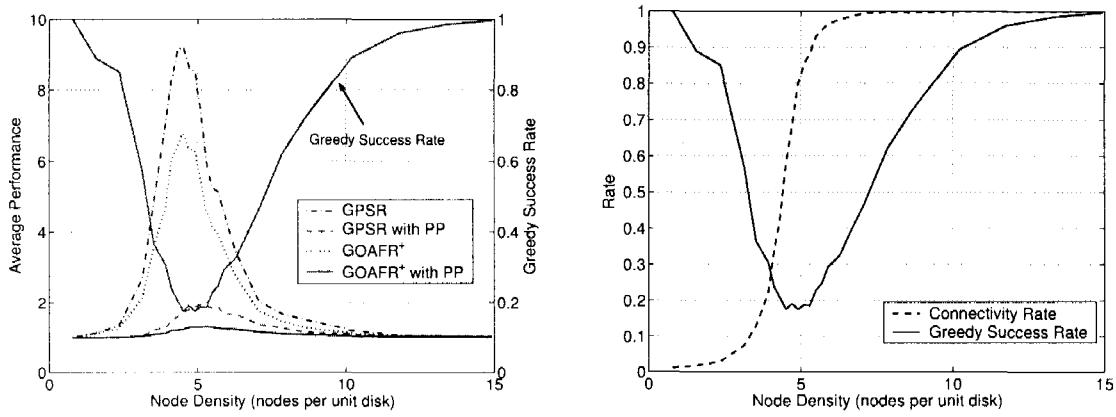


Figure 2.6. (a) Average performance of routing algorithms versus network density. The graphs are planarized using the GG algorithm. (b) The relationship of the connectivity and greedy success rates.

where K is the number of network realizations in which there exists a path from the source to the destination, among a total of 2000 realizations for a given network density. Figure 2.6 (a) and Figure 2.7 (a) show the average performance of the four algorithms versus the network density. The results in Figure 2.6 (a) are obtained when the GG (Gabriel Graph) algorithm is used for planarization, while in Figure 2.7 (a), the RNG (Relative Neighborhood Graph) algorithm is used for planarization. The corresponding relationships of the connectivity rate and greedy success rate of the two cases are depicted in Figs. 2.6 (b) and 2.7 (b). The greedy success rates are also plotted in Figure 2.6 (a) and Figure 2.7 (a), with respect to the right Y-axis. The connectivity rate is defined as $K/2000$. The greedy success rate is defined as the number of network realizations in which the source and destination are connected and the routing only involves greedy forwarding mode, divided by K. Note that the network realizations of Figure 2.6 and Figure 2.7 are independent.

Similar to [9], assume an ideal environment without collision in MAC layer because the objective is to evaluate the performance of routing algorithms. The simulation environment is implemented by C++. ns-2 is not used because there exists possible interference from other layers in an ns-2 environment and the number of nodes in the simulation can be too large to be handled efficiently by ns-2.



(a) (b)

Figure 2.7. (a) Average performance of routing algorithms versus network density. The graphs are planarized using the RNG algorithm. (b) The relationship of the connectivity and greedy success rates.

The following observations can be obtained from Figure 2.6 and Figure 2.7.

(1) The path pruning algorithm improves the performance of geographic routing significantly around a wide density range (approximately from 3 to 7 nodes per unit disk), where the greedy success rate is low as expected. For example, in average the routing algorithm with path pruning can reduce as much as about 80% hops compared to their counterparts without path pruning in the GG case, and the reduction of hops can be as much as about 85% in the RNG case. The maximum performance improvement due to the path pruning algorithm comes at a critical density region between $4\sim5$ nodes per unit disk. The observation of critical density region and the performance of GPSR and GOAFR$^+$ is consistent to those of [8, 9].

(2) For the relatively low and high network density, all protocols, with or without path pruning, perform approximately the same. This is because at very low network density, the source and the destination are rarely connected (which can be seen from the low connectivity rate); if they are connected, they are very likely direct neighbors, and thus the routing only involves greedy mode (which can be seen from the high greedy success rate). At very high network density, the source and the destination are usually connected and most routing involves only greedy forwarding mode.

TABLE 2.2

95% Performance Confidence Intervals of GPSR with and without PP (under GG, 'UB' denotes upper bound and 'LB' denotes lower bound)

Node Density	GPSR			GPSR with PP		
	UB	Mean	LB	UB	Mean	LB
3.14	2.59	2.14	1.69	1.05	1.03	1.01
3.93	6.72	6.00	5.28	1.28	1.24	1.20
4.32	8.34	7.65	6.96	1.47	1.40	1.33
4.91	7.61	7.04	6.47	1.86	1.78	1.70
5.89	3.74	3.49	3.24	1.85	1.77	1.69
7.07	2.19	2.03	1.87	1.48	1.41	1.34

(3) Comparing the results of the GG and RNG cases, we find that GPSR and GOAFR$^+$ perform better in the GG case than in the RNG case within the critical density region. This is due to that more edges are removed in the planarization step using RNG than using GG, and thus GPSR/GOAFR$^+$ in the RNG case has to go around bigger faces during the detouring mode.

(4) In the density region when the greedy success rate is lowest, GPSR and GOAFR$^+$ have their respective worst performance. From this point, as density continues to increase, greedy success rate becomes higher, and the effectiveness of the path pruning algorithm may be lower than the previous case. Finally as the greedy success rate comes close to one, the path pruning algorithm may only reduce a few hops or none.

In addition, the 95% confidence intervals of the average performance for GPSR and GOAFR$^+$ at several node densities, with and without path pruning, are shown in Tables 2.2, 2.3. The GG algorithm is used for graph planarization here. In these tables, the upper and lower bounds of the confidence intervals are listed. It can be observed that the performance confidence intervals of GPSR and GOAFR$^+$ are relatively large, especially at the density range where there is a high probability of detouring mode. In contrast, the corresponding confidence intervals

TABLE 2.3

95% Performance Confidence Intervals of GOAFR$^+$ with and without PP (under GG, 'UB' denotes upper bound and 'LB' denotes lower bound)

Node Density	GOAFR$^+$			GOAFR$^+$ with PP		
	UB	Mean	LB	UB	Mean	LB
3.14	2.26	1.91	1.56	1.02	1.01	1.00
3.93	4.97	4.44	3.91	1.12	1.10	1.08
4.32	6.10	5.67	5.24	1.16	1.14	1.12
4.91	5.76	5.41	5.06	1.28	1.26	1.24
5.89	2.96	2.80	2.64	1.27	1.25	1.23
7.07	1.69	1.62	1.55	1.13	1.12	1.11

of GPSR and GOAFR$^+$ with path pruning are significantly smaller. This is another benefit of the path pruning algorithm.

2.2.3 Routing Overhead and Scalability

It is proved earlier that if only greedy forwarding mode is involved in routing, it is not necessary to apply the path pruning algorithm. If both greedy forwarding mode and detouring mode are involved, the path pruning algorithm only needs to be applied in the detouring mode, which means that only nodes in the detouring mode need to maintain three states. By doing so, the performance improvement is almost the same as if the path pruning algorithm is applied to all nodes. This is verified by Figure 2.8, where we compare the average performance of two cases: applying the pruning algorithm in both modes, and applying the pruning algorithm only in detouring mode. The parameter configuration is the same as the one in Section 2.2.2. From Figure 2.8, we can see that no significant difference in the performance can be observed for the two cases. However, applying the path pruning algorithm only to the detouring mode has much lower overhead than applying it to both modes. Therefore, the path pruning algorithm is designed to be applied only in the detouring mode.

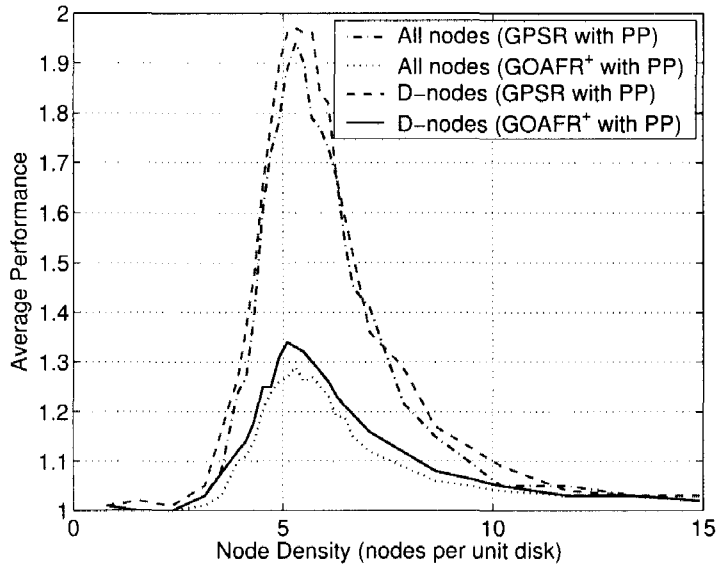


Figure 2.8. Comparison of average performance when the path pruning algorithm is applied to only nodes in the detouring mode (indicated by "D-nodes") and to all nodes. The graphs are planarized using the GG algorithm.

Next, plot the average overhead when the path pruning algorithm is applied in the detouring mode only. Average overhead is defined as the ratio of the number of nodes that need to maintain three states ($n.hop$, $n.next$, and $m.id$) and the number of total nodes on the original route, averaged over 2000 topology realizations for a given network density. Figure 2.9 depicts the temporary overhead and steady-state overhead of the path pruning algorithm when applied to GPSR and GOAFR$^+$ protocols. When the first packet is delivered using GPSR or GOAFR$^+$, every node in the detouring mode needs to maintain three states. This is the temporary overhead, because some nodes will drop their states after a certain time as no subsequent packet is routed through them. The density region of the peaks of the temporary overheads in Figure 2.9 do not completely match the worst of the average performance of GPSR and GOAFR$^+$ algorithms in Figure 2.6 and Figure 2.7. In fact, the network density for peak temporary overhead is a little lower than the density for worst average performance. This is because Figure 2.9 plots the percentage of nodes in the detouring mode, and a peak percentage of

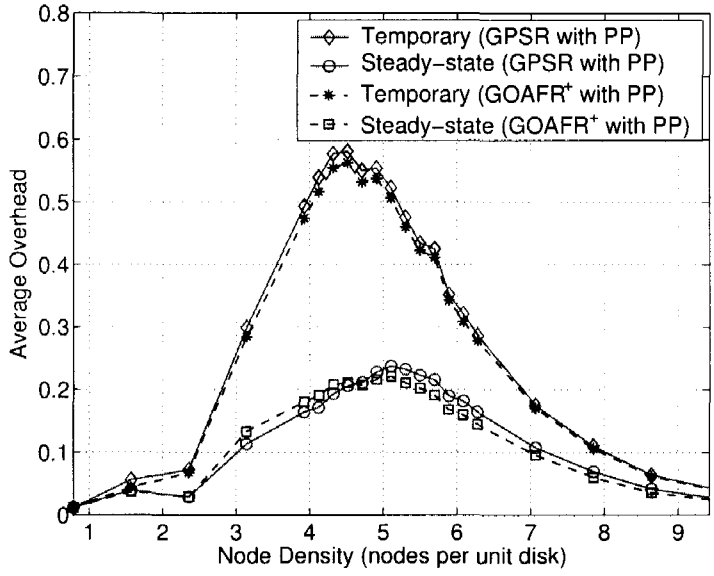


Figure 2.9. Average temporary and steady-state overhead of the path pruning algorithm. The graphs are planarized using the GG algorithm.

such nodes do not necessarily result in longest routing path *relative* to the optimal path. When network density is low, it is possible that even the optimal path has to go through certain "detours".

In the steady-state, only those detouring nodes remained on the pruned path need to maintain states. Define steady-state overhead as the ratio of the number of nodes that need to maintain states on the pruned path and the number of total nodes in the original GPSR or GOAFR$^+$ route. From Figure 2.9 we observe that in average, up to about 23% nodes of the original route need to store extra information. Furthermore, Tables 2.4, 2.5 list the upper and lower bounds of the 95% confidence intervals of the average temporary and steady-state overheads for GPSR and GOAFR$^+$ with path pruning, where the GG algorithm is used for graph planarization. Again, these tables show that the path pruning algorithm narrows the confidence intervals.

Note that although geographic routing protocols such as GPSR and GOAFR$^+$ have restricted themselves to using only completely stateless nodes, the proposed path pruning algorithm has relaxed this condition by adding a little storage over-

TABLE 2.4

95% Overhead Confidence Intervals of GPSR with PP (under GG)

Density	Temporary			Steady-state		
	UB	Mean	LB	UB	Mean	LB
3.14	0.36	0.30	0.24	0.13	0.11	0.09
3.93	0.52	0.49	0.46	0.18	0.16	0.14
4.32	0.60	0.58	0.56	0.20	0.19	0.18
4.91	0.57	0.55	0.53	0.24	0.23	0.22
5.89	0.37	0.35	0.33	0.20	0.19	0.18
7.07	0.19	0.18	0.17	0.11	0.11	0.11

head (three passively maintained states on a small subset of nodes on the route) but no extra communication and computation overhead are needed. Considering the significant performance improvement as shown in Figure 2.6 and Figure 2.7, the extra storage overhead of the proposed path pruning algorithm is well justified.

Furthermore, in the pruning algorithm, nodes in the detouring mode have to store the next hop on a per destination basis. If a node lies in a local minimum (dead-end), it is likely that several packets for different destinations around the same geographic area will get stuck there and consequently the size of the routing table will be dependent on the amount of traffic. However, the path pruning algorithm can be adapted so that the nodes store next hops based on the geographic area of the destination, instead of individual destination nodes, and thus subsequent packets to a nearby destination could also take advantage of the states in the nodes. This will reduce the overall network routing overhead in the presence of multiple source-destination pairs. In this sense, the path pruning algorithm is also a scalable one.

To evaluate the effect of the path pruning algorithm on the delivery rate when packets are under the TTL constraint, some simulations were done. For details about those simulations, please refer to [33,34].

TABLE 2.5

95% Overhead Confidence Intervals of GOAFR$^+$ with PP (under GG)

Density	Temporary			Steady-state		
	UB	Mean	LB	UB	Mean	LB
3.14	0.34	0.28	0.22	0.16	0.13	0.10
3.93	0.50	0.47	0.44	0.19	0.18	0.17
4.32	0.58	0.55	0.52	0.22	0.21	0.20
4.91	0.55	0.54	0.53	0.22	0.22	0.22
5.89	0.36	0.34	0.32	0.18	0.17	0.16
7.07	0.18	0.17	0.16	0.10	0.10	0.10

2.3 Conclusion

This chapter proposed a simple yet effective path pruning strategy to reduce the number of hops for route discovered by existing geographic routing protocols. With the help of three states passively maintained by a subset of nodes on the route, the proposed algorithm is capable of reducing a large portion of hops. Simulation results have demonstrated that the path pruning algorithm has low complexity of implementation but gives improved routing performance when applied to geographic routing for wireless sensor networks.

CHAPTER 3

ANCHOR-BASED GEOGRAPHIC ROUTING

The previous chapter proposes a path pruning strategy to shorten paths found by existing geographic routing algorithms. Another method to cut paths short is to use anchor-based geographic routing [23, 25], where anchors or waypoints are set as relay nodes. A packet is routed from the source to the destination through a sequence of anchor nodes. The anchor list can be set in advance [23], or adaptively obtained and fed back to the source after the first packet is routed [25]. In [23], two methods are proposed to obtain a path with anchors: one is GMPD (Geographic Map-based Path Discovery) and the other is FAPD (Friend Assisted Path Discovery). In GMPD, it is assumed that all the nodes in the network have the density map of the whole network. This is usually not possible for large-scale sensor networks. In FAPD, a path is first evaluated and if it does not perform well, FAPD responders try to find an anchor-based path. This method needs to maintain an assistant network, which may be of high overhead. In [25], three waypoint-based routing schemes are proposed to improve the efficiency of routing, among which the most effective one uses nodes on the convex hull of the path as waypoints. However, it needs to maintain a long list of nodes in the packet even if the packet never enters detouring mode and the computation involved in the convex hull forming process is complicated.

Furthermore, the existing anchor-based geographic routing algorithms such as those in [23, 25], do not take into account the lifetime of the nodes. Once the list of anchors between a source/destination pair is set, the path from the source to the destination usually does not change unless the network topology changes. Moreover, some nodes are frequently used as anchors and their energy may run

out very quickly. As a result, the lifetime of the network is reduced.

This chapter proposes a Projection Distance-based Anchor scheme (refer it as PDA hereafter) to generate the anchor list based on the projection distance of nodes in detouring mode. The projection is with respect to the virtual line linking the source and destination nodes. Compare the proposed technique for anchor-based geographic routing to that of [25], and demonstrate that the former achieves better performance in terms of path length and number of anchors. Furthermore, the method does not need to keep a long list of nodes if the packet does not enter detouring mode.

To address the uneven distribution of energy consumption, dynamic anchors are generated for the same source-destination link by introducing a random shift to the location of each anchor when each packet is routed. Because the shift is randomly generated, different packets may be routed through a different list of anchors. This allows different nodes to be involved in the routing process and the energy consumption is better distributed among nodes in the network. It is shown analytically that this strategy improves the network lifetime. In the simulation, we compare PDA with and without this lifetime-improving strategy in random network topologies. The results show that this strategy increases the number of packets delivered in the network lifetime given the same initial energy of nodes.

The remainder of this chapter is organized as follows: In Section 3.1, the proposed PDA algorithm is described. The lifetime-improving strategy is discussed in Section 3.2. This strategy is further analyzed in Section 3.3. Simulations and comparisons are given in Section 3.4 and 3.5. Section 3.6 concludes this chapter.

3.1 The PDA Algorithm

To obtain an anchor list adaptively, route the first packet of a burst from the source to the destination using an existing geographic routing algorithm, such as GPSR [5] and GOAFR$^+$ [9]. During the routing of the first packet, an anchor list is built. After the first packet is delivered, the anchor list is sent back to the

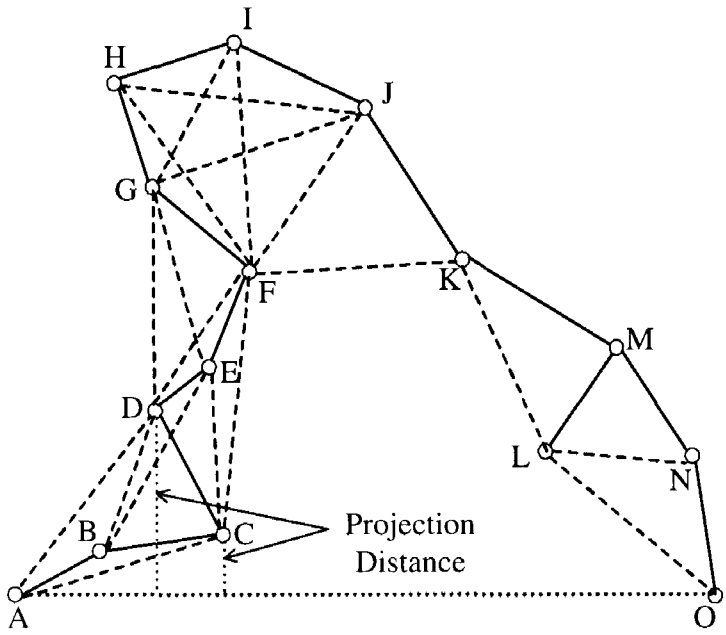


Figure 3.1. A routing example. Node A is the source, and node O is the destination.

source from the destination, and the list is embedded into subsequent packets. A subsequent packet is then routed from the source to the first anchor node, then to the second anchor node, and so on, until it reaches the destination.

A typical routing example involving both greedy and detouring modes is given in Figure 3.1, where A is the source node and O is the destination node, solid lines represent links remained after planarization and dash lines represent links removed due to planarization. The dotted line AO represents the virtual line linking the source and the destination. Use these line notations throughout the dissertation unless it is declared otherwise. Suppose that GPSR is the routing protocol. A packet will be first forwarded from A to C in greedy mode. At node C, since none of the neighbors of C is closer to the destination than itself, the packet will be switched to detouring mode. In GPSR, perimeter routing is used as the detouring strategy.

In detouring mode, only solid lines may be used. The route discovered by GPSR in this example is $\langle A, C, D, E, F, G, H, I, J, K, L, O\rangle$, which is clearly not

optimal. If some anchor nodes can be used as intermediate destinations, then the path may be shortened. For example, assuming that I is an anchor, a packet is routed from A to I first, then from I to O. With anchor I, the route discovered by GPSR becomes $\langle A, D, G, H, I, J, K, L, O\rangle$, which is significantly shorter than the original route. The objective is to develop an approach to obtain an anchor list adaptively as the first packet is routed.

Examining the nodes on faces that intersect the virtual line linking the source and destination on planar graphs of random topologies, it is found that the distance from the nodes to the virtual line usually increases first and then decreases. It does not fluctuate frequently. Inspired by this observation, it is proposed to use the change of the distance from nodes to the virtual line as an indicator when choosing anchors.

In general, suppose the path of the first packet is $\langle n_0, n_1, n_2, \cdots, n_{m-1}, n_m\rangle$, where n_0 is the source and n_m is the destination. If the routing process never enters detouring mode (i.e., only greedy mode is involved in the path discovery), it is unnecessary to set anchors because the path is already suboptimal [7]. Otherwise, we obtain a list of anchor nodes using the following PDA algorithm.

For a node n_i, denote its projected point on $\overrightarrow{n_0 n_m}$ as n_i'. Then, define n_i's *projection distance* as

$$\psi(n_i) = \begin{cases} |\overrightarrow{n_i n_i'}|, & n_i \text{ is on the left hand side of } \overrightarrow{n_0 n_m}, \\ -|\overrightarrow{n_i n_i'}|, & \text{otherwise.} \end{cases} \tag{3.1}$$

Suppose that n_{l_j} and n_{u_j} are the first node and the last node of the jth detouring sequence in the path $\langle n_0, n_1, n_2, \cdots, n_{m-1}, n_m\rangle$, respectively. Suppose loops have been removed. For the jth detouring sequence, the anchor list Γ_j is obtained as follows:

$$\Gamma_j = \{n_i : l_j < i < u_j, (\psi(n_i) - \psi(n_{i-1}))(\psi(n_{i+1}) - \psi(n_i)) \leq 0\} \cup \{n_{l_j}, n_{u_j}\}. \tag{3.2}$$

In other words, the anchor list contains all local minimum or maximum nodes in detouring mode with respect to the projection distance to the virtual line linking

TABLE 3.1

The PDA Algorithm

```
1.    if n_i is in detouring mode
2.       if n_i is in the anchor list
3.            delete all nodes after n_i in the anchor list.
4.       end
5.       if n_i = n_m
6.            add n_i to the anchor list.
7.            send the list back to the source.
8.       else if n_{i-1}, n_{i+1} are in detouring mode
9.            if (ψ(n_i) − ψ(n_{i-1}))(ψ(n_{i+1}) − ψ(n_i)) ≤ 0
10.                add n_i to the anchor list.
11.           end
12.      else
13.           add n_i to the anchor list.
14.      end
15.   end
```

the source and destination. Note that because n_{i-1} and n_{i+1} are neighbors of n_i, their location information is known to n_i, and thus n_i can calculate $\psi(n_{i-1})$ and $\psi(n_{i+1})$ without additional communication overhead. The anchor list Γ for the routing path is then obtained as the union of all Γ_j and n_m, where nodes are sorted according to their subscripts in increasing order.

The anchor list is fed back to n_0. Now when n_0 sends a packet to n_m, the anchor list Γ is embedded in the packet head. The packet will be routed from n_0 to the first anchor, then from the first anchor to the second anchor, and so on, until it reaches n_m. The routing between any two consecutive anchors can use the original geographic routing protocol such as GPSR. Due to the use of anchors, the chance of entering lengthy detouring mode is significantly reduced, since now greedy forwarding may be used at the segments where previously only detouring mode is possible.

The implementation of the PDA algorithm to obtain the anchor list is de-

tailed in Table 3.1. The algorithm is to be applied when the first packet is routed. n_i represents the node holding the packet. n_m denotes the destination. Besides the locations of source and destination (contained in the packet), an anchor list Γ is needed. In the anchor list Γ, each anchor node's location is recorded.

According to Table 3.1, if n_i is in detouring mode, the algorithm first checks whether n_i is already in the anchor list. If n_i is in the anchor list, it means that the node has been visited and a loop exists in the path. The loop is removed by deleting all nodes after n_i in the anchor list. If n_i is the destination, it is added to the anchor list and the list is returned to the source. If n_i's previous hop and next hop are in detouring mode, the algorithm checks if

$$(\psi(n_i) - \psi(n_{i-1}))(\psi(n_{i+1}) - \psi(n_i)) \leq 0. \tag{3.3}$$

If the inequality holds, n_i is a local minimum or maximum node and it is added to the anchor list. If either n_i's previous hop or next hop is in greedy mode, then n_i is either the first node or the last node of a detouring sequence, and the node is added to the anchor list.

Apply the PDA algorithm to the example given in Figure 3.1. According to GPSR, the first packet enters the detouring mode at node C, so C is added to the anchor list. When D holds the packet, because

$$(\psi(D) - \psi(C))(\psi(E) - \psi(D)) > 0,$$

D is not added to the anchor list. Similarly E, G, and H are not added to the anchor list. Because

$$(\psi(I) - \psi(H))(\psi(J) - \psi(I)) < 0,$$

I is added to the anchor list. After the first packet is delivered, the anchor list $\{C, I, J, O\}$ is obtained by the PDA algorithm. Subsequent packets are routed through

$$\langle A, C, F, I, J, K, L, O \rangle.$$

It is clear that the PDA algorithm obtains a shorter path compared to the one obtained by GPSR.

3.1.1 Comparison of PDA and OAS

Adaptive anchors are used in [25] for geographic routing, where three schemes are proposed: single-waypoint adaptive scheme, optimal adaptive scheme, and multi-point adaptive scheme. Refer the optimal adaptive scheme as OAS here. OAS performs best among the three schemes in [25]. The OAS algorithm obtains anchors based on the vertices of the convex hulls along the routing path. When all nodes on the path (or most of them) form a convex polygon, the anchor list found by OAS is very long. For example, in Figure 3.2, node A sends packets to node M. The first packet is routed by GPSR whose path is:

$$\langle A, C, D, E, F, G, I, L, M \rangle,$$

where nodes C, D, E, and F are in detouring mode. The OAS algorithm finds the convex hull formed by these nodes and obtains the following anchors:

$$A, D, E, F, G, I, L, M.$$

Using the proposed PDA algorithm, the following anchors are obtained: C, F, M. In this case, the PDA algorithm results in a shorter anchor list than OAS.

The path based on anchors will be almost the same as the path without anchors if the anchor list contains almost all nodes of the path along which the first packet traverses. Due to planarization, many links are removed (deleted links can only be used in greedy mode) and are not used in detouring mode. If the anchors are close to each other, they have less chance to use shortcuts. For the example in Figure 3.2, D, E, F are in the anchor list found by OAS. When D receives a packet, it sends the packet to next anchor E. Then, E sends the packet to F. This way, the packet misses the shortcut DF.

3.1.2 A Remark

Any two consecutive anchors in the anchor list found by the PDA algorithm can usually be routed in greedy mode without entering detouring mode. However,

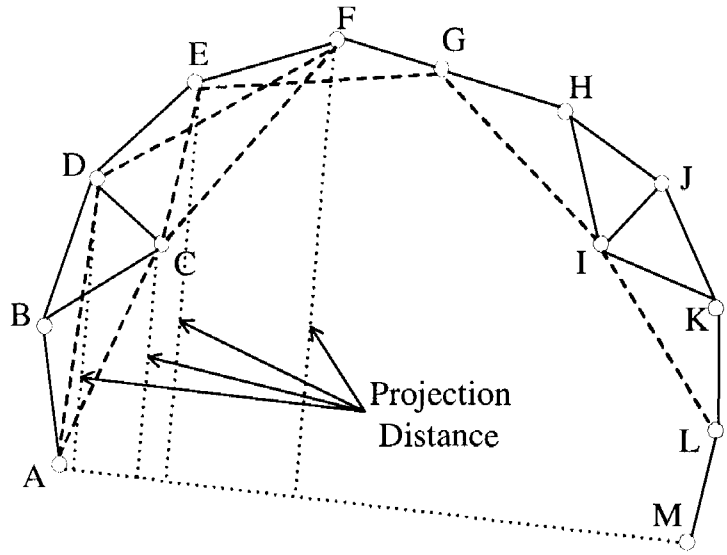


Figure 3.2. Comparison of PDA and OAS. Node A is the source, and node M is the destination.

the PDA algorithm can not always eliminate routing in detouring mode. You can see another example in Figure 3.3. Suppose that node S is to send a packet to node D. Based on GPSR and the PDA algorithm, the following anchors are obtained: E, F, D. With this anchor list, when S sends a packet to D, it will first forward the packet to E. When the packet comes to A, it will choose C instead of B since C is closer to E than B. C has no neighbor that is closer to E than C. Therefore, the packet enters detouring mode at C, and it is then routed along the dash line in Figure 3.3 according to the right hand rule of perimeter routing in GPSR. In this case, the packet goes through a longer path than the one without anchors. The similar problem exists in the OAS algorithm. For the example in Figure 3.3, S, G, and E are the first three anchors in the anchor list found by OAS. Hence succeeding packets are sent to G and then E. During the routing from G to E, the packets arrive at the local minimum C. At C, the packets enter detouring mode and they also go a longer path than the path without anchors.

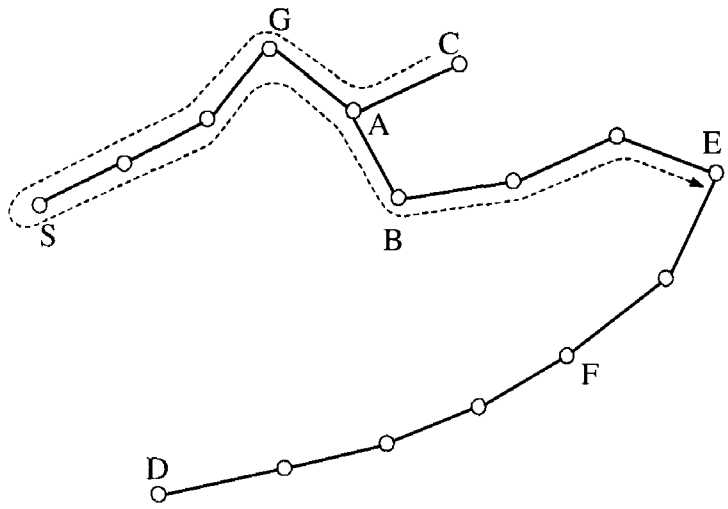


Figure 3.3. An example where the anchor-based path may be longer than the path without anchors. Node S is the source, and node D is the destination. The dash line indicates the path when routing from anchor C to E according to the right hand rule.

3.2 The Lifetime-Improving Strategy

One problem with PDA and other anchor-based geographic routing schemes is that the path between the source and destination pair changes little once the anchor list is generated. For example, in Figure 3.4, suppose the source node is S and the destination node is O. Using PDA, the anchor list $\{E, M, O\}$ can be obtained. The resulting path from S to O is

$$\langle S, A, B, E, G, H, M, P, Q, T, O \rangle.$$

The path usually does not change unless the network topology changes. Furthermore, regardless of the anchor-based geographic routing schemes, some nodes (e.g., nodes near the boundary of the network) tend to be selected as anchors for more than one source-destination pair. Both of these observations indicate that some nodes are used for routing more frequently than others. As a result, their energy may run out much faster than other nodes.

To deal with the uneven distribution of energy consumption and prolong the lifetime of the network, this subsection proposes a lifetime-improving strategy

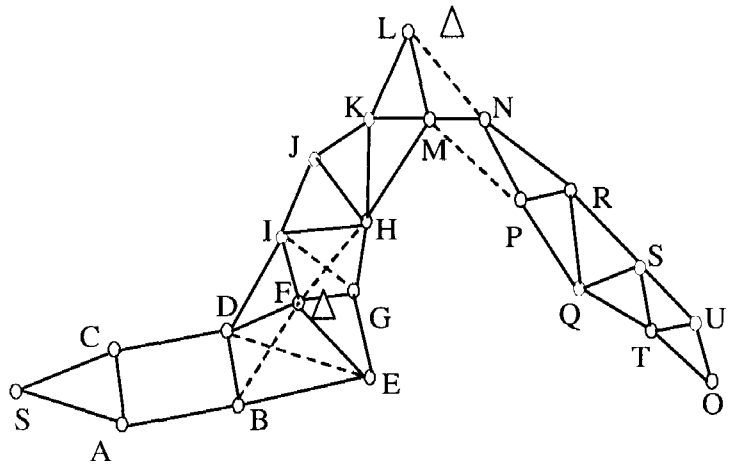


Figure 3.4. A PDA routing example with lifetime-improving strategy. The original anchors obtained by PDA are $\{E, M, O\}$, and the two triangles represent the new anchors after introducing dynamic anchors.

by introducing dynamic anchors. In this strategy, the first packet is assumed to be routed by one of the existing geographic routing algorithms. After the successful transmission of the first packet, a list of anchors is obtained if the packet ever enters the detouring mode. The anchors are represented with a list of coordinates:

$$(x_1, y_1), (x_2, y_2), \cdots, (x_k, y_k),$$

where k is the number of anchors and (x_k, y_k) is the destination. To obtain a dynamic anchor list, we introduce some random shift to the coordinates of each anchor except the destination. The random shift follows a 2-dimensional Gaussian distribution with respect to each anchor's coordinates. The distribution function is as follows:

$$f(n_x, n_y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(n_x^2+n_y^2)}{2\sigma^2}}. \tag{3.4}$$

Each time the source sends a packet, it randomly generates $k-1$ random shifts:

$$\{(n_{x_1}, n_{y_1}), (n_{x_2}, n_{y_2}), \cdots, (n_{x_{k-1}}, n_{y_{k-1}})\},$$

and the coordinates of the anchors in the dynamic anchor list are:

$$\{(x_1 + n_{x_1}, y_1 + n_{y_1}), \cdots, (x_{k-1} + n_{x_{k-1}}, y_{k-1} + n_{y_{k-1}}), (x_k, y_k)\}.$$

The source node first sends a packet destined for (x_k, y_k) to $(x_1 + n_{x_1}, y_1 + n_{y_1})$ instead of (x_1, y_1), and then the node at $(x_1 + n_{x_1}, y_1 + n_{y_1})$ sends the packet to $(x_2 + n_{x_2}, y_2 + n_{y_2})$. The process continues until the packet reaches the destination. It may occur that there is no node at location $(x_i + n_{x_i}, y_i + n_{y_i})$. If this is the case, consider that the packet has arrived at $(x_i + n_{x_i}, y_i + n_{y_i})$ if the distance from the packet's current location to $(x_i + n_{x_i}, y_i + n_{y_i})$ is smaller than a threshold γ. A packet is forwarded to the next anchor as soon as it enters detouring mode.

For example, in Figure 3.4, node S sends a packet to node O. If this is the first packet from S to O, the packet is routed with GPSR or GOAFR$^+$. The first packet goes through the path

$$\langle S, A, B, E, G, H, M, P, Q, T, O \rangle.$$

Assuming that PDA is used, the anchor list $\{E, M, O\}$ is obtained. For the second packet, introduce some random shift to the coordinates of E and M, and obtain the dynamic anchors indicated by triangles in the figure (denote the triangle corresponding to E as E', similarly the other as M'). Then, the source S first tries to send the packet to E' and the packet goes through the path $\langle S, C, D \rangle$. Since the distance from node D to E' is smaller than the communication range (assume γ equals the communication range), it can be considered that the packet has arrived at E'. Next, the packet is sent to the next anchor M'. It goes through the path $\langle D, I, J, K, L \rangle$. When the packet arrives at L, we consider that it has reached the anchor M'. Finally, the packet is sent to the destination O through $\langle L, N, R, S, U, O \rangle$. By introducing dynamic anchors, we see that the second packet does not go through anchors E and M. This way the lifetime of anchors is prolonged. Comparing the new path with the original one, it is clear that most of nodes in the original path are not used in the new path. This way, the nodes in the network are used alternately. Therefore, the energy consumption is distributed more evenly among the nodes in the network, and the network lifetime of the network is prolonged.

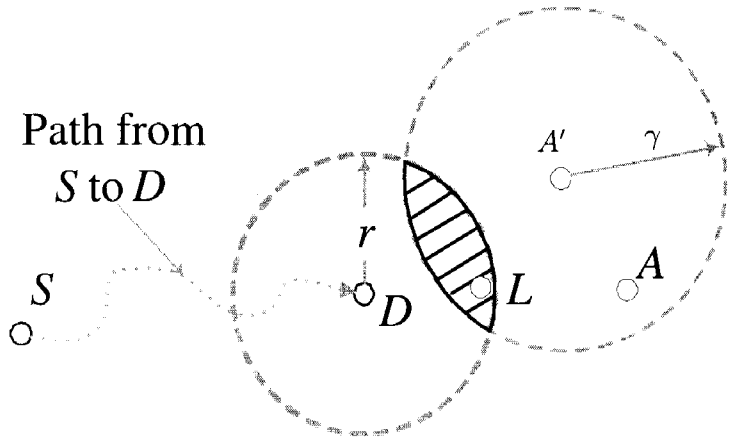


Figure 3.5. Example of routing for the last hop of the path from S to A'.

3.3 Analysis of the Lifetime-Improving Strategy

This section analyzes the probability that the original anchor is not used after introducing the random shift. Table 3.2 lists the parameters and notations used in the analysis.

Suppose the nodes are distributed randomly in the network with a uniform density function. The probability density function of D with respect to A' is $\Phi(x, y)$. As shown in Figure 3.5, D is the penultimate node on the path from S to A', and the distance $|DA'|$ must be smaller than $\gamma + r$. Otherwise the packet can not reach the circle centered at A' with radius γ after D's transmission. Let $\mathcal{H}$ denote the intersection region of the two circles centered at D and A'. There must be at least one node located within $\mathcal{H}$. Otherwise the packet can not reach the circle centered at A' after D's transmission for none of D's neighbors is within the circle.

If A is located outside $\mathcal{H}$, A can not be used as the last node on the path from S to A'. If A is located within $\mathcal{H}$, A can not be used as the last node if and only if within $\mathcal{H}$ there is a node that has a smaller distance to A' than $|AA'|$.

Consider the above two cases separately. First, consider the first case: A is located outside $\mathcal{H}$. When $|AA'| \geq \gamma$, A can not be used as the last node. Then,

$$P_1 = P(|AA'| \geq \gamma) = \oint_{x^2+y^2 \geq \gamma^2} f(x, y)dxdy. \tag{3.5}$$

TABLE 3.2

List of Parameters of the Lifetime-Improving Strategy

σ	The variance of the random shift
A	The original anchor with coordinates (x_A, y_A).
S	The anchor right before anchor A (if A is the first anchor, S is the source)
(n_{x_A}, n_{y_A})	The random shift introduced at anchor A
A'	The new anchor with coordinates $(x_{A'}, y_{A'}) = (x_A + n_{x_A}, y_A + n_{y_A})$
γ	Within which we regard the packet has arrived at the new anchor
L	The first node on the path from S to A' whose distance to A' is less than γ
D	The node right before L on the path from S to A'.
r	The communication range of each node
ρ	Network density (nodes per unit disk)

When $|AA'| < \gamma$, A must be outside of the circle centered at D. In other words, the distance between A and D must be greater than r. Let Υ denote

$$(p-x)^2 + (q-y)^2 \geq \gamma^2.$$

Using κ to denote $x^2 + y^2$ and ζ to denote $p^2 + q^2$, obtain

$$\begin{aligned} P_2 &= P(|AA'| < \gamma \ \& \ |AD| > r) \\ &= \oint_{\kappa<\gamma^2} \oint_{\zeta>r^2,\Upsilon} \Phi(p-x, q-y) f(x,y) dpdqdxdy. \end{aligned} \tag{3.6}$$

Second, consider the case that A is located within $\mathcal{H}$. Given A', D and a random point (x, y), define a function $\Psi(x, y)$ as the distance from point (x, y) to point A'. Then,

$$\begin{aligned} P_3 &= P(A \text{ is located within } \mathcal{H} \text{ and there is a node within } \mathcal{H} \text{ with } \Psi(x,y) < \Psi(0,0)) \\ &= \oint_{\kappa<\gamma^2} \oint_{\zeta\leq r^2,\Upsilon} \mathcal{W}\Theta(x,y,p,q) dpdqdxdy, \end{aligned} \tag{3.7}$$

where $\mathcal{W} = \Phi(p-x, q-y) f(x,y)$, and $\Theta(x,y,p,q)$ denotes the probability that there is at least one node (m, n) in $\mathcal{H}$ with $\Psi(m,n) < \Psi(0,0)$. Let (x, y) be the coordinates

of A', and (p, q) be the coordinates of D. Define a region as following:

$$\mathcal{G} = \{(m-x)^2 + (n-y)^2 < \gamma^2, (m-p)^2 + (n-q)^2 \leq r^2, \Psi(m,n) < \Psi(0,0)\}. \quad (3.8)$$

Suppose the size of $\mathcal{G}$ is $C(x, y, p, q)$ and the total area of the network is Ω. Because the nodes are uniformly distributed, a node is located within $C(x, y, p, q)$ with probability $C(x, y, p, q)/\Omega$. Since the network density is ρ, the total number of nodes in the network is $\rho\Omega/(\pi r^2)$. Therefore, the probability that no node is located within $\mathcal{G}$ is

$$P_{\mathcal{G}} = \left(1 - \frac{C(x, y, p, q)}{\Omega}\right)^{\rho\frac{\Omega}{\pi r^2}}, \quad (3.9)$$

and the following equation can be obtained

$$\Theta(x, y, p, q) = 1 - P_{\mathcal{G}}. \quad (3.10)$$

Substituting $\Theta(x, y, p, q)$ into (3.7), obtain

$$P_3 = \oint_{\kappa<\gamma^2} \oint_{\zeta\leq r^2,\Upsilon} \mathcal{W}\,(1 - P_{\mathcal{G}})\, dpdqdxdy. \quad (3.11)$$

Finally, the probability that A is not used as the last node in the path from S to A' is

$$P = P_1 + P_2 + P_3. \quad (3.12)$$

The following proves $0 < P < 1$.

$$\begin{aligned} P &= \oint_{\kappa\geq\gamma^2} f(x,y)dxdy + \oint_{\kappa<\gamma^2} \oint_{\zeta>r^2,\Upsilon} \mathcal{W}dpdqdxdy + \oint_{\kappa<\gamma^2} \oint_{\zeta\leq r^2,\Upsilon} \mathcal{W}(1-P_{\mathcal{G}})dpdqdxdy \\ &< \oint_{\kappa\geq\gamma^2} f(x,y)dxdy + \oint_{\kappa<\gamma^2} \oint_{\Upsilon} \mathcal{W}dpdqdxdy \\ &< \oint_{\kappa\geq\gamma^2} f(x,y)dxdy + \oint_{\kappa<\gamma^2} f(x,y)dxdy = 1. \end{aligned} \quad (3.13)$$

Because all the integral functions in P_1, P_2, and P_3 are non-negative, P_1, P_2, and P_3 are non-negative. Therefore $P \geq 0$. Furthermore,

$$\begin{aligned} P_1 &= \oint_{\kappa\geq\gamma^2} f(x,y)dxdy = \oint_{\kappa\geq\gamma^2} \frac{1}{2\pi\sigma^2} e^{-\frac{\kappa}{2\sigma^2}} dxdy > \oint_{4\gamma^2\geq\kappa\geq\gamma^2} \frac{1}{2\pi\sigma^2} e^{-\frac{\kappa}{2\sigma^2}} dxdy \\ &> \oint_{4\gamma^2\geq\kappa\geq\gamma^2} \frac{1}{2\pi\sigma^2} e^{-\frac{4\gamma^2}{2\sigma^2}} dxdy = \frac{1}{2\pi\sigma^2} e^{-\frac{2\gamma^2}{\sigma^2}} (4\pi\gamma^2 - \pi\gamma^2) > 0. \end{aligned} \quad (3.14)$$

Therefore $P > 0$, and the proposed strategy improves the lifetime of the network with nonzero probability. Notice that P_1 and P_2 have nothing to do with network density ρ. When ρ increases, $\Theta(x, y, p, q)$ increases and consequently P_3 increases, and thus P increases and the probability that the network lifetime is prolonged increases.

3.4 Simulation of PDA and OAS

First compare PDA and OAS in terms of path length, number of anchors, and computational complexity. The lifetime-improving strategy is not applied. The simulation configuration is similar to that in Section 2.2. Here the total number of nodes generated in the square field ranges from 100 to 1100, which corresponds to network densities ranging from 0.79 to 8.6 nodes per unit disk of area π. For a given network density, 5000 realizations of network graphs are generated.

3.4.1 Average Relative Path Length

To evaluate the average performance improvement of the proposed algorithm, the proposed PDA algorithm and OAS [25] are compared when they are applied to GPSR and GOAFR$^+$.

The relative path length of an algorithm A on a network N_i is defined as

$$P_A(i) = \frac{h_A(N_i, n_{s_i}, n_{d_i})}{h_D(N_i, n_{s_i}, n_{d_i})},$$

where $h_A(N_i, n_{s_i}, n_{d_i})$ is the number of hops of the route obtained by routing algorithm A on network N_i with source node n_{s_i} and destination node n_{d_i}; $h_D(N_i, n_{s_i}, n_{d_i})$ is the number of hops of the shortest path between n_{s_i} and n_{d_i} on network N_i found by the Dijkstra algorithm [32]. The average relative path length of algorithm A is given by

$$\overline{P}_A = \frac{1}{K} \sum_{i=1}^{K} P_A(i),$$

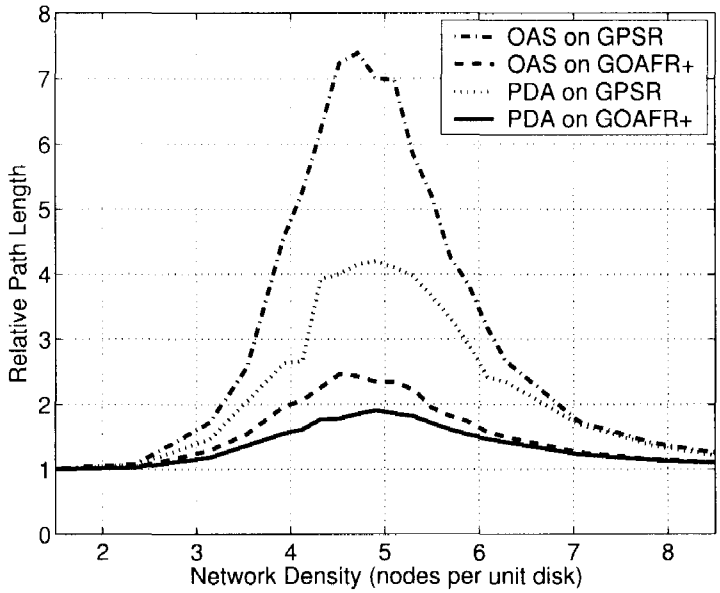


Figure 3.6. Performance comparison of the proposed PDA algorithm and the OAS algorithm, when both are applied to GPSR and GOAFR$^+$. "PDA" – the projection distance-based anchor algorithm, "OAS" – the optimal adaptive scheme.

where K is the number of network realizations in which there exists a path from the source to the destination, among a total of 5000 realizations for a given network density.

Compare the relative path length of the proposed PDA algorithm and the OAS algorithm of [25], when both algorithms are applied to GPSR and GOAFR$^+$. The results are shown in Figure 3.6. In Figure 3.6, "OAS on GPSR" means that the first packet is routed using GPSR, the OAS algorithm is used to obtain the anchor set when the first packet is routed, and the plotted curve is the relative path length of the anchor-based route. The lower the curve is, the better the performance is.

It can be seen from Figure 3.6 that relatively longer paths occur when the network density is around four to six nodes per unit disk, which is the density region with a high probability of detouring mode in GPSR or GOAFR$^+$ as shown in [9]. It can also be observed that the proposed algorithm outperforms the OAS algorithm in both the GPSR case and the GOAFR$^+$ case by a large margin. For example, when applied to GPSR at a network density of about five nodes per unit

TABLE 3.3

95% Confidence Intervals of the Relative Path Length of OAS and PDA on GPSR and GOAFR$^+$ ('UB' denotes upper bound and 'LB' denotes lower bound)

Network Density		3.14	3.93	4.32	4.91	5.89	7.07
OAS on GPSR	UB	1.913	5.153	6.708	7.438	4.060	1.822
	Mean	1.716	4.565	6.188	6.998	3.827	1.732
	LB	1.518	3.976	5.669	6.558	3.593	1.642
OAS on GOAFR$^+$	UB	1.374	2.095	2.364	2.415	1.774	1.284
	Mean	1.286	1.957	2.258	2.338	1.726	1.265
	LB	1.198	1.819	2.152	2.260	1.679	1.245
PDA on GPSR	UB	1.607	2.864	4.231	4.428	3.062	1.769
	Mean	1.460	2.630	3.926	4.202	2.907	1.688
	LB	1.312	2.397	3.621	3.977	2.753	1.608
PDA on GOAFR$^+$	UB	1.222	1.611	1.820	1.956	1.556	1.245
	Mean	1.176	1.552	1.771	1.912	1.526	1.230
	LB	1.131	1.492	1.723	1.869	1.496	1.215

disk, the algorithm has a relative path length of 4, which is significantly shorter than OAS's relative path length 7.5.

In Table 3.3, we show the 95% Confidence Intervals of the Relative Path Length of OAS and PDA on GPSR and GOAFR$^+$. It can be seen that the variation of the relative path length of PDA is smaller than that of OAS under the same network density with the same underlying routing algorithm. It means PDA performs more stable than OAS in terms of average relative path length.

3.4.2 Average Length of Anchor List

The following compares the average length of the anchor lists of the proposed algorithm and OAS. Among 5000 network realizations, we choose realizations using anchor nodes and calculate the average number of anchor nodes used in these realizations. The result is plotted in Figure 3.7. It can be seen that the average length reaches its peak value around network density 5.

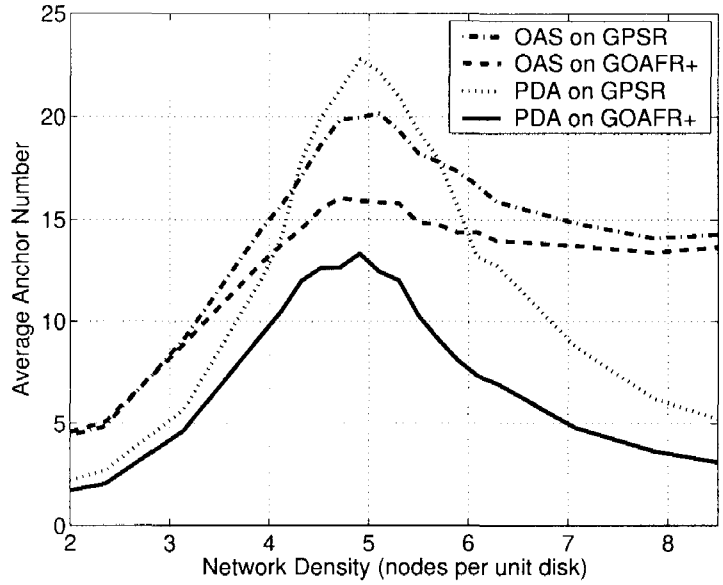


Figure 3.7. Comparison of average numbers of anchors of the proposed PDA algorithm and the OAS algorithm, when both are applied to GPSR and GOAFR$^+$.

When both PDA and OAS are applied to GOAFR$^+$, the average number of anchors of PDA is always smaller than that of OAS. Moreover, a large margin between them is seen. With GPSR, the average number of anchors of PDA is much smaller than that of OAS at all network densities except between 4 and 6, where PDA has a longer anchor list than OAS does (the biggest difference between PDA and OAS is only 3). Under critical densities $(4, 6)$, irregular faces occur frequently. Projection distance fluctuates very often and more nodes are added to the anchor list. That is probably why our algorithm has a longer anchor list than OAS in average in this density region. Compared with the reduction in relative path length, the difference here in the number of anchors is ignorable. In addition, in most density regions, PDA results in a shorter anchor list than OAS.

3.4.3 Computation and Storage Overhead

The OAS needs to calculate at least one angle whenever a new node is visited after the first node in detouring mode. While in PDA, only the projection distance is calculated. It is much easier than to calculate an angle because calcu-

lating an angle needs to use tangent function (it is more complex than common multiplication and division). Moreover, parameters of the line defined by source and destination nodes can be calculated at the first node in detouring mode and then be stored in the packet.

For OAS, the first packet needs to maintain a list of all nodes visited no matter whether it ever enters detouring mode. It is apparently a waste of storage space and bandwidth for many destinations can be reached in greedy mode only. Moreover, when the path is long, the packet size increases significantly. No extra cost is incurred in the proposed algorithm if the packet never enters detouring mode. Even if the packet entering detouring mode, decide whether to keep a node in the anchor list when visiting it. So the computation and storage overhead of PDA is smaller than that of OAS.

3.5 Simulation of the Lifetime-Improving Strategy

To validate the lifetime-improving strategy, compare two routing algorithms: PDA with and without random shift. In the following simulations, the total number of nodes generated in the square field ranges from 500 to 1000, which corresponds to network densities ranging from 3.9 to 7.85 nodes per unit disk of area π. For a given network density, 200 realizations of network graphs are generated. In each case, randomly choose a number of connected source-destination pairs in each graph, and each source tries to send ten packets to its corresponding destination. For simplicity, assume that each node has an initial energy of 100 units, and one transmission (i.e., forwarding a packet once) costs one energy unit. If a node runs out of energy, it is unusable anymore. If a node is unusable, an alternate routing has to be found. In [35], a variety of definitions of network lifetime are discussed. Here assume that the network lifetime ends when 20% or more nodes in the network exhaust their energy. For the PDA algorithm with the lifetime-improving strategy, the random shift of anchors follows a two-dimensional Gaussian distribution with a standard deviation of one unit distance.

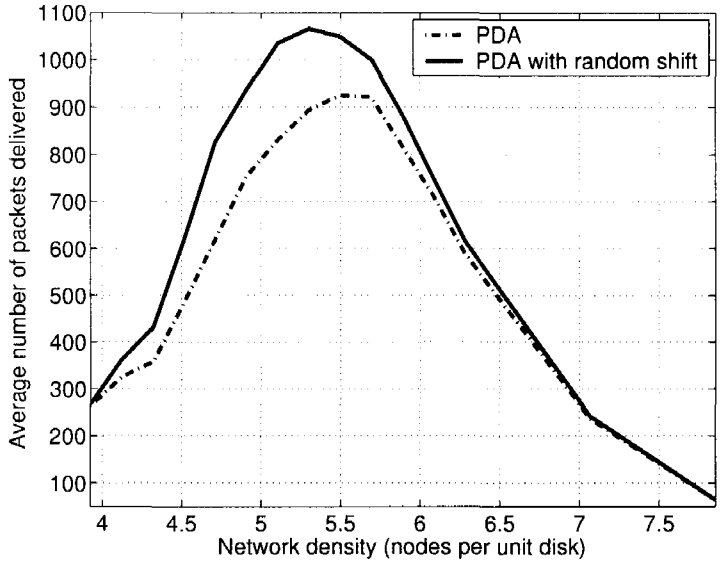


Figure 3.8. Average number of packets delivered using PDA with and without random shift.

3.5.1 Comparison of Network Lifetime

In each network graph realization, randomly choose source-destination pairs as long as the network is alive, and each source sends ten packets to its destination. Record the number of packets delivered in the lifetime of the network. The results are plotted in Figure 3.8. As the network density increases, the number of packets delivered increases first and then decreases. Figure 3.8 shows that the proposed strategy can increase the number of packets delivered in a critical density region of 4$\sim$6 nodes per unit disk, which is the density region where there is a high probability of detouring mode involved in the routing as shown in [9].

It is interesting that there are fewer packets delivered in high network densities. At high network densities, there is a high probability that only greedy forwarding is involved in the routing [9], and thus the proposed strategy has no effect due to the fact that PDA only builds an anchor list when the detouring mode is involved. In this case, it is likely that some nodes are used more frequently for greedy routing and thus the network lifetime is shorter than the case of density region of 4$\sim$6 nodes per unit disk.

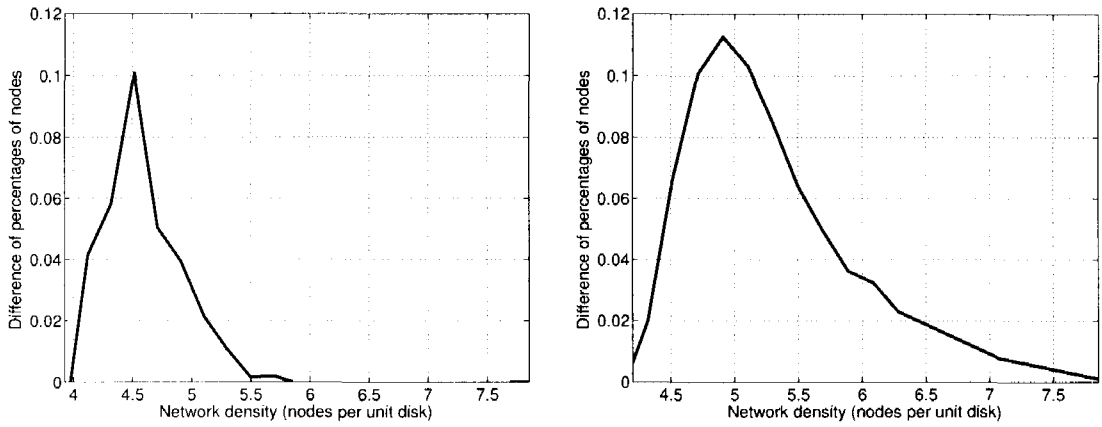


Figure 3.9. Difference of percentages of nodes with a certain energy level after 1000 packets delivered using PDA with and without random shift: (a) below 30 units; (b) above 70 units.

3.5.2 Comparison of Energy Consumption

This test compares the percentage of nodes of certain energy levels after a fixed number of packets delivered. For each network graph realization, randomly choose 100 source-destination pairs and each source sends 10 packets to its destination. If a source and destination link becomes unconnected before 10 packets are delivered, generate more source-destination pairs to make sure that the total number of packets delivered is 1000. After 1000 packets are delivered, we counted the percentage of nodes in network with an energy level below 30 units, and those with an energy level above 70 units. Figure 3.9 (a) shows the average difference of percentage of nodes with an energy level below 30 units, where the difference is defined as the percentage of nodes for PDA minus that for PDA with random shift. Figure 3.9 (b) shows the average difference of percentage of nodes with an energy level above 70 units, where the difference is defined as the percentage of nodes for PDA with random shift minus that of PDA. It can be seen from Figs. 3.9 (a) and Figure 3.9 (b) that the proposed strategy of random shift can reduce up to 10% of nodes with energy level below 30 units, and it can increase up to 11% of nodes with energy level above 70 units. Again, the most significant improvement

appears around the critical density region of 4~6 nodes per unit disk. In summary, the results show that the proposed strategy is effective in prolonging the lifetime of the networks.

3.5.3 Average Number of Packets per Source-Destination

In this test, fix the network density to five nodes per unit disk. Randomly choose source-destination pairs for each graph realization. The number of packets sent by each source to its destination ranges from 2 to 30. Then count the total number of packets delivered before 10% of nodes in the network exhaust their energy.

Figure 3.10 depicts the average number of packets delivered using PDA with random shift and without random shift. It can be observed that PDA with random shift improves the network lifetime for a broad range of numbers of packet per source-destination link. Note that when the number of packet sent by each source is only one, PDA with random shift is the same as PDA without random shift, since anchors are only used from the second packet (assume the anchor list has been sent back to the source node when the second packet is generated).

3.6 Conclusion

A simple yet efficient method is proposed to obtain an anchor list, which is based on the projection distance of nodes on the path of the first packet with respect to the virtual line linking the source and destination. Use local minimum nodes as anchors. The PDA algorithm can be applied to any non-flooding geographic routing protocols to shorten the routing path. Simulations show that the proposed algorithm's average path length is much shorter than that of OAS, and fewer anchors are used in the proposed approach than in OAS for most network density regions. Moreover, the computation involved in PDA is less complex than that of existing anchor-based schemes. Furthermore, to alleviate uneven distribution of energy consumption, dynamically change the anchors for every packet sent

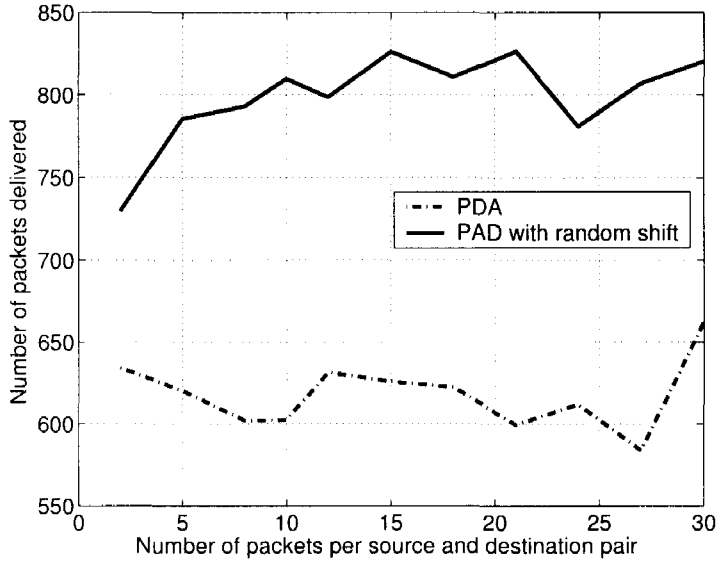


Figure 3.10. Total number of packets delivered versus the number of packets per source-destination pair.

by introducing a random shift to each anchor. Since the shift is randomly generated, different packets may be routed with different paths, and thus avoiding anchors being overly used. This allows energy consumption to be more evenly distributed in the network. Simulation results show that the proposed strategy improves the network lifetime, especially in a critical density region of 4~6 nodes per unit disk, which is the density region where the detouring mode is most likely involved in geographic routing.

CHAPTER 4

LOCATION-BASED MULTICAST

So far, our discussion focuses on unicast. Multicast routing poses special challenges compared with unicast routing. To minimize the total number of transmissions used in multicasting, it is critical to decide when to send a packet through different paths. If a packet is sent to each multicast destination separately, the multicast routing becomes unicast routings and it is often energy inefficient due to a large number of transmissions. Therefore, when to transmit a packet in multiple routes is of critical importance in the design of energy efficient multicast protocols. At the same time, because nodes are usually low-power with limited computation capacity, computation complexity must be taken into account in the design of multicast routing protocols. In this chapter, the objective is to design a multicast routing protocol that keeps the total number of transmissions small while being computationally efficient.

Various multicast routing protocols have been proposed [36–42]. For example, GMR (Geographic Multicast Routing) [42] is a heuristic neighbor selection scheme based on cost over progress metric. It preserves the good properties of geographic unicast routing while efficiently delivering multicast data packets to multiple destinations. It is a localized protocol without any type of flooding. It is shown in [42] that the computation complexity of GMR is $\mathcal{O}(\min(n^3, \ell^3)n\ell)$, where n is the number of neighbors of the current node and ℓ is the number of destinations associated with the current node.

This chapter proposes a novel multicast routing algorithm for wireless sensor networks, namely DCGM (Destination Clustering Geographic Multicast), by clustering destinations which can share the same next hop. Different from most

forward greedy routing [43, 44], where the current node chooses the node closest to the destination as the next hop for the packet corresponding to a destination in greedy mode, the proposed protocol considers all nodes which provide progress toward a destination in determining which node should be used as the next hop. Moreover, it considers not only destinations associated with the packet in greedy mode but also destinations associated with the packet in detouring mode to share their next hops.

It is proved that the complexity order of the algorithm is $\mathcal{O}(n\ell)$, which is much lower than that of GMR. Using simulation, it is shown that DCGM outperforms GMR in terms of average number of transmissions while reducing the computation complexity significantly. To further reduce the number of transmissions, apply k-means clustering strategy to GMR and DCGM. The strategy improves the performance of GMR and DCGM by dividing the destinations into many clusters and sending the packet first to the closest destination in each cluster, which then sends the packet to other nodes in the cluster. Simulation results show that the strategy can reduce the number of transmissions up to 35% percent.

The rest of the chapter is organized as follows. Section 4.1 provides a relevant literature review on multicast routing in wireless sensor networks. The DCGM algorithm is described and analyzed in Section 4.2. k-means is introduced to further improve the performance in 4.3. Section 4.4 compares DCGM with GMR by simulations.

4.1 Related Work on Multicast

Several multicast routing protocols have been proposed for ad-hoc and sensor networks. In DSM (Dynamic Source Multicast) [36], each node floods the network with information about its own position, therefore each node knows the position of all other nodes in the ad-hoc network. This approach can result in optimal multicasting but is not scalable, since it incurs high communication overhead when the number of nodes in the network is large.

DDM (Differential Destination Multicast) [37] uses position information to perform multicasting. Destinations are enclosed in the data packet and each node decides the next hop for each packet according to the distance of the destination measured in number of hops recorded in its unicast routing table. DDM is also not scalable as each node needs to maintain its unicast routing table for all possible destinations.

In [38], Mizumoto et al. proposed a protocol to send multicast messages to multiple geographic destination regions rather than to multiple destinations. It needs to calculate the cost of each possible set of subgroups and to maintain a routing table in each node. GGP (Geometry-driven Geocasting Protocol) is proposed to deal with geocasting with multiple targets in [39]. It uses Fermat points to construct a multicast tree. GGP works well in high-density sensor networks but not in low-density sensor networks. The PBM (Position Based Multicast) routing protocol proposed in [40] is a generalization of GFG (Greedy-Face-Greedy) routing to operate on multiple destinations. It builds a multicast tree. The shape of the tree may vary from the shortest path tree to an approximation of a minimum cost multicast tree according to different values of λ (λ is a weight parameter for two criteria in the protocol). However, it is difficult to determine the optimal value for λ and no method was given to calculate the optimal value. Furthermore, PBM needs to evaluate all possible neighbor subsets. The cost increases exponentially with the increase of the number of neighbors. SPBM (Scalable Position-Based Multicast) [41] is a hierarchy protocol based on group information. Each node needs to maintain a global member table containing entries for the three neighboring squares in each level from level 0 up to level $(L-1)$.

The following describes GMR [42] in more detail as an example of position based multicasting protocols. GMR defines a cost over progress metric. The cost is the number of selected neighbors. The progress measures the advance towards destinations.

The neighbor selection algorithm starts with a set of destinations

$$\{D_1, D_2, \cdots, D_\ell\}$$

in greedy mode. First, by grouping those destinations for which the same neighbor provides the most advance into the same subset, a set partition $\{M_1, M_2, \cdots, M_\iota\}$ is obtained. Each M_i has its own cost-progress ratio $\frac{C_i}{P_i}$ and the overall cost-progress ratio is

$$r = \frac{\sum_{i=1}^{\iota} C_i}{\sum_{i=1}^{\iota} P_i}, \tag{4.1}$$

where C_i is the cost of the subset M_i and it may vary according to different definitions of the cost. P_i is the total progress made by nodes in the subset M_i.

GMR attempts to find the set partition with the optimal ratio r. To do that, it tests all pairs (M_i, M_j) and finds the pair with the best reduction in cost-progress ratio, and merges these two sets to obtain a new set partition. The above procedure is repeated till there is no more reduction in cost-progress ratio. The packet is forwarded according to the final set partition.

For packets in detouring mode, GMR checks whether the current node is closer to any of the destinations than the node where the multicast perimeter mode started. If so, the destination is removed from destination list for which the packet is forwarded in detouring mode and is added to the destination list for which the packet is forwarded in greedy mode. Otherwise, the next hop is determined by the right-hand or left-hand rule. If multiple destinations use the same next hop, the packet is forwarded to the next hop with one transmission.

4.2 Destination Clustering Geographic Multicast

The GMR algorithm only applies when the packet is in greedy mode, but not in detouring mode. Let look at an example. In Figure 4.1, node S is to send a packet to destinations $\{J, K\}$. S first forwards the packet to A since A is the only neighbor of S. A forwards the packet to B. For destination K, the packet can not be routed in greedy mode any more. Suppose that the right-hand rule is used to

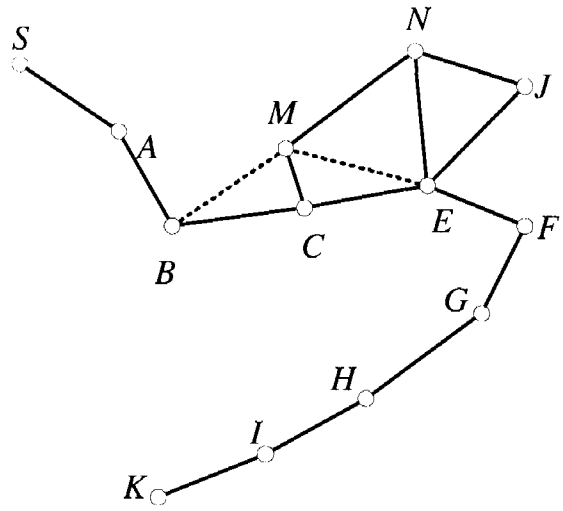


Figure 4.1. A multicast routing example. Node S is the source and J, K are the destinations.

explore the face, the packet for K is forwarded from B to C. For destination J, the packet can continue to be routed in greedy mode. According to GMR, B forwards the packet for J to node M. However, if the packet for J is also forwarded to node C, the packet for the two destinations (J and K) can share the following path CE. When the packet arrives at E, it can be sent through EJ to reach the destination J. The packet for destination K is routed with the right-hand rule to F. This way, the total number of transmissions is reduced.

The following presents DCGM, the proposed neighbor selection algorithm to be used by the current node to decide which subset of its neighbors to forward the multicast packet. This algorithm considers all the destinations no matter the packet is in greedy mode or detouring mode.

4.2.1 The DCGM Algorithm

First, introduce the packet format used in our geographic multicast routing protocol. The packet format is similar to that described in [42]. The header contains the location of the sender and a list of fields. The fields contain the following information:

- Neighbor Location. It is the location of the forwarding neighbor for the destinations in the current field.

- A Flag. It indicates whether the packet for these destinations is in greedy mode or detouring mode.

- Destination Location. If the packet is in greedy mode, it may contain multiple destinations. If the packet is detouring mode, there is only one destination in it.

- Detouring Information. If the packet is in greedy mode, it is null. Otherwise, it contains the information needed for the detouring mode. It may vary according to different routing algorithms. For example, with GPSR, it contains information such as the location where the packet last enters detouring mode and the location where the packet enters the current face.

All destinations in greedy mode are stored in the same field if their next forwarding node is the same.

Given ℓ destinations $\{D_1, D_2, \cdots, D_\ell\}$, suppose that the current node is N_0 and the neighbors of N_0 are $\{N_1, N_2, \cdots, N_n\}$. Define an $n \times \ell$ matrix $\boldsymbol{P}$ with the (i,j)th element being

$$\boldsymbol{p}_{ij} = \begin{cases} 1, & \text{when the packet for } D_j \text{ is in greedy mode, and} \\ & N_i \text{ is closer to } D_j \text{ than } N_0, \\ 1, & \text{when the packet for } D_j \text{ is in detouring mode,} \\ & \text{and } N_i \text{ is the next hop for } D_j, \\ 0, & \text{otherwise,} \end{cases} \tag{4.2}$$

where $1 \leq i \leq n, 1 \leq j \leq \ell$. If the packet for destination D_j is in greedy mode, $\boldsymbol{p}_{ij}$ indicates whether we can use N_i as the next hop for destination D_j. If the same packet for destination D_j is in detouring mode, $\boldsymbol{p}_{ij}$ indicates whether N_i is the next hop for D_j by the right-hand or left-hand rule.

TABLE 4.1

The Initialization of $\boldsymbol{P}$ and $\boldsymbol{Q}$

```
1.    for j = 1 to ℓ          /*process each destination in the packet */
2.       if D_j is in greedy mode
3.          for i = 1 to n
4.             if N_i is closer to D_j than N_0
5.                p_ij = 1.
6.                q_ij = |N_0 D_j| − |N_i D_j|.
7.                P(i) = P(i) + 1.
8.                Q(i) = Q(i) + q_ij
9.             else
   /*By greedy routing, N_i can not be used as the next hop for destination D_j */
10.               p_ij = 0.
11.               q_ij = 0.
12.            end
13.         end          /*end for i = 1 to n*/
14.      else          /*D_j is in detouring mode*/
15.         decide the next hop N_h for D_j.
            /*By the left-hand or right-hand rule */
16.         for i = 1 to n
17.            q_ij = 0.
   /*In detouring mode, no neighbor can provide advancement for the destination*/
18.            if N_i is the next hop N_h for D_j
19.               P(i) = P(i) + 1.
20.               p_ij = 1.
21.            else
   /*By detour routing, N_i can not be used as the next hop for destination D_j */
22.               p_ij = 0.
23.            end
24.         end          /*end for i = 1 to n*/
25.      end
26.   end          /*end for j = 1 to ℓ*/
```

Define another $n \times \ell$ matrix $\boldsymbol{Q}$ with the (i,j)th element being

$$\boldsymbol{q}_{ij} = \begin{cases} |N_0 D_j| - |N_i D_j|, & \text{when the packet for } D_j \text{ is in greedy mode,} \\ & \text{and } N_i \text{ is closer to } D_j \text{ than } N_0, \\ 0, & \text{otherwise,} \end{cases} \tag{4.3}$$

where $1 \leq i \leq n, 1 \leq j \leq \ell$. $|AB|$ denotes the distance between node A and node B. We call $\boldsymbol{Q}$ the advancement matrix. If the packet for destination D_j is in greedy mode, $\boldsymbol{q}_{ij}$ means how much advancement the packet can make if N_i is used as the next hop for destination D_j. If the packet for destination D_j is in detouring mode, $\boldsymbol{q}_{ij}$ is always set to 0.

The algorithm to initialize $\boldsymbol{P}$ and $\boldsymbol{Q}$ is given in Table 4.1. In the initialization algorithm, calculate the number of destinations for which each neighbor can be used as the next hop, that is,

$$\boldsymbol{P}(i) = \sum_{j=1}^{\ell} \boldsymbol{p}_{ij}, 1 \leq i \leq n. \tag{4.4}$$

Similarly, calculate the advancement each neighbor can provide, i.e.,

$$\boldsymbol{Q}(i) = \sum_{j=1}^{\ell} \boldsymbol{q}_{ij}, 1 \leq i \leq n. \tag{4.5}$$

When a node receives a multicast packet, it first calculate $\boldsymbol{P}$ and $\boldsymbol{Q}$, as well $\boldsymbol{P}(i)$ and $\boldsymbol{Q}(i)$, for $i = 1, \ldots, n$. Note that to obtain $\boldsymbol{P}$ and $\boldsymbol{Q}$, a node only needs to know the location of its neighbors and the information contained in the packets as describe earlier. Next, the node executes the DCGM algorithm to select a subset of its neighbors to forward the packet, which is shown in Table 4.2. First, the neighbor with the maximal $\boldsymbol{P}(i)$, among $i = 1, \ldots, n$, is chosen. If more than one neighbor gives the same maximal value, the neighbor with the maximal $\boldsymbol{Q}(i)$ among the neighbors with the maximal $\boldsymbol{P}(i)$ is chosen. Suppose that the ith neighbor is selected. Then, all the destinations whose corresponding value in the ith row of $\boldsymbol{P}$ is 1, set their corresponding columns in $\boldsymbol{P}$ and $\boldsymbol{Q}$ to 0 and set their next hops to the selected neighbor. After $\boldsymbol{P}$ and $\boldsymbol{Q}$ are updated, another neighbor is selected

TABLE 4.2

The DCGM Algorithm

```
1.          repeat
2.              h = -1.
3.              maxNb = 0.
4.              advancement = 0.
    /* find the maximal P(i) and set h to the index of the maximal value */
5.              for i = 1 to n
6.                  if P(i) > maxNb
7.                      h = i.
8.                      maxNb = P(i).
9.                      advancement = Q(i).
10.                 else if P(i) = maxNb and advancement > Q(i)
11.                     h = i.
12.                     advancement = Q(i).
13.                 end
14.             end
    /* set corresponding columns in P of the assigned destinations for N_h */
15.             if maxNb > 0
16.                 for j = 1 to ℓ
17.                     if p_hj = 1
18.                         assign the next hop for D_j to N_h.
19.                         for i = 1 to n
20.                             P(i) = P(i) − p_ij.
21.                             Q(i) = Q(i) − q_ij
22.                             p_ij = 0.
23.                             q_ij = 0.
24.                         end
25.                     end
26.                 end
27.             end
28.         until maxNb = 0
```

as next hop using the same procedure, and this procedure is continued until all destinations have been assigned next hops.

Specifically, in line 2 of Table 4.2, initialize the index of the neighbor with the maximal $\boldsymbol{P}(i)$. In line 3, initialize the number of destinations that can be assigned to N_h. In line 4, initialize the advancement of neighbor N_h. In lines $5 \sim 14$, obtain the maximal $\boldsymbol{P}(i)$ and $\boldsymbol{Q}(i)$. After that, if the number of destinations that can be assigned to N_h is not zero, assign each destination whose corresponding value in row h of $\boldsymbol{P}$ is 1 to neighbor N_h. Update $\boldsymbol{P}(i)$ and $\boldsymbol{Q}(i)$, and set all these destinations' corresponding columns in $\boldsymbol{P}$ and $\boldsymbol{Q}$ to 0, as shown in lines $15 \sim 27$. This process is repeated. When the algorithm stops, there should be a next hop corresponding to each destination, which is proved in Section 4.2.3.

4.2.2 A Routing Example

Revisit the topology in Figure 4.1. Suppose that the current node is B and the destinations are $\{K, N, F\}$. B has three neighbors A, C, and M. Since there is no neighbor of B closer to destination K than B, the packet for destination K enters detouring mode. Use the right-hand rule in the planarized graph to determine the next hop for destination K, which is C. Therefore $\boldsymbol{p}_{AK} = 0$, $\boldsymbol{p}_{CK} = 1$ and $\boldsymbol{p}_{MK} = 0$. Because B is in detouring mode, $\boldsymbol{q}_{AK} = 0$, $\boldsymbol{q}_{CK} = 0$ and $\boldsymbol{q}_{MK} = 0$. For destination N, nodes C and M are closer to it than B. Therefore $\boldsymbol{p}_{AN} = 0$, $\boldsymbol{p}_{CN} = 1$, $\boldsymbol{p}_{MN} = 1$, $\boldsymbol{q}_{AN} = 0$, $\boldsymbol{q}_{CN} = |BN| - |CN|$, and $\boldsymbol{q}_{MN} = |BN| - |MN|$. Similarly, obtain the parameters associated with destination F.

Therefore, $\boldsymbol{P}$ and $\boldsymbol{Q}$ are initialized as

$$\boldsymbol{P} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}, \tag{4.6}$$

$$\boldsymbol{Q} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & |BN| - |CN| & |BF| - |CF| \\ 0 & |BN| - |MN| & |BF| - |MF| \end{bmatrix}. \tag{4.7}$$

Then obtain $\boldsymbol{P}(1) = 0$, $\boldsymbol{P}(2) = 3$ and $\boldsymbol{P}(3) = 2$, among which $\boldsymbol{P}(2)$ has the maximal value. From the second row of matrix $\boldsymbol{P}$, it can be seen that all neighbors can use C as the next hop. Therefore, set all columns in $\boldsymbol{P}$ to 0. When the outer loop (lines $2 \sim 27$) in Table 4.2 is executed once more, $\mathrm{maxNb} = 0$. The algorithm stops. The packet includes $\{K, N, F\}$ in the header and is sent to node C. Node C first initializes its $\boldsymbol{P}$ and $\boldsymbol{Q}$, then decides which neighbor to use as the next hop corresponding to each destination.

4.2.3 Discussion of DCGM

This subsection gives some properties of the proposed algorithm and analyzes its complexity.

Proposition 1 *For each destination, DCGM assigns one and only one neighbor of the current node as the next hop if the number of neighbors of the current node is nonzero.*

Proof: There are only two modes: greedy mode or detouring mode for the packet with respect to a destination. If the packet for a destination is in greedy mode, there is at least one neighbor closer to the destination than the current node (there is at least one 1 in the jth column of $\boldsymbol{P}$). For the case of detouring mode, the current node can definitely find a next hop because the number of neighbors of the current node is nonzero. If the number of neighbors is 1, the current node forwards the packet to its only neighbor. If the number of neighbors is greater than 1, the current node finds its first counterclockwise or clockwise neighbor using the right-hand or left-hand rule. Suppose that after the DCGM algorithm is executed, no neighbor forwards the packet for destination D_j. The only operation that may change the elements in the jth column is in lines 22 and 23. If D_j has not been assigned to any neighbor, the elements in the jth column do not change. maxNb is definitely greater than 0 and the DCGM algorithm will not stop. It contradicts the assumption. Therefore a next hop is assigned corresponding to each destination after the DCGM algorithm.

To prove that there is only one forwarding neighbor corresponding to each destination, it must be shown that at most one neighbor is assigned as the next hop of the current node to each destination. Suppose that for D_j, two neighbors N_i and N_m $(m \neq i)$ of the current node are assigned as the next hop by the DCGM algorithm. If the next hop is first assigned to N_i. Then, all elements in the jth column become 0 after lines $19 \sim 24$. No operation in the algorithm changes an element in $\boldsymbol{P}$ from 0 to 1, and hence the jth column stays unchanged. If there is another neighbor N_m that is also assigned as the next hop corresponding to destination D_j, $\boldsymbol{p}_{mj}$ must not be 0. It is a contradiction because all elements of the jth column of $\boldsymbol{P}$ are 0 after D_j is assigned to N_i. Therefore, the assumption that there are two or more neighbors assigned as the next hop for the packet corresponding to one destination is wrong. There is at most one neighbor assigned to the packet for each destination.

Proposition 2 *The outer loop of the DCGM algorithm executes at most* $\min(n, \ell) + 1$ *times.*

Proof: Here an outer loop is defined as an operation sequence from line 2 to line 27 in Table 4.2. First look at the case when $\ell \geq n$. After each outer loop except the last one, there is one neighbor assigned as the next hop corresponding to some destinations. When one neighbor is selected as the neighbor with the maximal $\boldsymbol{P}(i)$, all the elements of its corresponding row in $\boldsymbol{P}$ become 0. Therefore if the same neighbor is chosen once more, it must be the case that maxNb $= 0$ and the algorithm stops. That is the last loop of the algorithm. Hence if there are n neighbors, the outer loop may be executed at most $n + 1$ times.

If $\ell < n$, prove that the outer loop can be executed at most $\ell+1$ times. In each outer loop except the last one, maxNb > 0. At least one packet for a destination is assigned to a neighbor. As the previous proposition shows, corresponding to each destination, only one neighbor can be assigned as the next hop. Therefore the outer loop executes at most ℓ times with maxNb > 0. For the last loop, maxNb $= 0$. In

total the outer loop is executed at most $\ell + 1$ times.

Proposition 3 *The complexity of DCGM is $\mathcal{O}(n\ell)$, where n is the number of neighbors of the current node and ℓ is the number of destinations associated with the current node.*

Proof: It is not difficult to see that the complexity to initialize $\boldsymbol{P}$, $\boldsymbol{Q}$, $\boldsymbol{P}(i)$, $\boldsymbol{Q}(i)$, for $i = 1, \ldots, n$ is $\mathcal{O}(n\ell)$. For the DCGM algorithm, the outer loop executes at most $\min(n, \ell) + 1$ times. The complexity of lines $5 \sim 14$ is $\mathcal{O}(n)$. The complexity of the algorithm because of this part is $\mathcal{O}(\min(n, \ell)n)$. From the first execution of the outer loop (lines $2 \sim 27$) to the last execution of the outer loop, suppose that maxNb is $p_1, p_2, \cdots, p_m$ and 0 respectively ($m + 1$ is the number of times the outer loop is executed). As known from lines $17 \sim 25$, lines $18 \sim 24$ will be executed only if the next hop corresponding to destination D_j is assigned to the current selected neighbor N_h. Proposition 1 shows that for each destination, the next hop is assigned once and only once. Therefore, obtain

$$\sum_{i=1}^{m} p_i = \ell. \tag{4.8}$$

For each assigned next hop in an outer loop, lines $18 \sim 24$ is executed once and the number of operations is $4n + 1$. There are p_i assigned packets in the ith round and the total number of operations in the ith round of the outer loop is $p_i(4n + 1)$. The total number of operations in the algorithm because of lines $15 \sim 27$ is

$$\sum_{i=1}^{m} p_i(4n + 1) = 4n\ell + \ell. \tag{4.9}$$

It can be concluded that the complexity order of DCGM is $\mathcal{O}(n\ell)$.

4.3 k-means Clustering Strategy

Both GMR and DCGM perform inefficiently around network density of four nodes per unit disk. To reduce the number of transmissions, apply the k-means clustering strategy to GMR and DCGM. It divides the destinations into many clusters and sends the packet to the closest destination in each cluster first, which then sends the packet to other nodes in the cluster.

There are many geometric clustering algorithms such as k-means [45], fuzzy C-mean [46], hierarchical clustering [47] and mixture of Gaussians [48]. The k-means method is a well-known algorithm for its running speed and its simplicity. Given a set of n data points, the algorithm uses a local search approach to partition the points into k clusters. A set of k initial centers is chosen arbitrarily. Each point is then assigned to the center closest to it and the centers are recomputed as centers of mass of their assigned points. The above process is repeated till convergence. It is shown that no partition occurs twice during the course of the algorithm. Therefore, the algorithm is guaranteed to terminate.

With k-means, the source node divides the destinations into k clusters if the number of destinations is larger than k. If the number of destinations is less than k, each destination forms a cluster. In each cluster, choose the node closest to the source as the representative of the cluster. The source node first sends the packet to the representatives. The representative in each cluster then sends the packet to all other destinations in the cluster. From the source to all representatives and from each representative to all other destinations in the cluster, the packet is routed with DCGM or GMR.

For nodes other than the source node, if it is a representative of some cluster, remove it from the destination list and add other nodes in the cluster to the list. If a representative is unreachable, choose another representative for the cluster and the new representative is responsible for sending the packet to other destinations in the cluster. When the current node has no neighbor or there is no node closer to the destination than the current node itself after the packet traverses the boundary of a face, the current node determines that the destination is unreachable.

Look at an example. In Figure 4.2, node S is to send a packet to destinations $\{H, J, K\}$. S first forwards the packet to A since A is the only neighbor of S. A forwards the packet to B. For destination K, the packet is routed to L because L is the only neighbor of B closer to K. For destination J, the packet can use either C or M as the next hop. For destination H, the packet is routed to C because C

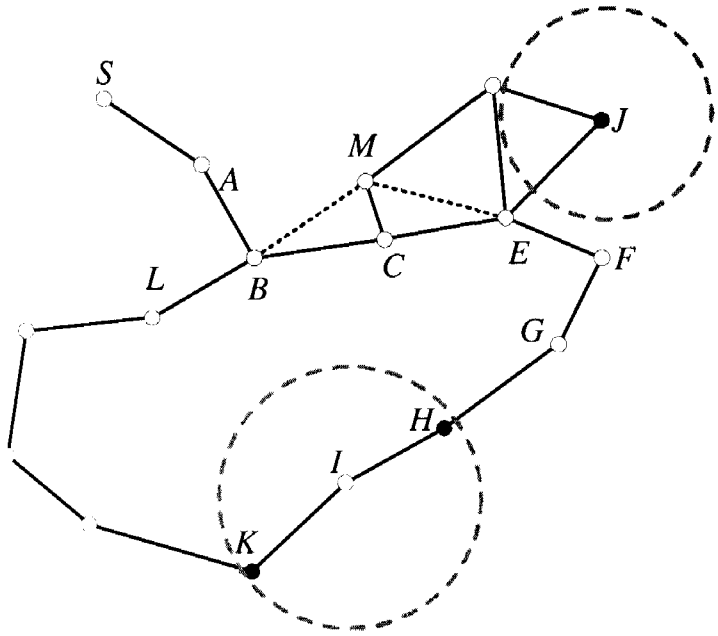


Figure 4.2. An example in which the clustering strategy can reduce the number of transmissions (S is the source and H, J, K are destinations).

is the only neighbor of B closer to H. DCGM uses C as the next hop for both H and J. After that, both the packet for K and the packet for H can not be routed in greedy mode any more. Suppose that the right-hand rule is used to explore the face. Because L is the first node in detouring mode for destination K, decide the next hop with respect to K. Therefore, the packet for K is forwarded from L to B. The following route is $\langle B, C, E, F, G, H, I, K\rangle$. Similarly for destination H, the route is $\langle C, E, F, G, H\rangle$, and for destination J, the route is $\langle C, E, J\rangle$. Clearly, the route for K and the route for H share some hops. But DCGM or GMR does not take advantage of that. With the clustering strategy, suppose the destinations are divided into two clusters. Then, H, K will be in one cluster and J will be in the other cluster. H is chosen as the representative of its cluster because H is the closer destination to S in the cluster. J is the only node in its cluster and it is the representative of the cluster. S sends the packet to H and J. The route for H is $\langle S, A, B, C, E, F, G, H\rangle$, and the route for J is $\langle S, A, B, C, E, J\rangle$. When the packet arrives at H, it is sent to K by H. The route is $\langle H, I, K\rangle$. It can be seen that by the

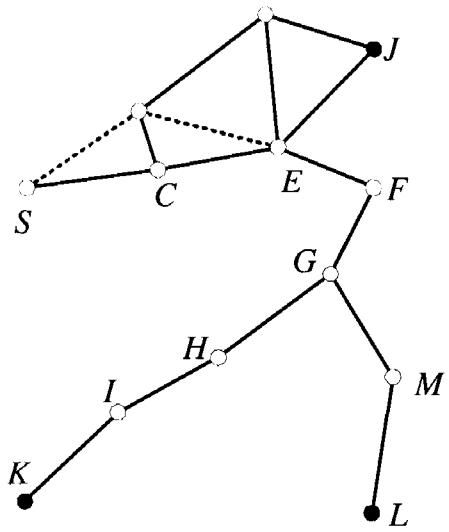


Figure 4.3. An example in which the clustering strategy can not reduce the number of transmissions (S is the source and J, K, L are the destinations).

proposed strategy the total number of transmissions is reduced.

4.3.1 Discussions

With the clustering strategy, the multicast task is divided into many small subtasks. Without this strategy, the number of destinations may be very large. After applying the k-means method to all destinations, the source node needs to send the packet only to k or fewer representatives. With the decreasing number of destinations, DCGM and GMR need less time to execute. Correspondingly, intermediate nodes also need to handle fewer destinations than before.

Although the worst-case running time of k-means is superpolynomial [49], it converges very quickly in practice using Lloyd's algorithm. Moreover, the clustering is executed only once at the source node. Compared with the performance improvement, the clustering strategy is a good trade-off between computation complexity and energy consumption.

In most cases, the proposed strategy can reduce the number of transmissions. Nevertheless, our strategy can not guarantee the improvement in the number of transmissions. Let look at an example shown in Figure 4.3. S is the source

node and J, K, L are the destinations. Without clustering, the packet is routed as follows: $\langle S, C, E, F(J), G, H(M, L), I, K\rangle$, where $F(J)$ means that E sends the packet to F, J with one transmission and $H(M, L)$ means that G sends the packet to H, M with one transmission and M sends the packet to L. The total number of transmissions is eight. Suppose $k = 2$. With k-means, we can divide the destinations into two clusters $\{K, L\}, \{J\}$. In the first cluster, K is chosen as the representative because it is closer to the source. In the second cluster, J is the representative. S sends the packet to K and J. They can share the first two hops C and E. When the packet arrives at E, E sends the packet to F and J at the same time. Since destination J has received the packet, the packet is sent to destination K by the following route: $\langle G, H, I, K\rangle$. Since K is the representative of cluster $\{K, L\}$, it sends the packet to L as follows: $\langle K, I, H, G, M, L\rangle$. The total number of transmissions is 12, which is larger than that without clustering (8 transmissions). If the number of destinations in a cluster is relative small, it may be better not to cluster them. This way it can increase the probability of sharing paths with other destinations.

4.4 Simulation Results

The simulation configuration is similar to that in Section 2.2. The total number of nodes generated in the square field ranges from 100 to 1300, which corresponds to network densities ranging from 0.79 to 10.3 nodes per unit disk of area π. For a given network density and a given number of destinations, 1000 realizations of network graphs are generated. For each realization, the source and destinations are randomly selected. The underlying geographic unicast routing protocol is GPSR.

4.4.1 Comparison of Average Number of Transmissions

The number of transmissions measures the efficiency of the paths selected and the amount of energy consumed. One transmission is one hop from a node to one or more of its neighbors. Use the number of transmissions to estimate the

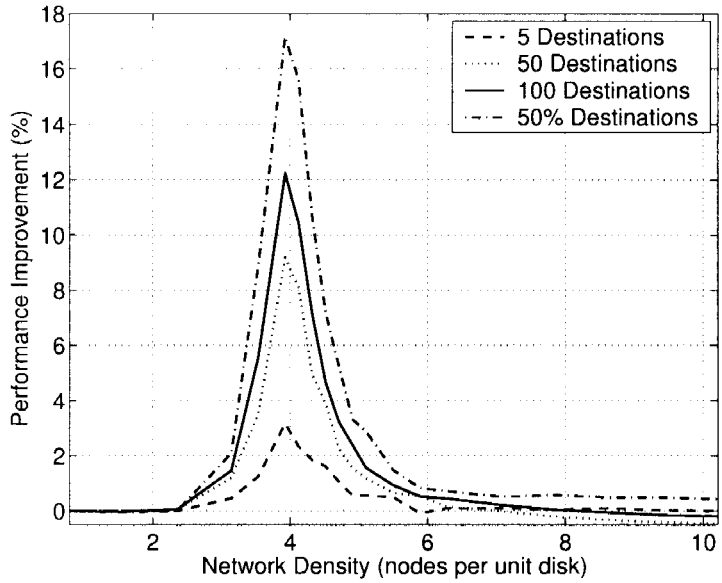


Figure 4.4. Percentage of performance improvement of DCGM over GMR with 5, 50, 100 and 50% destinations.

amount of energy consumed. The smaller the number of transmissions is, the less the network energy is consumed. Compare the average number of transmissions of GMR and DCGM.

For each network density, run 1000 realizations in each of four configurations: 5 destinations, 50 destinations, 100 destinations, 50% destinations (50% nodes in the network are destinations). For a given network density and a given number of destinations, in the ith realization, one packet is sent from the source to the destinations connected with the source. Suppose that the number of transmissions used by GMR is u_i, the number of transmissions used by DCGM is v_i and the number of destinations connected with the source is k_i. Define the difference of average number of transmissions as follows:

$$\tau = \frac{1}{t}\left(\sum_{i=1}^{t}\frac{u_i}{k_i} - \sum_{i=1}^{t}\frac{v_i}{k_i}\right). \tag{4.10}$$

where t is the number of realizations in which at least one destination is connected with the source. The percentage of performance improvement of DCGM over GMR is then obtained by normalizing τ with respect to the maximum of the

average number of transmissions of GMR over all densities. Figure 4.4 shows the simulation results. From the figure, it can be seen that the performance improvement increases first as the network density increases. It arrives at its peak value when the network density is around four nodes per unit disk. As the density continues to increase, the performance improvement decreases. Figure 4.4 shows that DCGM outperforms GMR in the density range of $3 \sim 5$ nodes per unit disk. This is because at this density range, there is a high probability of detouring as shown in [9]. It can also be seen that in general, the performance improvement increases with the increasing number of destinations.

4.4.2 Comparison of Average Energy Consumption

The previous test assumes equal energy consumption in each transmission. Here test the case where energy consumption varies as a function of the transmission distance. Use the same wireless communication model as that in [50]. A node may use different power levels for different multicast tasks. The energy needed for the communication between two nodes is d_{ij}^{α}, where node i and node j are within each other's communication range, d_{ij} is the distance between them, and α is a parameter that typically takes a value between 2 and 4. Here use $\alpha = 4$ as derived in the work performed by Rodoplu and Meng [51]. Normalize the energy in one transmission as follows:

$$e_{ij} = \frac{d_{ij}^{\alpha}}{r^{\alpha}}, \tag{4.11}$$

where r is the communication range. Therefore $0 < e_{ij} \leq 1$.

For a given density and number of destinations, suppose that the total energy consumed in the ith realization by GMR is x_i and the total normalized energy consumed by DCGM is y_i. Define the difference of average energy consumption as follows:

$$\zeta = \frac{1}{t}\left(\sum_{i=1}^{t}\frac{x_i}{k_i} - \sum_{i=1}^{t}\frac{y_i}{k_i}\right). \tag{4.12}$$

The percentage of energy saving of DCGM over GMR is obtained by normalizing ζ with respect to the maximum of the average energy consumption of GMR over all

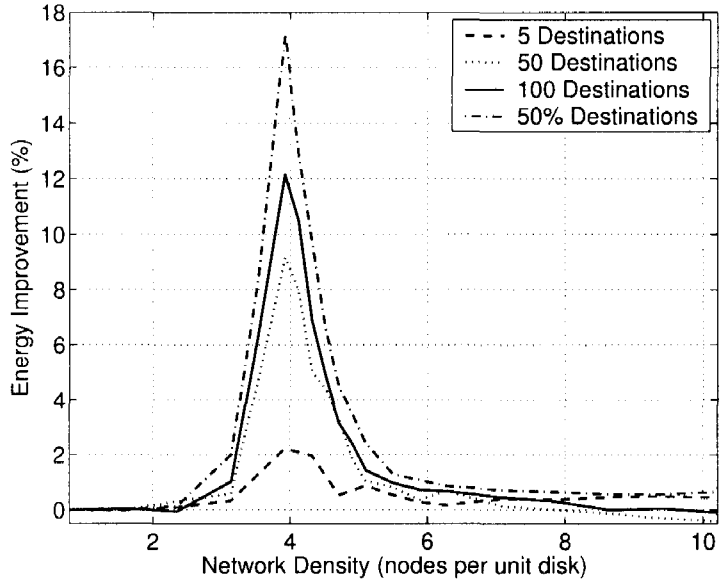


Figure 4.5. Percentage of energy saving of DCGM over GMR with 5, 50, 100 and 50% destinations.

densities. Figure 4.5 shows the simulation results. It can be seen that again DCGM outperforms GMR when there is a high probability of detouring.

For the jth network density, define

$$\sigma_j = \frac{1}{t}\sum_{i=1}^{t}\frac{x_i}{k_i}. \tag{4.13}$$

The energy improvement of DCGM over GMR is defined as

$$\frac{\zeta}{\max_{j\geq 1}(\sigma_j)}. \tag{4.14}$$

For each network density, run 1000 realizations. Four configurations are simulated: 5 destinations, 10 destinations, 25 destinations, 50% destinations.

4.4.3 Comparison of Computation Complexity

To compare the complexity of the proposed algorithm with that of GMR, choose 10% nodes in the network as destinations. For each network, run the first-hop routing 100 times. By first-hop routing, it means that only the source node's forwarding neighbors are computed. For each network density, 1000 networks

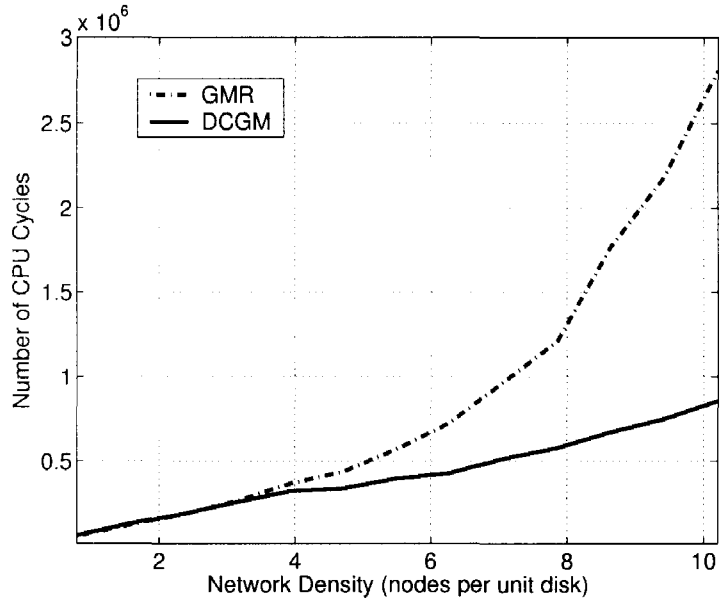


Figure 4.6. Comparison of computation complexity of GMR and DCGM when 10% nodes are randomly chosen as destinations in the network.

are generated. Calculate the average number of CPU cycles used for 100 first-hop routings. The simulation results are shown in Figure 4.6. From the figure, it can be seen that the number of CPU cycles used by DCGM changes slowly. The number of CPU cycles used by GMR increases significantly as the network density increases. For example, when the network density is about ten nodes per unit disk, the number of CPU cycles used by GMR is three times of that used by DCGM. When the network density is lower than four, it can be seen that the number of CPU cycles used by DCGM is almost the same as that used by GMR.

To see how computation complexity changes as the number of destinations changes, fix the network density to seven and change the number of destinations from 5 to 95. For each number of destinations, generate 1000 realizations and average the number of CPU cycles used for all realizations. In each realization, perform 100 first-hop routings with fixed source and destinations. The simulation results are shown in Figure 4.7. The result confirms that DCGM has much lower computation complexity than GMR.

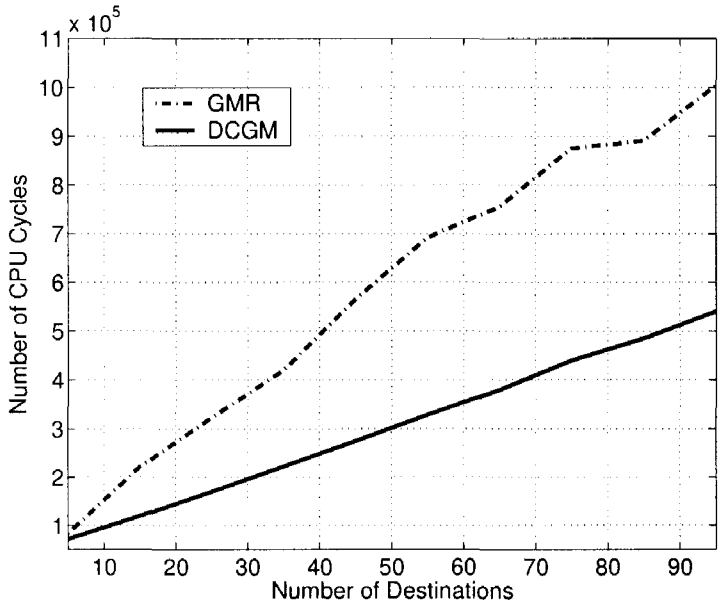


Figure 4.7. Comparison of computation complexity of GMR and DCGM with different numbers of destinations when the network density is seven nodes per unit disk.

4.4.4 Comparison of the Average Number of Transmissions With and Without k-means

The number of transmissions measures the efficiency of the paths selected and the amount of energy consumed. One transmission is one hop from a node to one or more of its neighbors. The smaller the number of transmissions is, the less the energy is consumed. To evaluate the effect of the clustering strategy, run simulations for DCGM and GMR with and without clustering. Use DCGM$^+$ to denote DCGM with clustering and GMR$^+$ to denote GMR with clustering. For the same realization of a network, use the same source and the same set of destinations to route a packet with GMR, DCGM, GMR$^+$ and DCGM$^+$ respectively. Only send a packet to the destinations connected with the source.

For each network density, run 1000 realizations in each of two configurations: 50 destinations, 50% destinations (50% nodes in the network are destinations). The number of clusters k is fixed at 12 unless otherwise specified.

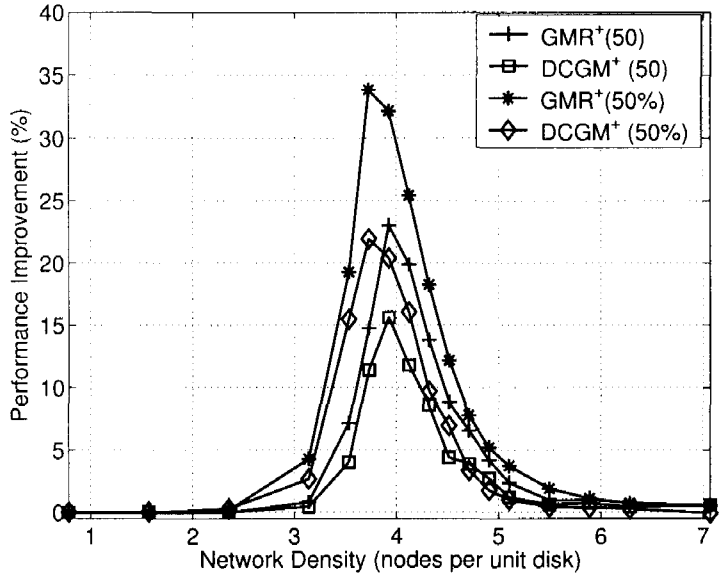


Figure 4.8. Comparison of performance improvement of GMR$^+$, DCGM$^+$ when the number of destinations is 50 and 50%.

For a given network density and a given number of destinations, in the ith realization, one packet is sent from the source to the destinations connected with the source. Suppose that the number of transmissions given by GMR is u_i, the number of transmissions given by GMR$^+$ is v_i, and the number of destinations connected with the source is ℓ_i. Define the difference of average number of transmissions as follows:

$$\tau = \frac{1}{t}\left(\sum_{i=1}^{t}\frac{u_i}{\ell_i} - \sum_{i=1}^{t}\frac{v_i}{\ell_i}\right). \tag{4.15}$$

where t is the number of realizations in which at least one destination is connected with the source. The percentage of performance improvement of GMR$^+$ over GMR is then obtained by normalizing τ with respect to the maximum of the average number of transmissions of GMR over all densities. Similarly, obtain the performance improvement of DCGM$^+$ over DCGM.

Figure 4.8 shows the result of performance improvement. It can be seen that the performance improvement increases with the increasing density first and arrives at the maximum at around network density of four nodes per unit disk. It decreases afterwards. Since there is a high probability of face routing around

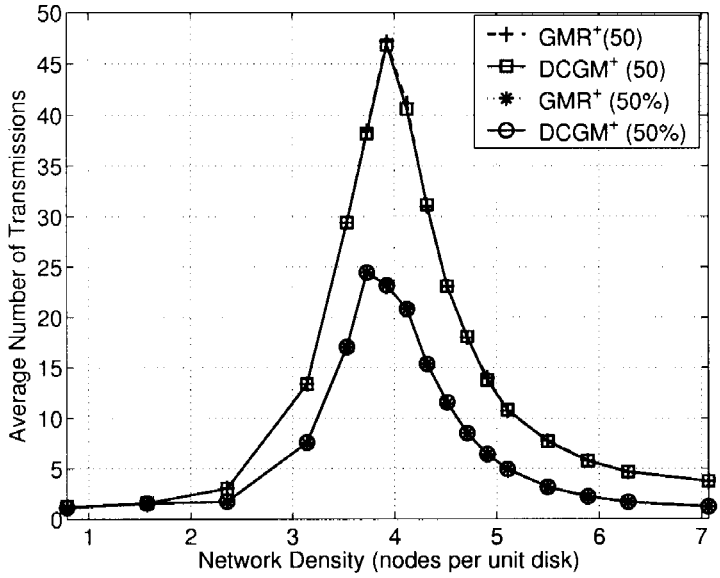


Figure 4.9. Comparison of the average number of transmissions of GMR^+ and $DCGM^+$ when the number of destinations is 50 and 50%.

this density, clustering can save some repeated face routings by routing the packet to the representative in the cluster first. Therefore, the improvement around network density of four nodes per unit disk is the biggest. When the density is around four nodes per unit disk, the performance improvement increases with the increasing of the number of destinations. The improvement is up to 35%. In Figure 4.9, compare the average number of transmissions under different densities and different numbers of destinations. It can be seen that GMR^+ obtains almost the same average number of transmissions with $DCGM^+$ under the same density and the same number of destinations. At the same network density, the average number of transmissions decreases with the increasing of the number of destinations. It is probably because the destinations are distributed more evenly when the number of destinations is increased. For each cluster, there are enough destinations to share the path from the source to the representative.

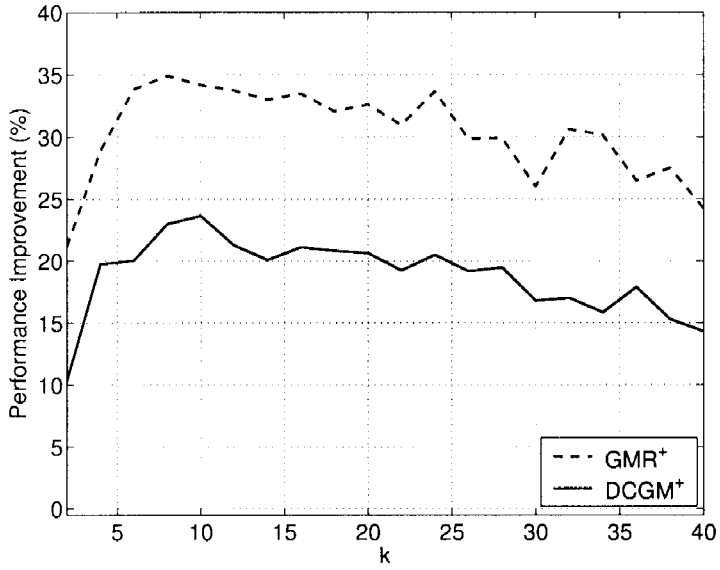


Figure 4.10. Comparison of performance improvement of GMR$^+$, DCGM$^+$ under different k when the number of destinations is 200.

4.4.5 Effects of the Number of Clusters

The relationship between performance improvement and cluster number k was also studied. Fix the network density to 4.1 nodes per unit disk and the number of destinations to 200. The number of clusters ranges from 2 to 40. For each case, run 1000 realizations. The simulation results are shown in Figure 4.10. Here τ is normalized with respect to the average number of transmissions of GMR at the corresponding number of destinations, so is DCGM. It can be seen that the performance improvement increases first and clustering achieves the highest performance improvement when k is around 10. It decreases afterwards. Therefore, k should be set around 10. At the same k, GMR$^+$ obtains higher improvement than DCGM$^+$.

4.4.6 Effects of the Number of Destinations

How the performance changes when the number of destinations varies was studied. The same configuration as section 4.4.5 is used. Here vary the number of destinations instead of cluster number k. The number of destinations ranges from

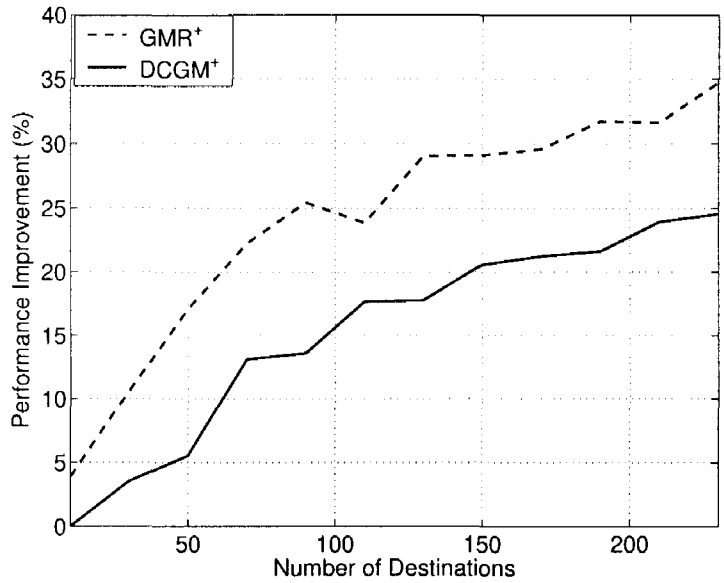


Figure 4.11. Comparison of performance improvement of GMR$^+$, DCGM$^+$ under different numbers of destinations when the number of clusters k is 12.

10 to 230 and k is fixed to 12. The simulation results are shown in Figure 4.11. It can be seen that the performance improvement increases with the increasing of the number of destinations.

4.4.7 Effects of the Average Size of Clusters

The average size of clusters is defined as the ratio of the number of destinations to k, which is denoted by ϑ. For each realization, adjust k according to the number of destinations connected with the source node. Suppose the number of connected destinations is ω, then $k = \lceil \omega/\vartheta \rceil$. The simulation configuration is similar to that of section 4.4.5. Use four different average sizes of cluster: 3, 6, 24, 48. Given ϑ and the number of destinations, we simulate 1000 realizations. The simulation results are shown in Figure 4.12. It can be seen that the performance improvement at any cluster size increases when the number of destinations increases. In the simulation, it can be observed that at any given number of destinations except when the number of destinations is small, the performance improvement increases first with the increasing of the average size of clusters and arrives at the

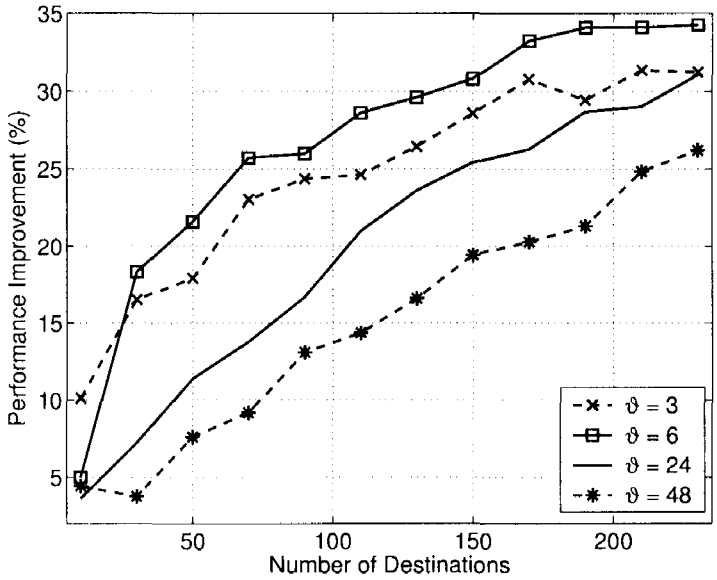


Figure 4.12. Comparison of performance improvement of GMR$^+$ under different average sizes of clusters.

peak when $\vartheta = 6$. It decreases afterwards. When the number of destinations is ten, there is no difference between the improvements with $\vartheta = 6, 24, 48$, while the performance improvement is higher when $\vartheta = 3$. This is because there is a lower probability of forming a cluster when ϑ is high.

4.5 Conclusion

To reduce the computation complexity of the existing geographic multicast routing protocols, a novel multicast algorithm is proposed, DCGM, which clusters destinations that can use the same next hop. It reduces the computation complexity significantly while obtaining better performance than existing protocols in terms of average number of transmissions. To further reduce the number of transmissions, apply clustering strategy to GMR and DCGM. It improves the performance of GMR and DCGM by dividing the destinations into many clusters and sending the packet first to the closest destination in each cluster, which then sends the packet to other nodes in the cluster. Simulation results show that the strategy can

reduce the number of transmissions up to 35% percent.

CHAPTER 5

LIFETIME-AWARE GEOGRAPHIC ROUTING

In previous chapters, the main objective is to find some short paths between the source and the destination so that the total number of hops is small. This chapter proposes a routing algorithm that balances the energy consumption of all nodes to prolong the lifetime of the network.

Most application scenarios for sensor networks involve battery-powered nodes with limited energy resources. Recharging or replacing the sensor battery may be difficult, or even impossible in harsh working environments. How to keep the network alive as long as possible is becoming a challenge in designing sensor networks. This can be approached from different layers [52] such as MAC layer, network layer and transport layer. In this chapter, this issue is addressed from the network layer. Network lifetime is prolonged by improving the routing protocols.

Park and Sahni [53] prove that the problem of routing packets in a wireless sensor network so as to maximize network lifetime is NP-hard. Many heuristic algorithms have been proposed to maximize network lifetime. Most of the algorithms are either centralized algorithms or need to make extensive use of broadcasting. Centralized algorithms are not scalable and algorithms using broadcasting waste a lot of energy in exchanging information between sensor nodes.

In wireless sensor network routing, high link reliability independent of distance within the physical radio range is commonly assumed. A statistical link layer model is developed in [54] and it shows the existence of a large "transitional region" where the link quality has high variance, including both good and highly unreliable links. The existence of such unreliable links would result in a high packet drop rate, resulting in drastic reduction of delivery rate or increased

energy wastage if retransmissions are employed.

How to prolong the lifetime of wireless sensor networks using geographic routing with a realistic link layer model remains largely unexplored. In [55], three energy-aware geographic forwarding schemes are proposed to improve the network lifetime by considering the residual energy of neighbors in deciding the next hop. It assumes an ideal link layer. In [16], the authors use a realistic link loss model to analyze an effective forwarding distance for both ARQ (automatic repeat request) and No-ARQ scenarios. Simulations show that the product of the packet reception rate (PRR) and the distance traversed towards destination is the optimal forwarding metric for the ARQ case and is a good metric even without ARQ. It minimizes the energy consumed per packet. If all the packets are routed through the best path, the nodes in that path will die quickly while other nodes remain intact. The network lifetime is seldom maximized with this metric.

In this chapter, a realistic link layer model is applied to wireless sensor networks and a new geographic routing algorithm is proposed to prolong the lifetime of wireless sensor networks. Define a function of four important factors: PRR (Packet Reception Rate), forwarding history, progress and remaining energy. With this function, evaluate each neighbor and forward a packet to the neighbor with the optimal value. Simulation results show that the proposed algorithm can usually double the lifetime of wireless sensor networks compared with PRR*D in [16].

The rest of the chapter is organized as follows: review related works in Section 5.1. Four important factors in geographic routing are given and the proposed algorithm is given in Section 5.2 and it is verified in Section 5.3. Conclusion is drawn in Section 5.4.

5.1 Existing Work

The lifetime of a network can be improved through awake/asleep schedule like [56]. However, our focus here is to prolong the lifetime of a network by improving the routing scheme. Moreover, any routing algorithm can be combined

with the awake/asleep schedule.

Many energy-efficient routing algorithms [52, 57–61] have been developed to maximize the lifetime of wireless ad-hoc and sensor networks.

Five metrics are proposed in [52]. They are: using a minimum-energy path (a path whose sum of the edge weights is minimum), maximizing time to network partition, minimizing variance in node energy levels, minimizing the node cost of each transmission (the cost of a node is some function of the energy used so far by that node) and minimizing maximum node cost. Only the first metric and the fourth metric have been implemented. It did not use a realistic link layer model in the performance evaluation. [57] considers only link cost and the advance a neighbor can make when the current node selects next hop. The algorithms proposed in [58–61] are either centralized or need to make extensive use of broadcasting. They are not suitable for geographic routing.

In [62, 63], maximizing lifetime is identified as a linear programming problem. Nevertheless, it needs knowledge of the rate at which each node generates information. The algorithm based on connected dominating sets [64] needs to maintain the dominating sets. The clustering-based algorithm [65] needs to maintain the clusters. Both incur high overhead.

Although there are many papers addressing the lifetime of wireless sensor networks, there are only a few papers discussing the lifetime of wireless sensor networks with geographic routing. PTKF (Partial Topology Knowledge Forwarding) [66] is proposed to improve the routing efficiency of current forwarding rules. Every node has its own knowledge range and calculates the path to the destination based on a shortest weighted path algorithm (such as Bellman-Ford's). It does not use any realistic link model and does not consider the lifetime of network.

Stojmenovic and Lin [67] present a power-cost efficient routing algorithm in geographic routing. Each node decides the next hop based on the estimation of the power needed to send the packet to the destination and remaining lifetime of neighbors.

GEAR (Geographical and Energy-Aware Routing) [68] attempts to balance the energy consumption across all its neighbors by minimizing the learned cost value of its neighbors. The learned cost is calculated with the distance to the target region and the consumed energy. In [69], the authors propose several greedy routing algorithms for ad hoc networks based on a realistic physical layer model. Simulation results show the performance is close to the shortest path algorithms under high densities.

A statistical link layer model is developed in [54] and it shows the existence of a large "transitional region" where the link quality has high variance, including both good and highly unreliable links. The existence of such unreliable links would result in a high rate of packet drops, resulting in drastic reduction of delivery rate or increased energy wastage if retransmissions are employed. Couto *et al.* [70] consider the transitional region and introduce a new metric ETX (expected number of transmissions) to improve the throughput.

5.2 Lifetime-Aware Geographic Routing

5.2.1 Problem Formulation

Assume that all nodes have the same maximal communication range. Every sensor has the same initial energy and no energy is consumed during packet reception.

A routing request is a pair (v_s, v_t), where v_s is the source node for the packet that is to be routed and v_t is the destination node. Let $r_1, r_2, \cdots (r_i = (v_{si}, v_{ti}))$ be a sequence of routing requests. The lifetime of a network is the maximum j before the network dies such that routing requests $r_1, \cdots, r_j$ are successfully routed.

There are four important factors that must be taken into account when designing a lifetime-aware routing protocol. The following discusses these factors one by one.

5.2.2 Packet Reception Rate

Use the link layer model derived in [54]:

$$PRR(d) = \left(1 - \frac{1}{2}e^{-\frac{\gamma(d)}{2\times 0.64}}\right)^{\rho 8f}, \tag{5.1}$$

where d is the transmitter-receiver distance, γ is the signal to noise ratio (SNR), ρ is the encoding ratio and f is the frame length. This particular equation resembles a MICA2 mote. Since the mote uses non-coherent frequency shift keying as the modulation technique and Manchester as the encoding scheme, $\rho = 2$. f usually takes the value 50. $\gamma(d)$ can be calculated as follows:

$$\gamma(d)_{dB} = P_{tdB} - PL(d)_{dB} - P_{ndB}, \tag{5.2}$$

where P_n is the noise floor,

$$PL(d) = PL(d_0) + 10nlog_{10}\left(\frac{d}{d_0}\right) + X_\sigma, \tag{5.3}$$

d_0 is a reference distance, and n is the path loss exponent and X_σ is a zero-mean Gaussian random variable (in dB) with standard deviation σ. $PL(d_0)$ can be calculated for a distance of $d_0 = 1$ meter. $PL(d_0) = 55$ dB. Here set the environmental parameters to $n = 3.0$ and $\sigma = 3.8$.

Figure 5.1 shows samples from the link layer model. Here the distance is normalized with respect to the maximal communication range. It can be seen that when the distance between two nodes is small (< 0.2), the PRR (Packet Reception Rate) is almost one. When the distance is greater than 0.8, PRR is almost zero. From 0 to 0.2, it is called the connected region. From 0.2 to 0.8, it is called the transitional region.

When choosing the next hop for the current node, the energy should be consumed as little as possible. Since the energy consumed for each hop is assumed to be the same, the number of retransmissions should be as small as possible. The exact number of retransmissions needed to send a packet successfully with only PRR can not be obtained. Use the expected number of retransmissions to estimate

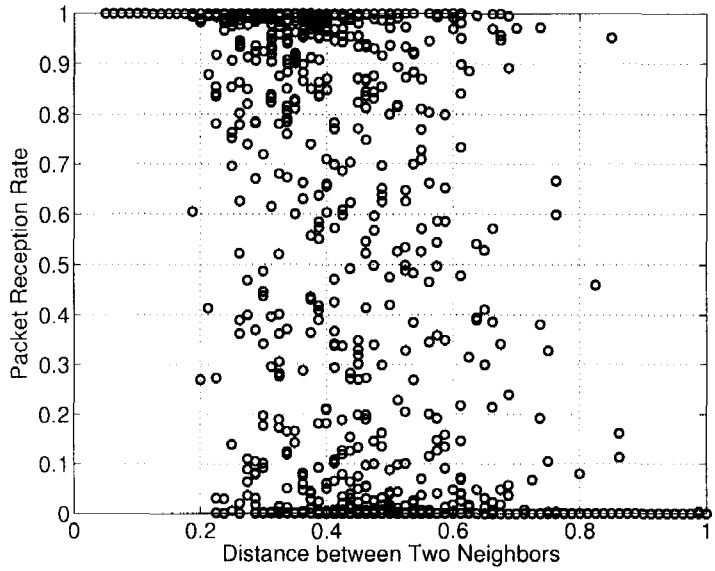


Figure 5.1. Samples from a realistic link loss model with connected region 0 to 0.2 and transitional region 0.2 to 0.8.

the amount of energy consumed. Use x to denote the number of transmissions to send a packet successfully (the number of retransmissions is $x - 1$).

For the ARQ (automatic repeat request), the expectation of x can be calculated as follows:

$$E(x) = \sum_{x=1}^{\infty} x * PRR(1 - PRR)^{x-1} = \frac{1}{PRR} \tag{5.4}$$

Therefore the bigger the PRR is, the smaller the number of transmissions is.

5.2.3 Forwarding History

Another factor is to balance the outgoing traffic out of a node. Let us look at an example in Figure 5.2. Suppose node S needs to forward a packet to destination D. The packet may come from S's neighbors or originate from S. The forwarding history is as follows: $(A, 3), (B, 6), (C, 2), (E, 5), (F, 4)$. The first element in each pair is a neighbor and the second element is the number of packets the current node has forwarded to this neighbor. To balance the traffic among different directions, prefer the neighbors with small numbers of packets forwarded to them by the current

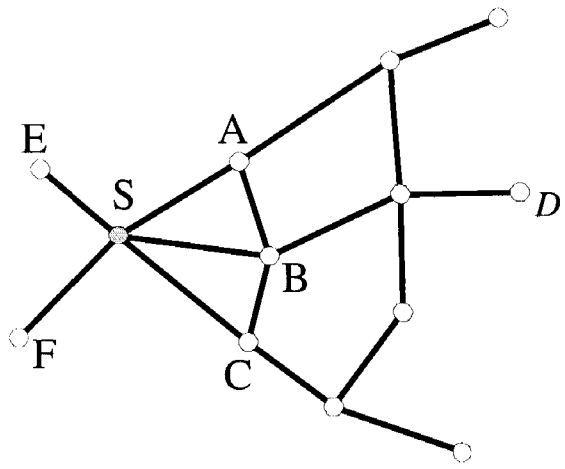


Figure 5.2. An example to decide the next hop based on the forwarding history, S is the current node and D is the destination node.

node.

Use N to denote the current node and N_i $(1 \leq i \leq m)$ to denote the neighbors of N, where m is the number of neighbors. Define the forwarding portion of a neighbor N_i as the ratio of the number of packets forwarded to N_i by N to the number of packets forwarded by N. Use $\delta(N_i)$ to represent the forwarding portion of N_i. To guarantee there is no loop in the path, only neighbors closer than N to the destination can be chosen as the next hop. From these neighbors, choose one with the smallest forwarding portion.

In Figure 5.2, let $N_1 = A, N_2 = B, N_3 = C, N_4 = E, N_5 = F$. From the forwarding history, it is known that node S has forwarded 3 packets to A. The total number of packets forwarded to these neighbors by S is $3+6+2+5+4 = 20$. So $\delta(N_1) = \frac{3}{20} = 0.15$. Similarly, we can obtain $\delta(N_2) = 0.3, \delta(N_3) = 0.1, \delta(N_4) = 0.25, \delta(N_5) = 0.2$. A, B, C are closer to the destination than S while E, F are not. Choose the next hop only from A, B, C. $\delta(N_3)$ is the smallest, so choose C as the next hop.

To analyze the proposed strategy based on forwarding history, define a m-tuple sequence as follows: $F_0, F_1, F_2, \cdots$, where $F_i = (q_{i1}, q_{i2}, \cdots, q_{im})$. F_i represents the number of packets forwarded to each neighbor by N when N has for-

For $j \neq h$,

$$q_{(\ell+1)j} = q_{\ell j} = \left\lfloor \frac{\ell}{m} \right\rfloor = \left\lfloor \frac{\ell+1}{m} \right\rfloor,$$
$$q_{(\ell+1)h} = q_{\ell h} + 1 = \left\lfloor \frac{\ell}{m} \right\rfloor + 1 = \left\lfloor \frac{\ell+1}{m} \right\rfloor + 1.$$

Therefore the statement is true for $i = \ell + 1$.

Case 2: $\ell = m \left\lfloor \frac{\ell}{m} \right\rfloor + m - 1$ and $\left\lfloor \frac{\ell+1}{m} \right\rfloor = \left\lfloor \frac{\ell}{m} \right\rfloor + 1$. There is one and only one h, where $q_{\ell h} = \left\lfloor \frac{\ell}{m} \right\rfloor$. $\forall j \neq h, q_{\ell j} = \left\lfloor \frac{\ell}{m} \right\rfloor + 1$. The current node chooses N_h as the next hop.

$$q_{(\ell+1)h} = q_{\ell h} + 1 = \left\lfloor \frac{\ell}{m} \right\rfloor + 1 = \left\lfloor \frac{\ell+1}{m} \right\rfloor.$$

For $j \neq h$, $q_{(\ell+1)h} = q_{\ell j} = \left\lfloor \frac{\ell}{m} \right\rfloor + 1 = \left\lfloor \frac{\ell+1}{m} \right\rfloor$. Therefore the conclusion holds.

Case 3: $\ell \neq m \left\lfloor \frac{\ell}{m} \right\rfloor + m - 1$ and $\ell \neq m \left\lfloor \frac{\ell}{m} \right\rfloor$. Obtain $\left\lfloor \frac{\ell+1}{m} \right\rfloor = \left\lfloor \frac{\ell}{m} \right\rfloor$. $\exists j, q_{\ell j} = \left\lfloor \frac{\ell}{m} \right\rfloor$. Suppose N_h is the next hop. Then,

$$q_{(\ell+1)h} = q_{\ell h} + 1 = \left\lfloor \frac{\ell}{m} \right\rfloor + 1 = \left\lfloor \frac{\ell+1}{m} \right\rfloor + 1. \tag{5.5}$$

For $j \neq h$, $q_{(\ell+1)h} = q_{\ell j} = \left\lfloor \frac{\ell}{m} \right\rfloor + 1 = \left\lfloor \frac{\ell+1}{m} \right\rfloor + 1$ or $q_{(\ell+1)h} = q_{\ell j} = \left\lfloor \frac{\ell}{m} \right\rfloor = \left\lfloor \frac{\ell+1}{m} \right\rfloor$.

The statement also holds for this case. Thus, $\forall i \geq 0, 1 \leq j \leq m$,

$$q_{ij} = \left\lfloor \frac{i}{m} \right\rfloor \text{ or } q_{ij} = \left\lfloor \frac{i}{m} \right\rfloor + 1.$$

From Proposition 4, it is known that if choosing the neighbors according to the forwarding history only, the difference between the numbers of packets forwarded by neighbors is at most one. However, when deciding the next hop with greedy method, only choose neighbors from those closer to the destination than the current node. In the following, forwarding history under this constraint is analyzed.

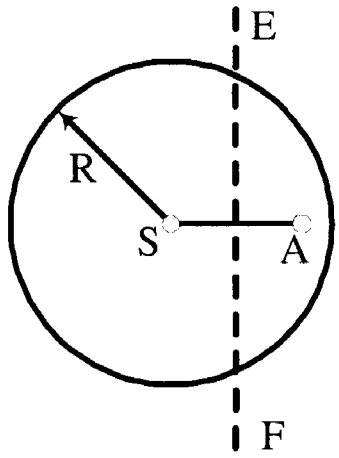


Figure 5.3. An example showing how to calculate the probability that a neighbor is a candidate for the next hop. R is the communication range, S is the current node, A is a neighbor of S and EF is a perpendicular bisector of line segment SA.

A neighbor can be a candidate for the next hop if and only if it is closer to the destination than the current node. In Figure 5.3, R is the communication range, S is the current node, A is a neighbor of S and EF is a perpendicular bisector of line segment SA. Suppose the destination is D. To satisfy $|SD| > |AD|$, D must locate on the right side of EF. Calculate the probability that A is a candidate for the next hop as follows:

$$P_1 = P(A \text{ is a candidate }) = \oint_{|DS|>|DA|} \phi(x,y)dxdy, \tag{5.6}$$

where $\phi(x,y)$ denotes the probability distribution function of the destination D with coordinates (x,y).

Suppose that the destinations of the packets forwarded by the current node are distributed uniformly within the deployment region. Let Ω represent the area on the right of EF within the deployment region. The bigger Ω is, the higher P_1 is. If the nodes are distributed in the deployment region uniformly, there are more nodes on the right side of EF when Ω is big.

Suppose that the number of packets forwarded by the current node is ω and the number of packets for which A is a candidate is z. Clearly, z follows the

binomial distribution. The probability density function is

$$\psi(z;\omega,P_1)=\begin{pmatrix}\omega\\ z\end{pmatrix}P_1^z(1-P_1)^{\omega-z}. \tag{5.7}$$

Then, the expected number of packets for which A is a candidate is $E(z)=\omega P_1$. The variance is $Var(z)=\omega P_1(1-P_1)$.

According to the *Central Limit Theorem*, approximate the distribution by the normal distribution with mean $\mathrm{E}(z)$ and variance $\mathrm{Var}(z)$ when ω is big enough. Therefore, we can obtain

$$P(z<E(z)+1.64\sqrt{Var(z)})=0.95. \tag{5.8}$$

That is to say z is upper bounded by

$$E(z)+1.64\sqrt{Var(z)}=\omega P_1+1.64\sqrt{\omega P_1(1-P_1)} \tag{5.9}$$

with 95% probability.

Since being a candidate does not mean a node will definitely be chosen as the next hop, the number of packets for which a node is chosen as the next hop is upper bounded by (5.9) with a probability of at least 95%.

In the following, it will be shown how our strategy performs by simulation. In a 5×5 unit square region, generate nodes uniformly. The number of nodes ranges from 100 to 1000. For each network density, randomly generate 1000 maps. Randomly choose a source. The source sends 1000 packets to 1000 randomly chosen destinations. After that, obtain the maximal number and the minimal number among the numbers of packets the nodes have forwarded. The average number of packets each node forwards can also be calculated. Define the maximal forwarding ratio as the maximal number of packets divided by the average number of packets and define the minimal ratio as the minimal number of packets divided by the average number of packets. The simulation results for node number 100, 300, 600, 1000 are shown in Figure 5.4. It can be seen that most maximal forwarding ratios are below two when the node number is 100. When the node number increases to

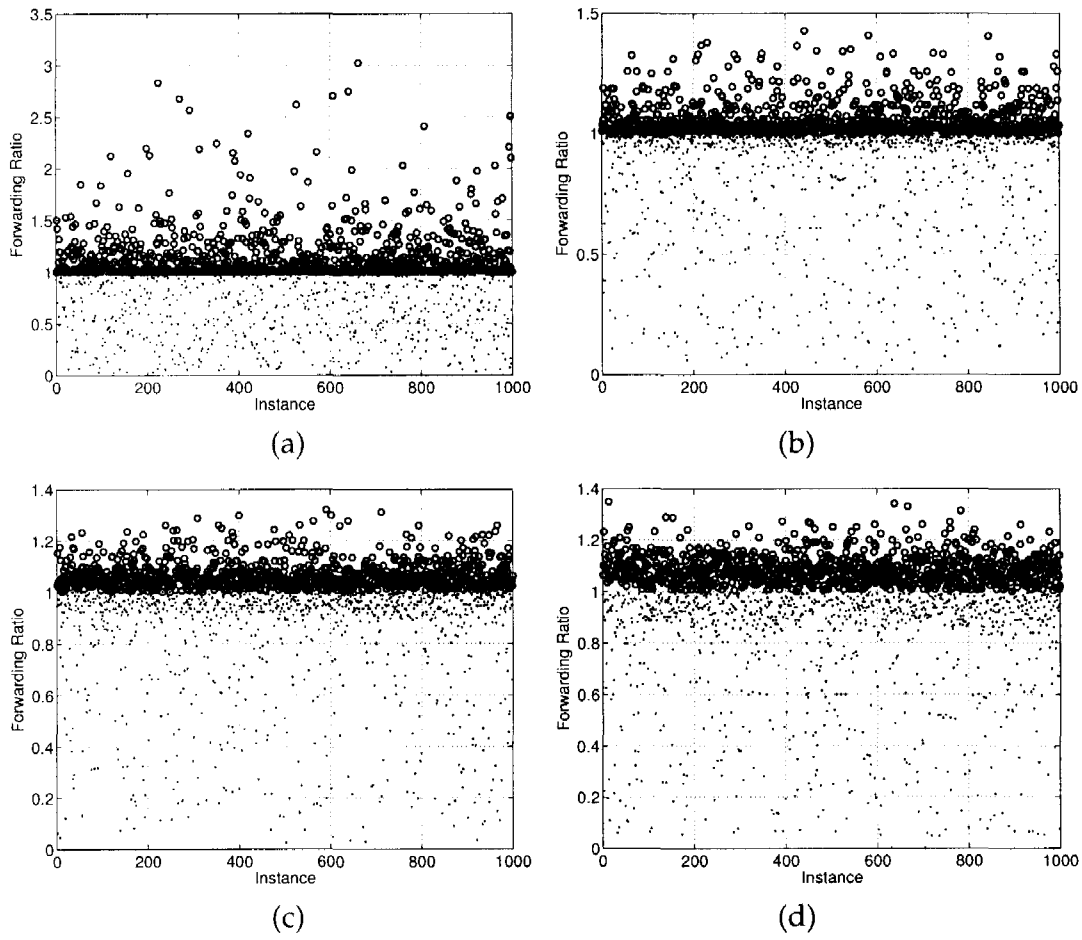


Figure 5.4. Maximal forwarding ratios and minimal forwarding ratios of 1000 instances (a) 100 nodes; (b) 300 nodes; (c) 600 nodes; (d) 1000 nodes. Small circles represent maximal forwarding ratios and small dots denote minimal forwarding ratios.

600, all the maximal ratios are below 1.5. When the node number keeps increasing, all the maximal ratios are smaller than 1.4. It means that the number of packets forwarded by each node does not vary much. None of the nodes die very quickly.

5.2.4 Progress

Another important factor in the process of routing is progress. Here define the progress a packet makes if it is forwarded to neighbor N_j as follows:

$$Q(N_j) = \frac{|ND| - |N_jD|}{R}, \tag{5.10}$$

where D is the destination, R is the communication range and $|\cdot|$ denotes the Euclidian Distance.

With all other factors the same, the bigger the progress, the better. If the average progress a node makes is big, the total number of hops from the source to the destination is small and the energy consumed is low.

5.2.5 Remaining Energy

To prolong the lifetime of wireless sensor networks, remaining energy should be taken into account. Some nodes may have more tasks than others and the objective is to propose a method to distribute the tasks evenly and balance the energy consumption among nodes. This factor is different from the forwarding history factor. First, forwarding history considers only the packets forwarded by the current node while remaining energy considers not only the energy consumed by packets forwarded by the current node but also the energy consumed by packets forwarded by other nodes.

Second, forwarding history tries to balance the traffic in each direction with respect to the current node. Remaining energy considers the balance of energy consumption among its neighbors. Some neighbors may transmit few packets but consume much energy because of low quality links. Some neighbors may transmit many packets but consume lot less energy.

When the number of retransmissions is not limited, the difference of remaining energy between two nodes may be very huge. For example, the PRRs of all neighbors of the current node may be very low. A neighbor's energy may be depleted before it transmits one packet successfully while other neighbors consume no or a little energy.

Suppose the number of retransmissions is Γ, the remaining energy of neighbor N_j $(1 \leq j \leq m)$ is e_{ij}^- $(e_{ij}^- \geq 0)$ when a neighbor is chosen by N for the ith packet and the remaining energy of neighbor N_j $(1 \leq j \leq m)$ is e_{ij}^+ $(e_{ij}^+ \geq 0)$ right after the neighbor chosen by N for the ith packet has forwarded the packet. Assume that

N_j $(1 \leq j \leq m)$ has no energy consumption caused by forwarding other packets between the time a neighbor is chosen by N for the ith packet and the time the neighbor has forwarded the packet successfully to its next hop. Each neighbor has the same initial energy. The follow proposition can be obtained.

Proposition 5 *If the current node has at least two neighbors and always choose the neighbor with the most remaining energy as the next hop, and* $\forall i > 1, 1 \leq j \leq m$, $\max_j e_{ij}^- - \min_j e_{ij}^- \neq 0$, $\max_j e_{ij}^+ - \min_j e_{ij}^+ \leq \max\{\max_j e_{ij}^- - \min_j e_{ij}^-,\ \Gamma + 1\}$.

Proof: Suppose that N_h is chosen as the next hop among the neighbors with the maximal remaining energy. Obtain $e_{ih}^- = \max_j e_{ij}^-$. After N_h transmits the packet to its next hop successfully, its energy becomes $e_{ih}^+ = e_{ih}^- - \tau$, where τ is the number of transmissions needed by N_h to transmit the packet successfully to the next hop. Two cases need to be considered.

Case 1: $\max_j e_{ij}^- - \min_j e_{ij}^- \geq \Gamma + 1$. Since the maximal number of retransmissions is Γ, $\tau \leq \Gamma + 1$. Moreover, $\max_j e_{ij}^- - \min_j e_{ij}^- \geq \Gamma + 1$,

$$e_{ih}^+ = e_{ih}^- - \tau = \max_j e_{ij}^- - \tau \geq \min_j e_{ij}^-. \tag{5.11}$$

While

$$e_{ij}^- = e_{ij}^+ \ (j \neq h), \tag{5.12}$$

obtain $\min_j e_{ij}^- = \min_j e_{ij}^+$, and

$$\max_j e_{ij}^+ \leq \max_j e_{ij}^-, \tag{5.13}$$

because $e_{ih}^+ \leq e_{ih}^-$. The following can be obtained:

$$\begin{aligned} \max_j e_{ij}^+ - \min_j e_{ij}^+ &= \max_j e_{ij}^+ - \min_j e_{ij}^- \\ &\leq \max_j e_{ij}^- - \min_j e_{ij}^- \\ &= \max\{\max_j e_{ij}^- - \min_j e_{ij}^-,\ \Gamma + 1\}. \end{aligned} \tag{5.14}$$

Case 2: $\max_j e_{ij}^- - \min_j e_{ij}^- < \Gamma + 1$. Since $e_{ih}^+ \leq e_{ih}^-$ and Equation (5.12) is true, Equation (5.13) still holds.

If $\tau \leq \max_j e_{ij}^- - \min_j e_{ij}^-$, the following holds

$$e_{ih}^+ = e_{ih}^- - \tau \geq \min_j e_{ij}^- \;\Rightarrow\; \min_j e_{ij}^+ = \min_j e_{ij}^-, \tag{5.15}$$

and

$$\begin{aligned}\max_j e_{ij}^+ - \min_j e_{ij}^+ &= \max_j e_{ij}^+ - \min_j e_{ij}^- \\ &\leq \max_j e_{ij}^- - \min_j e_{ij}^- < \Gamma + 1 \\ &= \max\{\max_j e_{ij}^- - \min_j e_{ij}^-,\ \Gamma + 1\}.\end{aligned}$$

If $\tau > \max_j e_{ij}^- - \min_j e_{ij}^-$, obtain

$$e_{ih}^+ = e_{ih}^- - \tau < \min_j e_{ij}^- \;\Rightarrow\; \min_j e_{ij}^+ = e_{ih}^+, \tag{5.16}$$

and

$$\begin{aligned}\max_j e_{ij}^+ - \min_j e_{ij}^+ &= \max_j e_{ij}^+ - e_{ih}^+ \leq \max_j e_{ij}^- - e_{ih}^+ = e_{ih}^- - e_{ih}^+ = \tau \\ &\leq \Gamma + 1 = \max\{\max_j e_{ij}^- - \min_j e_{ij}^-,\ \Gamma + 1\}.\end{aligned}$$

In summary, the statement holds.

From Proposition 5, it is known that the remaining energy strategy always tries to make the difference smaller or keeps it within a small range.

5.2.6 LAGR (Lifetime-Aware Geographic Routing)

From the previous discussion, it is known that the above four factors are very important for the lifetime of a wireless sensor network with geographic routing. The proposed method defines a function based on these four factors such that

$$\Upsilon(N_j) = PRR(N_j) + \alpha(1 - \delta(N_j)) + \beta Q(N_j) + \eta e_i(N_j),$$

where $1 \leq j \leq m, e_i(N_j)$ is the normalized energy of N_j by dividing the energy of N_j by its initial energy, and α, β, η are three non-negative constants. Here fix the weight for PRR to 1 and change the weights for the other three factors to obtain the optimal network lifetime.

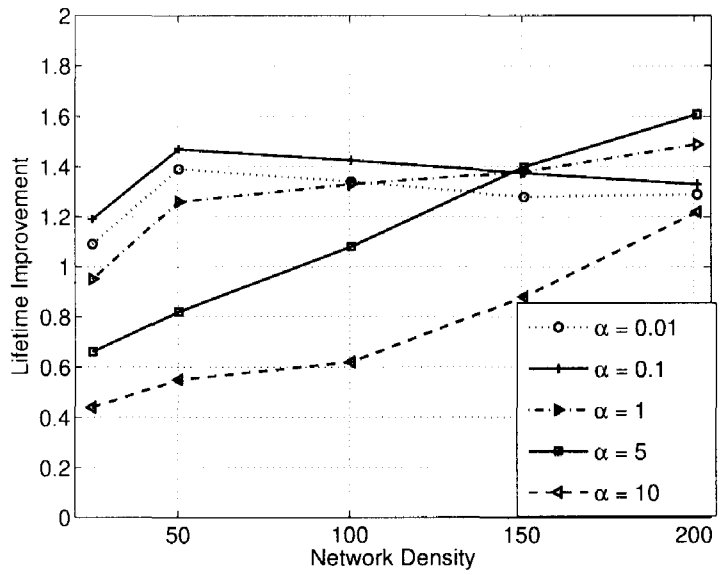


Figure 5.5. The lifetime improvement of LAGR over PRR*D when the weight for the forwarding history α changes ($\beta = 1, \eta = 1$).

Since $E(x) = \frac{1}{PRR}$, PRR should be as big as possible.

The smaller the forwarding portion $\delta(N_j)$ is, the more chance N_j is given to become the next hop. Here use $1 - \delta(N_j)$ to be consistent with PRR.

For the progress, prefer nodes with large progress so that the total number of hops from source to destination is small.

Nodes with more remaining energy are preferred in the next hop decision. This way, the energy consumption among current node's neighbors can be balanced. The following section shows how to obtain optimal values for α, β, η by extensive simulations.

5.3 Simulation

To evaluate the performance improvement of the proposed LAGR algorithm, the proposed algorithm and PRR*D in [16] are compared. The simulation configuration is similar to that in Section 2.2. The communication range of each node is fixed at one unit distance. One point should be noted here that there is no

strict value for the communication range (radio range) similar to the values used for ideal channel models. Here define a nominal communication range in order to calculate the density as a function of that range.

The network topology is generated by randomly and uniformly placing nodes on a square field of side length five units. The network density changes as we change the number of nodes in the square field. In the simulation, the number of nodes generated in the square field ranges from 200 to 1600, which corresponds to network densities ranging from 25 to 200 nodes per unit disk of area π.

For a node to be considered a neighbor, it must have at least PRR = 0.01 (set this threshold just to distinguish nodes having acceptable communication channels with the current node from nodes having bad communication channels with the current node). Otherwise, the number of retransmissions may be very high. Density is a good estimate for the number of nodes within the communication range of a node. According to the link layer model [54], many nodes within the communication range of a node may have very low PRR. That is why the density in the simulations is so high.

Each node has the same initial energy 100 units. The maximal number of retransmissions is ten unless stated otherwise. Assume a network dead if more than 10% nodes in the network are out of their energy. The lifetime of a network realization is the number of packets delivered till the network is dead. For a given network density, 5000 realizations of network graphs are generated. For the same routing sequence, route the packets with PRR*D and LAGR respectively. Suppose the number of packets delivered by PRR*D in the ith network is κ_i and the number of packets delivered by LAGR is ν_i. Only the case in which PRR*D delivers at least one packet (i.e. $\kappa_i > 0$) is considered. For each network, calculate the ratio of the number of packets delivered by LAGR to that by PRR*D. Average the ratios over all networks delivering at least one packet and we obtain the lifetime improvement. Suppose the number of networks in which $\kappa_i > 0$ is λ. Then the lifetime

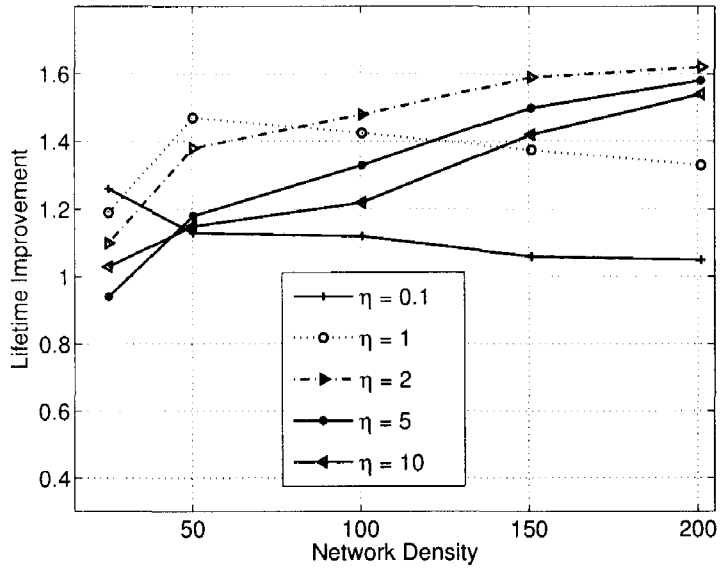


Figure 5.6. The lifetime improvement of LAGR over PRR*D when the weight for the remaining energy η changes ($\beta = 1, \alpha = 0.1$).

improvement of LAGR over PRR*D is defined as follows:

$$\zeta = \frac{1}{\lambda} \sum_{i=1}^{\lambda} \frac{\nu_i}{\kappa_i}. \tag{5.17}$$

5.3.1 Effects of Weights for Different Factors

First, fix $\beta = 1, \eta = 1$ and change α to see how the lifetime improvement changes. The results are shown in Figure 5.5. It can be seen that when $\alpha = 1, 5, 10$, the lifetime improvement increases as the network density increases. When $\alpha = 0.01, 0.1$, the lifetime improvement increases first then decreases as the network density increases. When the network density is low, the lifetime improvement increases as α decreases. When $\alpha \leq 0.1$, the lifetime improvement changes little. When the network density is high, the lifetime improvement of LAGR with $\alpha = 1$ is a little better than that with $\alpha = 0.1, 0.01$. Considering the lifetime improvement over all densities, set $\alpha = 0.1$.

Next set $\alpha = 0.1, \beta = 1$ and change η. The simulation results are shown in Figure 5.6. It can be seen that the lifetime improvement decreases with the in-

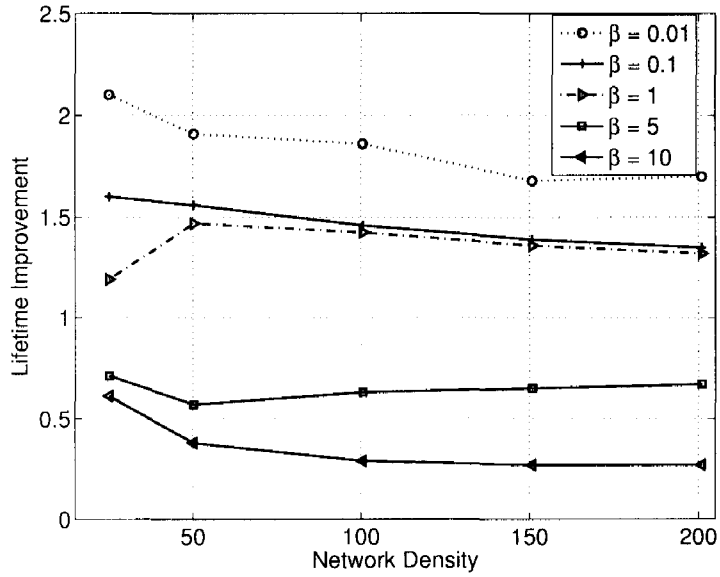


Figure 5.7. The lifetime improvement of LAGR over PRR*D when the weight for the progress β changes ($\eta = 1, \alpha = 0.1$).

creasing of network density when $\eta = 0.1$. It increases first and arrives at the peak value at density 50, then decreases slightly afterwards when $\eta = 1$. When $\eta > 1$, the lifetime improvement increases with the increasing of network density. It is probably because there are more choices for the next hop and the energy consumption is distributed more evenly among nodes when the network increases. Actually, remaining energy is an important factor. When η is small, it can be seen that a reduction of lifetime improvement is seen with the increase of network density.

Based on this simulation, choose $\eta = 1$. The simulation results are plotted in Figure 5.7 when the weight for the progress, β changes. It can be seen that the lifetime improvement increases with the decreasing of β at any given density. Given β, the lifetime improvement changes a little when the density changes. The maximal lifetime improvement is about 2.1. It means the proposed method can double the lifetime of network compared with that with PRR*D. $\alpha = 0.1, \beta = 0.01$, and $\eta = 1$ will be used in the following simulations.

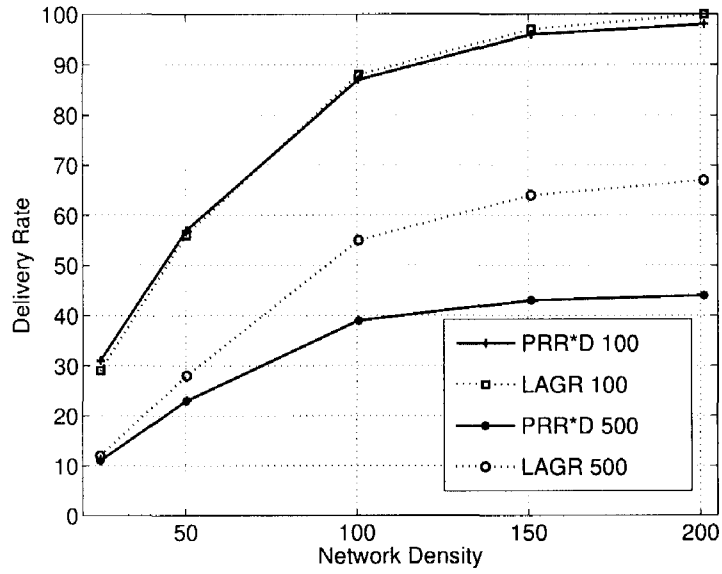


Figure 5.8. The delivery rate of LAGR and PRR*D at different numbers of packets: 100, 500, where $\alpha = 0.1, \beta = 0.01$, and $\eta = 1$.

5.3.2 Delivery Rate and Energy Efficiency

To evaluate delivery rate, energy efficiency of PRR*D and $LAGR$, use two metrics defined in [55]: delivery rate and energy efficiency. Delivery rate is defined as the percentage of packets sent by the source node which reach the destination node. Energy efficiency is defined as the number of packets delivered to the destination for each unit of energy spent by the network. Test two different cases: 100 packets and 500 packets. For each case, randomly generate 5000 networks at any given density. For the first case, randomly choose 100 pairs of source and destination in each network. For each pair, send a packet from the source to the destination. After sending 100 packets, calculate the delivery rate of each network. Average the delivery rates of all networks at a given network density and obtain the average delivery rate at that density. For the second case, do similarly and generate a different number of packets for each network. The simulation results are shown in Figure 5.8. It can be seen that the delivery rate of LAGR is almost the same as that of PRR*D when only 100 packets are generated for each network. The delivery rate increases as the network density increases. That is because the

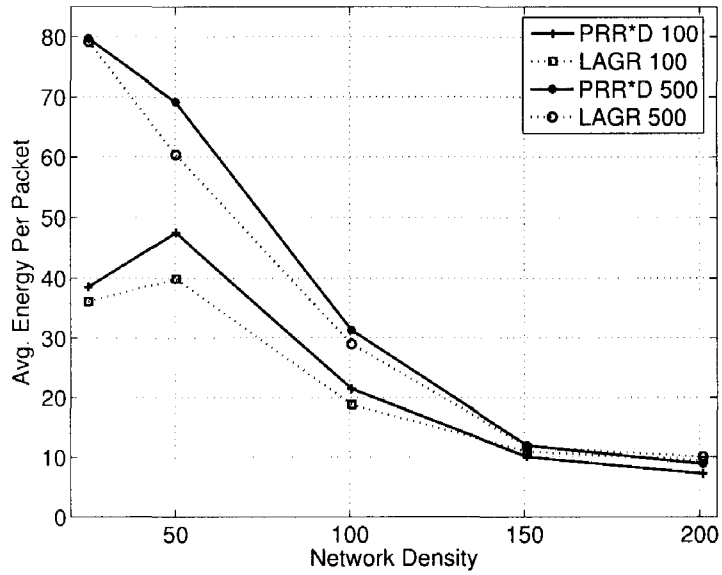


Figure 5.9. The average energy consumed per packet of LAGR and PRR*D at different numbers of packets: 100, 500, where $\alpha = 0.1, \beta = 0.01$, and $\eta = 1$.

probability of greedy failure is higher when the network density is low. When 500 packets are generated for each network, the difference between the delivery rate of LAGR and PRR*D increases as the network density increases. It is probably because some nodes die quickly with PRR*D, and LAGR can balance the energy consumption better when the network density increases.

The energy efficiency is usually very small and varies significantly when the network density changes. To visualize the simulation results better, the inverse of energy efficiency in the plot is used. The energy efficiency is obtained similarly as delivery rate. Define the inverse of energy efficiency as average energy consumed per packet. The simulation results are shown in Figure 5.9. It can be seen that LAGR costs a smaller amount of energy than PRR*D. The average energy consumed per packet decreases as the network density increases. It is because the current node has not many choices for the next hop when the network density is low. When the network is low, there is either no path or the distance between two hops is large. If there is no path, the packet may consume some energy but it is not included in the total number of packets delivered successfully. When the distance

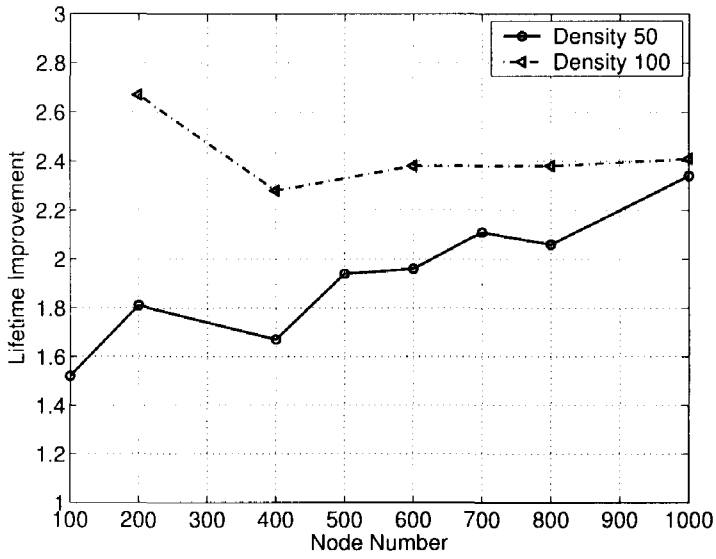


Figure 5.10. The lifetime improvement under different network sizes with fixed network density: 50 and 100, where $\alpha = 0.1, \beta = 0.01$, and $\eta = 1$.

is large, the PRR is probably low and for the same number of hops it consumes more energy. For the 500-packet case, the peak does not occur at the lowest density. It is probably because more failures of delivery at the lowest density are due to the isolated cases. The total amount of energy consumed at the lowest density for the same number of routing failures is less.

5.3.3 Effects of Network Size

In this subsection, fix the network density and change the network size to observe the lifetime improvement on different network sizes. Two network densities: 50, 100 are used. Node number varies from 100 to 1000. Given a density and node number, simulate 5000 networks. The simulation results are shown in Figure 5.10. It can be seen that the lifetime improvement increases when the network size increases at network density 50 and there is not much difference between the improvements at network density 100. This is due to the fact that there are usually no multiple paths between the source and destination when the network density is low and the network size is small. If there are few choices for the path, there is

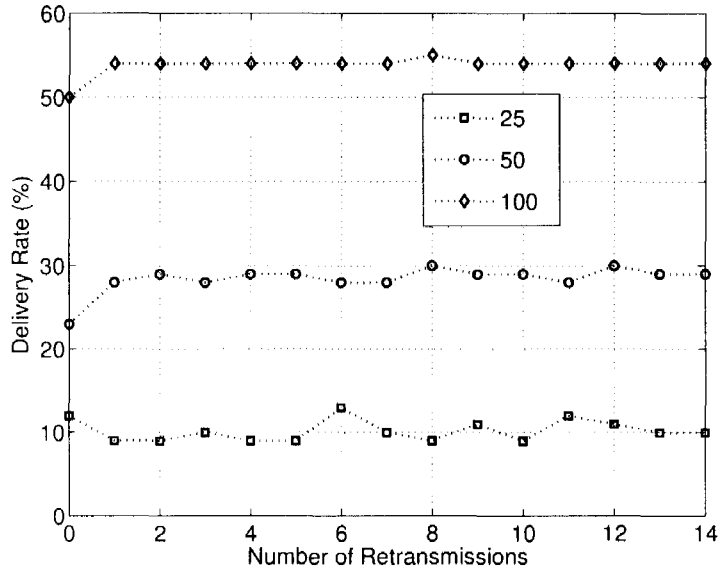


Figure 5.11. The delivery rate of LAGR under different numbers of retransmissions with network density: 25, 50 and 100, where $\alpha = 0.1, \beta = 0.01$, and $\eta = 1$.

not much difference between different algorithms and the lifetime improvement is small.

5.3.4 Effects of the Number of Retransmissions

To evaluate the effects of the number of retransmissions, we vary the number of retransmissions from 0 to 14 under network density 25, 50 and 100. The simulation results are shown in Figures 5.11 and 5.12. It can be seen from Figure 5.11 that the delivery rate is almost constant at a given network density. The number of retransmissions has little impact on the delivery rate. From Figure 5.12, when the network density is 100, the average energy consumed keeps constant at given network density. When the network density is 25 or 50, it increases slightly when the number of retransmissions increases.

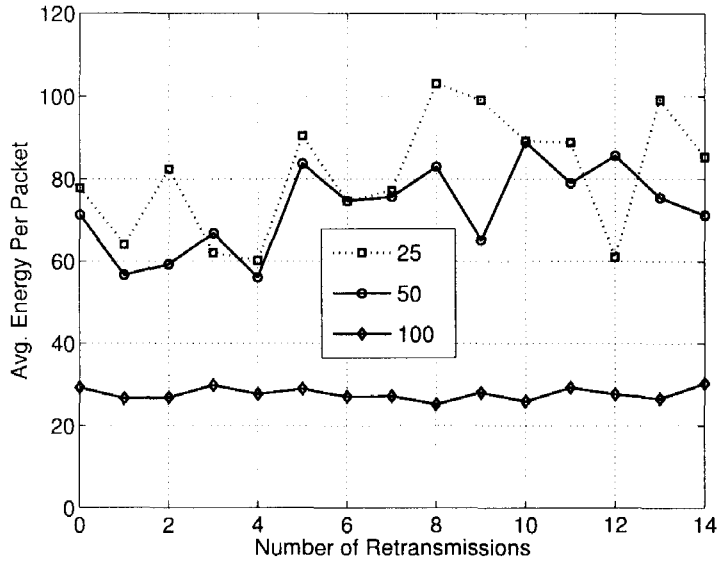


Figure 5.12. The average energy consumed per packet of LAGR under different numbers of retransmissions with network density: 25, 50 and 100, where $\alpha = 0.1, \beta = 0.01$, and $\eta = 1$.

5.4 Conclusion

To prolong the lifetime of geographic routing wireless sensor networks with a realistic link layer model, four important factors (PRR, forwarding history, progress and remaining energy) in geographic routing are studied and a new geographic routing algorithm: LAGR is proposed. Simulation results show that LAGR can usually double the lifetime of wireless sensor networks compared with PRR*D.

CHAPTER 6

CONCLUSION

This dissertation develops a framework for energy efficient geographic routing. This framework includes a path pruning strategy, an anchor-based routing protocol, a geographic multicast algorithm and a lifetime-aware geographic routing algorithm. The proposed methods are studied systematically and their effectiveness is demonstrated by extensive computer simulations.

First, an efficient path pruning strategy is proposed to reduce the excessive number of hops caused by the detouring mode of geographic routing protocols. The path pruning algorithm finds routing shortcuts by exploiting the channel listening capability of wireless nodes, and is able to reduce a large number of hops with the help of a little state information passively maintained by a subset of nodes on the route. Algorithm properties are also discussed and simulation results are provided to demonstrate the effectiveness of the proposed algorithm in shortening the routing path and improving delivery rate when it is applied to existing geographic routing protocols.

Second, an anchor-based geographic routing protocol is proposed, where anchors are set as relay nodes. A packet is routed from the source to the destination through a sequence of anchor nodes. The anchor list is obtained based on the projection distance of nodes in detouring mode with respect to the virtual line linking the source and destination. For existing anchor-based schemes, once an anchor list is obtained, the path from the source to the destination usually do not change unless the network topology changes, which may lead to the quick depletion of the energy for some nodes. To better distribute energy consumption among nodes in the network and thus prolong the network lifetime, a random shift is introduced to

the location of anchors to obtain virtual anchors for each packet sent. Simulation results show that our projection distance based algorithm outperforms existing anchor-based algorithms with shorter paths and fewer anchors in random network topology. They also demonstrate that the lifetime-improving strategy with virtual anchors is effective in increasing the number of packets delivered in the lifetime of sensor networks.

Third, a location-based multicast algorithm, namely the Destination Clustering Geographic Multicast (DCGM) is proposed for wireless sensor networks. Geographic routing is efficient in providing scalable unicast routing in resource-constrained wireless sensor networks. However, its applications in multicast routing remain largely unexplored. The idea of DCGM is to cluster destinations that can share the same next hop, and then iteratively select the next hop as the neighbor with the maximum number of destinations. It is proved that the complexity of DCGM is $\mathcal{O}(n\ell)$, where n is the number of neighbors of the current node and ℓ is the number of destinations associated with the current node. Simulation results show that DCGM achieves better performance than existing geographic multicast routing algorithms in terms of average number transmissions, with much lower computation complexity. To further reduce the number of transmissions, clustering strategy is applied to GMR and DCGM. It improves the performance of GMR and DCGM by dividing the destinations into many clusters and sending the packet first to the closest destination in each cluster, which then sends the packet to other nodes in the cluster. Simulation results show that the strategy can reduce the number of transmissions up to 35% percent.

Finally, a realistic link layer model is applied to wireless sensor networks and a new geographic routing algorithm is proposed to prolong the lifetime of wireless sensor networks. Maximizing the lifetime of wireless sensor networks under constrained resources is an interesting problem that has gained increasing attention. However, how to prolong the lifetime of wireless sensor networks with geographic routing remains largely unexplored. An ideal link layer model is as-

sumed in many methods in improving the lifetime of wireless sensor networks. In this dissertation, a realistic link layer model is introduced to our framework and a function consisting of four important factors: PRR (Packet Reception Rate), forwarding history, progress and remaining energy is defined. With this function, evaluate various characteristics of each neighbor and forward a packet to the optimal neighbor. Simulation results show that the proposed algorithm can usually double the lifetime of wireless sensor networks compared with existing approaches.

In summary, a comprehensive framework has been developed to cope with routing efficiency and lifetime that limit the performance of wireless sensor networks. It advances the basis theory, design and development of routing techniques in wireless communications. It will also impact the design and implementation of future routing protocols and communication systems.

REFERENCES

[1] V. D. Park and M. S. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," in *Proc. IEEE Conference on Computer Communications (INFOCOM)*, Apr. 1997, pp. 1405–1413.

[2] C. Perkins, "Ad-hoc on-demand distance vector routing," in *Proc. Military Communications Conference (MILCOM)* , Nov. 1997.

[3] Z. Haas, "A new routing protocol for the reconfigurable wireless networks," in *Proc. IEEE Int. Conf. on Universal Personal Communications*, Oct. 1997.

[4] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks," in *Proc. ACM International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (Dial-M)*, New York, Aug. 1999, pp. 48–55.

[5] B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *Proc. ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, Boston, MA, Aug. 2000, pp. 243–254.

[6] Y. B. Ko and N. H. Vaidya, "Location-aided routing (LAR) in mobile ad hoc networks," in *Proc. ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, Dallas, TX, Oct. 1998, pp. 66–75.

[7] F. Kuhn, R. Wattenhofer, and A. Zollinger, "Asymptotically optimal geometric mobile ad-hoc routing," in *Proc. ACM International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (Dial-M)*, Atlanta, GA, Sep. 2002, pp. 24–33.

[8] ——, "Worst-case optimal and average-case efficient geometric ad-hoc routing," in *Proc. ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, Annapolis, MD, Jun. 2003, pp. 267–278.

[9] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger, "Geometric ad-hoc routing: of theory and practice," in *Proc. ACM Symposium on the Principles of Distributed Computing (PODC)*, Boston, MA, Jul. 2003, pp. 63–72.

[10] I. A. Getting, "Perspective/navigation-the global positioning system," *IEEE Spectrum*, vol. 30, no. 12, pp. 36–38, 1993.

[11] N. Priyantha, A. Chakraborty, and H. Balakrishnan, "The cricket location-support system," in *Proc. ACM International Conference on Mobile Computing and Networking (MobiCom)*, Boston, MA, Aug. 2000.

[12] Y. B. Ko and N. H. Vaidya, "Geocasting in mobile ad hoc networks: Location-based multicast algorithms," Texas A&M University, Tech. Rep. TR-98-018, 1998.

[13] E. Kranakis, H. Singh, and J. Urrutia, "Compass routing on geometric networks," in *Proc. 11th Canadian Conference on Computational Geometry*, Aug. 1999, pp. 51–54.

[14] I. Stojmenovic and X. Lin, "Loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks," *IEEE Trans. Parallel and Distributed Systems*, vol. 12, no. 10, pp. 1023–1032, Oct. 2001.

[15] G. Xing, C. Lu, R. Pless, and Q. Huang, "On greedy geographic routing algorithms in sensing covered networks," in *Proc. ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, Tokyo, Japan, May 2004, pp. 31–42.

[16] K. Seada, M. Zuniga, A. Helmy, and B. Krishnamachari, "Energy-efficient forwarding strategies for geographic routing in lossy wireless sensor networks," in *Proc. ACM conference on Embedded networked sensor systems (SenSys)*, Baltimore, MD, Nov. 2004, pp. 108–121.

[17] K. Gabriel and R. Sokal, "A new statistical approach to geographic variation analysis," *Systematic Zoology*, vol. 18, pp. 259–278, Sep. 1969.

[18] X.-Y. Li, G. Calinescu, and P.-J. Wan, "Distributed construction of planar spanner and routing for ad hoc networks," in *Proc. IEEE Conference on Computer Communications (INFOCOM)*, Jun. 2002, pp. 1268–1277.

[19] G. Toussaint, "The relative neighborhood graph of a finite planar set," *Pattern Recognition*, vol. 12, pp. 261–268, 1980.

[20] P. Boone, E. Chavez, L. Gleitzky, E. Kranakis, J. Opatrny, G. Salazar, and J. Urrutia, "Morelia test: Improving the efficiency of the gabriel test and face routing in ad-hoc networks," in *Proc. SIROCCO (Lecture Notes in Computer Science, vol. 3104)*, 2004, pp. 23–34.

[21] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification*, IEEE Std. 802.11, 1997.

[22] J. Li, J. Jannotti, D. De Couto, D. Karger, and R. Morris, "A scalable location service for geographic ad-hoc routing," in *Proc. ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, Boston, MA, Aug. 2000, pp. 120–130.

[23] L. Blazevic, J. L. Boudec, and S. Giordano, "A location-based routing method for mobile ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 4, pp. 97–110, Mar.-Apr. 2005.

[24] P.-H. Hsiao and H. T. Kung, "Gravity routing in ad hoc networks: integrating geographical and topology-based routing," in *Proc. the 7th International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN)*, May 2004, pp. 397–403.

[25] H. Huang, "Adaptive algorithms to mitigate inefficiency in greedy geographical routing," *IEEE Communication Letters*, vol. 10, pp. 150–152, Mar. 2006.

[26] L. Zou, M. Lu, and Z. Xiong, "A distributed algorithm for the dead end problem of location based routing in sensor networks," *IEEE Trans. Vehicular Technology*, vol. 54, no. 4, pp. 1509–1522, Jul. 2005.

[27] C. Gui and P. Mohapatra, "Short: Self-healing and optimizing routing techniques for mobile ad hoc networks," in *Proc. ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, Annapolis, MD, Jun. 2003, pp. 279–290.

[28] S. Datta, I. Stojmenovic, and J. Wu, "Internal node and shortcut based routing with guaranteed delivery in wireless networks," *Cluster Computing*, vol. 5, pp. 169–178, Apr. 2002.

[29] N. B. Chang and M. Liu, "Revisiting the ttl-based controlled flooding search: optimality and randomization," in *Proc. ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, Tokyo, Japan, May 2004, pp. 85–99.

[30] D. Koutsonikolas, S. M. Das, H. Pucha, and Y. C. Hu, "On optimal ttl sequence-based route discovery in manets," in *Proc. IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW)*, Columbus, OH, Jun. 2005, pp. 923–929.

[31] "The network simulator - ns-2," http://www.isi.edu/nsnam/ns.

[32] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. The MIT Press, 2001.

[33] X. Liu, G. Zhao, M.-T. Sun, and X. Ma, "Path shortening for delivery rate enhancement in geographical routing via channel listening," in *Proc. IEEE Global Telecommunications Conference (GLOBECOM)*, San Francisco, CA, Nov. 2006, pp. 1–5.

[34] X. Ma, M.-T. Sun, G. Zhao, and X. Liu, "Improving geographical routing for wireless networks with an efficient path pruning algorithm," *IEEE Trans. Vehicular Technology*, vol. 1, pp. 246–255, 2006.

[35] D. M. Blough and P. Santi, "Investigating upper bounds on network lifetime extension for cell-based energy conservation techniques in stationary ad hoc

networks," in *Proc. ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, Atlanta, GA, Sep. 2002, pp. 183–192.

[36] S. Basagni, I. Chlamtac, and V. Syrotiuk, "Location aware, dependable multicast for mobile ad hoc networks," *Computer Networks*, vol. 36, pp. 659–670, Aug. 2001.

[37] L. Ji and M. S. Corson, "Differential destination multicast - a MANET multicast routing protocol for small groups," in *Proc. IEEE Conference on Computer Communications (INFOCOM)*, Anchorage, AK, Apr. 2001, pp. 1192–1202.

[38] A. Mizumoto, H. Yamaguchi, and K. Taniguchi, "Cost-conscious geographic multicast on manet," in *Proc. IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (Secon)*, Santa Clara, CA, Oct. 2004, pp. 44–53.

[39] S.-H. Lee and Y.-B. Ko, "Geometry-driven scheme for geocast routing in mobile ad hoc networks," in *Proc. IEEE Vehicular Technology Conference*, Melbourne, Australia, May 2006, pp. 638–642.

[40] M. Mauve, H. Füßler, J. Widmer, and T. Lang, "Position-based multicast routing for mobile ad-hoc networks," University of Mannheim, Tech. Rep. TR-03-004, 2003.

[41] M. Transier, H. Füßler, J. Widmer, M. Mauve, and W. Effelsberg, "Scalable position-based multicast for mobile ad-hoc networks," in *Proc. the First International Workshop on Broadband Wireless Multimedia*, San Jose, CA, Oct. 2004.

[42] J. Sanchez, P. Ruiz, and I. Stojmenovic, "GMR: Geographic multicast routing for wireless sensor networks," in *Proc. IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (Secon)*, Reston, VA, Sep. 2006.

[43] T.-C. Hou and V. Li, "Transmission range control in multihop packet radio networks," *IEEE Transactions on Communications*, vol. 34, no. 1, pp. 38–44, Jan. 1986.

[44] H. Takagi and L. Kleinrock, "Optimal transmission ranges for randomly distributed packet radio terminals," *IEEE Transactions on Communications*, vol. 32, no. 3, pp. 246–257, Mar. 1984.

[45] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, Mar. 1982.

[46] J. C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters," *Journal of Cybernetics*, vol. 3, no. 3, pp. 32–57, 1973.

[47] S. C. Johnson, "Hierarchical clustering schemes," *Psychometrika*, vol. 32, no. 3, pp. 241–254, Sep. 1967.

[48] N. L. A.P. Dempster and D. Rubin, "Maximum likelihood from incomplete data via theEM algorithm," *Journal of the Royal statistical Society*, vol. 39, no. 1, pp. 1–38, 1977.

[49] S. V. David Arthur, "How slow is the k-means method?" in *Proc. ACM Symposium on Computational Geometry*, Sedona, AZ, Jun. 2006, pp. 144–153.

[50] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "Energy-efficient broadcast and multicast trees in wireless networks," *Mobile Networks and Applications*, vol. 7, pp. 481 – 492, Dec. 2002.

[51] V. Rodoplu and T. H. Meng, "Minimum energy mobile wireless networks," in *Proc. IEEE International Conference on Communications (ICC)*, Atlanta, GA, Jun. 1998, pp. 1633–1639.

[52] S. Singh, M. Woo, and C. S. Raghavendra, "Power-aware routing in mobile ad hoc networks," in *Proc. ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, Dallas, TX, Oct. 1998, pp. 181–190.

[53] J. Park and S. Sahni, "An online heuristic for maximum lifetime routing in wireless sensor networks," *IEEE Trans. on Computers*, vol. 55, no. 8, pp. 1048–1056, Aug. 2006.

[54] M. Zuniga and B. Krishnamachari, "Analyzing the transitional region in low power wireless links," in *Proc. IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (Secon)*, Santa Clara, CA, Oct. 2004, pp. 517–526.

[55] T. L. Lim and G. Mohan, "Energy aware geographical routing and topology control to improve network lifetime in wireless sensor networks," in *Proc. IEEE International Conference on Broadband Networks*, Boston, MA, Oct. 2005, pp. 771–773.

[56] I. F. Akyildiz, M. C. Vuran, and Özgür B. Akan, "A cross-layer protocol for wireless sensor networks," in *Proc. IEEE Conference on Information Sciences and Systems (CISS)*, Princeton, NJ, Mar. 2006, pp. 1102–1107.

[57] S. Lee, B. Bhattacharjee, and S. Banerjee, "Efficient geographic routing in multihop wireless networks," in *Proc. ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, Urbana-Champaign, IL, May 2005, pp. 230–241.

[58] C.-K. Toh, H. Cobb, and D. Scott, "Performance evaluation of battery-life-aware routing schemes for wireless ad hoc networks," in *Proc. IEEE International Conference on Communications (ICC)*, Helsinki, Finland, Jun. 2001, pp. 2824–2829.

[59] A. Misra and S. Banerjee, "MRPC: Maximizing network lifetime for reliable routing in wireless environments," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC)*, Orlando, FL, Mar. 2002, pp. 800–806.

[60] J. Aslam, Q. Li, and D. Rus, "Three power-aware routing algorithms for sensor networks," in *Proc. IEEE Wireless Communications and Mobile Computing (WCNC)*, Mar. 2003, pp. 187–208.

[61] K. Kar, M. Kodialam, T. Lakshman, and L. Tassiulas, "Routing for network capacity maximization in energy-constrained ad-hoc networks," in *Proc. IEEE Conference on Computer Communications (INFOCOM)*, San Francisco, CA, Mar. 2003, pp. 673–681.

[62] J.-H. Chang and L. Tassiulas, "Routing for maximum system lifetime in wireless ad-hoc networks," in *Proc. 37-th Annual Allerton Conference on Communication, Control, and Computing*, Monticello, IL, Sep. 1999.

[63] ——, "Energy conserving routing in wireless ad-hoc networks," in *Proc. IEEE Conference on Computer Communications (INFOCOM)*, Tel Aviv, Israel, Mar. 2000, pp. 22–31.

[64] J. Wu, M. Gao, and I. Stojmenovic, "On calculating power-aware connected dominating sets for efficient routing in ad hoc wireless networks," in *Proc. IEEE International Conference on Parallel Processing (ICPP)*, Valencia, Spain, Sep. 2001, pp. 346–356.

[65] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proc. IEEE the 33rd Annual Hawaii International Conference on System Sciences HICSS*, Maui, HI, Jan. 2000.

[66] T. Melodia, D. Pompili, and I. F. Akyildiz, "Optimal local topology knowledge for energy efficient geographical routing in sensor networks," in *Proc. IEEE Conference on Computer Communications (INFOCOM)*, Hongkong, China, Mar. 2004, pp. 1705–1716.

[67] I. Stojmenovic and X. Lin, "Power-aware localized routing in wireless networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 12, no. 11, pp. 1122–1133, 2001.

[68] Y. Yu, R. Govindan, and D. Estrin, "Geographical and energy aware routing: a recursive data dissemination protocol for wireless sensor networks," UCLA Computer Science Department, Tech. Rep. UCLA/CSD-TR-01-0023, 2001.

[69] J. Kuruvila, A. Nayak, and I. Stojmenovic, "Hop count optimal position-based packet routing algorithms for ad hoc wireless networks with a realistic physical layer," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 6, pp. 1267–1275, Jun. 2005.

[70] D. S. J. D. Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," in *Proc. ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, San Diego, CA, USA, Sep. 2003, pp. 134–146.

APPENDICES

Notations

$E(\cdot)$	the expectation
$\binom{n}{i}$	the number of ways of selecting i from n
$\lvert\cdot\rvert$	the cardinality of a set or
	the distance between two points
$\oint\{\cdot\}$	the integration
$\overline{P}$	the average or mean
$\langle A, \cdots D\rangle$	a path from one node A to another node D
$\mathcal{O}(\cdot)$	the computation complexity
$G(V, E)$	a graph with vertex set V and edge set E

Abbreviations

ARQ	Automatic Repeat reQuest
CLT	Central Limit Theorem
CPU	Central Processing Unit
DCGM	Destination Clustering Geographic Multicast
DCGM$^+$	Destination Clustering Geographic Multicast with clustering
DDM	Differential Destination Multicast
DSM	Dynamic Source Multicast
ETX	Expected number of Transmissions
FAPD	Friend Assisted Path Discovery
GEAR	Geographical and Energy-Aware Routing
GFG	Greedy-Face-Greedy
GG	Gabriel Graph
GGP	Geometry-driven Geocasting Protocol
GMPD	Geographic Map-based Path Discovery
GMR	Geographic Multicast Routing
GMR$^+$	Geographic Multicast Routing with clustering
GOAFR	Greedy Other Adaptive Face Routing
GOAFR$^+$	Greedy Other Adaptive Face Routing$^+$
GPSR	Greedy Perimeter Stateless Routing
LAGR	Lifetime-Aware Geographic Routing
LB	Lower Bound
MAC	Media Access Control
NP-hard	Nondeterministic Polynomial-time hard
OAS	Optimal Adaptive Scheme
PBM	Position Based Multicast
PDA	Projection Distance-based Anchor scheme
PP	Path Pruning

PRR	Packet Reception Rate
PRR*D	Packet Reception Rate times Distance
PTKF	Partial Topology Knowledge Forwarding
RF	Radio Frequency
RNG	Relative Neighborhood Graph
SPBM	Scalable Position-Based Multicast
TTL	Time-to-live
UB	Upper Bound

CURRICULUM VITAE

NAME: Gang Zhao

ADDRESS: Department of Electrical and Computer Engineering
University of Louisville
Louisville, KY 40292

EDUCATION: Ph.D., Electrical and Computer Engineering
University of Louisville
2005-2008

M.S., Computer Science
Shanghai JiaoTong University
2002-2005

B.S., Computer Science
Shanghai JiaoTong University
1998-2002

PUBLICATIONS:

CONFERENCE PUBLICATIONS:

1. G. Zhao, X. Liu, and A. Kumar, "Geographic multicast with k-means clustering for wireless sensor networks," *Proc. IEEE Vehicular Technology Conference (VTC)*, pp. 233–237, Marina Bay Singapore, May 11-14, 2008.

2. G. Zhao, X. Liu, and A. Kumar, "Destination clustering geographic multicast for wireless sensor networks," *Proc. International Conference on Parallel Processing Workshops (ICPPW)*, pp. 48–48, Xi'an, China, Sep. 10-14, 2007.

3. G. Zhao, X. Liu and M.-T. Sun, "Energy-aware geographic routing for sensor networks with randomly shifted anchors", *Proc. IEEE Wireless Communica-*

tions and Networking Conference (WCNC), pp. 3456–3461, Hong Kong, Mar. 11-15, 2007.

4. X. Ma, X. Liu and G. Zhao, "Target localization and tracking in noisy binary sensor networks with known spatial topology", *Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. II, pp. 1029–1032, Honolulu, HI, Apr. 15-20, 2007..

5. G. Zhao, X. Liu and M.-T. Sun, "Anchor based geographic routing for sensor networks using projection distance", *Proc. International Symposium on Wireless Pervasive Computing (ISWPC)*, pp. 48–52, San Juan, Puerto Rico, Feb. 5-7, 2007.

6. X. Liu, G. Zhao, M.-T. Sun and X. Ma, "Path shortening for delivery rate enhancement in geographical routing via channel listening", *Proc. IEEE Global Telecommunications Conference (GLOBECOM)*, pp. 1–5, Nov. 27-Dec. 1, 2006.

7. M.-T. Sun, X. Ma, G. Zhao, and X. Liu, "Improving geographical routing for wireless networks with an efficient path pruning algorithm," *Proc. the Third Annual IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, pp. 246–255, Sep. 25-28, 2006.

8. G. Zhao, R. Barnard, P. Diver, R. Hritcu, A. Turkmen, X. Zhang, A. Langville and Z. Li, "Website volume prediction", *Industrial Mathematical & Statistical Modeling Workshop (IMSM)*, Raleigh NC, Aug. 2006.

9. G. Zhao, D. Zheng and K. Chen, "Design of single sign-on," *IEEE CEC04EAST*, Beijing, Sep. 2004.

10. G. Zhao, D. Zheng and K. Chen, "Key management in sensor networks," *ISNG*, Las Vegas, NV, Nov. 2004.

JOURNAL PUBLICATIONS

1. G. Zhao, X. Liu, and A. Kumar, "Destination clustering geographic multicast for wireless sensor networks," *International Journal of Sensor Networks*, submitted, Oct. 2007.

2. X. Liu, G. Zhao, and X. Ma, "Target localization and tracking in noisy binary sensor networks with known spatial topology," *Wiley Wireless Communications and Mobile Computing*, to appear, Mar. 2007.

3. G. Zhao, X. Liu, M.-T. Sun and X. Ma, "Energy efficient geographic routing with virtual anchors based on projection distance," *Elsevier Computer Communications*, vol. 31, pp. 2195–2204, Feb. 2008.

4. X. Ma, M.-T. Sun, G. Zhao, and X. Liu, "An efficient path pruning algorithm for geographical routing in wireless network," *IEEE Trans. Vehicular Technology*, to appear, May 2006.