Load Balanced and Constant Stretch Hole Bypassing Protocol for Wireless Sensor Networks

NGUYEN Phi Le

Doctor of Philosophy

Department of Informatics
School of Multidisciplinary Sciences
The Graduate University for Advanced Studies,
SOKENDAI

Load Balanced and Constant Stretch Hole Bypassing Protocol for Wireless Sensor Networks

by

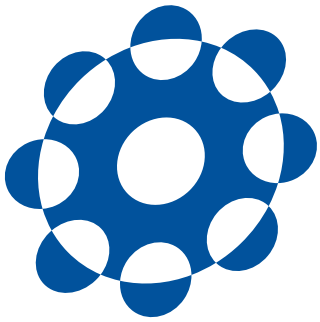
NGUYEN Phi Le

Dissertation

submitted to the Department of Informatics

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy



SOKENDAI (The Graduate University for Advanced Studies)

March 2019

Committee

Advisor	Dr. JI Yusheng
	Professor of National Institute of Informatics/SOKENDAI
Subadvisor	Dr. FUKUDA Kensuke
	Associate Professor of National Institute of Informatics/SOKENDAI
Subadvisor	Dr. KOIBUCHI Michihiro
	Associate Professor of National Institute of Informatics/SOKENDAI
Examiner	Dr. ABE Shunji
	Associate Professor of National Institute of Informatics/SOKENDAI
Examiner	Dr. KANEKO Megumi
	Associate Professor of National Institute of Informatics/SOKENDAI
Examiner	Dr. TEI Kenji
	Associate Professor of Waseda University

Abstract

Recent years have witnessed the emergence of wireless sensor networks (WSNs) which consist of tiny sensor nodes deployed over a region of interest to monitor and control the physical environment. Wireless sensor networks have been widely used in various domains such as military target tracking and surveillance, natural disaster relief, agricultural and environmental monitoring, biomedical health monitoring, etc. In this dissertation, we focus on large-scale WSNs which are based on the cooperation of a large number of sensor nodes. Typically, large-scale WSNs are used in monitoring applications such as agricultural monitoring, climate monitoring, forest monitoring, weather monitoring, etc.

Due to the short-range communication nature of sensor nodes, data transfer in WSNs is usually performed in the multi-hop communication paradigm. Hence, routing becomes one of the most important issues which has received intensive research attention. As sensor nodes are equipped with only limited and non-rechargeable batteries, conserving energy consumption is an important factor in designing routing protocol. Energy conservation can be achieved by *shortening the routing path* and *reducing the overhead* caused by control packets. In many applications, the network can't achieve its objective if all the nodes can't sense or report the sensed data, thus the death of even only one node may cause the network to operate un-functionally. Accordingly, *balancing traffic* over the network to extend the network lifetime is another important designing factor of routing protocol in WSNs.

Geographic routing has been widely used in wireless sensor networks because of its simplicity and efficiency resulted from its local and stateless nature. However, when subjected to routing holes (i.e., regions without sensor nodes that have communication capability), geographic routing suffers from the so-called local minimum phenomenon, where packets are stopped at the hole boundaries. The traditional scheme (named as

perimeter routing) is forwarding packets along the hole perimeters. However, this scheme satisfies neither of the requirements described above. Specifically, perimeter routing suffers from two serious problems: routing path enlargement and data concentration around the hole boundaries. Recently, several approaches have been proposed to address these two problems, wherein a common idea is to form a forbidden area around each hole from which packets are kept to stay away. Several proposals attempt to use simple forbidden areas which have certain selected shapes such as a circle, an ellipse, a hexagon, a quadrilateral, etc. Although the simplicity of these forbidden areas can help to reduce the control overhead, a significant enlargement on the routing path may occur due to the possibly significant difference between the forbidden area and the hole. To deal with the routing path enlargement problem, a recent proposal describes the forbidden area as the exact polygon whose vertices are the nodes on the hole boundary. However, this approach may incur with a significant control overhead since the information needed to represent the forbidden area may be very large when holes become complicated. Moreover, due to the static nature of the forbidden areas and routing paths, all of the existing protocols cannot solve the load imbalance problem thoroughly.

In this dissertation, we propose a novel protocol for bypassing multiple holes in wireless sensor networks which can balance the traffic over the network while ensuring the constant stretch property of routing paths. The key ideas behind our approach are to use elastic forbidden areas and dynamic routing paths. Specifically, to deal with the routing path enlargement problem we define the forbidden area of each hole as an equiangular polygon circumscribing the hole whose number of the vertices is determined on the basis of the required stretch upper bound. Moreover, to improve the load balance, we propose to use dynamic routing paths that vary for every packet, even for the packets of the same source-destination pair. The length of every routing path is controlled to guarantee the stretch upper bound. The theoretical analysis proves that the routing path stretch of the proposed protocol can be controlled to be as small as $1+\varepsilon$ (for any predefined $\varepsilon > 0$). The extensive experimental results show that our protocol strongly outperforms state-of-the-art protocols in term of load balancing.

The main contributions of the dissertation can be summarized as follows.

1. We analyze the geometrical characteristics of the forbidden area theoretically and figure out the relationship between the forbidden area and the routing path stretch. Based on the analysis results, we then propose an algorithm to construct the forbidden area which can guarantee the upper bound of routing paths. Moreover, the

information needed to represent the forbidden area is a constant which does not depend on the hole complexity.

2. Based on the proposed forbidden area, we propose a load balanced hole bypassing routing protocol (named BSMH) whose stretch is upper bounded by a predefined threshold $1+\varepsilon$, where ε is an arbitrary positive number.

3. We perform a theoretical analysis of the computational complexity of BSMH as well as the impact of parameter ε on the performance of BSMH.

4. We prove the constant stretch property of BSMH theoretically.

5. Finally, we conduct extensive experiments to evaluate the performance of BSMH and compare it with the state-of-the-art benchmarks.

To the best of our knowledge, BSMH is the first hole bypassing routing protocol addressing at the same time the three important designing factors described above, i.e., minimizing the routing path length, minimizing the control overhead and maximizing the load balance. Moreover, through both the theoretical analysis and practical experiments, the superiority of BSMH has been proved.

Dedicated to my Family.

Acknowledgements

Finally, I am going to the last milestone of my phd journey with this thesis becomes a reality. I would like to express my thanks to those who made this thesis possible.

First and foremost, I owe my deepest gratitude to my supervisor, Professor JI Yusheng, for her valuable advice, kind support and deeply understanding in both research and private issues. The trust and the encouragement that she gave me was a great motivation for me to always try my best. I really appreciate the advice and instructions she gave me during my past three years, which have grown me up. Without her, this thesis would not have been possible.

I also would like to express my sincere gratitude to my other supervisors, Professor ABE Shunji, Professor FUKUDA Kensuke, Professor KOIBUCHI Michihiro, Professor KANEKO Megumi and Professor TEI Kenji whose encouragement, insightful comments and hard questions helped me improve the thesis a lot.

I also like to acknowledge the NII international support team for the endless effort to provide a good research environment.

I wish to present my special thanks to my colleagues and my research team at Hanoi university of science and technology for their kind supports. My thanks go to Professor Nguyen Khanh Van for his useful discussions, and Professor Nguyen Thi Thu Trang, Professor Nguyen Thanh Hung for always being nice and willing to help me. I would like to thank all of the members in my research team who have made great contributions to this research.

I would like to show my warm thanks to all Kei lab members for the friendly atmosphere which made my every working day be comfortable, and for the thoughtful comments in the group meetings which helped improving my research.

I am deeply grateful to Vietnamese guys in NII who made my days full of joys. Especially, thank Nguyen, Ly, An for always being available whenever I need.

My special thanks send to my dearest friend who is always along with me and supports me.

Last but not least, from the bottom of my heart, I would like to send the warmest thanks to my family, especially my Parents, my Husband and my Sons for the understanding and sacrifice they always give me. My huge thanks and love send to my husband and my sons for the endurance they have been suffered during the past three years. Without them, I could not do anything.

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Preface

In the last decades, wireless sensor networks (WSNs) have gained increasing worldwide attention from both academic community and actual users. A WSN is composed of sensor nodes deployed over a region of interest to monitor and control the physical environment. Typically, a sensor node is equipped with four basic components: a sensing subsystem which consists of one or more sensors for data acquisition from the physical surrounding environment, a processing subsystem consisting of a memory for storing data and a controller to perform computation, a wireless communication component for data transmission and a power supply component. The sensor nodes collect information about physical phenomena such as temperature, light, sound, pressure, etc. from the surrounding environment and collaboratively transfer the sensory data to base stations for further processing. Recently, applications of WSNs have been widely used in various domains including military target tracking and surveillance [1][2], natural disaster relief [3, 4, 5, 6, 7], agricultural and environmental monitoring [8, 9, 10, 11, 12], biomedical health monitoring [13, 14, 15], etc.

In this dissertation, we focus on large-scale wireless sensor networks which are based on the cooperation of a significant number (e.g. thousands) of sensor nodes. Typically, large-scale wireless sensor networks can be found in monitoring applications such as agricultural monitoring, climate monitoring, forest monitoring, weather monitoring, battlefield surveillance, etc. With the proliferation in Micro-Electro-Mechanical Systems (MEMS) technology, the development of low-cost and multifunctional sensor nodes have become reality. The appearance of these low-cost and high multifunctional sensor nodes has promoted the broaden of large-scale wireless sensor networks and made it an important part of wireless sensor networks. Due to the short-range communication nature of the sensor nodes, in WSNs, data transfer is usually conducted in a multi-hop communication paradigm. Therefore, routing becomes a very important task which is needed for sending data between the nodes. Different from traditional networks, WSNs have their own resource constraints which pose many challenges on designing routing protocols. Specifically, wireless sensor networks endure from inherent limitations on the power supply, computational capacity and memory in each sensor node. The authors in [8] showed that most of the available sensors are equipped with the memory which is less than 640 Kb. Moreover, sensor nodes can only be equipped with a very limited power source which usually less than 5V [16][8]. In addition, it could be impossible or inconvenient to recharge the nodes' battery because the number of sensors is usually large and they may be deployed in remote, unattended, and hostile environments. Because of these resource limitations, routing protocols for wireless sensor networks should be made as simple as possible.

Geographic routing thus becomes one of the most popular routing protocol in wireless sensor networks due to its simplicity and efficiency. Geographic routing is a stateless routing protocol which does not use routing table, instead exploits the location information of 1-hop neighbors to make the routing decision. Such a routing protocol typically starts with a greedy strategy whereby each node chooses the next hop to be the neighboring node which is closer to the destination than the current node and closest to the destination among all the neighbors of the current node. Geographic routing performs well in dense wireless sensor networks. However, with the occurrence of holes (regions without working sensors) [17], this routing protocol suffers from a serious drawback called the *local minimum phenomenon* [18], i.e., the forwarding process is stopped at the hole boundary because there is no neighboring node closer to the destination than the current node. Holes are formed either due to the presence of some

geographical obstacles or because of the failure of sensor nodes due to various reasons such as battery depletion or the node being destroyed by external forces (e.g., by fire or earthquake). As wireless sensor networks (especially environmental monitoring networks) are usually deployed in harsh conditions which may contain many obstacles, holes often appear and become one of the most critical problems in dealing with routing in large-scale wireless sensor networks.

1.2 Criteria in designing routing protocol for large-scale WSNs

Due to the limitation on the power supplement of sensor nodes, energy conservation is one of the most critical issues in designing routing protocol for WSNs (see details in Section 2.3). The energy consumed by the routing process is comprised of two parts: the first one is for computation and the second one is for transmission. According to the experimental results shown in [19, 20], the energy expenditure for computation is insignificant compared to those consumed by transmission. Specifically, the energy needed for transmitting a single bit is approximately the same as that needed for processing a thousand operations [20]. Therefore, to conserve the energy consumption of sensor nodes, routing protocols must reduce the number and the size of packets that are transmitted. This can be done by shortening the routing path and reducing the number and the size of the control packets.

In wireless sensor networks, especially in environmental monitoring applications, every sensor node is usually required to accomplish two demanding tasks simultaneously, sensing and communicating. Specifically, in many WSN applications, besides communication capability, full sensing coverage is also required. In such scenarios, the network can achieve its objective only if all sensor nodes can sense and report the sensed data. Hence, the death of even only one node may cause the network to operate un-functionally. For examples, considering a battlefield surveillance wireless sensor network where the sensors have to detect and report any event occurring. As enemies can appear anywhere at any time, all sensors need to operate functionally to capture all events that may happen. As another example, considering an earthquake forecasting system where the prediction is given based on the data collected from the sensors. The death of a sensor locating at the critical region (e.g., the center of the earthquake) may

reduce the forecast accuracy significantly. As we don't know in advance which sensors will belong to the critical region, all sensors need to operate functionally at all the time. In the other word, it can be said that the lifetime of the network (i.e., the time duration during which the network operates properly) is decided by the lifetime of the sensor node which depletes the energy first. Hence, one of the most critical problems is to maximize the network lifetime, i.e., the minimum lifetime of sensor nodes. The network lifetime can be extended by applying various techniques such as scheduling sensor nodes to alternatively going to sleep state, adjusting the power level of sensor nodes, etc. In the context of routing protocol, to prolong the network lifetime, we need to evenly distribute the traffic load over all the nodes.

Therefore, we consider the following three provisions for the design of routing protocols in large-scale wireless sensor networks:

- Routing path length minimization.
- Control overhead maximization.
- Load balance maximization.

1.3 Existing hole bypassing protocols and their problems

The traditional scheme for bypassing the hole, which is called as *perimeter routing*, is to switch off the usual greedy forwarding mode and instead, manage to route packets along the hole boundary. The perimeter routing approach can be seen at very first works addressing hole problem such as [21][22]. However, the perimeter routing approach satisfies neither two first requirements described above. First, the routing paths along the hole perimeter are usually much longer than the shortest path, especially in the cases where the holes contain many concave areas. Indeed, the authors in [23] have proved that the routing path length may grow as much as $\Omega\left(c^2\right)$[1] when holes are present, where c is the optimal path length [23]. Secondly, the perimeter routing approach suffers from the load imbalance around the hole perimeter. Specifically, as a lot of packets will be sent along the hole boundary, the boundary nodes are imposed heavier traffic than the other nodes. Therefore, the boundary nodes are depleted of energy quickly and the network lifetime thereby is shortened.

[1] $\Omega(f(n))$ means that for large enough n, $\Omega(f(n)) \geq f(n)$ for some constant k.

To address the drawbacks of the perimeter approach, several methods have been proposed, wherein a common approach (which we named as *forbidden approach*) is to approximate the hole by a simple shape. This shape is often a minimum cover of the hole, and is considered as a special forbidden area (as we call it) from which the packets are made to stay away. The information of this forbidden area is disseminated to the surrounding area to establish a hole awareness. This hole awareness is then utilized to discover detour routes. In some proposals, the authors use the forbidden area that has a certain selected and simple shape, e.g., a circle [24], a hexagon [25], an ellipse [26] or a quadrilateral [27]. Although these approaches can alleviate the traffic concentration surrounding the holes, they may create traffic congestion around the perimeter of the forbidden area and thus cannot solve the load imbalance problem thoroughly. Moreover, the routing path enlargement problem remains unsolved because the difference between this forbidden area and the original hole can be quite large.

In a recent approach addressing the path enlargement problem [28], a hole is compactly described by a simple polygon, i.e., the smallest convex polygon covering the hole. Although this approach can guarantee the upper bound stretch[2] of all routes as $\Theta(c)$, where c is the path length of the optimal route, it still suffers from the load imbalance problem due to the traffic concentration around the convex polygon. Moreover, this approach cannot deal with the scenario where the sources and the destinations stay inside the forbidden area (we call this scenario as routing in the hole vicinity). Later on, the authors in [29] addressed the problem of routing in the hole vicinity by using the forbidden area as the polygon whose vertices are all the sensor nodes stay on the hole boundary. In spite of the fact that this is a rare method which can guarantee the upper bound stretch of routing paths in the hole vicinity scenario (i.e. the upper bound stretch is $\Theta\left(\frac{D}{\gamma}\right)$, where D is the diameter of the network and γ is the communication range of sensor nodes), it still faces the traffic congestion problem around the hole boundary. Moreover, this approach may result in extra resource overhead caused by disseminating and storing information of the forbidden areas when the holes become complicated.

In summary, it can be said that the existing hole bypassing routing protocols suffer from two problems as follows.

- The forbidden areas proposed so far are either too simple or too complicated. The too simple forbidden areas may cause a significant gap to the holes and thus lead

[2]The stretch of a routing path from a source node s to a destination t is the ratio of the number of hop counts of the routing path to the number of hop counts of the shortest path from s to t.

to the routing path enlargement problem. In the meanwhile, the too complicated forbidden areas may result in a large resource overhead in disseminating and storing information of the forbidden areas.

- None of the existing protocols can thoroughly solve both the two problems: routing path enlargement and load imbalance.

1.4 Our goals and approach

Motivated by the problems described above, in this dissertation, we focus on the hole problem in large-scale wireless sensor networks. We then propose a hole bypassing routing protocol which, to the best of our knowledge, is the first one addressing at the same time the three designing factors described in Section 1.2, i.e., minimizing the routing path length, maximizing the load balance and minimizing the control overhead. Specifically, we aim at designing a distributed geographic routing protocol that generates dynamic routing paths with the stretch upper bounded by a constant. This constant can be controlled to be as small as $1+\varepsilon$ (ε is a predefined positive number, which we call the *stretch factor*). The variation of the routing paths ensures the load balance over the network.

The keywords behind our approach include *adaptive forbidden areas* and *dynamic routing paths*. First, to solve the routing path enlargement problem while minimizing the setup overhead, we use *adaptive forbidden areas*. Specifically, our forbidden area of a hole is the hole's circumscribing convex polygon whose number of the vertices is determined on the basis of the required routing path stretch upper bound [3]. Clearly, the description information of such a forbidden area depends only on its number of vertices, and the difference between the forbidden area and the hole can be elastically made as small as desired by using a sufficiently large number of vertices. It means that we can control both the complexity and the approximation error of the forbidden areas by adjusting its' number of vertices. Specifically, given the stretch upper bound of $1+\varepsilon$, our protocol will decide the number of vertices of the forbidden area such that the stretch upper bound is guaranteed.

[3]The stretch of a routing path is defined by the ratio of the real routing path's hop counts to the shortest routing path's hop counts. The stretch of a routing protocol is the greatest stretch of routing paths determined by that protocol.

Second, our idea for dealing with load imbalance problem is to use *dynamic routing paths*. We attempt to design routing paths that vary for every packet, even the packets of the same source-destination pair. Specifically, for each packet, the source node determines a so-called base path which is a Euclidean path bypassing all the forbidden areas. The base path then is magnified using homothetic transformations to obtain a Euclidean routing path which will be used as the guideline for the packet. On the one hand, the homothetic centers are randomly chosen to provide the diversity of the routing path. On the other hand, the scale factors are controlled to guarantee the required routing path upper bound. The coordinates of the Euclidean path's vertices are inserted into the packet header as virtual anchors. The packet then is greedily forwarded towards the virtual anchors gradually until reaching the destination. Figure 1.1 sketches an overview of our approach.

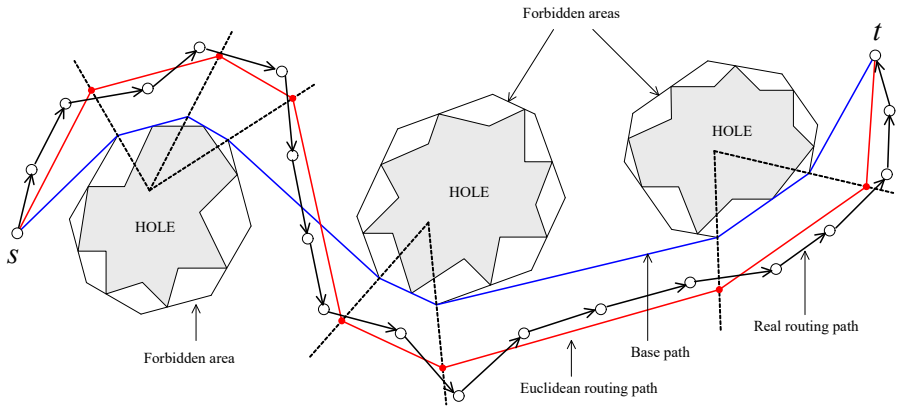


Figure 1.1: Illustration of our approach.

1.5 Main contributions

Our main contributions are as follows.

- First, we present a rigorous theoretical analysis on the geometrical characteristics of the forbidden area and figure out the relationship between the forbidden area and the routing path stretch. Based on the analysis results, we then propose a strategy to construct a forbidden area which can guarantee the stretch upper bound. Moreover, the information needed to represent the forbidden area is a constant which does not depend on the hole complexity.

- Second, based on the proposed forbidden area, we propose a load balanced hole bypassing routing protocol (named BSMH) whose stretch is upper bounded by a predefined threshold $1+\varepsilon$, where ε is an arbitrary positive number.
- Third, we perform theoretical analysis on the computational complexity of BSMH as well as the impact of the parameter ε on the performance of BSMH.
- Fourth, we prove the constant stretch property of BSMH theoretically.
- Finally, we conduct extensive experiments to evaluate the performance of BSMH and compare it with the state-of-the-art benchmarks.

1.6 Dissertation organization

Chapter 2 reviews the literature. Specifically, we introduce the wireless sensor networks and geographic routing in Section 2.1. Then, we describe the hole problem and present typical methods to alleviate the hole problem including perimeter routing and forbidden area-based approach in Section 2.4 and 2.5.

In Chapter 3, we describe our proposed protocol (BSMH) in details. We first present our proposed protocol to determine the forbidden areas in Section 3.3. Then, we describe our data forwarding scheme in the next sections. Specifically, in Section 3.4, we start by the simplest network topology which consists of only one hole, and the sources and the destinations of all packets stay outside of the hole's forbidden area. We move to a more complicated scenario where the sources and the destination may stay inside the hole's forbidden area in Section 3.5. In Section 3.6, we study the challenges in dealing with multiple holes and describe our strategies for bypassing multiple holes. Finally, based on the results attained from Sections 3.3, 3.4, 3.6, and 3.5, we propose our routing protocol in the most general network topology where multiple holes may exist and the sources and the destinations of packets may stay inside the holes' forbidden areas in Section 3.7.

We make a thorough theoretical analysis of BSMH in Chapter4. Specifically, we first investigate the computational complexity of BSMH in Section 4.1. Then, in Section 4.2, we analyze the impact of the scale factor (i.e., the required stretch upper bound) on the performance of BSMH. Finally, we prove the constant stretch property of BSMH in Section 4.3.

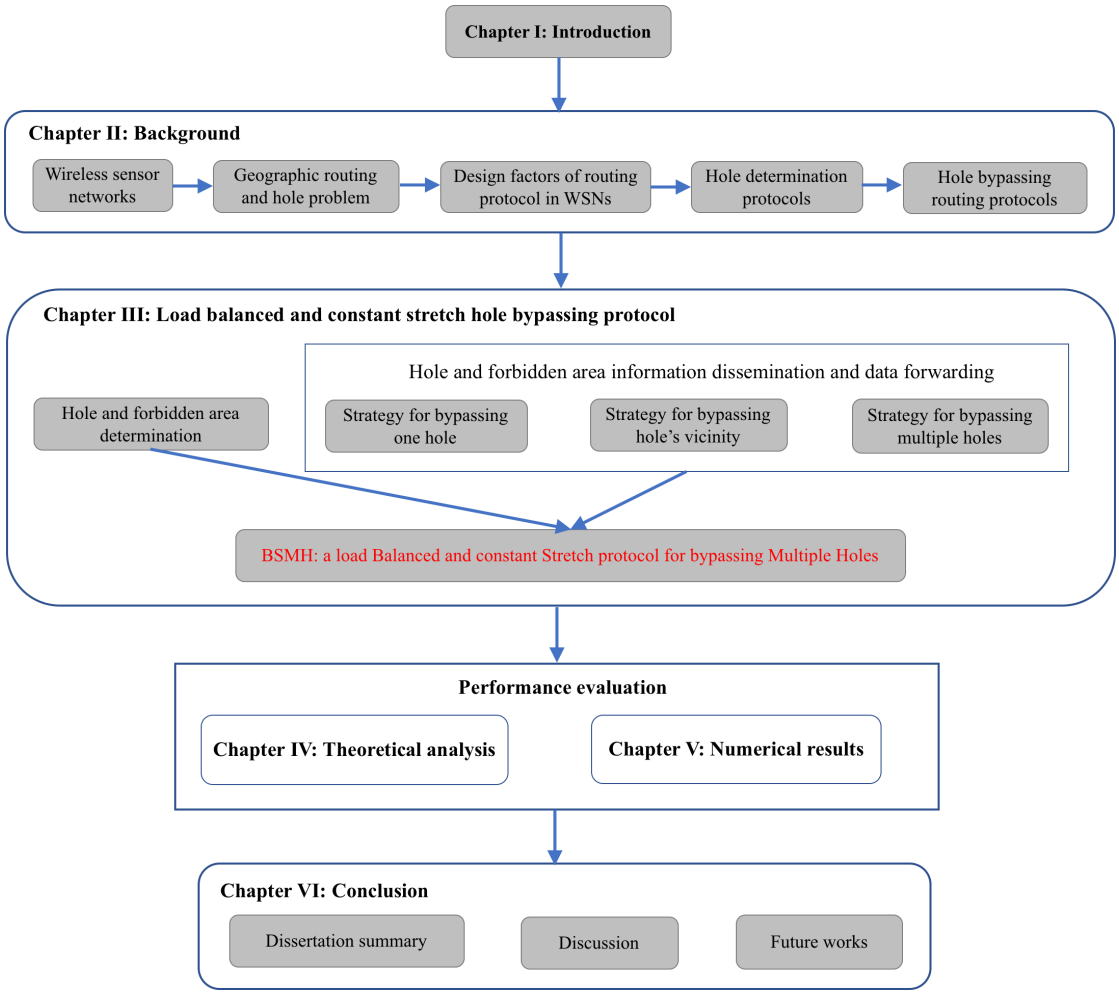


Figure 1.2: Illustration of the dissertation organization.

Chapter 4 presents extensive experiment results for comparing performance of BSMH and existing hole bypassing routing protocols.

In Chapter 6, we first summary the dissertation in Section 6.1. We then discuss some techniques to enhance the performance of our proposed protocol, as well as to customize the proposed protocol for dealing with dynamic holes in Section 6.2.1. Finally, Section 6.3 describes our future works briefly.

Figure 1.2 illustrates the dissertation organization.

2 Background

In this chapter, we describe the background of our research. We start by introducing the wireless sensor networks and their applications in Section 2.1. In Section 2.2, we describe geographic routing, the most popular routing protocol in wireless sensor networks, and present its most critical issue: routing hole problem. Section 2.3 presents our survey about criteria in designing routing protocols for WNSs. Then, in Section 2.4 we review the hole determination algorithms. Finally, we describe the hole bypassing protocols proposed in the literature and raise their problems in Section 2.5.

2.1 Wireless sensor networks

Recent years have witnessed the emergence of wireless sensor networks (WSNs) which consist of a large number of tiny sensor nodes equipped with the capabilities of sensing, processing and transmitting. In a WSN, sensor nodes collect information about physical phenomena such as temperature, humidity, vibrations, etc from the surrounding environment and transfer to the base stations for further processing (Fig.2.1). Typically, a sensor node is a tiny device including three components: sensing subsystem,

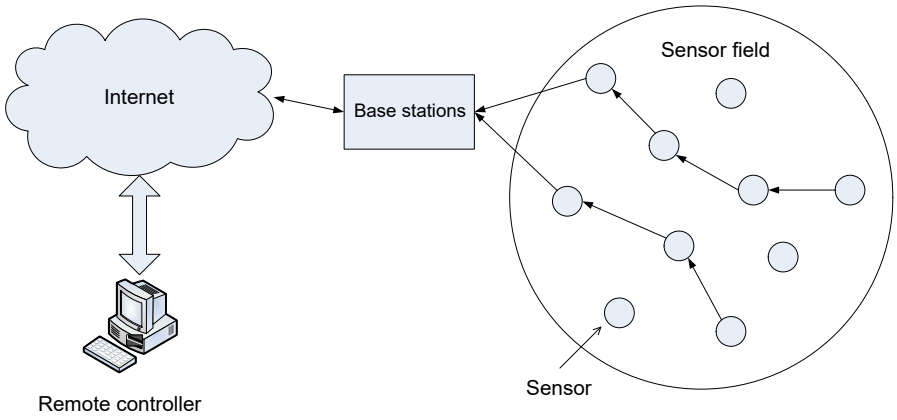


Figure 2.1: Wireless sensor network.

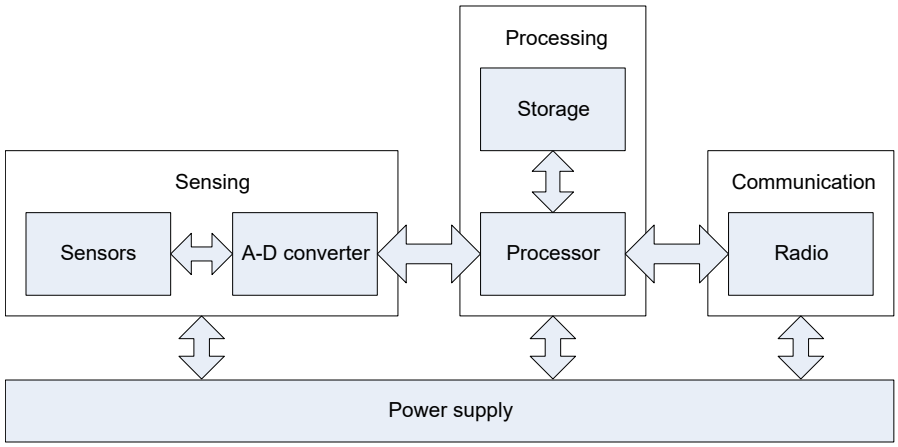


Figure 2.2: The components of a sensor node.

processing subsystem and communication subsystem as shown in figure 2.2).

The sensing subsystem consists of one or more sensors for data acquisition from the physical surrounding environment. The processing subsystem contains a memory which stores the sensed/aggregated and other data, and a controller performs computations before sending data to the other nodes. The wireless communication is the radio part which enable the communication between the sensor nodes and the base stations. Finally, the power source supplies the energy required by the device to perform all the tasks.

Different from early wired networks, sensor networks can contain a large number of nodes which may be up to hundreds or even thousands. Furthermore, the sensor nodes suffer from the inherent resource limitations. According to a survey in [8], sensor nodes

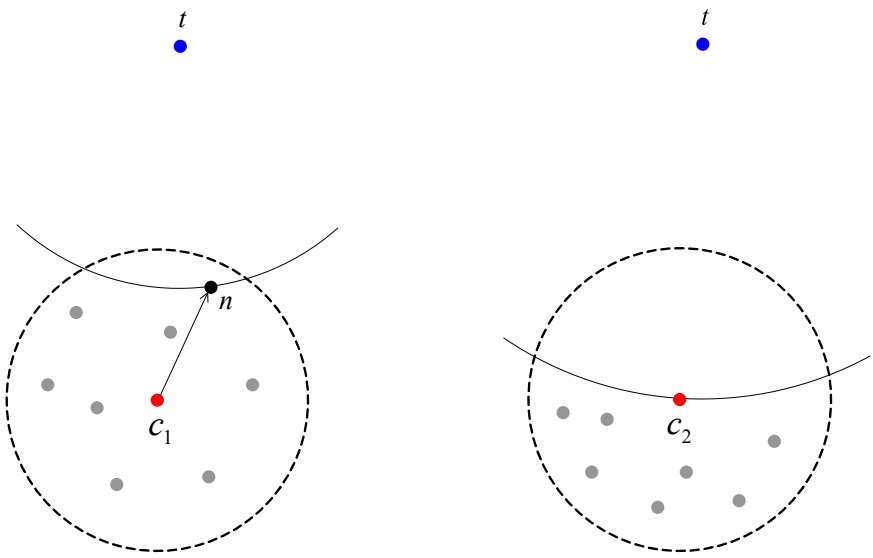
are equipped with only limited batteries that varies from $2.7V$ to $4.5V$. Moreover, this limited battery could be impossible or inconvenient to be recharged because the nodes may be deployed in a hostile or unpractical environment. In the same survey, the authors also showed that most of the available sensors is equipped with the memory which is less than 640 Kb.

Wireless sensor networks have a wealth of applications [30, 31, 32, 33, 34]. Especially, they are widely used in monitoring and investigating certain landscapes or environments which may be too large or remote for deploying wired network infrastructure or of too harsh conditions that are not suitable for traditional surveillance by human beings [3, 4, 5, 6].

2.2 Geographic routing and hole problem

Due to the short range wireless communication nature of the sensor nodes, data transmission in WSN is performed in multi-hop paradigm. Therefore, routing becomes an important issue which attracts a lot of attention from research community [35]. Inherent characteristics of WSN including limited energy supply, limited computation power of sensor nodes and the large scale of network topology pose many challenges to the design of routing protocol. Basically, to conserve the energy consumption, routing protocol should be made as simple as possible.

Geographic routing [36][21] which exploits the local geographical information at the GPS-equipped sensors is widely accepted for its simplicity and efficiency. Geographic routing algorithms typically assume: a) that each network node knows its own and its neighbors' positions and b) that the source of a message knows the destination's position. The simplest geographic routing protocol is the greedy forwarding where each node chooses the next hop to be the neighbor node with the most geographical advantage to the destination. Figure 2.3 illustrates the greedy forwarding strategy. In this figure, t is the destination and c_1 is an intermediate node on the routing path. Since n is a neighbor of c_1 which is closer to the destination than c_1, and closest to the destination among all the c_1's neighbors, n is chosen as the next hop. Geographic routing works well in networks where the nodes are deployed densely. However, with the occurrence of void areas (which are also called routing holes), i.e. the regions where the nodes have died out and hence, no longer participated in the network communication, geographic routing suffers from the so-called local minimum problem where there is no neighbor which is



(a) Greedy forwarding. (b) Local minimum phenomenon.

Figure 2.3: Geographic routing and local minimum problem.
The blue dot and the red dot represent the destination and the current node, respectively.
The dotted circle represents the sensing range of the current node, inside which are the current node's neighbors.
(a) The next hop (the black dot) is the neighbor which is closest to the destination.
(b) The packet is stuck since there is no neighbor which is closer to the destination than the current node.

closer to the destination than the current node [37]. For example, in Fig. 2.3(b), the packet will be stuck at c_2 since all the neighbors of c_2 are farther from the destination than c_2. In practice, the holes in wireless sensor networks can be formed either due to the presence of some geographical obstacles such as buildings, lakes or because of the failure of sensor nodes due to external destroying (e.g. fire, earthquake, etc).

2.3 Design factors of routing protocol in WSNs

We have conducted a survey over 120 articles about routing protocols to clarify the focus of research community on the designing factors. The articles are the conferences papers and journal articles published beyond 2010. According to our survey, the considered factors can be classified into 7 main groups as follows:

1. Delay: which includes metrics related to the end-to-end delay of packets such as average delay, maximum delay.

2. Routing path length: The length of a routing path is usually measured by the number of hop counts. This group consists of metrics such as average route length, maximum route length, average stretch, maximum stretch.

3. Packet delivery number/ratio: indicating the percentage or the number of packets/bytes that correctly arrive at the destinations. Typically, this group contains factors such as successful delivery rate, loss rate, on-time packet rate, the number of bits correctly received by the destinations.

4. Energy consumption: measuring the energy consumed by the nodes. This group contains various factors such as average energy consumption per node/per packet/per byte, total energy consumed by all the nodes, the average residual energy of all the nodes.

5. Load balance: indicating how well the load is distributed over the network. Typically, this group includes the factors such as deviation/variance/CDF of the residual/consumed energy of all the nodes, percentage/number of dead nodes over simulation time, maximum consumed/residual energy of a node, the number of nodes that become isolated. Some researches may not use specific quantify factors but they may visualize the load balance by using a distribution map of residual/consumed energy or the traffic load of all the nodes.

6. Network lifetime: There are numerous definitions about the network lifetime such as the time until the first node dies, the time until a certain percentage of nodes die, the time until the last node dies, the time until when the sink is unable to receive packet sent from the sources, the time until when nodes deplete a certain amount of their energy.

7. Control overhead: which includes all overhead spent on transmitting control packets or for processing and storing control data. Typically, this group consists of the following factors: the total number of control packets, energy consumption for controlling, overhead for storing the routing table and the neighbor table.

Besides the factors in these 7 groups, some other factors may be considered for special design objectives. For examples, a cluster-based routing protocol may consider the *ratio*

of single-node clusters as a design factor [38]; a routing protocol for networks with mobile sink may investigate the *convergence time* which is defined as the time required by a source to obtain the new location of a mobile sink [39]. However, these factors are minor and rarely used.

Protocol	Energy consumption	Delivery ratio	Delay	Network lifetime	Control overhead	Routing path length	Load balance	Others
EQSR[40]	✓	✓	✓					
EBRP[41]	✓	✓		✓			✓	
DRINA[42]		✓			✓			
EBGR[43]	✓	✓			✓			
-[44]				✓				
ENS_OR[45]	✓	✓		✓			✓	
Ring Routing[46]	✓		✓	✓				
LOCALMOR[47]		✓	✓	✓			✓	
EQGOR[48]		✓	✓		✓			✓
- [49]	✓			✓			✓	
QELAR[50]	✓	✓		✓	✓	✓	✓	
QoSNET[51]		✓	✓	✓			✓	
ALBA-R[52]	✓	✓	✓			✓		
Elastic[39]	✓	✓	✓		✓			✓
IGP[53]				✓				
BFP[54]		✓	✓			✓		
DASDR[55]		✓					✓	✓
GeoSpray[56]		✓	✓					✓
ILSR[57]					✓	✓		
QoS-PSO[58]	✓	✓	✓					✓
ERA[59]	✓			✓			✓	
HydroCast[60]		✓	✓		✓			✓
-[61]	✓	✓	✓		✓			
NC-RMR[62]	✓	✓					✓	
ADA[63]		✓						✓

Protocol	Energy consumption	Delivery ratio	Delay	Network lifetime	Control overhead	Routing path length	Load balance	Others
IHSBEER[64]	✓			✓			✓	
EFFORT[65]	✓		✓	✓				
MaxEW[66]				✓			✓	
VAPR[67]	✓	✓	✓					
ROL[68]	✓	✓	✓	✓			✓	
GMCAR[69]	✓	✓		✓	✓	✓	✓	
$(ACH)^2$[70]		✓	✓	✓			✓	
-[71]		✓		✓				
CMRP[72]	✓	✓	✓	✓	✓			
GMR[73]		✓						
GEDAR[74]	✓	✓	✓		✓			
AEMRP-LB[75]			✓				✓	
LASeR[76]	✓	✓	✓		✓			
-[77]	✓	✓	✓					
DFCR[78]	✓	✓					✓	
VCP[79]		✓	✓		✓	✓		
H2-DAB[80]	✓	✓	✓					
CAR[81]	✓							
TAR[82]		✓						
ProHet[83]		✓			✓	✓		
ECPSOA[84]	✓	✓	✓					
RASeR[85]	✓	✓	✓		✓			
ERP-SCDS[38]	✓			✓		✓		✓
ECDC[86]	✓							
CS Routing[87]	✓		✓			✓		
GRR[88]					✓	✓		
OMLRP[89]	✓	✓				✓		
E^2R^2[90]	✓	✓		✓			✓	
CBER[91]				✓				

Protocol	Energy consumption	Delivery ratio	Delay	Network lifetime	Control overhead	Routing path length	Load balance	Others
MQoSR[92]	✓	✓	✓	✓				
PWDGR[93]	✓		✓	✓		✓		
ABC[94]		✓	✓		✓			
EIGR[95]	✓	✓	✓		✓			
PRTR[96]	✓	✓	✓					
BIOSARP[97]	✓	✓			✓			
DACR[98]	✓	✓	✓	✓	✓			
-[99]	✓						✓	
DECAR[100]	✓			✓			✓	
OHCR, OHA[101]	✓			✓			✓	
SEDR[102]	✓			✓				✓
BVR-VCM[103]	✓	✓	✓		✓	✓		
MRFTM[104]	✓	✓	✓					✓
EDAL[105]	✓		✓	✓	✓			
CLB-Routing[106]				✓			✓	
BiO4SeL[107]	✓	✓		✓				
SAERP[108]	✓						✓	✓
RELAX[109]	✓	✓	✓				✓	
GBRR[110]	✓			✓				
TCOR[111]	✓	✓	✓					
ECLDel[112]		✓			✓	✓	✓	✓
BADCR[113]	✓			✓			✓	
MACRO[114]	✓	✓	✓					
TORP[115]	✓	✓				✓		
-[116]		✓	✓					
QDGRP[117]	✓	✓	✓					
PCOR[118]		✓						✓

Protocol	Energy consumption	Delivery ratio	Delay	Network lifetime	Control overhead	Routing path length	Load balance	Others
VGE–MGP[119]						✓		
G-STAR[120]		✓	✓		✓	✓		
HPS[121]						✓		✓
LRP-QS[122]		✓	✓		✓			
E-TRAIL[123]	✓	✓		✓				
-[124]	✓	✓						✓
LVGR[29]					✓	✓		
WCDS-DCR[125]	✓			✓				
DCR[126]	✓	✓	✓					
GAEMW[127]	✓	✓		✓			✓	
DGSI[128]							✓	
EVRP[129]	✓	✓	✓		✓	✓		
RE^2MR[130]	✓	✓	✓			✓		
RSSR [131]		✓				✓		✓
MBR, MBE, MBC[132]	✓		✓				✓	
CAGR[133]		✓	✓		✓			
EDGR[134]	✓	✓	✓	✓				
REACT[135]		✓						✓
GOAL[28]					✓	✓		✓
-[136]		✓					✓	
BHOP-GR[25]			✓		✓	✓		
HDAR[137]					✓	✓		
HDDL[138]						✓		✓
-[139]	✓	✓	✓					
DLBM[140]	✓		✓				✓	
BRIDGE[141]	✓	✓			✓	✓	✓	
RDM[142]	✓				✓			✓
ROT[143]	✓	✓		✓		✓		

Protocol	Energy consumption	Delivery ratio	Delay	Network lifetime	Control overhead	Routing path length	Load balance	Others
BVR-VCM[144]	✓	✓	✓		✓	✓		
LDMR[145]	✓	✓	✓					
-[146]	✓	✓						
EMGR[147]	✓	✓			✓			
R-GVI[148]	✓	✓	✓		✓	✓		
BVR-VRC[149]	✓	✓	✓		✓	✓		
RGP[150]	✓				✓	✓	✓	✓
ViP[151]		✓			✓	✓		
CDF[152]						✓	✓	
HBF[153]	✓	✓	✓	✓				
-[154]						✓	✓	✓

Table 2.1: Factors in designing routing protcols for WSNs.

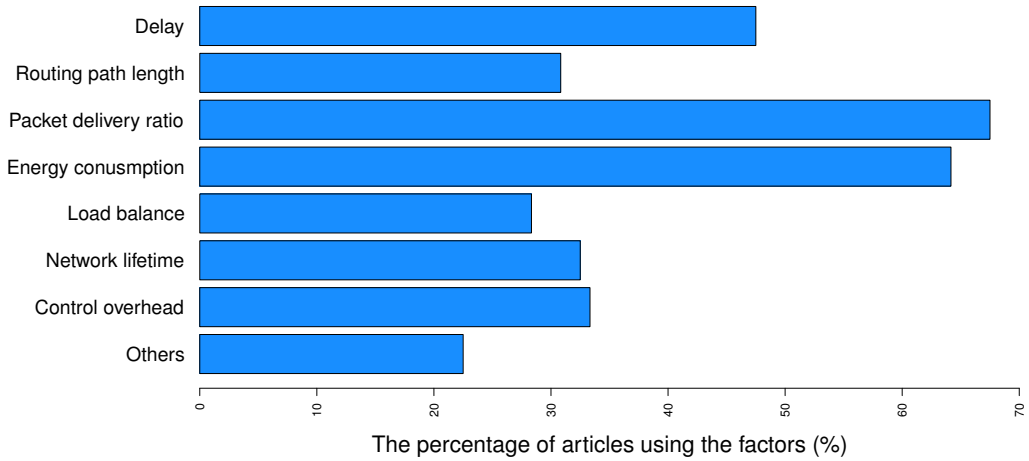


Figure 2.4: Usage of the factors.

Figure 2.4 summarizes the usage of the metrics.

As shown, the delivery ratio is the most important factor which has been investigated in more than 67% of the surveyed articles. This is obvious because the most important

mission of a routing protocol is to successfully delivery packets to the destination. The factor that gains the second-most concern is the energy consumption which has appeared in about 64.5% of the articles. This is due to the fact that sensor nodes have only very limited energy, thus conserving energy consumption is one of the most critical issues. 47.5% of the works tried to improve the packet delay and 30.8% of the works considered the routing path length as one of the design criteria. Load balance, network lifetime and control overhead gain less concern and each of them appeared in about 30% of the surveyed articles.

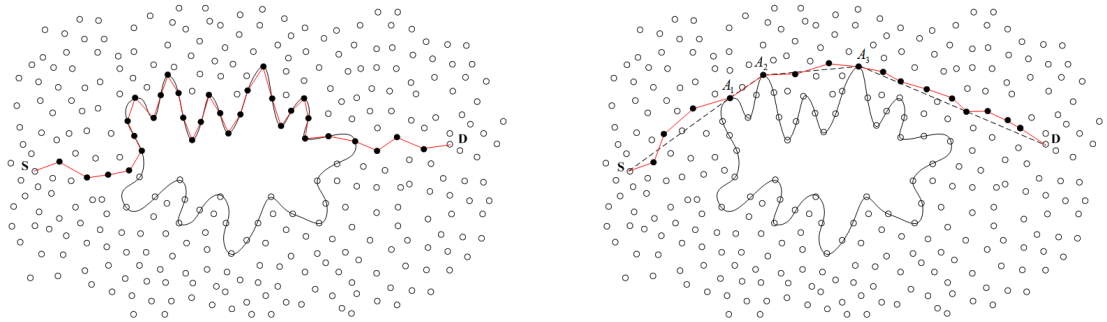
It is worth noting that the packet delay and routing path length have an extremely close relationship, i.e, the longer the routing path the higher the delay. Moreover, note that the energy expenditure of a sensor node consists of three components: sensing energy, processing energy and transmitting energy, where routing protocols contribute to only the last two components. In addition, according to the studies described in [19, 20], the energy consumed by data transmission is much expensive compared to that by computation, i.e., the energy needed for transmitting a single bit is approximately the same as that needed for processing a thousand operations. Consequently, in order to conserve energy consumption, routing protocols should reduce the number of transmitted packets. This is equivalent to reduce the routing path length.

Regarding the network lifetime, although there are numerous definitions, the *first node dead instant* has been used the most. Specifically, among 39 works that tried to extend the network lifetime, 82% of them (i.e., 32 articles) defined the network lifetime as the time duration until the first node dies. It means that in order to prolong the network lifetime, one should maximize the minimum lifetime of the sensor nodes. This can be done by balancing the traffic load evenly over all the nodes.

In consequence, the designing criteria of routing protocols in wireless sensor networks can be summarized as follows:

- Maximizing the number of packets successfully delivered
- Minimizing the routing path length
- Maximizing the load balance
- Minimizing the control overhead

2.4 Hole determination protocols



(a) A long route traditionally formed without the awareness about the hole.

(b) A short route cleverly formed by using the hole information.

Figure 2.5: Comparing possible routing paths, avoiding a large hole

Knowing the presence of a hole in advance can certainly help the nodes to find efficient routes going around the hole. Figure 2.5 illustrates this using a scenario with a large hole which has a rough face. Fig. 2.5(a) shows an unnecessarily long route which could be formed as without the awareness about the hole. While Fig.2.5(b) shows a much shorter route which could be formed by using the hole info.

An algorithmic approach for locating holes in WSN has been firstly introduced in [18] where Fang *et. al.,* proposed algorithms to obtain the exact boundary of the hole as a polygon with vertices being adjacent nodes on the hole side. The author first proposed a rule named TENT to identify the so-called stuck nodes where the local minimum phenomenon may happen. These stuck nodes actually are the nodes staying on the holes' boundaries. The authors also proposed an algorithm, named boundhole, to determine the hole boundary. The boundhole algorithm can be shortly described as follows. Each stuck node p initiates a HBD message (denoted for Hole Boundary Detection) which includes its location and sends it to p's leftmost node with respect to the stuck angle. The leftmost node can be defined as follows: p faces the area formed by the two rays of this angle, and uses the angle's bisector line to conduct counter-clockwise sweeping. The leftmost node is the first one that is met by the sweeping line. Upon receiving the HBD message, p's leftmost neighbor node writes its location into the message and passes it to the next-hop determined by the minimal sweeping angle from the previous hop. The HBD message will finally come back to node p from p's rightmost neighbor

with respect to the initial investigated angle, where the rightmost neighbor is defined in a similar way as the leftmost node. Basically, the HBD message creates a closed cycle that goes back to p.

There are many other hole detection algorithms have been proposed which can be classified into two main categories: geometric methods, and topo-logical methods. The geometrical approach uses the coordinates of the nodes and standard geometric tools such as the Delaunay or Voronoi diagrams to detect holes and their boundaries. Several simple distributed algorithms have been proposed in [155, 156] to detect the boundary nodes and build the routes around holes. In [157], Zhang *et. al.,* proposed an algorithm to detect the hole boundary nodes on the basis of Voronoi Diagram. The authors also described a method to calculate the accurate location of hole boundary by analyzing the sensing edges of the boundary nodes.

The topological approach requires neither coordinates nor localization of sensors. Instead, this approach uses topological properties such as the information of connectivity to identify the boundary sensors. In [158], W.C.Chu *et al.,* proposed a protocol to detect boundary nodes based on *contour line*. In their protocol, each node maintains a list of its 1-, 2- and 3-hops neighbors. Based on this list, each node can construct its 1-, 2-, and 3-contour lines. They argued that, a node s is a boundary node if its 2-contour can not enclose s. Ghrist and Muhammad *et al.,* [159] proposed a purely connectivity-based hole detection method. They constructed the Rips complex corresponding to the connectivity graph of the network and determined the hole by verifying whether the first homology group of the simplicial complex is trivial. Yan *et al.,* [160] adopted two types of simplicial complexes called Cech complex and Rips complex to identify holes and classify holes to be triangular and non-triangular. They also proposed a distributed algorithm on the basis of these complexes to determine the non-triangular holes. Another algorithm based on Rips complex has also been proposed in [161]. An obvious advantage of the topological approach is that it does not require accurate location information of sensor nodes. However, this approach suffers from a serious drawback that it can not determine the exact boundary of holes. In [162], F.Yan *et al.,* presented a closed-form expression of the upper bound and lower bound in using homology-based approach to detect holes. They also proposed a homology-based distributed protocol which can detect about 90% holes in about 90% cases. Other topological-based protocols can be found in [163, 164, 165].

2.5 Hole bypassing routing protocols

2.5.1 Perimeter routing approach

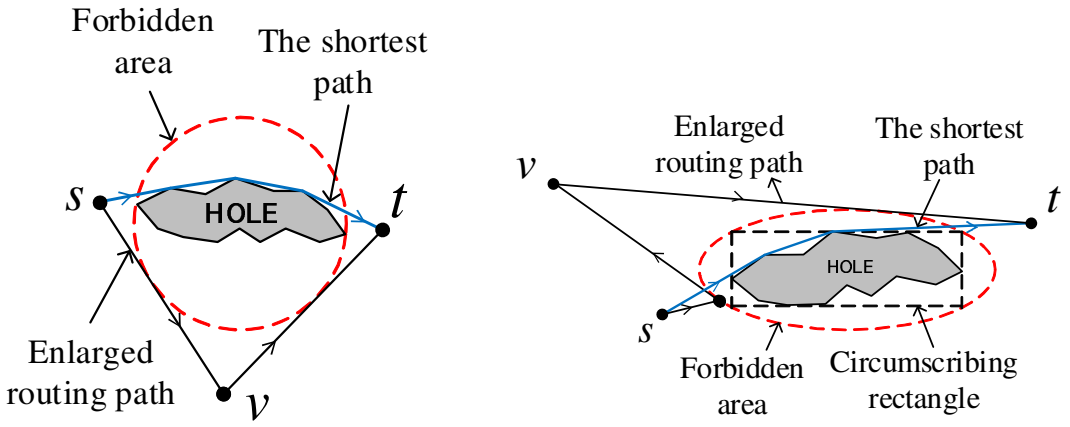
To bypass a hole, traditional schemes appropriately switch between greedy and perimeter forwarding modes, in the later of which the data packets are forwarded along the hole boundary. These proposals require a specific embedding of a planar graph (e.g., Gabriel Graph), a complicated procedure based on a restrictive assumption about the underlying graph (e.g., a unit-disk graph). Greedy Perimeter Stateless Routing (GPSR) [21] is such a typical routing protocol widely known in the networking community. Other perimeter routing approaches can be found at [166, 23, 22, 167]. Although these traditional approaches can alleviate the local minimum phenomenon, they face two critical problems. The first problem is the enlargement of routing paths and the second problem is the traffic concentration around the hole boundary. Subramanian et al. [168] showed that this perimeter routing could cause the throughput capacity of the whole network to significantly decrease due to the concentration of traffic on the face of the holes. On a planar network with no holes, Gupta and Kumar's seminal paper [169] shows that a uniform traffic demand of $\Theta(1/\sqrt{n\log n})$ is achievable. However, from [168] a large hole occupying a constant fraction of the network area can cause the throughput capacity to drop to just $O(1/n)$. Also in the paper, Subramanian et al. proposed a randomized routing scheme which can achieve a near-optimal throughput capacity, which however appears merely of theoretical interest (the overhead traffic can be extremely large in practice).

2.5.2 Forbidden area-based approach

To deal with these two problems, a new approach has been proposed which we call forbidden area approach. The main idea is to create a forbidden area around every hole from which all the packets are kept to stay away. Typically, forbidden area is a static region covering the hole and having simple shapes such as circle[24, 146], ellipse[26], hexagon[25], convex hull[150, 29] In [24], Fucai Yu *et al.,* proposed a routing scheme wherein the forbidden area is a circle covering the hole. First, the nodes on the boundary of the holes are identified using Boundhole algorithm [18] and a virtual circle which exactly covers the hole is formed. The information of the circle is disseminated to all nodes on the boundary of the hole. When a source node S wants to send a data

packet to a destination node *D*, it first sends the packet along the line *SD* by geographic routing. The node on the boundary of the hole receiving the data packet informs the source node the information of the virtual circle. Then, the source node calculates an anchor location which is the intersection of two tangent lines from *S* and *D* to the virtual circle. The source node forwards the data packet to the node which is closest to the anchor location by geographic routing and this node will forwards the data packet to the original destination. A routing scheme using virtual hexagon is proposed in [25]. Similar to [24], after identifying the boundary of a hole using the Boundhole algorithm, information of the center and the radius of the virtual circle which exactly covers the hole is calculated. This information is transferred to all nodes on the boundary of the hole. When a source node S has data, it initiates an observer packet and forwards it to the destination *SD* using greedy forwarding [21] and waits for a time delay *T*. If the observer packet arrives at a node on the boundary of the hole, this node forwards the information of the virtual circle back to the source node. The intermediate nodes on the way to the source node will save the information of the virtual circle and give it to a data packet on the same route. After time delay *T*, the source node forwards the data packet to the destination using greedy forwarding. When the data packet reaches the intermediate node which has virtual circle information, the intermediate node calculates a virtual hexagon which has the center and radius the same as the center and radius of the virtual circle and has one edge paralleling to *SD*. Finally, the intermediate node selects the detour points from the points of the virtual hexagon and forwards the packet to the original destination.

Instead of a virtual circle, in [26] a virtual ellipse covering the hole is calculated and the information of the ellipse is sent to all nodes inside the ellipse. Source node *S* initiates data packets and forwards them to the destination node *D* by geographic routing. When the data packet reaches a node on the boundary of the ellipse, the node on the boundary of ellipse calculates the location of a dynamical anchor point and forwards the packet to the anchor point. When the node closest to the anchor point receives the data packet, it forwards the data packet to the original destination. Other works using the forbidden area approach can be found in [132][146][150]. all of these approaches can reduce the data congestion on the hole boundary but they may create a new congestion area around the hole cover, instead. Moreover, routing paths may be enlarged due to the difference between the hole and the hole cover. Fig.2.6 illustrates the worst cases when the forbidden areas are in the shape of a circle or an ellipse.



(a) The forbidden area is a circle ([24]).

(b) The forbidden area is an ellipse ([26]).

Figure 2.6: Examples of routing path enlargement due to the difference between the hole and the forbidden area.

[153] is a rare protocol that uses dynamic forbidden area which is circle-shaped. Although this protocol can alleviate the traffic congestion around the forbidden area, it still suffer from routing path enlargement problem.

In [28], Won et al. proposed a protocol which can guarantee that the route stretch is upper bound by $\Theta(c)$, where c is the path length of the shortest path. In this protocol, the forbidden area is the convex hull of the hole. The packets then are routed along the shortest path from the source to the destination that goes through the vertices of the convex hull. This work is followed in [29] to address the problem of routing between nodes inside the concave regions of the holes. In [29], the authors proposed to describe the hole by a polygon whose vertices are all the nodes staying on the hole boundary. The routing path then is determined by the visibility graph whose vertices are vertices of the holes. The upper bound stretch was proved to $\Theta\left(\frac{D}{\gamma}\right)$, where D is the diameter of the network and γ is the communication range of sensor nodes). Although these two protocols are the rare ones that can provide stretch upper bound, they still suffer from the same problem, i.e., traffic concentration around the forbidden area, as the other protocols described above. Moreover, in these two protocols, especially [29], the information needed to represent the forbidden area depends on the holes and it may be significantly large when the holes become complicated. Consequently, they may cause extra overhead in disseminating and storing information of the forbidden areas.

Recently, Huang et al. tried to improve the load balance by exploiting energy information in making decision [134]. Specifically, before sending data packets, every source node sends two so-called burst packets towards the destination. The burst packets are hole-bypassing packets, one goes along the right-hand side and the other goes along the left-hand side of the holes. These packets collect information of two anchor lists along the routing path. Upon arriving at the destination, the burst packets, with the anchor lists embedded, are pushed back to the source. When a source node has a packet to send, it randomly chooses an anchor list and embeds the location of the anchors into the packet header. The packet then is forwarded towards the anchors gradually, where the next hop is chosen based on the location and the residual energy of the neighbors. Different from the other geographic routing protocols, EDGR requires nodes to periodically broadcast beacons to update the energy information.

2.5.3 Heuristic approach

There are several approaches that exploit heuristic methods to maintain the load balance between the nodes. Yu *et al.* [121] proposed a scheme to avoid the local minimum problem by identifying the nodes staying inside the concave areas and prevent them from participating in data delivery. Although these schemes can shorten the routing path, they continue to suffer from traffic congestion surrounding the hole. In [170], the next forwarder node is chosen based on a so-called forwarding factor. This factor is proportional with the residual energy and inversely proportional with the distance to the destination of the neighbors. Accordingly, neighbors with higher residual energy and shortest distance to the destination are more likely to be chosen. In [135], the next hop is chosen based on a self-election paradigm. Specifically, upon receiving a packet, all the neighbors of the sending node will start a waiting timer and wait for its timer to expires before broadcasting the packet. The waiting time is proportional to the destination. In order to alleviate the hole, all the nodes maintain a so-called eligible nodes table which consists of only 1-hop neighbor closer to the destination. The nodes can identify themselves as a node on the hole region if their eligible nodes table is empty. When a node which is further from the destination receives a packet from a hole node, it broadcast the packet to its neighbors, and this process is repeated until the packet arrives at a node closer to the base station. Another routing protocol addressing the local minimum problem was proposed by Petrioli et al. in [52]. In this protocol, each

node is assigned a color from a predefined color list. Upon receiving a packet, the nodes with even colors search for the next forwarder that has positive advancement, while the nodes with odd colors search for the next forwarder that has negative advancement to the destination. The loop-freedom of route determination of Rainbow has been proved theoretically.

Although all of these heuristic approaches can alleviate the data congestion on specific areas, they can not provide any guarantee on the route stretch.

We summary the pros and cons of some typical hole bypassing routing protocols in Tab.2.2.

Protocol	**Features**	**Stretch upper bound**	**Load balancing**	**Control overhead**
GPSR[21]	**Perimeter** routing	NO	Congestion around hole perimeter	Small
CIRCLE[146, 24]	Circle as the **forbidden area**	NO	Congestion around the forbidden areas' boundary	Small
HBF[153]	Dynamic circle as the **forbidden area**	NO	No specific congestion area	Small
ELLIPSE[26]	Ellipse as the **forbidden area**	NO	Congestion around the forbidden areas' boundary	Small
BHOP-GR[171]	Hexagon as the **forbidden area**	NO	Congestion around the forbidden areas' boundary	Small
RGP[150]	Convex hull as the **forbidden area**	NO	Congestion around the forbidden areas' boundary	Small
GOAL[28]	Convex hull as the **forbidden area**	$\Theta(c)$, c: the shortest path's length	Congestion around the forbidden areas' boundary	May be large, depending on the holes' complexity
LVGR[29]	Exact hole boundary as the **forbidden area**	$\Theta\left(\frac{D}{\gamma}\right)$, D: the diameter of the network, γ: the communication range of sensor nodes	Congestion around the forbidden areas' boundary	May be very large, depending on the holes' complexity
EDGR[134]	- Convex hull as the **forbidden area** - **Heuristic** based on energy and location information	NO	No specific congestion area	Very large due to periodically broadcasted beacons
HPS[121]	**Heuristic** based on location information	NO	Congestion around convex hulls' perimeter	Small
BECHA[170]	**Heuristic** based on location and energy information	NO	No specific congestion area	Very large due to periodically broadcasted beacons
REACT[135]	**Heuristic** based on the estimated distance to the destination	NO	No specific congestion area	Very small overhead but requires all destinations to have extra large communication ranges
ALBA-R[52]	**Heuristic** based on a coloring algorithm	NO	No specific congestion area	Small

Table 2.2: Pros and cons of existing hole bypassing routing protocols.

3

Load Balanced and Constant Stretch Hole Bypassing Protocol

In this chapter, we present our novel routing protocol for wireless networks where multiple holes may exists. Given an arbitrary positive number ε (which we call *stretch factor*), we aim at designing a hole bypassing protocol that can guarantee the stretch under $1+\varepsilon$, and that can balance traffic over the network. We first describe in Section 3.1 the network model, the assumptions and the notations used throughout this dissertation. Section 3.2 sketches the overview of our proposed protocol. Our algorithm for determining forbidden areas is presented in Section 3.3. The next three sections describe our strategies in determining routing path. Specifically, we start with the simplest network topology consisting of only one hole and all sources and destinations stay out side of the hole's forbidden area in Section 3.4. The routing strategy described in Section 3.4 can be seen as the basic idea of our routing approach. Then, we describe the challenges and our strategies to deal with the case when the sources and the destinations stay inside the forbidden area of a hole in Section 3.5. Section 3.6 shows our approach in dealing with multiple holes. Finally, based on the result obtained from Section 3.3, 3.4, 3.6 and 3.5,

we propose our routing protocol for the most general network topology where multiple holes may exist and the sources and the destination may stay inside the forbidden areas.

3.1 Preliminaries

3.1.1 Network model

We assume that each node knows its position (using GPS or other positioning services [172]) and its 1-hop neighbors (through the neighbor notification packets); in addition, the source node knows the position of the destination node. For theoretical analysis, we make a reasonable assumption that the considered network is sufficiently dense such that there are sensors everywhere apart from the considered hole. Given such an ideal situation, we can model the *geographical greedy routing path* between two given nodes s and t (in the dense area) as *the Euclidean line connecting s and t*. Figure 3.1 illustrates such an example. In this figure, s_1t_1 does not intersect the hole; the packets from s_1 can be greedily directed straightforward toward t_1, and thus, this greedy routing path can be modeled by the segment s_1t_1. In contrast, s_2t_2 intersects the hole, and thus, the packets from s_2 are greedily forwarded to v before arriving at t_2. Therefore, this routing path from s_2 to t_2 can be modeled by the broken line s_2vt_2.

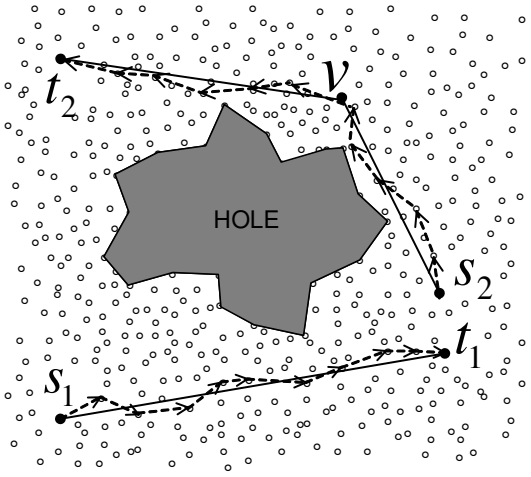


Figure 3.1: Illustration of the theoretical model.

3.1.2 Notations and definitions

A *routing hole* is defined as a non-self-intersecting polygon that has all vertices as sensor nodes $S_1, \ldots, S_n$ and satisfies the following conditions:

- Its interior does not contain any sensor nodes.
- The Euclidean distance between S_i and S_{i+1} is within the transmission range ($\forall i = \overline{1,n}; S_{n+1} \equiv S_1$).

Let $\mathscr{Q}$ be a polygon with the vertices of $Q_1, Q_2, ..., Q_n$ (sorted in the counterclockwise direction). We have the following definitions and notations. Notation $|.|$ denotes the Euclidean length, e.g., $|AB|$ is the Euclidean length between two points A and B; $|l|$ is the Euclidean length of the line l. $\widehat{Q}_i$ denotes the interior angle of $\mathscr{Q}$ at Q_i, i.e., $\widehat{Q}_i = \angle Q_{i-1}Q_iQ_{i+1}$ ($\forall i = 1, 2, ..., n$, $Q_0 \equiv Q_n, Q_{n+1} \equiv Q_1$), and $p_{\mathscr{Q}}$ denotes the perimeter of $\mathscr{Q}$. Let A, B be two points staying on the boundary of $\mathscr{Q}$; then, $\{A \sim B\}_{\mathscr{Q}}$ and $\{A \sim B\}^{-}_{\mathscr{Q}}$ denote the boundary segment of $\mathscr{Q}$ from A to B in the counterclockwise and clockwise directions, respectively.

Definition (Circumscribing polygon)
A circumscribing polygon of $\mathscr{Q}$ is defined as a convex polygon $\mathscr{P}$ such that $\mathscr{P}$ entirely covers $\mathscr{Q}$, and each edge of $\mathscr{P}$ contains at least one vertex of $\mathscr{Q}$.

Definition (Convex hull)
The convex hull of $\mathscr{Q}$ is defined as a circumscribing polygon of $\mathscr{Q}$ whose vertices are vertices of $\mathscr{Q}$. Note that a polygon may have many circumscribing polygons but only one convex hull. Moreover, the convex hull can be seen as the smallest circumscribing polygon.

Definition (Angle of a polygon to the x-axis)
The angle from the x-axis of $\mathscr{Q}$ is defined as the smallest angle of $\overrightarrow{Q_iQ_{i+1}}$ from the x-axis ($\forall i = 1, ..., n; Q_{n+1} \equiv Q_1$).

Definition (View-limit vertex)
Let N be an arbitrary point staying outside of $\mathscr{Q}$; then, the view-limit vertex from N to $\mathscr{Q}$ is defined as a vertex Q_i of $\mathscr{Q}$ such that the line passing through N and Q_i does not intersect $\mathscr{Q}$.

Clearly, for each node N staying outside of a polygon $\mathscr{Q}$, there are two such view-limit vertices from N to $\mathscr{Q}$.

Definition (View-limit angle)
Let Q_i, Q_j be the two view-limit vertices from a node N to a polygon $\mathcal{Q}$; then, the view-limit angle from N to $\mathcal{Q}$ (denoted as $v_{\mathcal{Q}}(N)$) is defined as the angle made between two arrays $\overrightarrow{NQ_i}, \overrightarrow{NQ_j}$, which contains $\mathcal{Q}$.

Definition (Distance to a polygon)
The distance from N to $\mathcal{Q}$ (denoted as $d_{\mathcal{Q}}(N)$) is defined as the smallest distance from N to an edge of $\mathcal{Q}$.

Let $\mathfrak{R}$ be a routing protocol, and let (s,t) be an arbitrary source-destination pair. Let *r(s,t)* be a routing path defined by $\mathfrak{R}$, and let *o(s,t)* be the theoretical shortest routing path from *s* to *t*. We have the following definitions regarding term *stretch*.

Definition (Hop count stretch)
The hop count stretch (or stretch, for short) of the route $r(s,t)$ is the ratio of the number of hop count of $r(s,t)$ to that of $o(s,t)$.
The stretch of the routing protocol $\mathfrak{R}$ is the maximum stretch of all routing path determined by $\mathfrak{R}$.

Definition (Euclidean stretch)
The Euclidean stretch (or E-stretch, for short) of the route $r(s,t)$ is the ratio of the Euclidean length of $r(s,t)$ to that of $o(s,t)$.
The Euclidean stretch of the routing protocol $\mathfrak{R}$ is the maximum Euclidean stretch of all routing path determined by $\mathfrak{R}$.

Table 3.1 summaries the notations and Fig.3.2 illustrates the definitions.

Table 3.1: List of notations introduced in this thesis

Notation	*Description*
$\|.\|$	Euclidean length
$\{A \sim B\}_{\mathcal{Q}}$	Boundary segment of $\mathcal{Q}$ from A to B in the counterclockwise direction.
$\{A \sim B\}_{\mathcal{Q}}^{-}$	Boundary segment of $\mathcal{Q}$ from A to B in the clockwise direction.
$\widehat{Q_i}$	Interior angle of $\mathcal{Q}$ at its vertex, Q_i
$p_{\mathcal{Q}}$	Perimeter of $\mathcal{Q}$
$v_{\mathcal{Q}}(N)$	A view-limit angle from N to $\mathcal{Q}$
$d_{\mathcal{Q}}(N)$	Distance from N to $\mathcal{Q}$
$l_{\mathcal{Q}}(s,t)$	The shortest Euclidean path from s to t staying outside of $\mathcal{Q}$

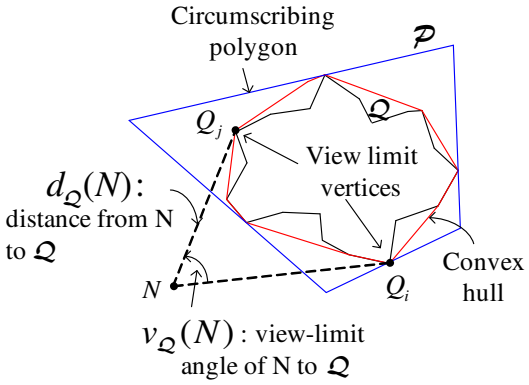


Figure 3.2: Illustration of definitions.

3.2 Protocol overview

We assume that the nodes do not know the hole location in advance. Therefore, before forwarding data, we need some setup phases that help to determine the hole as well as the forbidden areas and broadcast their information to the surrounding nodes. Our protocol consists of three phases as follows.

- The first phase is to detect the hole boundaries and construct the forbidden area for each hole.
- In the second phase, the information of the holes and the forbidden areas is disseminated to the nodes in the network. To reduce the overhead, we do not broadcast full information of all the holes to all nodes in the network. Instead, the broadcast information received by a node depends on the node's location. Intuitively, a node closer to a hole will receive more detail information about the hole.
- Once the first two phases finish, the nodes utilize hole information to make the routing decision. Specifically, for every packet, the source node first determines a base path to the destination. This base path is an Euclidean path that bypasses all the forbidden areas whose information stored in the local memory of the source node. The base path then is magnified using homothetic transformations to obtain the Euclidean routing path which will act as the guide line for the packet. The homothetic centers are chosen randomly to conserve the diversity of the routing paths, while the scale factors are controlled to guarantee the stretch upper bound.

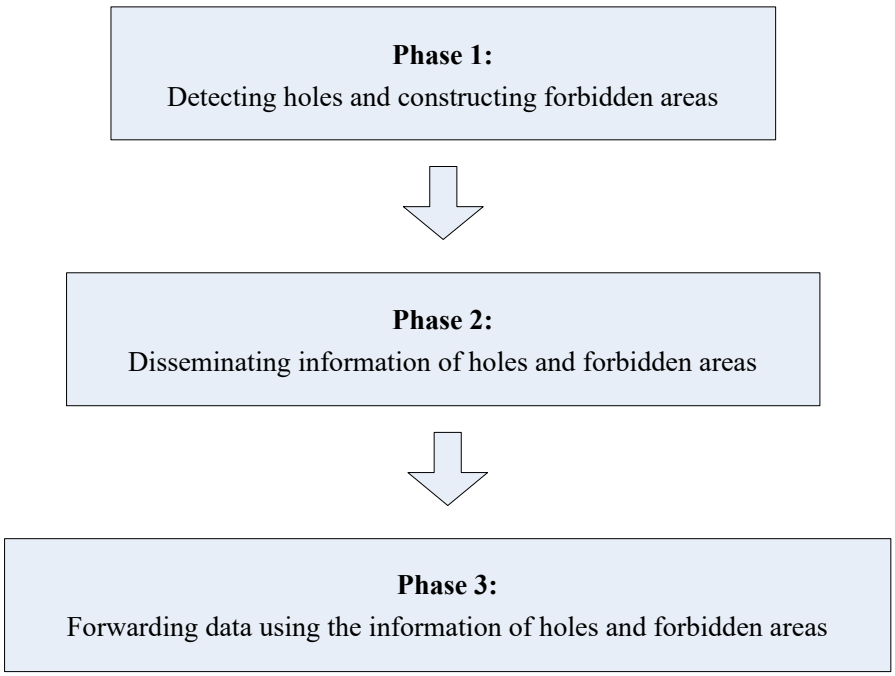


Figure 3.3: Flow of our proposed protocol.

If the source node has full information about all the holes, then the packet is forwarded along the Euclidean routing path until reaching the destination. Otherwise, the packet is forwarded along the Euclidean routing path until arriving at an intermediate node which has more detail information about the holes. The intermediate node then redetermines the routing path to the destination and forwards the packet along the new routing path until reaching the destination.

Figure 3.3 illustrates the flow of our proposed protocol.

3.3 Hole and forbidden area determination

3.3.1 Theoretical basis

Before performing theoretical analysis on the geometrical characteristic of forbidden area we present some definitions. Let $\mathcal{Q}$ be a polygon and s, t be two points on the plane.

Definition (Shortest Euclidean path)
The shortest $\mathcal{Q}$-bypassing Euclidean path of (s,t) (or $l_{\mathcal{Q}}(s,t)$ for short) is defined as the shortest broken line from s to t that does not intersect the interior of $\mathcal{Q}$.
Definition (Polygon stretch)
Let $\mathcal{P}$ be a polygon covering $\mathcal{Q}$, then the polygon stretch of $\mathcal{P}$ to $\mathcal{Q}$ for (s,t) is defined as the ratio of $|l_{\mathcal{P}}(s,t)|$ and $|l_{\mathcal{Q}}(s,t)|$ (i.e., $\frac{|l_{\mathcal{P}}(s,t)|}{|l_{\mathcal{Q}}(s,t)|}$).
The polygon stretch of $\mathcal{P}$ to $\mathcal{Q}$ is defined as the maximum polygon stretch of $\mathcal{P}$ to $\mathcal{Q}$ for all (s,t) (i.e., $\max_{\forall(s,t)} \frac{|l_{\mathcal{P}}(s,t)|}{|l_{\mathcal{Q}}(s,t)|}$).

Forbidden areas play an important role in the routing scheme. A simple forbidden area (e.g. circle, ellipse) can reduce the overhead in disseminating and storing its information but increases the hole bypassing route length due to the large difference between the hole and the forbidden area. In contrast, a tight forbidden area can reduce the length of the routing path but may require more information to describe and thus causes a large overhead. An ideal forbidden area should satisfy the following three requirements:

i) **Minimum polygon stretch:** Note that in our protocol, packets are kept to stay away from all the forbidden areas. Therefore, the E-stretch of the routing protocol is lower bounded by the polygon stretch of the forbidden area to the hole. Consequently, to guarantee the upper bound of the route stretch, the optimal forbidden area should have the minimal polygon stretch to the hole.

ii) **Minimum overhead:** It is obvious that the overhead caused by disseminating and storing the information of the forbidden area is proportional to its description information. Therefore, to reduce the overhead, the description information of the forbidden area should be minimized.

Unfortunately, there always exists a trade-off among these three requirements; thus, finding such an ideal forbidden area is impossible. For example, to ensure the first requirement, packets should be routed along the shortest paths and thus the nodes on these shortest paths will be imposed a heavier traffic than the others. This is obviously contrary to the last requirement. To satisfy the second requirement, the forbidden area should be as simple as possible. However, a simple forbidden area may cause a large difference with the original hole, thus results in a long routing path and can not satisfy the first requirement. Due to this trade-off, finding such an ideal forbidden area is impossible. In the following, we propose a strategy for constructing a forbidden area that can guarantee the following properties:

i*) **Constant polygon stretch:** Its polygon stretch to the hole is upper bounded by a constant, and this constant can be controlled to be as small as desired.

ii*) **Constant overhead:** Its description information is upper bounded by a constant, which does not depend on the hole size.

Forbidden areas that have fixed shapes, such as circle or ellipses, may have very little description information, but they cannot reflect the complexity of the hole and thus may enlarge the routing path. Therefore, we attempt to construct a forbidden area as a polygon with variable size. Clearly, the description information of the polygon is directly proportional to the number of vertices, and the difference between the polygon and the hole can be made as small as possible by using a sufficiently large number of vertices.

First, we consider property i* of our forbidden area; then, a natural question is that, for a given upper bound of the polygon stretch to the hole, which polygon covering the hole has the smallest number of vertices?

Lemma 1

Let $\mathscr{G}$ be the convex hull and $\mathscr{P}$ be a circumscribing polygon of a polygon $\mathscr{Q}$. Then the following statements are true:

1. *The view limit vertices of an arbitrary point N (which stays outside of $\mathscr{G}$) to $\mathscr{Q}$ and $\mathscr{G}$ are identical.*
2. *Let $Q_{i_1}, Q_{i_2}, ..., Q_{i_k}$ be the vertices of $\mathscr{Q}$ staying on the edges of $\mathscr{P}$ then Q_{i_j} must be vertices of $\mathscr{G}$ ($\forall j = 1, ..., k$).*
3. *$\mathscr{P}$ is a circumscribing polygon of $\mathscr{G}$.*

Lemma 2

Let $\mathscr{Q}$ be a polygon and $\mathscr{G}$ be the convex hull of $\mathscr{Q}$. Let (s,t) be two arbitrary points staying outside of $\mathscr{G}$ such that st intersects $\mathscr{Q}$. Denote G_{s_1}, G_{s_2} and G_{t_1}, G_{t_2} as the view-limit vertices of s and t to $\mathscr{G}$ such that G_{s_1}, G_{t_1} stay on the right side and G_{t_2}, G_{s_2} stay on the left side of $\overrightarrow{st}$, respectively. Then, $l_{\mathscr{Q}}(s,t) \equiv l_{\mathscr{G}}(s,t)$, moreover, they are the shorter one of $s\{G_{s_1} \sim G_{t_1}\}_{\mathscr{G}}t$ and $s\{G_{s_2} \sim G_{t_2}\}_{\mathscr{G}}^{-}t$ (Figure 3.4).

Lemma 2 implies that, to minimize the description information, the forbidden area should be a convex polygon because, for any concave polygon, its convex hull has the same polygon stretch to the hole but a smaller number of vertices.

Now, regarding property ii*, given a number of vertices, we will investigate which convex polygon covers the hole and has the smallest polygon stretch to the hole.

Lemma 3

Let $\mathscr{Q}$ be a polygon. Assume that $\mathscr{P}$ is the polygon whose polygon stretch to $\mathscr{Q}$ is the smallest among all convex n-gons (i.e., polygons whose number of vertices is n) covering $\mathscr{Q}$. Then, $\mathscr{P}$ must be a circumscribing polygon of $\mathscr{Q}$.

Proof (of Lemma 3)

Figure 3.5 illustrates our proof. We prove by contradiction. Let $P_1, P_2, ..., P_n$ be the vertices of $\mathscr{P}$. Suppose that $\mathscr{P}$ is not a circumscribing polygon of $\mathscr{Q}$, then there exist an edge of $\mathscr{P}$, say P_iP_{i+1}, that does not intersect $\mathscr{Q}$. Suppose Q_j is the one whose distance to P_iP_{i+1} is the smallest among all vertices of $\mathscr{Q}$. Draw a line going through Q_j and paralleling with P_iP_{i+1}, then this line does not intersect the interior of $\mathscr{Q}$. Denote the intersections of this line with $P_{i-1}P_i$ and $P_{i+1}P_{i+2}$ as $P_i^{'}$ and $P_{i+1}^{'}$, respectively. Denote by $\mathscr{P}^{'}$ the polygon whose vertices are $P_1, P_2, ...P_{i-1}, P_i^{'}, P_{i+1}^{'}, P_{i+2}, ..., P_n$. Then, $\mathscr{P}^{'}$ stays inside of $\mathscr{P}$ and covers $\mathscr{Q}$. Accordingly, $\mathscr{P}^{'}$ has a smaller polygon stretch to $\mathscr{Q}$ but the same number of the vertices as $\mathscr{P}$.

From the above analysis, we know that, to minimize the routing path stretch and the overhead, the forbidden area should be a circumscribing polygon of the hole. In the following, we present more details on circumscribing polygons to determine what type of circumscribing polygon can guarantee a constant polygon stretch to the hole.

Lemma 4

Let $P_1P_2P_3$ be a triangle; then, $|P_2P_3| \geq (|P_1P_2| + |P_3P_1|) \sin \frac{\widehat{P_1}}{2}$.

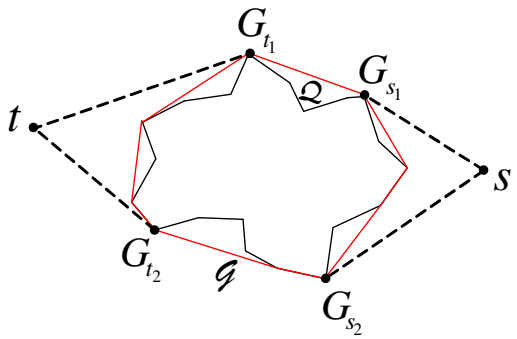


Figure 3.4: Illustration of Lemma 2.

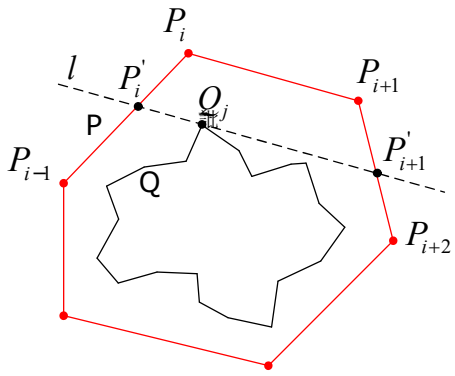


Figure 3.5: Illustration of the proof of Lemma 3.

Proof (of Lemma 4)
Using triangular cosine formula and Cauchy equality, we have:

$$\begin{aligned}
|P_2P_3|^2 &= |P_1P_2|^2 + |P_3P_1|^2 - 2|P_1P_2||P_3P_1|\cos\widehat{P_1} \\
\Rightarrow |P_2P_3|^2 &\geq (|P_1P_2| + |P_3P_1|)^2 \left(\frac{1-\cos\widehat{P_1}}{2}\right) \\
\Rightarrow |P_2P_3|^2 &\geq (|P_1P_2| + |P_3P_1|)^2 \sin^2\frac{\widehat{P_1}}{2} \\
\Rightarrow |P_2P_3| &\geq (|P_1P_2| + |P_3P_1|)\sin\frac{\widehat{P_1}}{2}
\end{aligned}$$

The proof is done.

Theorem 1
Let $\mathscr{Q}$ be a polygon, and let $\mathscr{P}$ be its circumscribing polygon. Denote by $P_1, P_2, ..., P_n$ the vertices of $\mathscr{P}$ (ordered in the counterclockwise direction). Assume that $P_{r_1}, ..., P_{r_m}$ are the vertices of $\mathscr{P}$ that are not vertices of $\mathscr{Q}$. Then, the polygon stretch of $\mathscr{P}$ to $\mathscr{Q}$ does not exceed $\frac{1}{sin(\rho/2)}$, where $\rho = \min_{i=\overline{1,m}} \widehat{P_{r_i}}$.

Proof (of Theorem 1)
Figure 3.6 illustrates our proof. Let s, t be two arbitrary points staying outside of $\mathscr{P}$; in the following, we will prove that $\frac{|l_{\mathscr{P}}(s,t)|}{|l_{\mathscr{Q}}(s,t)|} \leq \frac{1}{sin(\rho/2)}$. Denote $\mathscr{G}$ as the convex hull of $\mathscr{Q}$ and assume that $G_1, ..., G_m$ are the vertices of $\mathscr{G}$. According to Lemma 1 and Lemma 2, $l_{\mathscr{Q}}(s,t) \equiv l_{\mathscr{G}}(s,t)$, and they can be expressed by $s\{G_u \sim G_{u+k}\}_{\mathscr{G}}t$, where G_u and G_{u+k} are the view-limit vertices of s and t to $\mathscr{G}$, respectively. Without loss of generality, we

assume that G_u, G_{u+k} stay on the right side of $\overrightarrow{st}$. Assume that $P_v, ..., P_{v+h}$ are all the vertices of $\mathscr{P}$ that stay on the right side of $\overrightarrow{st}$. Then, the broken line $s\{P_v \sim P_{v+h}\}_{\mathscr{P}}t$ does not intersect the hole, thus

$$|l_{\mathscr{P}}(s,t)| \leq |s\{P_v \sim P_{v+h}\}_{\mathscr{P}}t| \tag{3.1}$$

$h \leq 1$ is a special case and can be proven easily. Therefore, in the following, we only consider the cases when $h \geq 2$. According to Lemma 1, $\mathscr{P}$ is also a circumscribing polygon of $\mathscr{G}$; therefore, each edge of $\mathscr{P}$ must contain at least one vertex of $\mathscr{G}$. Assume that $G_{u+w_j}, ..., G_{u+w_j+\rho_j}$ ($\rho_j \in \{0,1\}, w_j \in \mathbb{N}$) are the vertices of $\mathscr{G}$ staying on $P_{v+j}P_{v+j+1}$ ($\forall j = 0, .., h-1$). Then,

$$|s\{G_u ... G_{u+k}\}_{\mathscr{G}}t| \geq |sG_{u+w_0}| + \sum_{j=0}^{h-2} \left|\{G_{u+w_j} \sim G_{u+w_{j+1}}\}_{\mathscr{G}}\right| + \left|G_{u+w_{h-1}}t\right| \tag{3.2}$$

According to Lemma 4, we have

$$\begin{aligned} |sG_{u+w_0}| &\geq \sin\frac{\angle sP_vG_{u+w_0}}{2}\left(|sP_v| + |P_vG_{u+w_0}|\right) \\ &\geq \sin\frac{\rho}{2}\left(|sP_v| + |P_vG_{u+w_0}|\right) \qquad (3.3) \\ \left|G_{u+w_{h-1}}t\right| &\geq \sin\frac{\angle G_{u+w_{h-1}}P_{v+h}t}{2}\left(\left|G_{u+w_{h-1}}P_{v+h}\right| + |P_{v+h}t|\right) \\ &\geq \sin\frac{\rho}{2}\left(\left|G_{u+w_{h-1}}P_{v+h}\right| + |P_{v+h}t|\right) \qquad (3.4) \end{aligned}$$

Similarly, for all $0 \leq j \leq h-2$ such that P_{v+j} is not a vertex of $\mathscr{G}$, we have

$$\left|\{G_{u+w_j} \sim G_{u+w_{j+1}}\}_{\mathscr{G}}\right| \geq \sin\angle G_{u+w_j}P_{v+j+1}G_{u+w_{j+1}} \times \left(\left|G_{u+w_j}P_{v+j+1}\right| + \left|P_{v+j+1}G_{u+w_{j+1}}\right|\right)$$

$$\Rightarrow \left|\{G_{u+w_j} \sim G_{u+w_{j+1}}\}_{\mathscr{G}}\right| \geq \sin\frac{\rho}{2} \times \left(\left|G_{u+w_j}P_{v+j+1}\right| + \left|P_{v+j+1}G_{u+w_{j+1}}\right|\right) \tag{3.5}$$

For any $0 \leq j \leq h-2$ such that P_{v+j+1} is a vertex of $\mathscr{G}$, P_{v+j+1} coincides with G_{u+w_j} and $G_{u+w_{j+1}}$, and thus, (3.5) becomes trivial.

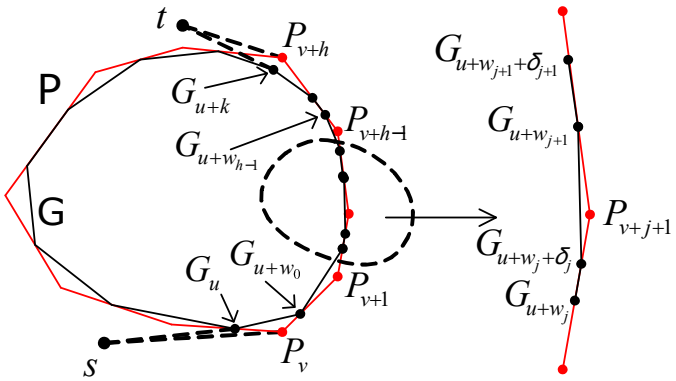


Figure 3.6: Illustration of the proof of Theorem 1.

By substituting (3.3), (3.4) and (3.5) into (3.2), it can be deduced that

$$|s\{G_u \sim G_{u+k}\}_{\mathscr{G}} t| \geq \sin\frac{\rho}{2}\{(|sP_u| + |P_v G_{u+w_0}|)$$
$$+\sum_{j=0}^{h-2}\left(\left|G_{u+w_j}P_{v+j+1}\right| + \left|P_{v+j+1}G_{u+w_{j+1}}\right|\right) + \left(\left|G_{u+w_{h-1}}P_{v+h}\right| + |P_{v+h}t|\right)\}$$

$$\Rightarrow |s\{G_u \sim G_{u+k}\}_{\mathscr{G}} t| \geq \sin\frac{\rho}{2}|s\{P_v \sim P_{v+h}\}_{\mathscr{P}} t| \tag{3.6}$$

From (3.1) and (3.6), finally, we have

$$|l_{\mathscr{Q}}(s,t)| = |s\{G_u \sim G_{u+k}\}_{\mathscr{G}} t|$$
$$\geq \sin\frac{\rho}{2}|s\{P_v \sim P_{v+h}\}_{\mathscr{P}} t| \geq \sin\frac{\rho}{2}|l_{\mathscr{P}}(s,t)|$$

In other words, $\frac{|l_{\mathscr{P}}(s,t)|}{|l_{\mathscr{Q}}(s,t)|} \leq \frac{1}{sin(\rho/2)}$.

Accordingly, if the forbidden area is an equiangular polygon, then its polygon stretch to the hole is upper bounded by a constant, $\frac{1}{\sin\frac{(n-2)\pi}{2}}$, where n is the number of vertices ($n \geq 3$). Consequently, it can be seen that a circumscribing polygon of the hole with equal angles can guarantee both property i* (the polygon stretch is upper bounded by $\frac{1}{\sin\frac{(n-2)\pi}{2}}$) and property ii* (its description information is $O(n)$, which does not depend on the hole size) of our forbidden area. Hereafter, we call the forbidden areas constructed by our algorithm the core polygons of the holes.

3.3.2 Distributed core polygon construction

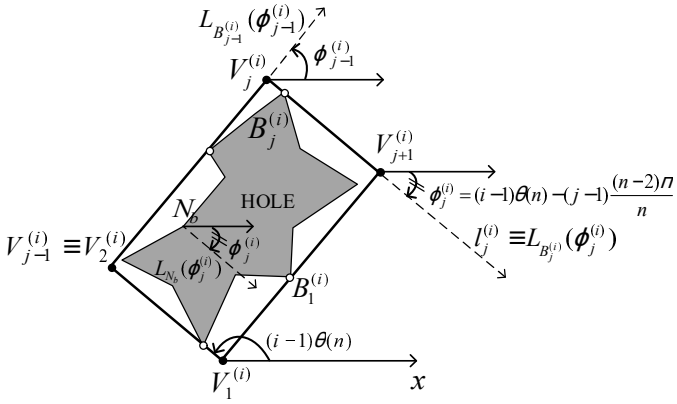
In the following, we describe our protocol to locate a hole and construct the core polygon (i.e., the forbidden area) whose polygon stretch to the hole are upper bound by $1+\varepsilon$. Let n be an integer such that $\frac{1}{\sin\frac{(n-2)\pi}{2n}} < 1+\varepsilon$ ($n \geq 3$), then the core polygon of a hole is a polygon satisfying that:

- It is a circumscribing polygon of the hole, and its number of vertices does not exceed n.
- All of its angles (that are not angles of the hole's convex hull) equal to $\frac{(n-2)\pi}{n}$.

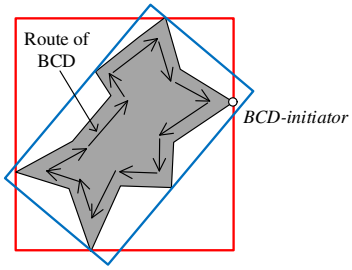
Obviously, for each hole $\mathscr{H}$ there are many polygons satisfying the above conditions. We call a polygon satisfying the above three conditions a *core polygon candidate of* $\mathscr{H}$. Intuitively, the core polygon should be the one closest to $\mathscr{H}$ (i.e., the difference between them is the smallest). Therefore, our algorithm will create a set of core polygon candidates and then selects the one with the smallest area as the core polygon. Specifically, the core polygon candidates will be the ones whose angles to the x-axis form an arithmetic sequence. Denote by $\theta(n)$ the common difference and by k_n the size of this arithmetic sequence ($\theta(n) > 0; k_n \in \mathbb{Z}$, $\theta(n)$ and k_n are fixed for all holes); then, the angle of the i^{th} core polygon from the x-axis will be $(i-1)\theta(n)$ ($i = 1,...,k_n$). Note that the angle from the x-axis of the core polygon candidates is always less than or equal to $\frac{2\pi}{n}$; therefore, $\theta(n)$ and k_n should be chosen such that $k_n\theta(n) < \frac{2\pi}{n}$. In our protocol, k_n is a predefined parameter, and $\theta(n)$ is calculated from k_n as $\frac{2\pi}{n(k_n+\zeta)}$, where ζ is a small positive number.

Figure 3.7(a) illustrates the i^{th} core polygon whose vertices are $V_1^{(i)}, V_2^{(i)}, ..., V_n^{(i)}$ (sorted in the counterclockwise order). We call a hole boundary node staying on the boundary of a core polygon a *core node* of that core polygon. We denote by $\varphi_j^{(i)}$ the angle between array $\overrightarrow{V_j^{(i)}V_{j+1}^{(i)}}$ and the x-axis, by $B_j^{(i)}$ a core node remaining on $V_j^{(i)}V_{j+1}^{(i)}$ ($\forall i = 1,...,k_n; j = 1,...,n$). For an arbitrary node N and an angle γ, we denote $L_N(\gamma)$ as the line crossing N such that its angle to the x-axis equals to γ. The following statements are true:

1. $V_j^{(i)}$ is the intersection of $L_{B_{j-1}^{(i)}}(\varphi_{j-1}^{(i)})$ and $L_{B_j^{(i)}}(\varphi_j^{(i)})$.



(a) Determining the i^{th} core polygon candidate. The white dots are the core nodes, the black dots are the vertices of the core polygon candidate.



(b) There are two core polygon candidates. The blue one is selected as the core polygon since its area is the smallest.

Figure 3.7: Illustration of core polygon determination protocol.

2. For $\forall j \in \{1,...,n\}$, if we draw a line $l_j^{(i)}$ going through $V_j^{(i)}$ and $V_{j+1}^{(i)}$, then the hole must stay on the left side of $l_j^{(i)}$ (regarding the direction of $\overrightarrow{V_j^{(i)}V_{j+1}^{(i)}}$).

3. A hole boundary node N_b stays on $V_j^{(i)}V_{j+1}^{(i)}$ if and only if $L_{N_b}(\varphi_j^{(i)})$ (i.e. the line going through N_b and being parallel with $V_j^{(i)}V_{j+1}^{(i)}$) coincides with $l_j^{(i)}$.

Statement 1 indicates that the core polygons can be determined by identifying the core nodes. Statements 2 and 3 show that the necessary and sufficient condition for a boundary node N_b being a core node of the i^{th} core polygon is, there exists $1 \leq j \leq n$ such that the hole stays on the left side of $L_{N_b}(\varphi_j^{(i)})$. Note that $\varphi_j^{(i)} = (i-1)\theta(n) + (j-1)\frac{2\pi}{n} (\forall i = 1,...,k_n; j = 1,...,n)$ because all the interior angles (at the vertices) of the core polygons equal to $\frac{(n-2)\pi}{n}$. Based on the above observations, we propose a distributed core polygon construction algorithm as follows.

All nodes determine whether they are on the boundary of the hole using the TENT rule described in [18][1]. Each hole boundary node then creates a Hole Boundary Approximation (HBA) packet and sends to its left neighboring node. These HBA packets are

[1]It was proved that a sensor node p is on the hole boundary if there exit two adjacent neighbors u, v of p such that the center of the circumcircle of triangle puv is out of transmission range of p. The TENT rule checks all adjacent neighbor pairs of a sensor node p to identify whether there exits any neighbor pair that satisties the above condition. For more details, see [18].

then forwarded by the right-hand rule[2] described in [18] to determine the hole boundary as well as the core polygons. Note that multiple HBA packets can be created, thus in order to avoid overhead, we will eliminate the late-coming, redundant packets. Consequently, there is only one HBA packet (whose creator discovers the hole first) that can go around the hole without being dropped by intermediate nodes. Denote the creator of this HBA packet as H_0. In the following, we consider only the HBA packet created by H_0. It is worth noting that the original right-hand rule can discover only the hole boundary. Therefore, to determine the core polygons, we customize the rule as follows. The HBA packet contains information of the core nodes (i.e., $B_j^{(i)}$ $(\forall i = 1,...,k_n; j = 1,...,n)$). The creator of the HBA packet (i.e., H_0) initializes $B_j^{(i)}$ $(\forall i = \overline{1,k_n}; \forall j = \overline{1,n})$ as its own coordinates. When a node N_b on the hole boundary receives the HBA packet, it determines if it is a core node of the i^{th} core polygon $(\forall i = 1,...,k_n)$ by checking whether there exists any $1 \leq j \leq n$ such that all the core nodes found so far stay on the left side of $L_{N_b}(\varphi_j^{(i)})$. If it is, then it updates its coordinates to $B_j^{(i)}$[3]. After all $B_j^{(i)}$ $(i = 1,...,k_n; j = \overline{1,n})$ have been updated, N_b forwards HBA packets to the next hole boundary node using the right-hand rule. After $\mathscr{H}$'s *BCD-initiator* receives the HBA packet back, it obtains a list of $B_j^{(i)}$ $(\forall i = \overline{1,k_n}; \forall j = \overline{1,n})$. *BCD-initiator* then determines the vertices of $\mathscr{H}$'s core polygon candidates based on the above-mentioned Statement 1. After that, *BCD-initiator* selects the core polygon candidate whose area is the smallest as $\mathscr{H}$'s core polygon. Figure 3.7(b) shows an example when $n = 4$ and $k_n = 2$. In this figure, the red polygon is the first core polygon candidate, whose angle from the x-axis is 0, and the blue polygon is the second core polygon candidate, whose angle from the x-axis is approximately equal to $\frac{\pi}{4}$. The blue one is chosen as the core polygon.

3.3.3 Section summary

In this section, we studied the relationship between the forbidden area and the stretch upper bound. Based on the analysis results, we proposed a strategy to construct the forbidden area. The forbidden area of a hole, called the core polygon, is a convex polygon satisfying the following conditions:

[2]Suppose the current hole boundary node is t_i and its previous hole boundary node is t_{i-1}, then the right hand rule determines the next hole boundary node t_{i+1} as follows. Draw a ray l with direction $t_i t_{i-1}$ and sweep it around t_i counterclockwise, then t_{i+1} is the first 1-hop neighbor of t_i hit by l.

[3]The pseudo code is presented by Algorithm 1 in the Appendix.

- It circumscribes the hole.
- Its all angles equal $\frac{(n-2)\pi}{n}$, where n is such an integer that $\frac{1}{\sin\frac{(n-2)\pi}{2n}} < 1+\varepsilon$.

We also proposed a distributed protocol for locating the hole boundaries and constructing the core polygons.

Given the holes and the core polygons determined, in the next sections, we present our strategies for utilizing hole and core polygon information to forward data packets.

3.4 Strategy for bypassing one hole

In this section, we start with the simplest network topology, where there is only one hole, and the sources and the destinations of all the packets stay outside of the hole's core polygon. By using the protocol proposed in Section 3.3, the hole and the core polygon is determined. Now, we will describe how the information of the core polygon can be used to determine the routing path and forward data packets.

3.4.1 Core polygon information dissemination

After the first phase, the node on the hole boundary (hereafter, termed as hole boundary nodes) determined the core polygon. Let $\mathscr{C}$ denote the determined core polygon. At the second phase, the information of $\mathscr{C}$ (i.e., coordinates of all the vertices) will be disseminated. To reduce the dissemination overhead, the dissemination region is limited to the nodes near the hole. Specifically, the dissemination process is stopped at node N satisfying the following condition:

$$\cos\frac{v_{\mathscr{C}}(N)}{2} > \frac{1}{1+\varepsilon} + \frac{p_{\mathscr{C}}\left(1-\sin\frac{(n-2)\pi}{2n}\right)}{d_{\mathscr{C}}(N)} \tag{3.7}$$

Where $v_{\mathscr{C}}(N)$ and $d_{\mathscr{C}}(N)$ are the view-limit angle and the distance from N to $\mathscr{C}$ (see Section 3.1 for the definitions), respectively. Note that when node N is further from the hole, the left side of (3.7) tends to increase, whereas the right side tends to decrease. Therefore, (3.7) will hold when N is sufficiently far from the core polygon. Condition (3.7) is designed so that the dissemination region is sufficiently large to ensure the constant-stretch of routing paths (as will be proved in Section 3.4.3).

The details of the protocol are given below[4]. Node H_0 starts by creating a *hole core information (HCI)* packet that conveys the coordinates of the vertices of all k_n core polygons. H_0 then broadcasts this HCI packet to its 1-hop neighbors. Any node receiving an HCI packet will perform the following process:

- **Step 1 - Add the information of the core polygon to the local memory:** The node checks if the information of the core polygon in the HCI packet thas been stored in its local memory. If "yes", then it must already received a HCI packet before, thus it simply drops the current HCI packet. Otherwise, it adds information of the core polygons to the local memory and proceeds to the next step.
- **Step 2 - Check for termination:** The node checks whether condition (3.7) is satisfied. If "yes", then it drops the HCI packet. Otherwise, it proceeds to Step 3.
- **Step 3 - Forward HCI packet:** The node broadcasts the HCI packet to its 1-hop neighbors.

3.4.2 Data forwarding

After the *core polygon dissemination* phase, the nodes can be divided into two kinds: the first kind (which we call *hole-aware node*) consists of nodes that have information of core polygons, and the second kind (which we call *blind node*) consists of the other nodes. If a packet is initiated at a blind node, then it is forwarded toward the destination using geographic greedy routing until arriving at the first hole-aware node. This first hole-aware node is called the *sub-destination* node. Assume that s and t are the source and destination, respectively, and t' is the sub-destination node (in case s is a blind node). Below, we show how s (or t') can utilize the core polygon information to make a routing decision.

Our data forwarding protocol consists of three steps: First, s (resp. t') determines the so-called base path (resp. the sub-base path) which is the shortest path from s (resp. t') to t that bypasses the core polygon. Then, the base path (or the sub-base path) is magnified by using a homothetic transformation. The center O of the transformation is chosen randomly inside the core polygon, and the scale factor $\xi \leq 1$ is computed based on the distance between the source and the destination. The image obtained from

[4]The pseudo code is presented by Algorithm 2 in the Appendix.

the homothetic transformation is called the Euclidean routing path which acts as the guidance line for the packet. Specifically, the locations of the vertices of the Euclidean routing path are inserted into the packet as the *virtual anchors*. Finally, the packet is then forwarded toward these virtual anchors greedily until reaching the destination. Note that because the virtual anchors are not necessarily sensor nodes, the packet will be forwarded to the nodes nearest to the virtual anchors[5].

The base path (or the sub-base path) can be determined as follows. Node s (or t') determines the view limit vertices v_s^l, v_s^r and v_t^l, v_t^r of s (or t') and t to $\mathscr{C}$ such that v_s^l, v_t^l stay on the left side and v_s^r, v_t^r stay on the right side of $\overrightarrow{st}$ (or $\overrightarrow{t't}$), respectively. The base path (or the sub-base path) is chosen as the shorter of $s\{v_s^r \sim v_t^r\}_{\mathscr{C}}t$ (or $t'\{v_s^r \sim v_t^r\}_{\mathscr{C}}t$) and $s\{v_s^l \sim v_t^l\}_{\mathscr{C}}^{-}t$ (or $t'\{v_s^l \sim v_t^l\}_{\mathscr{C}}^{-}t$).

Let C_s and C_t be the view limit vertices of s (or t') and t to $\mathscr{C}$ that belong to the base path, respectively. Let $a = |\{C_s \sim C_t\}_{\mathscr{C}}|$; $b = |OC_s| + |OC_t|$; $c = |sC_s| + |C_t t|$ if s is a hole aware node and $z = |t'C_s| + |C_t t|$, otherwise. Then ξ is computed as follows.

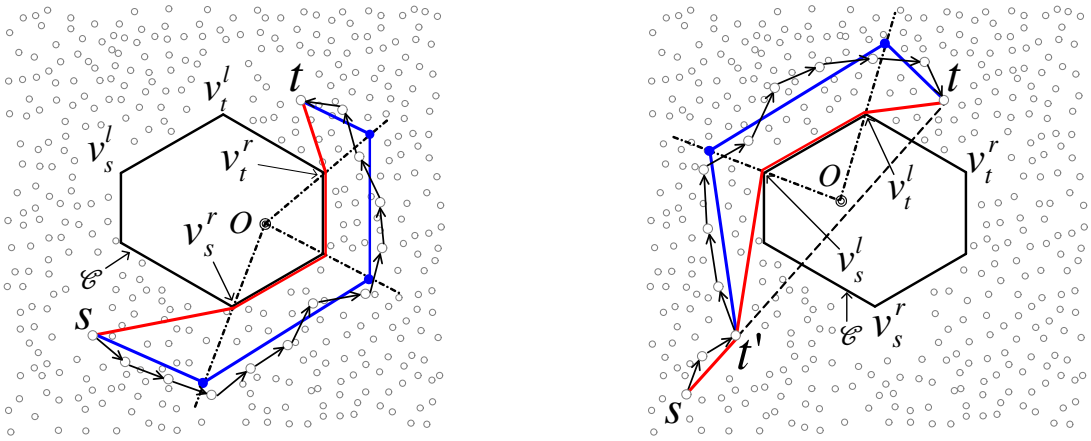
$$\xi = 1 + (\alpha - 1)\frac{a+c}{a+b} \tag{3.8}$$

where ξ is a parameter determined as follows.

- $\alpha = (1+\varepsilon)sin\frac{(n-2)\pi}{2n}$ if x is a source node.
- $\alpha = Max\left\{1, (1+\varepsilon)sin\frac{(n-2)\pi}{2n}cos\frac{v_{\mathscr{C}}(t')}{2} + \frac{\left((1+\varepsilon)sin\frac{(n-2)\pi}{2n}cos\frac{v_{\mathscr{C}}(t')}{2} - 1\right)|st'|}{|l_{\mathscr{C}}(t',t)|}\right\}$, otherwise.

Figure 3.8 illustrates an example. In this figure, the red solid lines represent the shortest paths bypassing the core polygon, and the dotted black lines represent the real routing paths. Figure 3.8(a) illustrates the forwarding scheme when the source node is a hole-aware node. In this figure, v_s^r, v_1, v_t^r are the virtual anchors and A_1, A_2, A_3 are the nodes nearest to them. Therefore, the routing path will be $s \to A_1 \to A_2 \to A_3 \to t$. An example in which the source is a blind node is presented in Figure 3.8(b). In this figure, the packet is first greedily forwarded toward the destination t until reaching the sub-source node t'. From t', the packet is forwarded along the shortest path bypassing the core polygon (i.e., $t' \to A_1 \to A_2 \to t$).

[5]The pseudo code is presented by Algorithm 3 in the Appendix.



(a) When the source node is a hole-aware node.

(b) When the source node is a blind node.

Figure 3.8: Examples of the routing scheme.
The red lines represent the base paths, the blue lines represent the Euclidean routing path. The blue dots are virtual anchors.

3.4.3 Theoretical analysis

In this section, we will show that the E-stretch of the routing path generated by our protocol is upper bounded by the predefined threshold $1+\varepsilon$.

Lemma 5

Let $\mathscr{Q}$ be a convex polygon, and let $\mathscr{P}$ be its circumscribing polygon. Denote $P_1, P_2, ..., P_n$ as the vertices of $\mathscr{P}$ (ordered in the counterclockwise direction). Assume that $P_{r_1}, ..., P_{r_m}$ are the vertices of $\mathscr{P}$ that are not vertices of $\mathscr{Q}$. Then, $p_{\mathscr{P}} \leq \frac{1}{\sin\frac{\rho}{2}} p_{\mathscr{Q}}$, where $\rho = \min_{i=\overline{1,m}} \widehat{P_{r_i}}$.

Proof

First, we prove the following statement (see Figure 3.9(a) for the illustration):

Let P_1, P_2, P_3 are three vertices (ordered in the counterclockwise direction) of a triangular $\mathscr{P}$. Let $Q_1, ..., Q_n$ be n points staying inside of $\mathscr{P}$ such that $P_1Q_1, Q_2, ..., Q_nP_3$ forms a convex polygon (denote this polygon as $\mathscr{Q}$). Then, $|\{P_1 \sim P_3\}_{\mathscr{P}}| \leq \frac{1}{\sin\frac{\angle P_1P_2P_3}{2}} |\{P_1 \sim P_3\}_{\mathscr{Q}}|$.

According to Lemma 4, we have: $|\{P_1 \sim P_3\}_{\mathscr{P}}| \leq \frac{1}{\sin\frac{\angle P_1P_2P_3}{2}} |P_1P_3|$.

Note that, $|P_1P_3| \leq |\{P_1 \sim P_3\}_{\mathscr{Q}}|$. Therefore, $|\{P_1 \sim P_3\}_{\mathscr{P}}| \leq \frac{1}{\sin\frac{\angle P_1P_2P_3}{2}} |\{P_1 \sim P_3\}_{\mathscr{Q}}|$.

Now, we prove the proposition (see Figure 3.9(b) for the illustration).

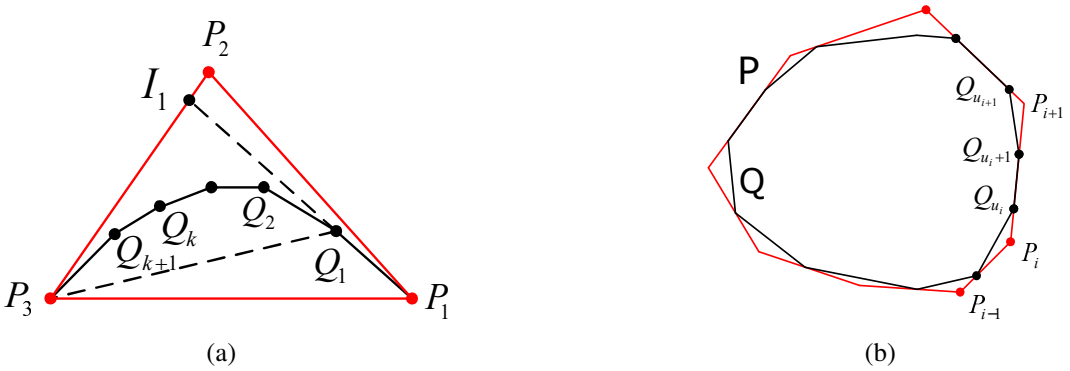


Figure 3.9: Illustration of the proof of Lemma 5.

Denote by $Q_{u_i},...,Q_{u_i+\delta_i}$ the vertices of $\mathscr{Q}$ staying on P_iP_{i+1}, where $i=1,...,n$, $\delta_i \in \{0,1\}$, $P_{n+1} \equiv P_1$. For each $i \in \{1,..,n\}$ such that P_{i+1} is not a vertex of $\mathscr{Q}$, consider triangular $Q_{u_i}P_{i+1}Q_{u_{i+1}}$ ($\forall i=1,...,n$), then according to the above statement, we have:

$$\left|\{Q_{u_i} \sim Q_{u_{i+1}}\}_{\mathscr{P}}\right| \leq \frac{1}{\sin\frac{\angle Q_{u_i}P_{i+1}Q_{u_{i+1}}}{2}} \left|\{Q_{u_i} \sim Q_{u_{i+1}}\}_{\mathscr{Q}}\right| \tag{3.9}$$

Note that, if P_{i+1} is a vertex of $\mathscr{Q}$ then $Q_{u_i} \equiv P_{i+1} \equiv Q_{u_{i+1}}$ and thus (3.9) becomes trivial.

By summing up inequality (3.9) for $\forall i=1,...,n$ the proposition is proved.

Lemma 6

Let $\mathscr{Q}$ be a convex polygon, and let $\mathscr{P}$ be another convex polygon which entirely covers $\mathscr{Q}$. Then, for any two arbitrary points (s,t) staying outside of $\mathscr{P}$, the following two statements are true:

1. $|l_{\mathscr{P}}(s,t)| - |l_{\mathscr{Q}}(s,t)| \leq p_{\mathscr{P}} - p_{\mathscr{Q}}$.
2. *The E-stretch of $\mathscr{P}$ to $\mathscr{Q}$ for $(s,t) \leq 1 + \frac{p_{\mathscr{P}}-p_{\mathscr{Q}}}{|l_{\mathscr{Q}}(s,t)|}$.*

Proof

Figure 3.10 illustrates the proof. Denote by $Q_1,...,Q_n$ the vertices of $\mathscr{Q}$, and let $P_1,...,P_m$ be the vertices of $\mathscr{P}$, which are ordered in the counterclockwise direction. Denote by $s\{Q_u \sim Q_{u+k}\}_{\mathscr{Q}}t$ the shortest path between s and t that bypasses $\mathscr{Q}$, and assume that this path stays on the right side of $\overrightarrow{st}$. According to Proposition 1, Q_u and Q_{u+k} must be the view-limit vertex of s and t to $\mathscr{Q}$, respectively. Denote by $P_v,..,P_{v+h}$ all vertices of $\mathscr{P}$ that stay on the right side of $\overrightarrow{st}$; then, $l_{\mathscr{P}}(s,t) \leq s\{P_v \sim P_{v+h}\}_{\mathscr{P}}t$. Denote by Q'_u

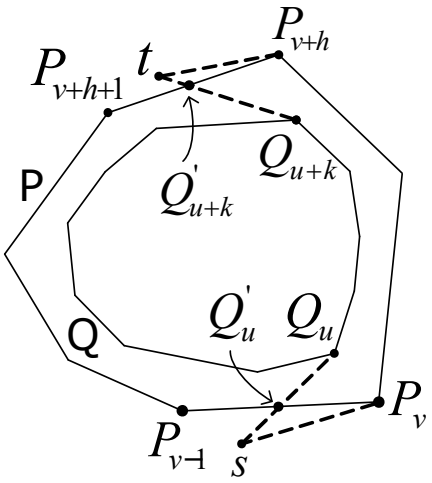


Figure 3.10: Illustration of the proof of Lemma 6.

and Q'_{u+k} the intersection of sQ_u and tQ_{u+k} with $P_{v-1}P_v$ and $P_{v+h}P_{v+h+1}$, respectively. Then,

$$l_{\mathscr{P}}(s,t) \leq s\{P_v \sim P_{v+h}\}_{\mathscr{P}} t \leq \left|sQ'_u\right| + \left|\{Q'_u \sim Q'_{u+k}\}_{\mathscr{P}}\right| + \left|Q'_{u+k}t\right| \tag{3.10}$$

It is obvious that

$$\left|Q_{u+k}Q'_{u+k}\right| + \left|\{Q'_{u+k} \sim Q'_u\}_{\mathscr{P}}\right| + \left|Q'_uQ_u\right| \geq |\{Q_{u+k} \sim Q_u\}_{\mathscr{Q}}| \tag{3.11}$$

Note that $p_{\mathscr{P}} = \left|\{Q'_u \sim Q'_{u+k}\}_{\mathscr{P}}\right| + \left|\{Q'_{u+k} \sim Q'_u\}_{\mathscr{P}}\right|$. Therefore, from (3.11), it is deduced that

$$\left|\{Q'_u \sim Q'_{u+k}\}_{\mathscr{P}}\right| \leq p_{\mathscr{P}} - |\{Q_{u+k} \sim Q_u\}_{\mathscr{Q}}| + \left|Q_{u+k}Q'_{u+k}\right| + \left|Q'_uQ_u\right| \tag{3.12}$$

Note that $p_{\mathscr{Q}} = |\{Q_u \sim Q_{u+k}\}_{\mathscr{Q}}| + |\{Q_{u+k} \sim Q_u\}_{\mathscr{Q}}|$. Therefore, (3.12) is equivalent to

$$\left|\{Q'_u \sim Q'_{u+k}\}_{\mathscr{P}}\right| \leq p_{\mathscr{P}} - p_{\mathscr{Q}} + |\{Q_u \sim Q_{u+k}\}_{\mathscr{Q}}| + \left|Q_{u+k}Q'_{u+k}\right| + \left|Q'_uQ_u\right| \tag{3.13}$$

By adding (3.10) and (3.13), we have

$$l_{\mathscr{P}}(s,t) \leq p_{\mathscr{P}} - p_{\mathscr{Q}} + l_{\mathscr{Q}}(s,t) \tag{3.14}$$

or $l_{\mathscr{P}}(s,t) - l_{\mathscr{Q}}(s,t) \leq p_{\mathscr{P}} - p_{\mathscr{Q}}$. Statement (1) is proved. Statement (2) can be deduced directly from statement (1) because the E-stretch of $\mathscr{P}$ to $\mathscr{Q}$ for (s,t) equals $1 + \frac{|l_{\mathscr{P}}(s,t)| - |l_{\mathscr{Q}}(s,t)|}{|l_{\mathscr{Q}}(s,t)|}$.

Theorem 2

For every source and destination pair (s,t) staying outside of the core polygon, the stretch of routing paths determined by our strategy does not exceed $1+\varepsilon$.

Proof

Let $\mathscr{R}(s,t)$ and $\mathscr{L}(s,t)$ denote the routing path determined by our protocol and the shortest routing path, respectively. Note that $\mathscr{L}(s,t) = l_{\mathscr{H}}(s,t)$. Moreover, according to our model (described in Section 3.1.1), the routing path stretch can be approximated by $\frac{|\mathscr{R}(s,t)|}{|\mathscr{L}(s,t)|}$, this is equivalent to $\frac{|\mathscr{R}(s,t)|}{|l_{\mathscr{H}}(s,t)|}$. In the following, we will prove that $\frac{|\mathscr{R}(s,t)|}{|l_{\mathscr{H}}(s,t)|} \leq 1+\varepsilon$.

Case 1: s is a hole aware node

Let $\mathscr{B}(s,t)$ be the base path determined by s, then $\mathscr{B}(s,t) = l_{\mathscr{C}}(s,t)$ (i.e., the shortest path from s to t that bypasses the core polygon $\mathscr{C}$). Suppose that $\mathscr{B}(s,t) = s\{C_s \sim C_t\}_{\mathscr{C}} t$, where C_s and C_t are the view limit vertices of s and t to $\mathscr{C}$, respectively. Also suppose that R_s and R_t are two vertices belonging to $\mathscr{R}(s,t)$ that are the images of C_s and C_t through the homothetic transformation, respectively. According to Theorem 1, we have the following:

$$|\mathscr{B}(s,t)| = |l_{\mathscr{C}}(s,t)| \leq \frac{1}{sin\frac{(n-2)\pi}{2n}} |l_{\mathscr{H}}(s,t)| \tag{3.15}$$

Denote $\{R_s \sim R_t\}$ as the image of $\{C_s \sim C_t\}_{\mathscr{C}}$ through the homothetic transformation, then $\mathscr{R}(s,t)$ can be decomposed into three paths: sC_s, $\{R_s \sim R_t\}$ and $C_t t$. Obviously, we have:

$$|\{R_s \sim R_t\}| = \xi |\{C_s \sim C_t\}_{\mathscr{C}}| \tag{3.16}$$

Moreover, using the triangular cosine formula, we have:

$$|sR_s| \leq |sC_s| + |C_s R_s| = |sC_s| + (\xi - 1)|OC_s| \tag{3.17}$$

$$|R_t t| \leq |C_t t| + |C_t R_t| = |C_t t| + (\xi - 1)|OC_t| \tag{3.18}$$

From (3.16), (3.17), (3.18), it is deduced that:

$$|\mathscr{R}(s,t)| \leq |sC_s| + |C_t t| + \xi |\{C_s \sim C_t\}_{\mathscr{C}}| + (\xi - 1)(|IC_s| + |IC_t|) = z + \xi x + (\xi - 1)y \tag{3.19}$$

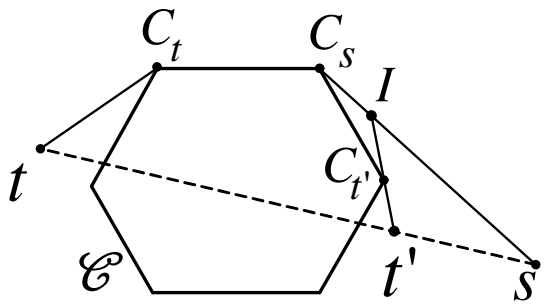


Figure 3.11: Illustration of the proof of Theorem 2.

By substituting the value of ξ in (3.8) into (3.19), we obtain:

$$|\mathscr{R}(s,t)| \leq \alpha |\mathscr{B}(s,t)| = (1+\varepsilon) sin\frac{(n-2)\pi}{2n} |\mathscr{B}(s,t)| \tag{3.20}$$

From (3.15) and (3.20), the theorem is proved ■

Case 2: s is a blind node

We first prove the following statement:

$$|l_{\mathscr{C}}(s,t)| \geq cos\frac{v_{\mathscr{C}}(t')}{2} \left(|st'| + |l_{\mathscr{C}}(t',t)|\right) \tag{3.21}$$

According to Proposition 2, $l_{\mathscr{C}}(t',t)$ can be expressed by $t'\{C_{t'} \sim C_t\}_{\mathscr{C}} t$, where $C_{t'}, C_t$ denote the view-limit vertices of t' and t to $\mathscr{C}$, respectively. Without loss of generality, assume that $C_{t'}, C_t$ remain on the right-hand side of $\overrightarrow{t't}$, as shown in Figure 3.11. Denote C_v as the view-limit vertex of s to $\mathscr{C}$ staying on the right-hand side of $\overrightarrow{t't}$, and I as the intersection of $\overrightarrow{t'C_{t'}}$ and sC_v.

According to Proposition 4, we have

$$|sI| \geq \sin\frac{\angle st'I}{2} \left(|st'| + |t'I|\right) \tag{3.22}$$

Note that

$$\angle st'I = \pi - \angle It't \geq \pi - v_{\mathscr{C}}(t') \tag{3.23}$$

Therefore, from (3.22), it can be deduced that

$$|sI| \geq \cos\frac{v_{\mathscr{C}}(t')}{2} \left(|st'| + |t'I|\right) \tag{3.24}$$

On the other hand, we have

$$|l_{\mathscr{C}}(s,t)| \geq \left|s\{C'_t \sim C_t\}_{\mathscr{C}}t\right| = |sI| + |IC_v| + \left|\{C'_t \sim C_t\}_{\mathscr{C}}\right| + |C_t t| \tag{3.25}$$

From (3.24) and (3.25), it is deduced that

$$\begin{aligned} |l_{\mathscr{C}}(s,t)| &\geq \cos\frac{v_{\mathscr{C}}}{2}\left(\left|st'\right| + \left|t'I\right| + |IC_v| + \left|\{C'_t \sim C_t\}_{\mathscr{C}}\right| + |C_t t|\right) \\ &= \cos\frac{v_{\mathscr{C}}}{2}\left(\left|st'\right| + |l_{\mathscr{C}}(t',t)|\right) \end{aligned} \tag{3.26}$$

Let $\mathscr{B}(t',t)$ and $\mathscr{R}(t',t)$ be the sub-base path and the Euclidean routing path determined by t'. Note that $\mathscr{B}(t',t)$ is the shortest path from t' to t, i.e., $l_{\mathscr{C}}(t',t)$. Since $\mathscr{R}(t',t)$ is the image of $\mathscr{B}(t',t)$ through the homothetic transformation with the scale factor of ξ, using the same proof as shown in the Case 1, it can be deduced that:

$$|\mathscr{R}(t',t)| \leq \xi|\mathscr{B}(t',t)| = \alpha|l_{\mathscr{C}}(t',t)| \tag{3.27}$$

The routing path from s to t can be decomposed into two parts: the first path is the routing path from s to t' (whose Euclidean length can be approximated by $|st'|$), and the second part is the routing path from t' to t (whose Euclidean length can be approximated by $|\mathscr{R}(t',t)|$). Therefore, we have the following:

$$|\mathscr{R}(s,t)| \leq |st'| + \alpha|l_{\mathscr{C}}(t',t)|| \tag{3.28}$$

Since α obtains the value of 1 or $(1+\varepsilon)sin\frac{(n-2)\pi}{2n}cos\frac{v_{\mathscr{C}}(t')}{2} + \frac{\left((1+\varepsilon)sin\frac{(n-2)\pi}{2n}cos\frac{v_{\mathscr{C}}(t')}{2}-1\right)|st'|}{|l_{\mathscr{C}}(t',t)|}$, in the following we prove the theorem in these two cases.

1) If $\alpha = (1+\varepsilon)sin\frac{(n-2)\pi}{2n}cos\frac{v_{\mathscr{C}}(t')}{2} + \frac{\left((1+\varepsilon)sin\frac{(n-2)\pi}{2n}cos\frac{v_{\mathscr{C}}(t')}{2}-1\right)|st'|}{|l_{\mathscr{C}}(t',t)|}$:

According to Theorem 1, we have the following:

$$|l_{\mathscr{H}}(s,t)| \geq \sin\frac{(n-2)\pi}{2n}|l_{\mathscr{C}}(s,t)| \tag{3.29}$$

From (3.26) and (3.29), we obtain:

$$|l_{\mathscr{H}}(s,t)| \geq \sin\frac{(n-2)\pi}{2n}\cos\frac{v_{\mathscr{C}}}{2}\left(\left|st'\right| + |l_{\mathscr{C}}(t',t)|\right) \tag{3.30}$$

By substituting the value of ξ into (3.28), we have: $|\mathscr{R}(s,t)| \leq (1+\varepsilon)|l_{\mathscr{H}}(s,t)|$.
1) If $\xi = 1$:

According to Lemma 6, we have:

$$|l_{\mathscr{C}}(s,t)| - |l_{\mathscr{H}}(s,t)| \leq p_{\mathscr{C}} - p_{\mathscr{H}} \tag{3.31}$$

Moreover, according to Lemma 5, we have:

$$p_{\mathscr{C}} \leq \frac{1}{\sin\frac{(n-2)\pi}{2n}} p_{\mathscr{H}} \tag{3.32}$$

From (3.31) and (3.32), is is deduced that:

$$|l_{\mathscr{H}}(s,t)| \geq |l_{\mathscr{C}}(s,t)| - \left(1 - \sin\frac{(n-2)\pi}{2n}\right) p_{\mathscr{C}} \tag{3.33}$$

From (3.26) and (3.33), we obtain the following:

$$|l_{\mathscr{H}}(s,t)| \geq \cos\frac{v_{\mathscr{C}}}{2}\left(|st'| + |l_{\mathscr{C}}(t',t)|\right) - \left(1 - \sin\frac{(n-2)\pi}{2n}\right) p_{\mathscr{C}} \tag{3.34}$$

Note that as $\alpha = 1$, we have: $\xi = 1$, and thus $|\mathscr{R}(s,t)| = |st'| + |l_{\mathscr{C}}(t',t)|$. Therefore,

$$\frac{|\mathscr{R}(s,t)|}{|l_{\mathscr{H}}(s,t)|} \leq \frac{|st'| + |l_{\mathscr{C}}(t',t)|}{\cos\frac{v_{\mathscr{C}}}{2}\left(|st'| + |l_{\mathscr{C}}(t',t)|\right) - \left(1 - \sin\frac{(n-2)\pi}{2n}\right) p_{\mathscr{C}}} \tag{3.35}$$

Because t' is the first hole-aware node that receives the packet, the previous node of t' must be a blind node. This means that t' has information of the hole core polygon but its 1-hop neighbor does not. Therefore, t' must stay on the border of the dissemination region, and it must satisfy condition (3.7), i.e., $cos\frac{v_{\mathscr{C}}(t')}{2} > \frac{1}{1+\varepsilon} + \frac{p_{\mathscr{C}}\left(1-sin\frac{(n-2)\pi}{2n}\right)}{d_{\mathscr{C}}(t')}$. Thus, from (3.35), it is deduced that $\frac{|\mathscr{R}(s,t)|}{|l_{\mathscr{H}}(s,t)|} < 1+\varepsilon$. The proof is completed ■

3.4.4 Section summary

In this section, we addressed the hole bypassing problem in the simplest scenario, where the network contains only one hole and all the sources and the destination stay outside of the core polygon. We proposed a hole information dissemination strategy, in which

nodes close to the hole receive the core polygon information, while nodes far from the hole do not. If a packet is originated at a node without hole information, it is greedily forwarded towards the destination until arriving at an intermediate node which has core polygon information. The first node on the routing path that has core polygon information (i.e., either the source node or an intermediate node) conducts the following strategy to forward the packet:

- The node determines the so-called base path which is the shortest path to the destination that bypasses the core polygon.
- The base path then is magnified using a homothetic transformation whose center is chosen randomly inside the core polygon, while the scale factor is controlled to guarantee the stretch upper bound. The image obtained by this homothetic transformation is called the Euclidean routing path.
- The vertices of the Euclidean routing path are inserted to the packet and act as virtual anchors. The packet then is greedily forwarded towards the virtual anchors until reaching the destination.

A thorough theoretical analysis on the stretch of routing path determined by our strategy has been conducted and it was shown that the stretch is upper bounded by a predefined threshold.

3.5 Strategy for bypassing hole's vicinity

In the two previous sections, we have solved the hole bypassing problem when all the sources and the destinations staying outside of the holes' core polygons. Unfortunately, in practice, the sources and the destination may fall into the core polygons. In such cases, the routing strategies proposed in these two sections can not be applied. To this end, in this section, we tackle the problem of forwarding packets in the vicinity of a hole. Specifically, we focus on a network topology consisting of one hole, and the sources and destinations of packets may stay inside the hole's core polygon.

3.5.1 Challenges and strategies

There are two challenges when tackling the routing problem in the hole's vicinity. The first one is how to determine the base paths satisfying the path length condition (i.e. the

length of a base path does not exceed the predefined times that of the shortest path). The second one is how to scale up the base path such that the image obtained by the homothetic transformations (i.e., Euclidean routing path) does not intersect the holes' interior. We describe our strategies to deal with these two problems in the following.

3.5.1.1 Base path determination strategy

Let us consider the most complicated case when both the source s and the destination t stay inside the holes' convex hulls. Let us denote the holes whose convex hulls contain s and t as $\mathscr{H}(s)$ and $\mathscr{H}(t)$, respectively, and denote $\mathscr{V}(s)$ and $\mathscr{V}(t)$ as the convex hulls of $\mathscr{H}(s)$ and $\mathscr{H}(t)$, respectively. Then intuitively, the shortest hole-bypassing Euclidean path from s to t can be decomposed into three parts. The first one, which we call as *go-out path*, is the one from s to the first point residing on the boundary of $\mathscr{V}(s)$ (let us denote this point as G_s). The second one, which we call as *go-between path*, is the one connecting G_s to a point on the boundary of $\mathscr{V}(t)$ (let us denote this point as G_t). The last one, which is called as *go-in path*, is the one connecting G_t and t. We note that to determine the go-out path and go-in path, one needs to know the holes' full information. However, as the go-between path contains only nodes residing on the convex hulls of the holes, determining the go-between path requires only information of the holes' convex hulls. Furthermore, as will be shown in Lemma 1, if one doesn't need to determine the exact shortest path from s to t, but just wants to determine a path whose length satisfying the path length condition, then one can replace the convex hulls by simple convex polygons that have a smaller number of vertices than the convex hulls, e.g., core polygons. Accordingly, we propose a hole information dissemination strategy as follows.

- Broadcast full information of a hole to nodes staying inside the hole or close to the hole boundary.
- For the nodes staying far from the hole boundary, broadcast only information of the hole's core polygon.

3.5.1.2 Euclidean routing path determination strategy

The second challenge is how to choose the homothetic centers such that the image of the base path obtained through the homothetic transformations does not intersect the hole's

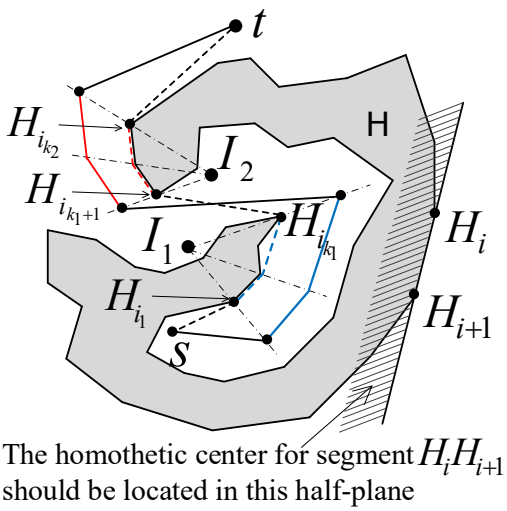


Figure 3.12: Strategy to choose the homothetic centers.
The bold dashed line represents the shortest path, while the bold solid line represents the real routing path, from s to t. The blue segment (i.e., $H_{i_1}...H_{i_{k_1}}$) is a negative segment, thus its homothetic center, I_1, resides on its left side. The red segment (i.e., $H_{i_{k_1+1}}...H_{i_{k_2}}$) is a positive segment, thus its homothetic center, I_2, resides on its right side..

interior. Obviously, the strategies used in the two previous sections (i.e., the homothetic centers are arbitrary points inside the core polygons) can not be applied in this context. We observe that in a homothetic transformation, the homothetic center and the image reside on opposite sides with respect to the object. Therefore, to prevent the image from intersecting with the hole, homothetic centers should reside on the same side with the hole with respect to the shortest path. Suppose the vertices on the hole boundary are indexed by such ordered that the hole stays on the right side of the boundary regarding the increasing direction of the indexes. Then, if H_i and H_j are two consecutive hole boundary nodes on the shortest path, the homothetic center should reside on the right side of $\overrightarrow{H_iH_j}$ if $i < j$, but on the left side of $\overrightarrow{H_iH_j}$ if $i > j$. Obviously, there may be no single homothetic center that can satisfy this requirement for all nodes on the shortest path. Therefore, we divide the shortest path into segments, each of which is comprised of boundary nodes ordered in the ascending order (i.e., called as *positive segment*) or descending order (i.e., called as *negative segment*) of the indexes. For each segment, we use a specific homothetic center, where the homothetic center of a positive segment resides on the right side and the homothetic center of a negative segment resides on the left side of that segment. Upon determining homothetic centers, each segment will be

magnified by using a homothetic transformation with the chosen center and a specific scale factor. The combination of the images of the segments through the homothetic transformations forms the routing path. Fig. 3.12 shows an example. In this figure, $sH_{i_1}...H_{i_{k_2}}t$ is the shortest path from s to t. $H_{i_1}...H_{i_{k_1}}...H_{i_{k_2}}$ is divided into two segments: $H_{i_1}...H_{i_{k_1}}$ and $H_{i_{k_1+1}}...H_{i_{k_2}}$. $H_{i_1}...H_{i_{k_1}}$ is a negative segment (i.e., $i_1 > ... > i_{k_1}$), thus its homothetic center (i.e., I_1) stays on its left side. Whereas, $H_{i_{k_1+1}}...H_{i_{k_2}}$ is a positive segment (i.e., $i_{k_1+1} < ... < i_{k_2}$), thus its homothetic center (i.e., I_2) stays on its right side. $H_{i_1}...H_{i_{k_1}}$ is magnified using a homothetic transformation with the center of I_1, while $H_{i_{k_1+1}}...H_{i_{k_2}}$ is magnified using a homothetic transformation with the center of I_2. The combination of the images obtained from these transformations forms the routing path.

3.5.2 Hole and core polygon information dissemination

To reduce the dissemination overhead, our protocol is designed so that only the nodes close to a hole receive the full information of the hole (hereafter we call this full information the hole information), and the other nodes that stay far from the hole receive only the information of the hole's core polygon (hereafter, we call this information the core polygon information). We call the region where the hole information is broadcast as *vicinity region* of the hole and define as follows.

Definition (Vicinity region)
Let $\mathscr{H}$ be a hole and $\mathscr{C}$ be its core polygon. Denote by $p_{\mathscr{C}}$ the perimeter of $\mathscr{C}$, and $e_{\mathscr{C}}$ the length of the longest edge of $\mathscr{C}$. Then, the vicinity region of $\mathscr{H}$ consists of all nodes whose shortest distance to every vertex of $\mathscr{C}$ is not greater than $\frac{p_{\mathscr{C}}}{2\varepsilon} + \frac{e_{\mathscr{C}}}{2}$. (Fig.3.13)

After the first phase, the hole information and the core polygon information of the hole has been stored at the *BCD-initiator*. Now, the *BCD-initiator* creates a *hole core information (HCI)* packet that conveys the hole information (i.e., the coordinates of all nodes residing on the hole boundary), and the core polygon information (i.e., the coordinates of all vertices of the core polygon). The *BCD-initiator* then broadcasts its HCI packet to its 1-hop neighbors. When a node receives an HCI packet, it checks if the hole or the core polygon stored in the HCI message has been already stored in its local memory. If "yes", it simply drops the HCI packet (because it has been received the same HCI message before). Otherwise, it conducts the following process:

- If it stays inside the vicinity region of the hole stored in the HCI packet, then it

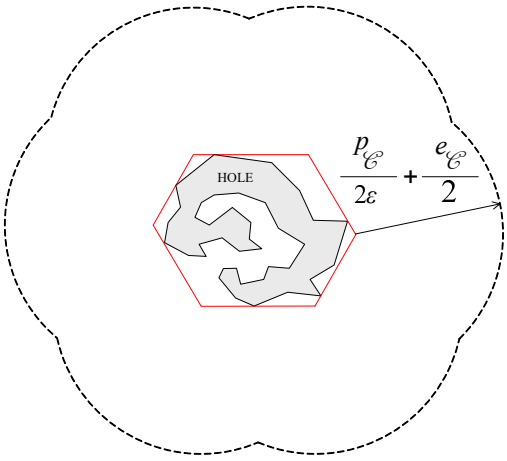


Figure 3.13: Vicinity region of the hole is the regions limited by the dotted line.

stores the hole information into the local memory and broadcasts the HCI message to the neighbors.

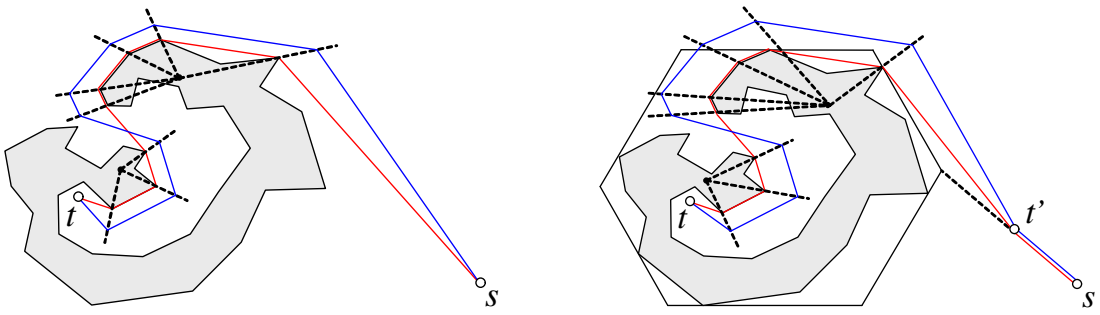
- Otherwise, it stores the core polygon information into the local memory, removing the hole information from the HCI message before broadcasting to the neighbors.

3.5.3 Data forwarding

After the second phase, all the nodes in the vicinity region of the hole have hole information, while the other nodes have only information of the core polygon. We call nodes with hole information as hole aware nodes, and nodes with only core polygon information as blind nodes. In this section, we will show how a source node s can exploit this information to make a routing decision to a destination t.

Similar to the data forwarding strategies described in the previous sections, our data forwarding protocol consists of three steps: determining the base path, determining the Euclidean routing path, and forwarding the packet along the Euclidean path. Before going to the detail of each step, we sketch the overview. Let us denote the hole as $\mathscr{H}$ and the core polygon as $\mathscr{C}$. If s is a hole aware node, then s determines the base path as the shortest Euclidean path from s to t which bypasses $\mathscr{H}(s)$. This base path then is magnified using a transformation to obtain the Euclidean routing path. Finally, the packet is forwarded along the Euclidean routing path until reaching t.

Otherwise, if s is a blind node, then it first greedily forward packet towards the



(a) Case 1: s is a hole aware node.

(b) Case 2: s is a blind node.

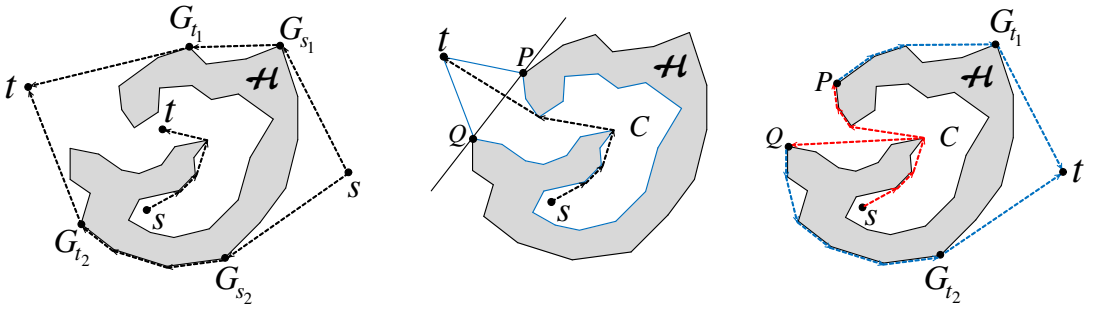
Figure 3.14: Data forwarding.
The red/orange paths lines are the base path/sub-base path, the blue/green lines are the Euclidean routing path/sub-Euclidean routing path.

nearest view-limit vertex of s to the core polygon $\mathscr{C}$ until reaching a node (let us call this node as t') that has the hole information of $\mathscr{H}$. t' is called the sub-destination node. Note that since the last vertex of the Euclidean routing path is a vertex of $\mathscr{C}$, such the sub-destination node always exists. Having the hole information of $\mathscr{H}$, t' determines a so-called *sub-base path* which is the shortest Euclidean path from t' to t, which bypasses $\mathscr{H}$. This sub-base path then is magnified by a transformation to obtain a so-called *sub-Euclidean routing path*. The packet then is greedily forwarded along the sub-Euclidean routing path until reaching t. See Fig.3.14 for the illustration.

3.5.3.1 Base path determination

Before going to the algorithm details, we start with the following observations. Let s and t be two points on the plane. Let $l_{\mathscr{H}}(s,t)$ denote the shortest path from s to t that bypasses $\mathscr{H}$.

Proposition 1 (outside-convex shortest path)
Suppose s and t stay outside or on the boundary of the convex hull $\mathscr{G}$ (of the hole $\mathscr{H}$) such that st intersects $\mathscr{G}$. Let G_{s_1}, G_{s_2}, G_{t_1}, and G_{t_2} denote the view-limit vertices of $\mathscr{G}$ with respect to s and t such that G_{s_1} and G_{t_1} are on the right side and G_{s_2} and G_{t_2} are on the left side of $\overrightarrow{st}$. Then, $l_{\mathscr{H}}(s,t)$ is the shorter one between $s\{G_{s_1} \sim G_{t_1}\}_{\mathscr{G}}t$ and $s\{G_{s_2} \sim G_{t_2}\}_{\mathscr{G}}t$ (Fig. 3.15(a)).



(a) If s and t stay outside of the convex hull, $L(s,t)$ is the shorter one between $s\{G_{s_1} \sim G_{t_1}\}_{\mathscr{G}}t$ and $s\{G_{s_2} \sim G_{t_2}\}_{\mathscr{G}}t$. If s and t stay inside the same concave region, $L(s,t)$ can be determined using the algorithm proposed in [173].

(b) P,Q are two gate-points of C. s and t can be considered as staying in the polygon whose boundary colored blue. $L(s,t)$ can be determined using the algorithm proposed in [173].

(c) $L(s,t)$ is either $s..P..G_{t_{i_1}}...t$ or $s..Q..G_{t_{i_2}}...t$. The red lines present the inside-convex parts, the blue lines present the outside-convex parts

Figure 3.15: Hole bypassing shortest path.

Proposition 2 (inside-convex shortest path)
Suppose s stays inside a concave region $\mathscr{V}$ of the hole $\mathscr{H}$, and t stays outside of $\mathscr{E}$ such that s and t reside on the same half-plane with respect to line PQ. Denote by P and Q the two gate-points of $\mathscr{V}$. Then, $l_{\mathscr{H}}(s,t)$ must pass through either P or Q (Fig. 3.15(c)).

Based on these observations, we propose an algorithm to determine the shortest path from s and t that bypasses the hole $\mathscr{H}$ (denoted as $l_{\mathscr{H}}(s,t)$) as follows:

- If both s and t stay outside or on the boundary of the hole's convex hull, then $l_{\mathscr{H}}(s,t)$ can be easily determined by using Proposition 1 (Fig. 3.15(a)).
- If s and t stay inside the same concave region, then $L(s,t)$ can be determined using the algorithm proposed in [173] [6] (Fig. 3.15(a)).
- If either s or t (e.g., s) stays inside a concave region (denoted as C), and s, t stay in opposite half-planes with respects to the line connecting the gate-points of C, then s,t can be considered as staying inside the polygon made by C and t. Therefore, $l_{\mathscr{H}}(s,t)$ can be determined by using the algorithm proposed in [173] (Fig. 3.15(b)).

[6] [173] proposed an algorithm to determine the shortest path between two points staying inside a polygon.

- If at least one of s and t (e.g., s) resides on a concave region (denoted as C), and both s and t stay in the same half-plane with respects to the line connecting the gate-points of C, then $l_{\mathscr{H}}(s,t)$ can be divided into the following two parts (see (Fig. 3.15(c)). The first one (namely, *inside-convex part*) connects s (or/and t) to the gate-point(s) of their concave region(s). The second one (namely, *outside-convex part*) connects either the two gate-points of the concave regions (in the case t also stays inside a concave region), or the gate-point of s's concave region with t. The inside-convex part can be determined using [173], while the outside-convex part can be determined using Proposition 1.

3.5.3.2 Determining the Euclidean routing path

Let x denote the source node if the source node is a hole aware node, and the sub-destination node if the source node is a blind node. After determining the base path (or the sub-base path), denoted as $\mathscr{B}(x,t)$, x constructs the Euclidean routing path (or sub-Euclidean routing path) by using homothetic transformations as follows. Suppose $\mathscr{B}(x,t) = xH_1H_2...H_nt$, then x divides $\mathscr{B}(x,t)$ into segments such that each of which is either positive-segment or negative-segment, i.e., $\mathscr{B}(x,t) = x\bigcup_{k=0}^{m} H_{i_k+1}H_{i_k+2}...H_{i_{k+1}}t$, $(i_0 = 0, i_{m+1} = n)$, s.t. $i_k + 1 < ... < i_{k+1}$ or $i_k + 1 > ... > i_{k+1}$ $(\forall k = \overline{0,m})$. We denote segment $H_{i_k+1}H_{i_k+2}...H_{i_{k+1}}$ as L_k. For each k, L_k is scaled up using a homothetic transformation whose center (denoted by O_k) and scale factor (denoted by ξ_k) are determined as follows. O_k is a random point that stays on the right side of L_k if L_k is a positive-segment, or on the left side of L_k if L_k is a negative-segment;

$$\xi_k = 1 + (\alpha - 1)\frac{a_k + c_k}{a_k + b_k} \tag{3.36}$$

where a_k is the length of segment L_k; $b_k = \left|O_kH_{i_k+1}\right| + \left|O_kH_{i_{k+1}}\right|$; and c_k is defined as follows.

$$c_k = \begin{cases} |sH_1| + \frac{1}{2}\left|H_{i_1}H_{i_1+1}\right|, \text{ if } k = 0 \\ \frac{\left|H_{i_k}H_{i_k+1}\right| + \left|H_{i_{k+1}}H_{i_{k+1}+1}\right|}{2}, \text{ if } k = \overline{1,m-1} \\ |H_nt| + \frac{1}{2}\left|H_{i_m}H_{i_{m+1}}\right|, \text{ if } k = m \end{cases}$$

α is the parameter defined as follows.

- $\alpha = 1+\varepsilon$ if x is a hole aware source node.
- $\alpha = Max\left\{1, \frac{(1+\varepsilon)L_0 - |st'|}{|l_{\mathscr{H}}(t',t)|}\right\}$, otherwise.

where $L_0 = Max\left\{|sv| - \frac{e_{\mathscr{C}}}{2}, |st'| + |l_{\mathscr{H}}(t',t)| - \frac{p_{\mathscr{C}}}{2}, |st'| + l_{\mathscr{G}} - e_{\mathscr{C}}\right\}$, v is the nearest view-limit vertex of s to $\mathscr{C}$.

We denote the image of L_k obtained by the transformation as $L_k^{'}$, i.e., $L_k^{'} = H_{i_k+1}^{'} H_{i_k+2}^{'} ... H_{i_{k+1}}^{'}$, where $H_{i_k+u}^{'}$ is the image of H_{i_k+u} ($\forall u = 1, ..., i_{k+1} - i_k$). The routing path from x to t (denoted as $\mathscr{R}(x,t)$) is the combination of $L_k^{'}$ ($\forall k = \overline{0,m}$), i.e., $L^{'}(s,t) = s \bigcup_{k=0}^{m} H_{i_k+1}^{'} H_{i_k+2}^{'} ... H_{i_{k+1}}^{'} t$. Note that when ξ_k is too large, $L_k^{'}$ may intersect with hole's interior. In such cases, x gradually reduces ξ_k by a factor of $\frac{3}{4}$ until $L_k^{'}$ disjoints of hole's interior.

After determining $\mathscr{R}(x,t)$, x inserts the location of $H_{i_j}^{'} (\forall j = \overline{1,n})$ into the packet header. The packet then is routed towards $H_{i_1}^{'}, H_{i_2}^{'}, ..., H_{i_n}^{'}, t$ sequentially using greedy forwarding algorithm.

3.5.4 Theoretical analysis

Let $\mathscr{R}(s,t)$ denote the routing path determined by our algorithm, and $\mathscr{L}(s,t)$ denote the shortest routing path. In this section, we will show that the routing path generated by our routing protocol is upper bounded by a constant, $1+\varepsilon$, i.e., $\frac{|\mathscr{R}(s,t)|}{|\mathscr{L}(s,t)|} \leq 1+\varepsilon$.

First, we will prove the following statement:

$$|\mathscr{R}(x,t)| \leq |\mathscr{B}(x,t)| \tag{3.37}$$

On the one hand, we have the following:

$$|\mathscr{R}(x,t)| = \left|xH_{i_1}^{'}\right| + \sum_{k=0}^{m}\left|L_k^{'}\right| + \sum_{k=1}^{m}\left|H_{i_k}^{'}H_{i_k+1}^{'}\right| + \left|H_{i_n}^{'}t\right| \tag{3.38}$$

On the other hand, we have:

$$\left|sH_1^{'}\right| \leq |sH_1| + (\xi_0 - 1)\,|O_0H_1| \tag{3.39}$$

$$\left|H_n^{'}t\right| \leq |H_nt| + (\xi_m - 1)\,|O_mH_n| \tag{3.40}$$

$$\left|L_k^{'}\right| = \xi_k\,|L_k| \quad \forall k = \overline{0,m} \tag{3.41}$$

$$\begin{aligned}\left|H_{i_k}^{'}H_{i_{k+1}}^{'}\right| &\leq \left|H_{i_k}^{'}H_{i_k}\right| + \left|H_{i_k}H_{i_{k+1}}\right| + \left|H_{i_{k+1}}H_{i_{k+1}}^{'}\right| \\ \Rightarrow \left|H_{i_k}^{'}H_{i_{k+1}}^{'}\right| &\leq (\xi_{k-1}-1)\left|O_{k-1}H_{i_k}\right| + \left|H_{i_k}H_{i_{k+1}}\right| \\ &\quad + (\xi_k - 1)\left|O_k H_{i_{k+1}}\right| \quad \forall k = \overline{1,m}\end{aligned} \tag{3.42}$$

From (3.38), (3.39), (3.40), (3.41) and (3.42) it can be deduced that,

$$\begin{aligned}|\mathscr{R}(x,t)| &\leq |xH_1| + \sum_{k=1}^{m}\left|H_{i_{j_k}}H_{i_{j_{k+1}}}\right| + |H_{i_n}t| \\ &\quad + \sum_{k=0}^{m}\xi_k|L_k| + (\xi_k - 1)\left(\left|I_k H_{i_{j_{k+1}}}\right| + \left|I_k H_{i_{j_{k+1}}}\right|\right)\end{aligned} \tag{3.43}$$

Moreover, note that:

$$|\mathscr{B}(x,t)| = |xH_1| + \sum_{k=1}^{m}\left|H_{i_k}H_{i_{k+1}}\right| + |H_n t| + \sum_{k=0}^{m}|L_k| \tag{3.44}$$

By substituting the value of ξ_k ($\forall k = \overline{0,m}$) from (3.36) into the right side of (3.43), we can deduce that:

$$|\mathscr{R}(x,t)| \leq \alpha\,|\mathscr{B}(x,t)| \tag{3.45}$$

Case 1: the source node is a hole-aware node.

In this case, $\mathscr{B}(s,t)$ is the same as $\mathscr{L}(s,t)$. Moreover, $\alpha = 1+\varepsilon$ (according to (3.36)). Therefore, from (3.45), it is deduced that $\frac{|\mathscr{R}(s,t)|}{|\mathscr{L}(s,t)|} \leq 1+\varepsilon$. The statement is proved.

Case 2: the source node is a blind node.

First, we will prove the following statement:

$$|\mathscr{L}(s,t)| \geq L_0 \tag{3.46}$$

(i.e., $L_0 = Max\left\{|st'| + |l_{\mathscr{H}}(t',t)| - \frac{p_{\mathscr{C}}}{2}, |st'| - \frac{e_{\mathscr{C}}}{2}, |st'| + l_{\mathscr{G}} - e_{\mathscr{C}}\right\}$).

Let v be the vertex of $\mathscr{C}$ whose distance to s is the smallest. Let u denote the intersection of $\mathscr{L}(s,t)$ with the boundary of $\mathscr{C}$, and V_1V_2 denote the edge of $\mathscr{C}$ containing u. Denote by $\mathscr{L}(s,u), \mathscr{L}(s,V_1), \mathscr{L}(s,V_2)$ the shortest Euclidean paths bypassing $\mathbb{H}$ from s to u, V_1, V_2, respectively. Sine v is the vertex closest to s, the following statement holds:

$$|sv| \leq Min\left\{|sV_1|, |sV_2|\right\} \tag{3.47}$$

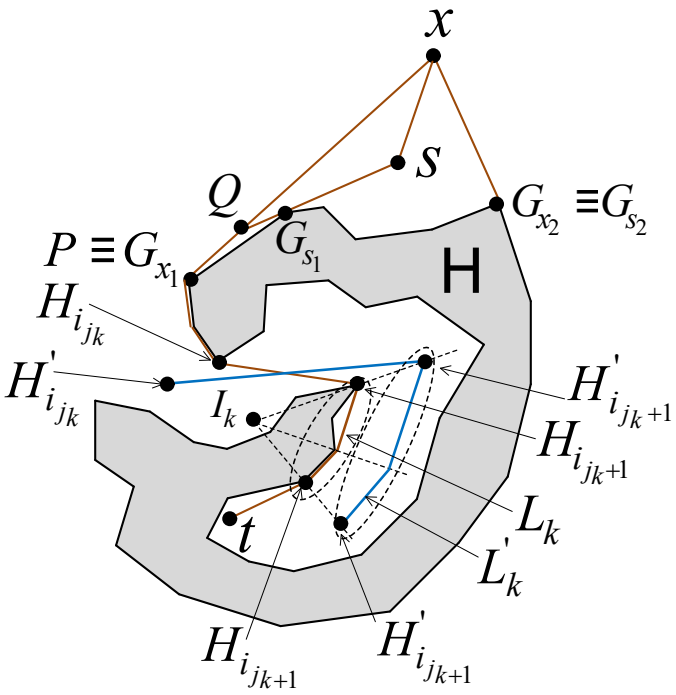


Figure 3.16: Proof of the constant stretch property.

Note that

$$|sV_1| \leq |su| + uV_1 \tag{3.48}$$

$$|sV_2| \leq |su| + uV_2 \tag{3.49}$$

Consequently, from (4.26),(4.30) and (4.31), we deduce that:

$$|sv| \leq |su| + \frac{e_{\mathscr{C}}}{2} \tag{3.50}$$

Obviously, $|\mathscr{L}(s,t)| \geq |su|$, thus (4.47) implies that:

$$|\mathscr{L}(s,t)| \geq |sv| - \frac{e_{\mathscr{C}}}{2} \tag{3.51}$$

Since $l_{\mathscr{H}}(t',t)$ is the shortest path from t' to t, its length must be not greater than $|l_{\mathscr{H}}(t',u)| + |l_{\mathscr{H}}(u,t)|$, where $l_{\mathscr{H}}(u,t)$ is the shortest path from u to t. Note that

$$|\mathscr{L}(s,t)| = |su| + |l_{\mathscr{H}}(u,t)| \tag{3.52}$$

thus,

$$|\mathscr{L}(s,t)| \geq |su| + |l_{\mathscr{H}}(t',t)| - |l_{\mathscr{H}}(t',u)| \tag{3.53}$$

Note that $|su| \geq |st'| - \frac{e_{\mathscr{C}}}{2}$ and $|l_{\mathscr{H}}(t',u)| \leq |\frac{p_{\mathscr{C}}}{2}|$, thus:

$$|\mathscr{L}(s,t)| \geq |st'| + |l_{\mathscr{H}}(t',t)| - \frac{p_{\mathscr{C}} + e_{\mathscr{C}}}{2} \tag{3.54}$$

Since l_G is the shortest Euclidean path from a vertex of $\mathscr{C}$ to t, we have:

$$\begin{aligned} |l_G| &\leq |l_{\mathscr{H}}(u,t)| + \frac{e_{\mathscr{C}}}{2} \\ \Rightarrow |l_{\mathscr{H}}(u,t)| &\geq |l_G| - \frac{e_{\mathscr{C}}}{2} \end{aligned} \tag{3.55}$$

As $|\mathscr{L}(s,t)| = |su| + |l_{\mathscr{H}}(u,t)|$, it is deduced that: $|\mathscr{L}(s,t)| \geq |sv| - \frac{e_{\mathscr{C}}}{2} + |l_{\mathscr{H}}(u,t)|$. From (4.34), we obtain the following:

$$|\mathscr{L}(s,t)| \geq |sv| + |l_G| - e_{\mathscr{C}} \tag{3.56}$$

Consequently, from (3.51), (3.54) and (3.56), we deduce that $|\mathscr{L}(s,t)| \geq L_0$ and Statement (3.46) is proved.

Now we are going to prove the theorem.

$\mathscr{R}(s,t)$ is comprised of two parts: the first path is from s to t', and the second path is from t' to t. Note that the second part is the image of $l_{\mathscr{H}}(t',t')$ through the homothetic transformation with the scale factor of α. Therefore, the length of the second part does not exceed α times that of $|l_{\mathscr{H}}(t',t')|$. Consequently, we have:

$$|\mathscr{R}(s,t)| \leq |st'| + \alpha |l_{\mathscr{H}}(t',t')| \tag{3.57}$$

According to the definition of α, i.e., $\alpha = Max\left\{1, \frac{(1+\varepsilon)L_0 - |st'|}{|l_{\mathscr{H}}(t',t)|}\right\}$, thus in the following we will show that:

$$|st'| + |l_{\mathscr{H}}(t',t')| \leq (1+\varepsilon)|\mathscr{L}(s,t)| \tag{3.58}$$

$$|st'| + \frac{(1+\varepsilon)L_0 - |st'|}{|l_{\mathscr{H}}(t',t)|}|l_{\mathscr{H}}(t',t')| \leq (1+\varepsilon)|\mathscr{L}(s,t)| \tag{3.59}$$

As $|\mathscr{L}(s,t)| \geq L_0$ (see (3.46)), (3.59) is trivial.

According to (3.54), we have:

$$|st'| + |l_{\mathscr{H}}(t',t)| \leq |\mathscr{L}(s,t)| + \frac{p_{\mathscr{C}}}{2} \tag{3.60}$$

As t' is the first hole-aware node (on the routing path from s to t), it must stay outside of the hole's vicinity region (if not, t' must broadcast hole information to its neighbors and thus t''s previous node must be a hole-aware node). Thus, the distance from t' to every point on the boundary of $\mathscr{C}$ must be not smaller than $\frac{p_{\mathscr{C}}}{2\varepsilon} + \frac{e_{\mathscr{C}}}{2}$. Consequently, we have the following:

$$|\mathscr{L}(s,t)| \geq |su| \geq |sV_1| - \frac{e_{\mathscr{C}}}{2} \geq \frac{p_{\mathscr{C}}}{2\varepsilon} \tag{3.61}$$

From (3.60) and (3.61), we obtain $|st'| + |l_{\mathscr{H}}(t',t)| \leq (1+\varepsilon)|\mathscr{L}(s,t)|$, thus (3.58) is proved ■

3.5.5 Section summary

In this section, we addressed the routing problem in the hole's vicinity. Specifically, we considered a network topology which consists of only one hole, the sources and the destinations of packets may stay inside the concave areas of the hole.

We figured out two challenges in dealing with this type of network topology as follows:

- How to efficiently disseminate information of the hole.
- How to construct the routing path by magnifying the base path such that the routing path does not intersect the hole's interior.

To deal with these challenges, we proposed the following strategies:

- In order to reduce the overhead, nodes close to the hole will receive full information of the hole, while nodes far from the hole receive only the information of the core polygon.
- If the source node has hole information, it determines the base path as the shortest path to the destination; otherwise, if the source node has only core polygon information, the base path is the shortest path from the source to a vertex of the core polygon.

- Before applying the homothetic transformations, the base path is divided into segments, each of which is either a positive or a negative segment. These segments are magnified separately by using different homothetic transformations.

The theoretical analysis results proved that the stretch of routing paths determined by our algorithm is always smaller than the predefined threshold $1+\varepsilon$.

3.6 Strategy for bypassing multiple holes

In this section, we are going to tackle a more complicate scenario where multiple holes may exist in the between of the sources and the destinations. We first figure out the challenges and then propose our strategies.

3.6.1 Challenges and strategies

Dealing with multiple holes is much more difficult compared with single hole. The reason is due to the high traffic load in the region in between of the holes (we call this region the *critical region*). The hardest challenge is how to reduce the traffic in this region. In the previous section, we have seen that the base path (which is used to create the routing path) is the shortest path bypassing the core polygon. However, in the context of networks with multiple holes, always use the same base path to construct the routing paths may cause a high traffic load in the critical region. To this end, we attempt to use dynamic base paths that vary for every packet. On the one hand, the base paths are probabilistically selected such that a path that is farther from the critical region is more likely to be chosen. On the other hand, the base path length is controlled to guarantee the required routing path upper bound. Specifically, the base paths are the Euclidean paths that bypass all the core polygons and have the length under a predetermined threshold. For each base path, we assign a so-called priority index which indicates how far is it from the holes. When a source has a packet to send, it probabilistically pickups one base path (with the propability being proportional with the path's priority index) and scales up the base path by using homothetic transformations. The scale factors of the transformations are controlled to guarantee the required upper bound of the stretch and the scale centers of the transformations are chosen randomly to guarantee the diversity of routing path.

Let s and t be a source and a destination, respectively. Suppose that s has information of all core polygons, in the following we describe our algorithm for determining the base path and constructing the routing path.

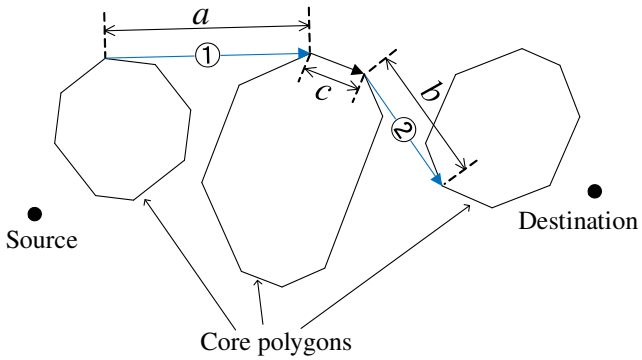
3.6.2 Base path determination

The source node s determines a set of base path candidates which are either the shortest core polygons-bypassing Euclidean path from s to t, or satisfies the following conditions:

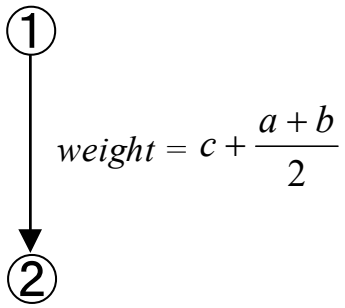
- Its length is less than the shortest path's length by a factor of $(1+\varepsilon) \times \sin\frac{(n-2)\pi}{2n}$.
- The angle between any segment connecting its two consecutive vertices and vector $\overrightarrow{st}$ is an obtuse angle.

Note that in WSNs, the destinations are usually predefined (i.e., the sinks), thus the source nodes may determine all base path candidates and store their information in the local memory in advance.

According to [174], the shortest core polygons-bypassing Euclidean path consists of a tangent line from s to the first core polygon involved, a tangent line from the last core polygon involved to t, some line segments with each bitangent to two consecutive involved core polygons, and some boundary segments along the perimeter of involved core polygons. To simplify the presentation, we consider s and t as two degenerated core polygons and we call both tangent lines (i.e., the lines involving s or t) and bitangent lines (i.e., the lines involving two core polygons) as *tangent segments*. Firstly, s constructs a directed weighted *tangent graph* whose each vertex corresponds to a tangent segment and two vertices are connected if they are tangent segments of the same core polygon. Moreover, the weight of an edge is defined as the sum of the length of the shortest core polygon boundary segment connecting the two corresponding tangent segments and the half of the two corresponding tangent segments' length (Fig. 3.17 shows an example of two vertices and the edge connecting them). Upon determining the tangent graph, s determines the shortest Euclidean path to t using Dijsktra algorithm. s also determines the other base path candidates based on the tangent graph by using breath first search algorithm. Upon determining the base path candidates, s goes to the second step where it will assign the priority index to each base path candidate as follows. Let $\mathfrak{C}$ be the convex hull of all core polygons (i.e., the convex polygon covering



(a) ① and ② are two tangent segments of the same core polygon.



(b) Tangent graph's vertices and edge corresponding to ① and ②.

Figure 3.17: Illustration of vertices and edge of the tangent graph.

all core polygons, whose all vertices are the vertices of the core polygons, see Fig.3.18 for illustration). Intuitively, the nodes staying inside $\mathfrak{C}$ tend to be imposed heavier traffic than the other nodes (i.e., because they are closer to the boundaries of multiple holes). Therefore, to balance traffic, the base path should be chosen such that a path that is further from $\mathfrak{C}$ will have more chances to be chosen. To this end, the *priority index* is a positive number that indicates how far is it from the convex hull. Specifically, for each base path candidate L_b, denote n as its total number of vertices (including the source and destination) and n^* as the number of its vertices staying outside of $\mathfrak{C}$'s interior, then the priority index of L_b is defined as $\frac{n*}{n}$ (e.g., in Fig. 3.3, the priority indexes of the base path candidate ① and ② are 1 and $4/8$, respectively).

3.6.3 Euclidean routing path determination

s probabilistically chooses from the set of all base path candidates to t a base path such that the probability for a base path candidate to be chosen is proportional with its priority index. Let us denote the chosen base path by $sC_1^1...C_1^{n_1}...C_m^1...C_m^{n_m}t$, where $C_j^1...C_j^{n_j}$ is a boundary segment of a core polygon $\mathscr{C}_j$. Then, for each $j \in \{1,...,m\}$, s magnifies segment $\left\{C_j^1 \sim C_j^{n_j}\right\}_{\mathscr{C}_j}$ using a homothetic transformation with the center (i.e., denoted as O_j) chosen randomly inside $\mathscr{C}_j$ and the scale factor (denoted as ξ_j) determined as follows:

$$\xi_j = 1 + \frac{(\alpha\theta - 1)\left(a_j + c_j\right)}{a_j + b_j} \tag{3.62}$$

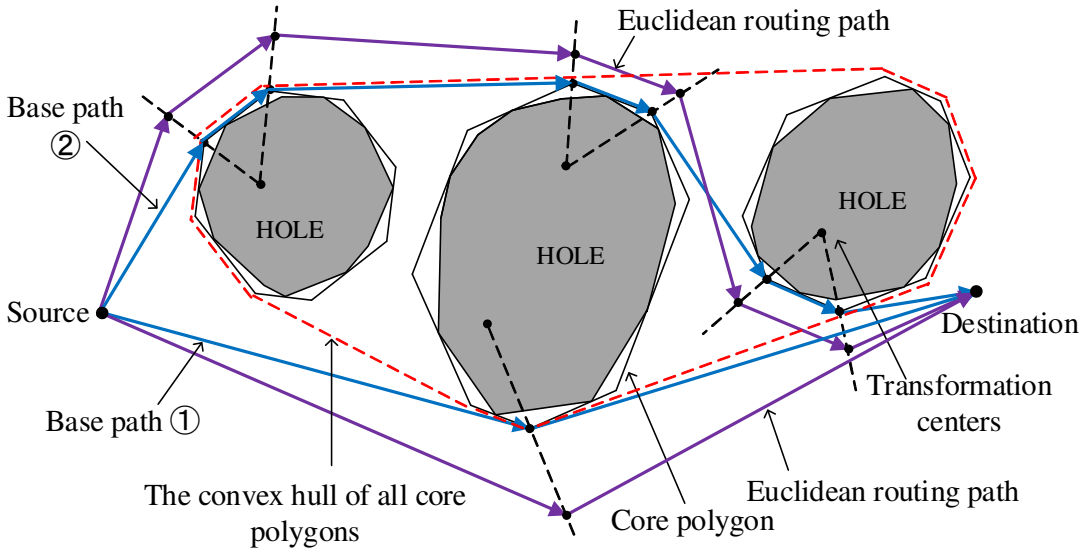


Figure 3.18: An example of routing paths determined by our protocol. The blue lines represent the base paths, the purple lines represent the Euclidean routing paths.

where θ is the ratio between the length of the shortest core polygons-bypassing Euclidean path and the chosen base path, $\alpha = (1+\varepsilon)\sin\frac{(n-2)\pi}{2n}$, $a_j = \left|\left\{C_j^1 \sim C_j^{n_j}\right\}_{\mathscr{C}_j}\right|$, $b_j = \left|O_jC_j^1\right| + \left|O_jC_j^{n_j}\right|$, and,

$$c_j = \begin{cases} \left|sC_j^1\right| + \frac{1}{2}\left|C_j^{n_j}C_{j+1}^1\right|, \text{ if } j = 0 \\ \frac{\left|C_{j-1}^{n_{j-1}}C_j^1\right| + \left|C_j^{n_j}C_{j+1}^1\right|}{2}, \text{ if } j = \overline{1,m-1} \\ \left|C_j^{n_j}t\right| + \frac{1}{2}\left|C_{j-1}^{n_{j-1}}C_j^1\right|, \text{ if } j = m \end{cases}$$

Figure 3.19 illustrates a homothetic transformation of one base path's segment. Denote by $R_1^1 \ldots R_j^{n_j}$ the image of $C_j^1 \ldots C_j^{n_j}$ through such homothetic transformations ($\forall j = \overline{1,m}$), then the Euclidean routing path is the combination of $R_j^1 \ldots R_j^{n_j}$ ($\forall j = \overline{1,m}$). The location of R_j^l ($\forall j = \overline{1,m}, l = \overline{1,n_j}$) are inserted into the packet header as *virtual anchors*. The packet is then forwarded toward these virtual anchors using geographic greedy forwarding, sequentially. Note that because the virtual anchors are not necessarily sensor nodes,

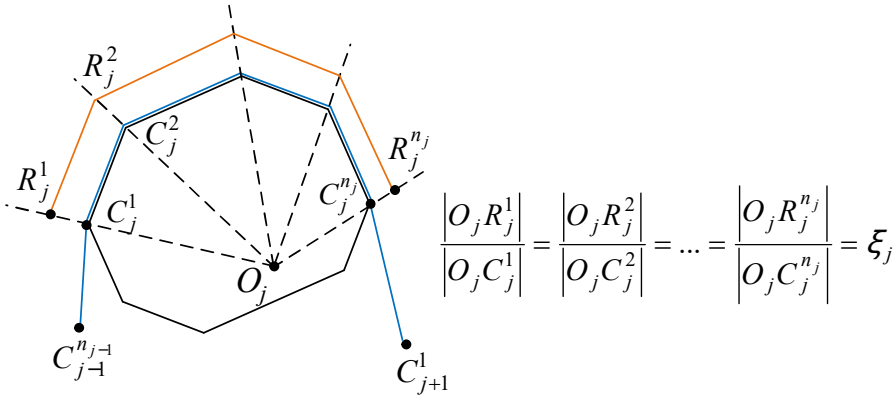


Figure 3.19: Homothetic transformation of a base path's segment.

the packet will be forwarded to the nodes nearest to the virtual anchors.

Although the base path (i.e., $sC_1^1...C_1^{n_1}...C_m^1...C_m^{n_m}t$) disjoints of all core polygons, its magnified image (i.e., $sR_1^1...R_1^{n_1}...R_m^1...R_m^{n_m}t$) may not. In that case, s has to reduce the scale factors of the transformations. Specifically, for every two consecutive virtual anchors $R_{v_1}^{u_1}$ and $R_{v_2}^{u_2}$ (v_1 may coincide with v_2) such that $R_{v_1}^{u_1}R_{v_2}^{u_2}$ intersects core polygons, s gradually reduces the scale factors ξ_{v_1}, ξ_{v_2} by a factor of γ (i.e., γ is a random number in the range of $(0,1)$) until segment $R_{v_1}^{u_1}R_{v_2}^{u_2}$ does not intersect any core polygon.

3.6.4 Theoretical analysis

Let s be a the source node, and t be a destination node, in the following we will prove that the stretch of a routing path from s to t is upper bounded by $1+\varepsilon$. We make an assumption that all core polygons are disjoint (i.e., there are no two core polygons intersecting). We consider four paths connecting the source and the destination: 1) the shortest Euclidean path that bypasses all the holes, 2) the shortest Euclidean path that bypasses all the core polygons, 3) the base path chosen by s, 4) the Euclidean routing path determined by s. We denote these four paths as L_h, L_c, L_b, L_r, respectively. Obviously, $|L_h| \leq |L_c| \leq |L_b| \leq |L_r|$, and the stretch of the routing path can be approximated by $\frac{|L_r|}{|L_h|}$ (according to our theoretical model described in Section 3.1).

Lemma 7
$\frac{|L_c|}{|L_h|} \leq \frac{1}{\sin\frac{(n-2)\pi}{2n}}$.

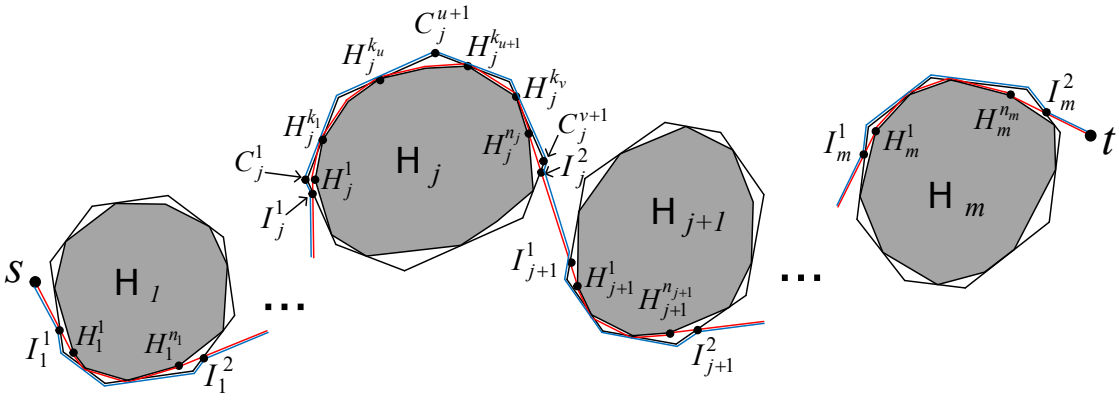


Figure 3.20: Illustration of L_h and $\mathfrak{L}$. The red line is L_h, the blue line is $\mathfrak{L}$.

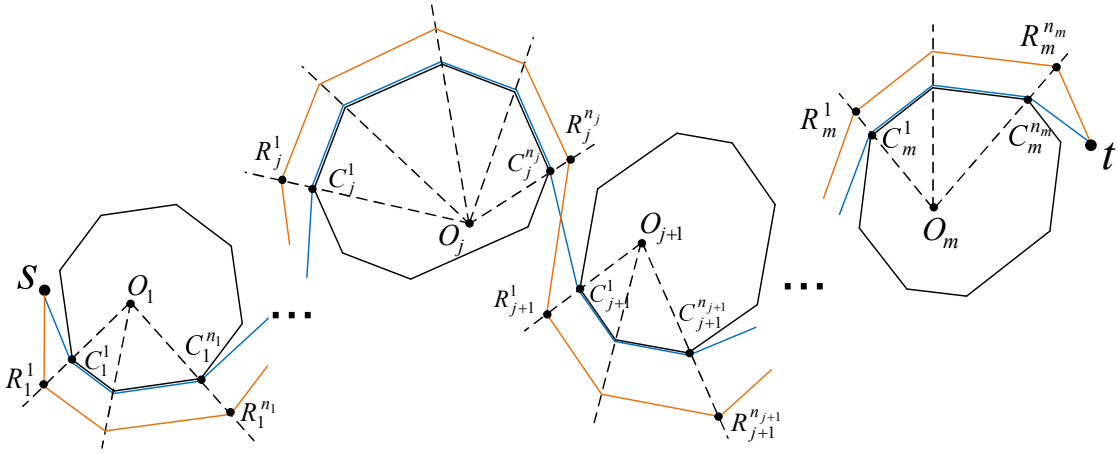


Figure 3.21: Illustration of L_b and L_r. The blue line is L_b, the orange line is L_r.

Proof

We will prove by showing that there exists a path that does not intersect interior of any core polygon and that has a length less than $\frac{1}{\sin\frac{(n-2)\pi}{2n}}|L_h|$ (*).

Suppose $L_h = sH_1^1...H_1^{n_1}...H_m^1...H_m^{n_m}t$, where $H_j^1...H_j^{n_j}$ is a boundary segment of hole $\mathscr{H}_j$ ($\forall j = \overline{1,m}$) (see Fig. 3.20). Note that, for each $j \in \{1,2,...,m-1\}$, segment $H_j^{n_j}H_{j+1}^1$ intersect $\mathscr{C}_j$ at at most two points and one of them is $H_j^{n_j}$. We denote the second intersection point as I_j^2, where $I_j^2 \equiv H_j^{n_j}$ if $H_j^{n_j}H_{j+1}^1$ intersects $\mathscr{C}_j$ at only one point. Similarly, $H_j^{n_j}H_{j+1}^1$ also intersects $\mathscr{C}_{j+1}$ at at most two points, one of them is H_{j+1}^1. We denote the other intersection point as I_{j+1}^1 which may coincides with C_{j+1}^1 if $H_j^{n_j}H_{j+1}^1$ intersects $\mathscr{C}_{j+1}$ at only one point. We also denote the intersection point of

sH_1^1 with $\mathscr{C}_1$ as I_1^1 and the intersection point of $H_m^{n_m}t$ with $\mathscr{C}_m$ as I_m^2. We will prove that, the path $\mathfrak{L} = sI_1^1 \left\{I_1^1 \sim I_1^2\right\}_{\mathscr{C}_1} ...I_j^1 \left\{I_j^1 \sim I_j^2\right\}_{\mathscr{C}_j} I_j^2 ....I_1^{i_m} \left\{I_m^1 \sim I_m^2\right\}_{\mathscr{C}_m} I_m^2 t$ (the direction of $\left\{I_j^1 \sim I_j^2\right\}_{\mathscr{C}_j}$ is the same as that of $\left\{H_j^1 \sim H_j^{n_j}\right\}_{\mathscr{H}_j}$) satisfies condition (*). Note that, $I_j^2 I_{j+1}^1$ is a part of $H_j^{n_j} H_{j+1}^1$ ($\forall j = \overline{1,m}$), thus they do not intersect any core polygon. $\mathfrak{L}$ is comprised of core polygons' boundary segments and the set of $I_j^2 I_{j+1}^1$ ($j = \overline{1,m}$), thus $\mathfrak{L}$ does not intersect any core polygon. Since all core polygons are disjoint, $\mathfrak{L}$ stays outside of all holes.

Now we will prove that $|\mathfrak{L}| \leq \frac{1}{sin\frac{(n-2)\pi}{2n}} |L_h|$. Note that, the length of $\mathfrak{L}$ can be decomposed as:

$$|\mathfrak{L}| = \left|sI_1^1\right| + \sum_{j=1}^{m} \left|\left\{I_j^1 \sim I_j^2\right\}_{\mathscr{C}_j}\right| + \sum_{j=1}^{m-1} \left|I_j^2 I_{j+1}^1\right| + \left|I_m^2 t\right| \tag{3.63}$$

The length of L_h can be decomposed as:

$$|L_h| = \left|sI_1^1\right| + \sum_{j=1}^{m} \left|I_j^1 \left\{H_j^1 \sim H_j^{n_j}\right\}_{\mathscr{H}_j} I_j^2\right| + \sum_{j=1}^{m-1} \left|I_j^2 I_{j+1}^1\right| + \left|I_m^2 t\right| \tag{3.64}$$

From (4.6) and (4.7), it is obviously that we can prove the lemma by showing that:

$$\left|\left\{I_j^1 \sim I_j^2\right\}_{\mathscr{C}_j}\right| \leq \frac{1}{sin\frac{(n-2)\pi}{2n}} \left|I_j^1 \left\{H_j^1 \sim H_j^{n_j}\right\}_{\mathscr{H}_j} I_j^2\right| \quad \forall j \in \{1,...,m\} \tag{3.65}$$

We denote $H_j^{k_1}, ..., H_j^{k_v}$ as the $\mathscr{H}_j$'s vertices that belong to $\left\{H_1^1 \sim H_j^{n_j}\right\}_{\mathscr{H}_j}$ and stay on the boundary of $\mathscr{C}_j$, and denote $C_j^u C_j^{u+1}$ as the edge of $\mathscr{C}_j$ that contains $H_j^{k_u}$ ($\forall u = \overline{1,v}$). Using triangular cosin formula, we have:

$$\begin{aligned} \left|H_j^{k_u} C_j^{u+1}\right| + \left|C_j^{u+1} H_j^{k_{u+1}}\right| &\leq \frac{1}{\sin\frac{\angle H_j^{k_u} C_j^{u+1} H_j^{k_{u+1}}}{2}} \left|H_j^{k_u} H_j^{k_{u+1}}\right| \\ &\leq \frac{1}{\sin\frac{(n-2)\pi}{2n}} \left|\left\{H_j^{k_u} \sim H_j^{k_{u+1}}\right\}_{\mathscr{H}_j}\right| \quad \forall j = \overline{1,m} \end{aligned} \tag{3.66}$$

Similarly, we have:

$$\left|I_j^1 C_j^1\right| + \left|C_j^1 H_j^{k_1}\right| \leq \frac{1}{\sin\frac{(n-2)\pi}{2n}} \left(\left|I_j^1 H_j^1\right| + \left|\left\{H_j^1 \sim H_j^{k_1}\right\}_{\mathscr{H}_j}\right|\right) \tag{3.67}$$

$$\left|I_j^2 C_j^{v+1}\right| + \left|C_j^{v+1} H_j^{k_v}\right| \leq \frac{1}{\sin\frac{(n-2)\pi}{2n}} \left(\left|I_j^2 H_j^{n_j}\right| + \left|\left\{H_j^{k_v} \sim H_j^{n_j}\right\}_{\mathscr{H}_j}\right|\right) \tag{3.68}$$

By summing up (4.9), (4.10) and (4.11) we obtain (4.8) ■

Lemma 8
$\frac{|L_r|}{|L_b|} \leq \alpha\theta.$

Proof
Suppose $L_b = sC_1^1 ... C_1^{n_1} ... C_m^1 ... C_m^{n_m} t$, where $C_j^1 ... C_j^{n_j}$ is a boundary segment of a core polygon $\mathscr{C}_j$ ($\forall j = \overline{1,m}$) (see Fig. 4.6). Suppose $L_r = sR_1^1 ... R_1^{n_1} ... R_m^1 ... R_m^{n_m} t$, where R_j^l is the image of C_j^l ($\forall j = \overline{1,m}, l = \overline{1,n_j}$) through a homothetic transformation with the scale factor ξ_j calculated by formula (4.12). The length of L_b and L_r can be decomposed as follows.

$$|L_b| = \left|SC_1^1\right| + \sum_{j=1}^{m} \left|\left\{C_j^1 \sim C_j^{n_j}\right\}_{\mathscr{C}_j}\right| + \sum_{j=1}^{m-1} \left|C_j^{n_j} C_{j+1}^1\right| + \left|C_m^{n_m} t\right| \tag{3.69}$$

$$|L_r| = \left|SR_1^1\right| + \sum_{j=1}^{m} \left|\left\{R_j^1 \sim R_j^{n_j}\right\}\right| + \sum_{j=1}^{m-1} \left|R_j^{n_j} R_{j+1}^1\right| + \left|R_m^{n_m} t\right| \tag{3.70}$$

Since $R_j^1 \sim R_j^{n_j}$ is the image of $\left\{C_j^1 \sim C_j^{n_j}\right\}_{\mathscr{C}_j}$ ($\forall j = \overline{1,m}$) through a homothetic transformation with the scale factor ξ_j, we have:

$$\left|\left\{R_j^1 \sim R_j^{n_j}\right\}\right| = \xi_i \left|\left\{C_j^1 \sim C_j^{n_j}\right\}_{\mathscr{C}_j}\right| \tag{3.71}$$

Using triangular inequality, we have:

$$\left|sR_1^1\right| \leq \left|sC_1^1\right| + \left|C_1^1 R_1^1\right| = \left|sC_1^1\right| + (\xi_1 - 1)\left|O_1 C_1^1\right| \tag{3.72}$$

$$\left|R_m^{n_m} t\right| \leq \left|C_m^{n_m} t\right| + \left|C_m^{n_m} R_m^{n_m}\right| = \left|C_m^{n_m} t\right| + (\xi_m - 1)\left|O_m C_m^{n_m}\right| \tag{3.73}$$

$$\begin{aligned}\left|R_j^{n_j} R_{j+1}^1\right| &\leq \left|R_j^{n_j} C_j^{n_j}\right| + \left|C_j^{n_j} C_{j+1}^1\right| + \left|C_{j+1}^1 R_{j+1}^1\right| \\ &= (\xi_j - 1)\left|O_j C_j^{n_j}\right| + \left|C_j^{n_j} C_{j+1}^1\right| + (\xi_{j+1} - 1)\left|O_{j+1} C_{j+1}^1\right|\end{aligned} \tag{3.74}$$

Summing up (4.18), (4.19), (4.20) and (4.21), we have:

$$|L_r| \leq \left|sC_1^1\right| + |C_m^{n_m}t| + \sum_{j=1}^{m-1}\left|C_j^{n_j}C_{j+1}^1\right| + \sum_{j=1}^{m}\left\{\xi_j\left|\left\{C_j^1 \sim C_j^{n_j}\right\}_{\mathscr{C}_j}\right| + (\xi_j - 1)\left(\left|O_jC_j^1\right| + \left|O_jC_j^{n_j}\right|\right)\right\} \quad (3.75)$$

The right side of (4.22) equals to:

$$\sum_{j=1}^{m} \xi_j a_j + \left(\xi_j - 1\right) b_j + c_j$$

By substituting value of ξ_j obtained from (4.12), we have $\xi_j a_j + \left(\xi_j - 1\right) b_j + c_j = \alpha\theta\left(a_j + c_j\right)$. Consequently, $|L_r| \leq \alpha\theta \sum_{j=1}^{m} a_j + c_j = \alpha\theta\,|L_b|$ ■

The stretch of the routing path is $\frac{|L_r|}{|L_h|}$, which is equivalent with $\frac{|L_r|}{|L_b|} \times \frac{|L_b|}{|L_c|} \times \frac{|L_c|}{|L_h|}$. According to Lemma 7 and 8, it is deduced that $\frac{|L_r|}{|L_h|} \leq \alpha\theta \times \frac{1}{\theta} \times \frac{1}{\sin\frac{(n-2)\pi}{2n}} = (1+\varepsilon)$ ■

3.6.5 Section summary

In this section, we addressed the routing problem in wireless sensor networks with multiple holes. We observed that the most challenge in dealing with multiple holes is solving the traffic concentration in the region between the core polygons (i.e., critical region). To tackle this challenge, we proposed to use dynamic base paths. Specifically, for each source-destination pair, instead of using only one base path for all packets, the source node determines a set of base path candidates. When the source node has a packet to send, it probabilistically select a base path from the candidates such that the candidate which is farther from the critical region has a higher probability to be chosen as the base path. The base path then is magnified to obtained the Euclidean routing path whose vertices will be inserted into packets as virtual anchors. Finally, the packet is greedily forwarded towards the virtual anchors gradually until reaching the destination.

3.7 BSMH: A Load Balanced and Constant Stretch Protocol for Bypassing Multiple Holes

Based on the results obtained from the previous sections, now we describe our routing protocol for the most general network topology where multiple holes may exist as well as the sources and the destinations of the packets may stay inside the holes' core polygons.

3.7.1 Phase 1: Hole and core polygon determination

Every node uses the protocol described in Section 3.3 to detect the holes and construct the core polygons. Specifically, all the nodes use TENT rule[18] to identify whether they are the stuck nodes (i.e., the nodes that may incur the local minimum phenomenon). Then, every stuck node initiate a HBA packet and send this packet around its hole by using RIGHT HAND rule[18]. The HBA packets collect the coordinate of all the nodes it has traversed. Accordingly, when a HBA packet comes back to its initiator, it has location of all nodes on the hole boundary. Note that, for the same hole, there may be multiple HBA created, thus in order to reduce the control overhead, we use an election algorithm to drop redundant HBA packet. The pseudo codes of the TENT rule is described in Algorithm 1. The format of the HBA packet is presented in Tab.3.2 and the pseudo code for forwarding the HBA packet is described in Algorithm 2.

3.7.2 Phase 2: Hole information dissemination

We use the strategy proposed in Section 3.5.2 to disseminate information of the holes and the core polygons. By which, the nodes staying inside a hole's vicinity region (see Definition 11) receive the hole information (i.e., the coordinates of all vertices of the hole), and the other nodes far from the hole receive only the information of the hole's core polygon (i.e., the coordinates of all vertices of the core polygon). Due to the characteristic of networks with multiple holes, we redefine the vicinity region as follows.

Definition (Vicinity region)
Let $\mathscr{H}$ be a hole and $\mathscr{C}$ be its core polygon. Denote by $p_{\mathscr{C}}$ the perimeter of $\mathscr{C}$, and $e_{\mathscr{C}}$ the length of the longest edge of $\mathscr{C}$. Then, the vicinity region of $\mathscr{H}$ consists of all nodes

ALGORITHM 1: TENT rule [18]

Input *p: current node.*
nb: 1-hop neighbors of p. R: communication range of the sensors.
Output stuck_angle_list*: list of stuck angles of p. //* each stuck angle is an angle of a hole consisting *p.*
nb.sort_in_counterclockwise_order ;
for *each pair of adjacent nodes u, v* **do**
  $l_1 \leftarrow$ perpendicular bisector of up ;
  $l_2 \leftarrow$ perpendicular bisector of vp ;
  $O \leftarrow$ intersection of l_1 and l_2;
  if $|Op| \leq R$ **then**
    stuck_angle_list.Add(upv)
  end
end
return *stuck_angle_list*

Table 3.2: HBA message

Field	*Size (bytes)*	*Content*
Type	1	BSMH_HBA
Previous Node	16	The location of the node sent the HBA message
Core nodes	$n \times 16$	An array consists the coordinates of n nodes on the hole boundary

ALGORITHM 2: HBA forwarding algorithm

Input *p: packet, c: the current node (i.e., node receiving the HBA packet), nb: 1-hop neighbors of c.* $i_{ID} \leftarrow$ the ID of p's initiator ;
$l_{ID} \leftarrow 0$;
if $l_{ID} > i_{ID}$ **then**
  c.drop(HBA);
end
else
  $l_{ID} \leftarrow i_{ID}$;
  next_hop $\leftarrow$ the first item of *nb* hit when sweeping $t_1 p$ by the counterclockwise direction;
  p.Add(c's coordinates);
  c.forward(p);
end

N whose shortest distance to $\mathscr{C}$, $d_{\mathscr{C}}(N)$, satisfies the following condition:

$$\frac{e_{\mathscr{C}}}{2\sin\frac{(n-2)\pi}{2n}} + \frac{p_{\mathscr{C}}}{2} \geq \left(1+\varepsilon-\frac{1}{\sin\frac{(n-2)\pi}{2n}}\right) d_{\mathscr{C}}(N) \tag{3.76}$$

Since the information of every hole is disseminated by the same algorithm, in the following, we describe the algorithm regarding one hole.

After the first phase, the initiator of the HBA packet has coordinates of all node on the hole boundary. This initiator creates a packet named HCI packet that convey information of all hole boundary nodes, and broadcast the HCI packets to the neighbors. When a node receives a HCI packet, it performs the following procedure:

- If the HCI packet contains information of a hole, then the node checks whether it already stored information of the hole in the local memory. If "yes", it simply drops the HCI packet. Otherwise, it stores information of the hole into the local memory. Moreover, if the node has already stored information of the hole's core polygon in the memory, then it removes the core polygon's information because now it already has information of the hole. The node then checks whether it stay inside the hole's vicinity. If "yes", it broadcast HCI packet. If "no", if replace the information of the hole by the information of the core polygon before broadcast the HCI packet.

- If the HCI packet contains information of a core polygon, then the node checks whether it already stored information of the same core polygon or a hole whose core polygon is the core polygon in the HCI packet. If "yes", the node simply drops the packet. Otherwise, the node stores information of the core polygon into the local memory and broadcast the HCI packet.

The format of the HCI packet is described in Table 3.3 and the pseudo code of our dissemination algorithm is shown in Algorithm 3.

3.7.3 Phase 3: Data forwarding

After the second phase, every node has hole information of all the holes whose vicinity regions contain the node and has core polygon information of all the other holes. In this section, we will show how a source node s can exploit this information to make a routing

Table 3.3: HCI message

Field	*Size (bytes)*	*Content*
Type	1	BSMH_HCI
info_type	1	indicating whether the information in the packet is a hole or a core polygon
vertices_coordinates	$n \times 16$	An array consists the coordinates of n vertices of the hole or the core polygon

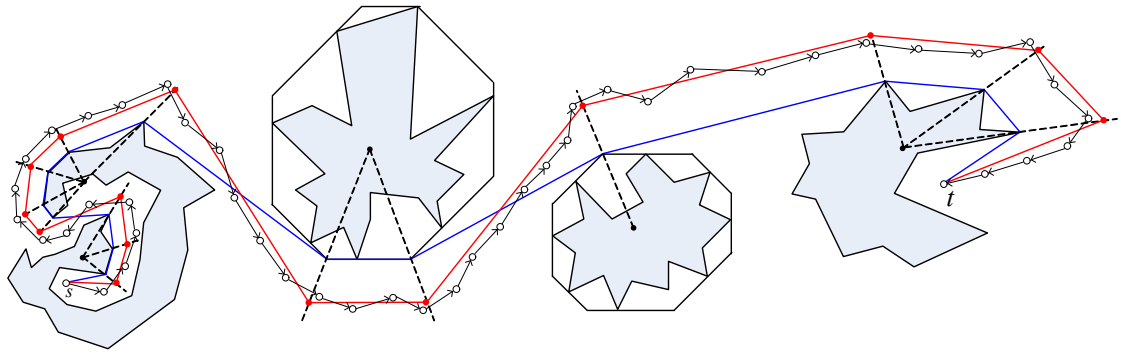
decision to a destination t. To simplify the presentation, we use the following notations. Let N and M be two arbitrary nodes ($N \neq M$), then $\mathscr{C}(N)$ denotes the core polygon containing N, and $\mathscr{H}(N)$ denotes the hole whose core polygon is $\mathscr{C}(N)$ ($\mathscr{H}(N) = \mathscr{C}(N) = \varnothing$ if N does not stay inside any core polygon). Let $\mathbb{C}(\overline{N})$ denote the set of all core polygons excepting $\mathscr{C}(N)$, and $\mathbb{C}(\overline{N},\overline{M})$ denote the set of all core polygons excepting $\mathscr{C}(N)$ and $\mathscr{C}(M)$. Similarly, let us denote $\mathbb{H}(\overline{N})$ the set of all holes excepting $\mathscr{H}(N)$, and $\mathbb{H}(\overline{N},\overline{M})$ the set of all holes excepting $\mathscr{H}(N)$ and $\mathscr{H}(M)$. N is said a M-aware node, if either $\mathscr{H}(M) = \varnothing$ or N has the hole information of $\mathscr{H}(M)$.

Our data forwarding protocol consists of three steps: determining the base path, magnifying the base path to obtain the Euclidean routing path, and forwarding the packet along the Euclidean path. Before going to the detail of each step, we sketch the overview. If s is a t-aware node, then s determines the base path as an Euclidean path from s to t which bypasses $\mathscr{H}(s)$, $\mathscr{H}(t)$and $\mathbb{C}(\overline{s},\overline{t})$. This base path then is magnified using a transformation to obtain the Euclidean routing path. Finally, the packet is forwarded along the Euclidean routing path until reaching t. Otherwise, if s is a t-blind node, then it doesn't have full information of $\mathscr{H}(t)$, thus it can't determine a Euclidean path to t which bypasses all the holes. Therefore, s determines the base path as an Euclidean path from s to a vertex of $\mathscr{C}(t)$ which bypasses $\mathscr{H}(s)$ and $\mathbb{C}(\overline{s})$. This base path then is magnified through a transformation to obtain the Euclidean routing path. The packet then is forwarded along the Euclidean routing path until reaching a node (let us call this node as t') that has the hole information of $\mathscr{H}(t)$. Note that since the last vertex of the Euclidean routing path is a vertex of $\mathscr{H}(t)$, such a node t' always exists. Having the hole information of $\mathscr{H}(t)$, t' determines a so-called *sub-base path* which is the shortest Euclidean path from t' to t, which bypasses $\mathscr{H}(t')$, $\mathscr{H}(t)$ and $\mathbb{C}(\overline{t},\overline{t'})$. This sub-base path then is magnified by a transformation to obtain a so-called *sub-Euclidean routing path*. The packet then is greedily forwarded along the sub-Euclidean routing path until

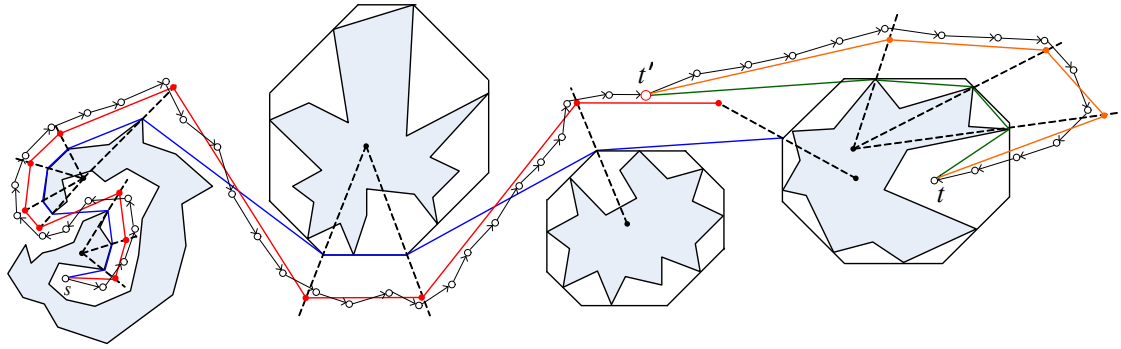
ALGORITHM 3: HCI dissemination

```
Input p: packet, c: current node local_core_list ← all core polygons stored c's local memory;
local_hole ← the hole stored in c's local memory;
new_core ← the core polygon stored in the HCI packet;
if c is the initiator of p then p.contains_a_hole = true;
if p.contains_a_hole then
    new_hole ← the hole stored in the HCI packet if local_hole != null then c.drop(HCI) ;
    else
        new_core ← create the core polygon from new_hole if local_core_list != null then
            foreach core ∈ local_core_list do
                if core == new_core then Remove(core) ;
            end
        end
        if c stay outside of the vicinity of new_hole then
            p.remove(new_hole); p.add(new_core); p.contains_a_hole = false;
        end
        c.add(new_hole); c.broadcast(HCI);
    end
end
else if !p.contains_a_hole then
    new_core ← the core polygon stored in the HCI packet;
    already_received_HCI = false;
    if local_hole != null then
        local_core ← create the core polygon from local_core ;
        if local_core == new_core then already_received_HCI = true;
    end
    foreach core ∈ local_core_list do
        if core == new_core then
            already_received_HCI = true;
            break;
        end
    end
    if already_received_HCI then c.drop(HCI) ;
    else
        c.add(new_core); c.broadcast(HCI);
    end
end
```

reaching t. See Fig.3.22 for the illustration.



(a) Case 1: s is a t-aware node.



(b) Case 2: s is a t-blind node.

Figure 3.22: Data forwarding. The blue/green lines are the base path/sub-base path, the red/orange paths are the Euclidean routing path/sub-Euclidean routing path.

3.7.3.1 Determining the base path candidate

Following the strategy proposed in Section 3.6.2, we use dynamic base path that is probabilistically chosen from a set of base path candidates. Suppose N and M be two nodes in the network, we denote $\mathscr{X}(N) = \mathbb{C}(\overline{N},\overline{M}) \cup \mathscr{H}(M) \cup \mathscr{H}(N)$ if N is a M-aware node, and $\mathscr{X}(N) = \mathbb{C}(\overline{N}) \cup \mathscr{H}(N)$, otherwise.

Definition (Γ-bounded path)
An Γ-bounded path from N to M is an Euclidean paths bypassing all elements of $\mathscr{X}(N)$ and satisfying the following conditions:

- *Its length does not exceed Γ.*
- *The angle between any segment connecting its two consecutive vertices and vector $\overrightarrow{NM}$ is an obtuse angle.*

M and N are said to be visible to each other if MN does not intersect the interior of $\mathbb{C}(\overline{s},\overline{t})$, $\mathscr{H}(s)$ and $\mathscr{H}(t)$.

Below is the algorithm to determine the base path.

First, s constructs the visibility graph $G=(V,E)$ as follows. If s is a t-aware node, then V consists of s, t and all the vertices of $\mathbb{C}(\overline{s},\overline{t})$, $\mathscr{H}(t)$ and $\mathscr{H}(s)$. Otherwise, if s is a t-blind node, then V consists of s, and all the vertices of $\mathbb{C}(\overline{s})$ and $\mathscr{H}(s)$. Two vertices of V is connected if they are visible to each other. s applies the breath first search algorithm on graph G to determine the base path candidates as follows.

- If s is a t-aware node, the base path candidates are all paths from s to t that are either the shortest path or $\sin\frac{(n-2)\pi}{2n}(1+\varepsilon)|\mathscr{L}(s,t)|$-bounded paths, where $\mathscr{L}(s,t)$ is the shortest path from s to t.

- If s is a t-blind node, the base path candidates are all paths from s to a vertices on $\mathscr{C}(t)$ that are $(\Gamma_1\times|\mathscr{L}(s,\mathscr{C}(t))|)$-bounded paths,
  where $\Gamma_1=Max\left\{1,\frac{\sin\frac{(n-2)\pi}{2n}\left((1+\varepsilon)d_{\mathscr{C}(t)}(s)-\frac{p_{\mathscr{C}(t)}}{2}\right)}{d_{\mathscr{C}(t)}(s)+\frac{e_{\mathscr{C}(t)}}{2}}\right\}$ and $\mathscr{L}(s,\mathscr{C}(t))$ denotes the shortest length of the shortest paths from s to a vertex of $\mathscr{C}(t)$.

Second, all base path candidates are assigned priority indexes as follows. Let $\mathbb{C}$ be the convex hull of all the core polygons. For a base path L_b, denote n as its total number of vertices (including the source and destination) and n^* as the number of its vertices staying outside of $\mathbb{C}$'s interior, then the priority index of L_b is defined as $\frac{n*}{n}$.

Finally, s probabilistically choose from the base path candidate set a base path such that the probability for a base path candidate to be chosen is proportional with its priority index.

Algorithm 4 describes the pseudo code for determining the base path candidate.

3.7.3.2 Determining the Euclidean routing path

Let N and M be two arbitrary points on the plane, and $\mathbb{P}$ be a set of polygons on the plane. Let l be a $\mathbb{P}$-bypassing broken line from N to M, that is comprised of boundary segments of $\mathbb{P}$'s polygons. In the following, we define a transformation $F(l,\mathbb{P},\alpha)$ that transforms l to a new $\mathbb{P}$-bypassing broken line l' whose length does not exceed α times that of l. α is call the scale factor of the transformation.

ALGORITHM 4: Base path candidate determination algorithm.

Input *s: source, d: destination,* ε*: the stretch factor*
Output *P: base path candidate set P, shortest_len* $\leftarrow$ Algorithm 5 (s, d) ;
if d_core == *null* || d_hole != *null* **then** $\Gamma_1 = \frac{(n-2)\pi}{2n}(1+\varepsilon)$;
else
  $p_{\mathscr{C}(t)} \leftarrow$ the perimeter of *d_core*; $e_{\mathscr{C}(t)} \leftarrow$ the length of the longest edge of *d_core*;
  $d_{\mathscr{C}(t)}(s) \leftarrow$ the shortest distance from s to an edge of *d_core*;
  $\Gamma_1 = Max\left\{1, \frac{\sin\frac{(n-2)\pi}{2n}\left((1+\varepsilon)d_{\mathscr{C}(t)}(s) - \frac{p_{\mathscr{C}(t)}}{2}\right)}{d_{\mathscr{C}(t)}(s) + \frac{e_{\mathscr{C}(t)}}{2}}\right\}$
end
for $p \in P$ **do**
  if $|p| > \Gamma_1 \times$ *shortest_len* **then** P.remove(p) ;
end
convex_hull $\leftarrow$ the convex hull of *all_core_polygon_list*;
$V_c \leftarrow$ vertices of *convex_hull*; $n \leftarrow$ the number of the vertices of V_c ;
for $p \in P$ **do**
  $n_1 \leftarrow$ the number of p's vertices belonging to V_c; $p.priority \leftarrow \frac{n_1}{n}$
end
return P

First, l is decomposed into segments, each of which is a boundary segment of a polygon of $\mathbb{P}$. Then, for every segment that is a boundary segment of a concave polygon, it is further divided into sub-segments, each of which is either positive-segment or negative-segment. Eventually, l is divided into segments/or sub-segments that are either boundary segments of convex polygons or positive/negative-segments of concave polygons. Let us denote the segments/sub-segments composing l as $L_1, ..., L_k$. Then, each L_i $(i = 1, ...k)$ is scaled up using a specific homothetic transformation denoted as F_i, whose scale center (denoted as O_i), and scale factor (denoted as ξ_i) are defined as follows. Suppose L_i consists of n_i vertices (n_i is a positive integer), and let us denote L_i's vertices as $L_i^1, ..., L_i^{n_i}$. If L_i is a boundary segment of a convex polygon, then O_i is a random point staying inside the polygon whose boundary containing L_i. Otherwise, O_i is a random point staying on the right side of L_i if L_i is a positive-segment, and staying on the left side of L_i if L_i is a negative-segment.

$$\xi_i = 1 + (\alpha - 1)\frac{z_i + x_i}{x_i + y_i} \tag{3.77}$$

ALGORITHM 5: Non-intersection path determination algorithm.

Input *s: source, d: destination*
Output *P: list of paths that do not intersect core polygons and holes stored in s; shortest_len: the shortest length of the base paths* s_core_list ← all core polygons stored at *s*;
s_hole_list ← the holes stored at *s*;
all_core_polygon_list ← *s_core_list*;
for hole ∈ s_hole_list **do**
  core ← create core polygon of *hole*; *all_core_polygon_list*.add(*core*);
end
for core ∈ all_core_polygon_list **do**
  if *s* ∈ core **then** *s_core* ← *core* ;
  if *d* ∈ core **then** *d_core*← *core* ;
end
if s_core *!=* *null* || d_core *!=* *null* **then**
  for hole ∈ all_hole_list **do**
    if s_core *!=* *null* *&* hole ∈ s_core **then** *s_hole* = hole ;
    if d_core *!=* *null* *&* hole ∈ d_core **then** *d_hole* = hole ;
  end
end
for core ∈ all_core_polygon_list *&* core *!=* s_core *&* core *!=* d_core **do**
  if *sd intersect* core **then** *obstacles*.add(*core*) ;
end
if s_core *!=* *null* **then** *obstacles*.add(*s_hole*) ;
if d_hole *!=* *null* **then** *obstacles*.add(*d_hole*) ;
else if d_core *!=* *null* **then** *obstacles*.add(*d_core*) ;
V ← all the vertices of *obstacles*;
V.add(*s*);
if d_hole *!=* *null* || d_core == *null* **then** *V*.add(*d*) ;
if d_hole == *null* *&* d_core *!=* *null* **then** *dest_list* ← vertices of *d_core* ;
else *dest_list* ← *d* ;
foreach $u, v \in V$ **do**
  if *uv does not intersect obstacles* **then** E.add((u, v)) ;
end
P ← all paths from *s* to each item of *dest_list*;
shortest_len ← the shortest length of *P* ;
return *P*, *shortest_len*

where $x_i = |L_i|$, $y_i = \left|O_iL_i^1\right| + \left|O_iL_i^{n_i}\right|$,

$$z_i = \begin{cases} \left|sL_i^1\right| + \frac{1}{2}\left|L_i^{n_i}L_{i+1}^1\right|, \text{ if } i = 1 & (3.78) \\ \frac{\left|L_i^1L_{i-1}^{n_{i-1}}\right| + \left|L_i^{n_i}L_{i+1}^1\right|}{2}, \text{ if } i = \overline{2,k-1} & (3.79) \\ \left|L_i^{n_i}t\right| + \frac{1}{2}\left|L_{i-1}^{n_{i-1}}L_i^1\right|, \text{ if } i = k & (3.80) \end{cases}$$

Let us denote by $L_i^{'}$ the image of L_i through the homothetic transformation F_i, then $F(l,\mathbb{P},\alpha) = \bigcup_{i=1}^{k} L_i^{'}$. Note that, when ξ_i is too large, $L_i^{'}$ (i.e., the image of L_i obtained by using F_i) may intersect $\mathbb{P}$. In such cases, ξ_i is gradually decreased by a factor of $\frac{3}{4}$ until $L_i^{'}$ bypasses $\mathbb{P}$. Algorithm 7 describes the pseudo code of $F(l,\mathbb{P},\alpha)$.

Let us denote the base path from s to t as $\mathscr{B}(s)$, Now by using the transformation F defined above, s constructs the Euclidean routing path as the image of $\mathscr{B}(s)$ through $F(\mathscr{B}(s),\mathbb{A}(s),\alpha_1)$, where $\mathbb{A}(s)$ and α_1 are defined as follows.

$$\mathbb{A}(s) = \begin{cases} \{\mathscr{H}(s),\mathscr{H}(t)\} \cup \mathbb{C}(\overline{s},\overline{t}), \text{ if } s \text{ is a } t\text{-aware node} & (3.81) \\ \{\mathscr{H}(s)\} \cup \mathbb{C}(\overline{s}), \text{ otherwise} & (3.82) \end{cases}$$

$$\alpha_1 = \begin{cases} \frac{1+\varepsilon}{\theta_1}\sin\frac{(n-2)\pi}{2n}, \text{ if } s \text{ is a } t\text{-aware node} & (3.83) \\ Max\left\{1, \frac{\sin\frac{(n-2)\pi}{2n}\left\{\frac{(1+\varepsilon)L_0}{\theta_2} - \frac{p_{\mathscr{C}(t)}}{2}\right\}}{\theta_1\left(L_0 + \frac{e_{\mathscr{C}(t)}}{2}\right)}\right\}, \text{otherwise} & (3.84) \end{cases}$$

where $p_{\mathscr{C}(t)}$ is the perimeter of $\mathscr{C}(t)$; $L_0 = Max\left\{\sin\frac{(n-2)\pi}{2n}\left|l_{\mathbb{C}}(s,\mathscr{C}(t))\right| - \frac{e_{\mathscr{C}(t)}}{2},\right.$ $\left.\sin\frac{(n-2)\pi}{2n}\left|l_{\mathbb{C}}(s,\mathscr{C}(t))\right| + |l_G| - e_{\mathscr{C}(t)}\right\}$, $l_{\mathbb{C}}(s,\mathscr{C}(t))$ and l_G are the shortest Euclidean paths from s and t to a vertex of $\mathscr{C}(t)$, respectively. θ_1 can be seen as the ratio of the base path length to the shortest path length, which is defined as follows. Specifically,

- $\theta_1 = \frac{|\mathscr{B}(s)|}{|l_{\mathbb{C}}(s,t)|}$, if s is a t-aware node.
- $\theta_1 = \frac{|\mathscr{B}(s)|}{|l_{\mathbb{C}}(s,\mathscr{C}(t))|}$, otherwise.

Algorithm 6 describes the pseudo code for determining the Euclidean routing path.

3.7.3.3 Forwarding the data packet

Denote by $sR_1...R_mx$ the Euclidean routing path constructed by s using the algorithm described above, where x coincides with t if s is a t-aware node, and x is a vertex of $\mathscr{C}(t)$, otherwise. Then, the coordinates of $R_1,...,R_m,x$ are inserted into the packet header as virtual anchors that will guide the packet.

If s is a t-aware node, then the packet is forwarded toward the virtual anchors using greedy algorithm until reaching t.

Otherwise, the packet is forwarded toward the virtual anchors using greedy algorithm until reaching the first t-aware node, hereafter we call this node as *sub-destination* and denote as t'. Then, t' determines the sub-base path, denoted as $\mathscr{B}(t^{'})$, which is an Euclidean path from t' to t that bypasses $\mathscr{H}(t^{'})$, $\mathscr{H}(t)$ and $\mathbb{C}(\overline{t'},\overline{t})$ and satisfies either one of the following conditions:

- $\mathscr{B}(t^{'})$ is the shortest path from t to a vertex of $\mathscr{C}(t)$.

- $\mathscr{B}(t^{'})$ is $(\Gamma_2 \times |\mathscr{L}(t',t)|)$-bounded path, where $\mathscr{L}(t',t)$ is the shortest path from t' to t and $\Gamma_2 = Max\left\{1, \frac{L_0(1+\varepsilon)}{\frac{\theta_1}{\sin\frac{(n-2)\pi}{2n}}\left(L_0+\frac{e_{\mathscr{C}}(t)}{2}\right)+\frac{p_{\mathscr{C}}(t)}{2}}\right\}$.

Let us denote $\mathbb{A}(t^{'})$ as the union of $\mathscr{H}(t^{'})$, $\mathscr{H}(t)$ and $\mathbb{C}(\overline{t'},\overline{t})$, then t' constructs the so-called sub-Euclidean routing path (denoted as $\mathscr{R}(t',t)$) which is the image of $\mathscr{B}(t^{'})$ through the transformation $F(\mathscr{B}(t^{'}),\mathbb{A}(t^{'}),\alpha_2)$, where α_2 is a parameter defined by

$$\alpha_2 = Max\left\{1, \frac{(1+\varepsilon)L_0 - |l_{t'}|}{|\mathscr{B}(t')|}\right\} \tag{3.85}$$

where $l_{t'}$ is the routing path from s to t'.

Finally, all the vertices of $\mathscr{R}(t',t)$ are inserted into the packets as virtual anchors. The packet then is greedily forwarded towards these virtual anchors gradually until reaching the detination t.

Algorithm 8 describes the pseudo code for forwarding data packet. The format of the data packet header is presented in Tab.3.4.

3.7.4 Section summary

In this section, we proposed our hole bypassing routing protocol, named BSMH. Our proposed protocol addresses the most complicated network scenario where multiple holes may exist, and the sources and the destinations of packets may stay inside the holes' core polygons. In order to reduce the overhead, we proposed a hole information dissemination protocol, in which the full information of each hole is broadcast to nodes staying inside the hole's vicinity region, while the core polygon information of the hole is broadcast to the other nodes. To guaranteeing the constant stretch upper bound while balancing traffic over the network, we proposed a data forwarding protocol which can be summarized as follows.

- For each source-destination pair, the source node determines a set of base path candidates, each of which is assigned a priority index. When having a packet to send, the source node exploits the priority indexes to probabilistically chose a base path from the candidate set.
- The base path then is magnified using a set of homothetic transformations whose centers are randomly chosen and the scale factors are controlled to guarantee the stretch upper bound. The result of the homothetic transformations is the Euclidean routing path.
- The packet then is forwarded along the Euclidean routing path until reaching the destination or arriving at an intermediate node which have more detail information about holes. In the latter case, the intermediate node redetermines the Euclidean routing path before forwarding the packet to the destination.

ALGORITHM 6: Euclidean routing path determination algorithm.

Input *P: the set of base paths, s: source, d: destination,* ε*: stretch factor*

Output *r: the Euclidean routing path* $\mathscr{B}(s) \leftarrow$ a base path probabilistically chosen from P;

$s\_core \leftarrow$ the core polygon containing s;

$s\_hole \leftarrow$ the hole whose core polygon is s_core ;

$d\_core \leftarrow$ the core polygon containing d;

$d\_hole \leftarrow$ the hole whose core polygon is d_core ;

$C \leftarrow$ all core polygons excepting s_core and d_core;

$\mathbb{A} \leftarrow C$;

if s_hole *!=* *null* **then**

  $\mathbb{A}$.add(s_hole);

end

if d_hole *!=* *null* || d_code == *null* **then**

  $\mathbb{A}$.add(d_hole);

  $l_{\mathbb{C}}(s,t) \leftarrow$ the shortest length of P ;

  $\theta_1 = \frac{|\mathscr{B}(s)|}{|l_{\mathbb{C}}(s,t)|}$;

  $\alpha_1 = \frac{1+\varepsilon}{\theta_1} \sin\frac{(n-2)\pi}{2n}$;

end

else if d_code *!=* *null* **then**

  $\mathbb{A}$.add(d_core);

  $l_{\mathbb{C}}(s,\mathscr{C}(t)) \leftarrow$ the shortest length of P ;

  $e_{\mathscr{C}(t)} \leftarrow$ the length of the longest edge of d_code;

  $p_{\mathscr{C}(t)} \leftarrow$ the perimeter of d_code;

  $\theta_1 = \frac{|\mathscr{B}(s)|}{|l_{\mathbb{C}}(s,\mathscr{C}(t))|}$;

  $L_0 = Max\left\{\sin\frac{(n-2)\pi}{2n}\,|l_{\mathbb{C}}(s,\mathscr{C}(t))| - \frac{e_{\mathscr{C}(t)}}{2}, \sin\frac{(n-2)\pi}{2n}\,|l_{\mathbb{C}}(s,\mathscr{C}(t))| + |l_G| - e_{\mathscr{C}(t)}\right\}$;

  $\alpha_1 = Max\left\{1, \frac{\sin\frac{(n-2)\pi}{2n}\left\{\frac{(1+\varepsilon)L_0}{\theta_1} - \frac{p_{\mathscr{C}(t)}}{2}\right\}}{\theta_1\left(L_0 + \frac{e_{\mathscr{C}(t)}}{2}\right)}\right\}$

end

$r \leftarrow F(\mathscr{B}(s), \mathbb{A}, \alpha_1)$;

return r

ALGORITHM 7: $F(l, \mathbb{P}, \alpha)$.

Input *l: a path,* $\mathbb{P}$*: set of polygons, alpha: scale factor*
Output l'*: scaled up path* $L_1, ..., L_k \leftarrow$ decompose l into segments, each of which is either positive-segment or negative-segment of a polygon belonging to $\mathbb{P}$;
for $i = \overline{1,k}$ **do**
  $P_i \in \mathbb{P} \leftarrow$ the polygon containing L_i;
  $L_i^1, ..., L_i^{n_i} \leftarrow$ the vertices of L_i;
  if P_i *is a convex polygon* **then** O_i is a random point inside P_i ;
  else if L_i *is a positive-segment* **then** O_i is a random point staying on the right side of L_i ;
  else O_i is a random point staying on the left side of L_i ;
  $x_i = |L_i|$; $y_i = \left|O_i L_i^1\right| + \left|O_i L_i^{n_i}\right|$;
  if $i = 1$ **then** $z_i = \left|sL_i^1\right| + \frac{1}{2}\left|L_i^{n_i} L_{i+1}^1\right|$;
  else if $i = \overline{2,k-1}$ **then**
    $z_i = \frac{\left|L_i^1 L_{i-1}^{n_{i-1}}\right| + \left|L_i^{n_i} L_{i+1}^1\right|}{2}$
  end
  else $z_i = \left|L_i^{n_i} t\right| + \frac{1}{2}\left|L_{i-1}^{n_{i-1}} L_i^1\right|$;
  $\xi_i = 1 + (\alpha - 1)\frac{z_i + x_i}{x_i + y_i}$;
  for $j = \overline{1,n_i}$ **do**
    $L_i^{j'}.x = \xi_i \times L_i^j.x + (1 - \xi_i) \times O_i.x$;
    $L_i^{j'}.y = \xi_i \times L_i^j.y + (1 - \xi_i) \times O_i.y$;
    while $L_i^{j'} L_i^{j'+1}$ *intersects* $\mathbb{P}$ || $L_i^{j'} L_i^{j'+1}$ *exceeds the network* **do**
      $\xi_i \leftarrow \frac{3}{4}\xi_i$
    end
    l'.add($L_i^{j'}$)
  end
end
return l'

Table 3.4: Data message

Field	*Size (bytes)*	*Content*
Type	1	BSMH_Data
forwarding_mode	1	indicating whether the packet is in aware mode or blind mode
vertices_coordinates	$a \times 16$	An array consists the coordinates of a virtual anchors

ALGORITHM 8: Data forwarding protocol.

Input *p: the packet, c: current node* $d \leftarrow$ the destination stored in p; $m \leftarrow$ forwarding mode of p ;
if *c is the source node* **then**
    c.Algorithm 4 *// Find the base path set* ;
    p.virtual_anchor_list $\leftarrow$ *c*.Algorithm 6 *// Determine the Euclidean routing path* ;
    if *c*.d_hole *!= null* || *c*.d_hole == *null* **then** $m \leftarrow d\_aware$; ;
    else $m \leftarrow blind$; ;
end
else
    if m == blind **then**
        if *c*.d_hole *!= null* **then**
            c_core_list $\leftarrow$ all core polygons stored at c; *c_hole_list* $\leftarrow$ the holes stored at c;
            all_core_polygon_list $\leftarrow$ *c_core_list*;
            for hole $\in$ c_hole_list **do**
                core $\leftarrow$ create core polygon of *hole*; *all_core_polygon_list*.add(*core*);
            end
            for core $\in$ all_core_polygon_list **do**
                if $s \in$ core **then** *c_core* $\leftarrow$ *core* ;
                if $d \in$ core **then** *d_core*$\leftarrow$ *core* ;
            end
            P, *shortest_len*$\leftarrow$ Algorithm 5(c, d);
            $\Gamma_2 = Max\left\{1, \frac{L_0(1+\varepsilon)}{\frac{\theta_1}{\sin\frac{(n-2)\pi}{2n}}\left(L_0+\frac{e_{\mathscr{C}}(t)}{2}\right)+\frac{p_{\mathscr{C}}(t)}{2}}\right\}$;
            for $p \in P$ **do**
                if $|p| > \Gamma_2 \times$ *shortest_len* **then** P.remove(p) ;
            end
            convex_hull $\leftarrow$ the convex hull of *all_core_polygon_list*;
            $V_c \leftarrow$ vertices of *convex_hull*; $n \leftarrow$ the number of the vertices of V_c ;
            for $p \in P$ **do**
                $n_1 \leftarrow$ the number of p's vertices belonging to V_c; $p.priority \leftarrow \frac{n_1}{n}$
            end
            $\mathscr{B}(t^{'}) \leftarrow$ a base path probabilistically chosen from P ;
            $l_{t'} \leftarrow$ the routing path from the source s to c; $\alpha_2 = Max\left\{1, \frac{(1+\varepsilon)L_0-|l_{t'}|}{|\mathscr{B}(t')|}\right\}$;
            p.virtual_anchor_list $\leftarrow F(\mathscr{B}(t^{'}), \mathbb{A}, \alpha_2)$; $m \leftarrow d\_aware$
        end
    end
end
current_virtual_anchor$\leftarrow$ the first item of *p.virtual_anchor_list*;
p.next_hop$\leftarrow$ getNextHopByGreedy(c, *current_virtual_anchor*);
if next_hop == *null* **then**
    if *p*.virtual_anchor_list == *null* **then** *current_virtual_anchor*$\leftarrow$ d ;
    else
        Remove the first item of *p.virtual_anchor_list*;
        current_virtual_anchor$\leftarrow$ the first item of *p.virtual_anchor_list*;
    end
end
p.next_hop$\leftarrow$ getNextHopByGreedy(c, *current_virtual_anchor*);
c.forward(p);

4

Theoretical analysis

In this chapter, we analysis BSMH theoretically. We first investigate the computational complexity of BSMH in Section 4.1 In Section 4.3 we describe a thoroughly proof of the constant stretch property of BSMH. Finally, in Section 4.2, we present a preliminary and intuitive analysis about the impact of stretch factor, i.e., ε, on the performance of BSMH.

4.1 Computational complexity

In this section, we analysis the computational complexity of each phase of BSMH. Throughout this section, we denote h the total number of the holes in the network, v the largest number of the nodes belonging to the boundary of a hole, n the number of the vertices of a core polygon, k_n the number of the core polygon candidates, b the maximum number of neighbors of a node.

4.1.1 Complexity of the hole determination algorithm

The hole determination algorithm consists of two steps: determining the stuck nodes (the nodes where local minimum phenomenon may happen, see Algorithm 1) and locating the hole boundary (Algorithm 2). The stuck node determination algorithm is conducted locally at every node. Specifically, every node first needs to sort its neighbor list by the clockwise order, and then it searches all of its two adjacent neighbors to find stuck angles. The computational complexity for sorting the neighbor list is $O(b \log b)$; The computational complexity for searching all of the adjacent neighbor pairs is $O(b)$. Consequently, the total computational complexity for determining the stuck nodes is $O(b \log b)$.

To locate the hole boundary, nodes on the hole boundary have to forward HBA messages. In order to determine the next hop of a HBA message, the node performs RIGHT HAND rule which investigates all neighbors of the current node to find the most-right neighbor. The computational complexity for determining the next hop by the RIGHT HAND rule is $O(b)$.

In consequence, the computational complexity of hole determination algorithm is $O(b \log b)$.

4.1.2 Complexity of the dissemination algorithm

The node receiving a HCI packet may perform the following tasks:

1. Checking whether the node has already received a HCI packet.
2. Constructing the core polygon of the hole contained in the HCI packet.
3. Checking whether the node stays inside the vicinity region of the hole contained in the HCI packet.

To conduct the first task, the node needs to investigate all of the core polygons and the holes stored in its memory. For each of them, the node identifies whether it is the same as the one contained in the HCI packet. The computational complexity for this task is $O(h \times v)$ (1).

In the second task, the node performs a loop over the vertices of the hole contained in the HCI packet. For each vertex, the node checks whether it is a core node (i.e., a node staying on the edges of the core polygon candidates). This job causes a computational

complexity of $O(v \times n \times k_n)$. Then, the node determines the vertices of all the core polygon candidates which are the intersection points of all lines going through the core nodes and making specific angles with the $x-$axis. This job requires a complexity of $O(n \times k_n)$. Finally, the node computes the area of each core polygon candidate and chooses the core polygon candidate with the smallest area to be the core polygon. The computational complexity of this job is $O(n \times k_n)$. In consequence, the total complexity of the second task is $O(v \times n \times k_n)$ (2).

In the third task, the node just needs to compute the distances to the core polygon stored in the HCI packet and compare them with the threshold value. Thus, the computational complexity of this task is $O(n)$ (3).

From (1), (2) and (3), it is deduced that the computational complexity of the dissemination algorithm is $O(v \times (h + nk_n))$.

4.1.3 Complexity of the data forwarding algorithm

In order to forward a data packet, the node may need to perform the following tasks:

1. Determining the base path candidate set (i.e., in the case the node is the source node) (Algorithm 4).

2. Determining the Euclidean routing path (i.e., in the case the node is the source node) (Algorithm 6).

3. Determining the next hop by using the greedy forwarding algorithm.

In order to determine the base path candidates, the node needs to identifying the hole containing the source and the destination. This job requires the computational complexity of $O(h \times n)$. Then, the node constructs the visibility graph whose vertices are the vertices of the core polygons that don't containing the source and the destination, and the vertices of the holes containing the source and the destination, and whose edges don't intersect the interior of the core polygons and holes. As the number of the vertices of the visibility graph does not exceed $2v + (h-2)n$, the computational complexity for constructing such a visibility graph does not exceed $O\left((2v + (h-2)n)^3\right)$. Based on the visibility graph, the node determines all the base path candidates by using breath-first search whose computational complexity is $O\left(2v + (h-2)n + (2v + (h-2)n)^2\right)$. Therefore, the computational complexity of the first task (i.e., finding the base path candidates) is

$O\left(h \times n + (2v + (h-2)n)^3 + 2v + (h-2)n + (2v + (h-2)n)^2\right)$ which is equivalent to $O\left((2v + (h-2)n)^3\right)$ (4).

Let c be the number of the base path candidates, then the complexity for choosing a base path from the candidate list is $O(c)$. To scale up the base path using the homothetic transformation, the node needs to determine the transformation centers, scale factors and the coordinates of the images. The computational complexity for these jobs is $O(m)$, where m is the number of vertices of the base path. As $m < 2v + (h-2)n$, the computational time complexity for the second task does not exceed $O(c + 2v + (h-2)n)$ (5).

Having determined the Euclidean path, the next hop is determined by using greedy algorithm whose computational complexity is only $O(b)$, where b is the number of 1-hop neighbors (6).

From (4), (5), and (6), the time complexity of data forwarding algorithm is $O\left((2v + (h-2)n)^3 + c + b\right)$.

4.2 Impact of the stretch factor ε

4.2.1 Routing path stretch

In BSMH, packets are forwarded along the Euclidean routing path which is the image of the base path through the homothetic transformations. The scale factors of the transformations, i.e., α_1 and α_2, are determined base on formulas (3.83), (3.84) and (3.85). Obviously, the scale factors are proportional to ε. Therefore, when increasing ε, the length of the Euclidean routing path increases and thus enlarges the routing path stretch.

4.2.2 Load balance

The relationship between ε and the network's load balance is not straightforward. In the following, we present a preliminary and intuitive theoretical analysis of this relationship. Let us consider a simple network topology consisting of only one hole and one source-destination pair as shown in Fig.4.1. Let us denote by $\mathscr{C}$ the hole, by s the source node and by t the destination node. Denote by l_1 and l_2 the two Euclidean paths that bypass $\mathscr{C}$ and that have all the vertices be the vertices of $\mathscr{C}$ as shown in Fig.4.1. Suppose

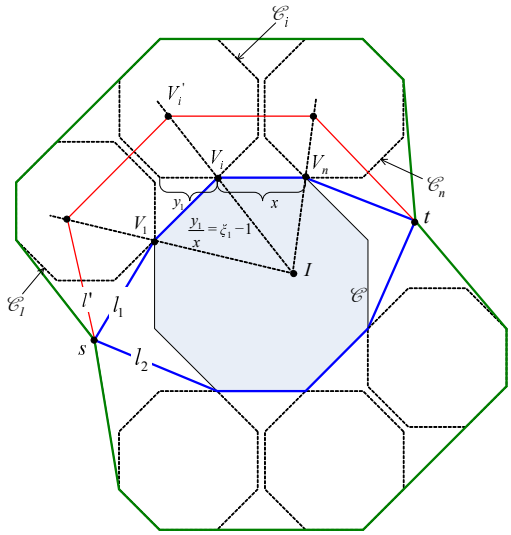


Figure 4.1: The trajectory of Euclidean routing paths.
The base paths l_1 and l_2 are colored blue, the Euclidean path l' is colored red.

that l_1 is shorter than l_2. Then, if ε is sufficiently large, the base path candidates are both l_1 and l_2. Otherwise, the base path candidates consist of only l_1. Suppose l' is a Euclidean path. Without loss of generality, we assume that l' is the image of l_1 through a homothetic transformation with the center I and the scale factor ξ. Note that ξ is defined by Formula (3.77)(3.83), and it can be written as:

$$\xi = \gamma_1 \varepsilon + \gamma_2 \tag{4.1}$$

where γ_1 and γ_2 are constants that the same for all packets from s to t. Suppose $V_1, ..., V_n$ are the vertices of $\mathscr{C}$ that are vertices of l_1. Let V_i' be the image of V_i through the homothetic transformation ($\forall i = 1, ..., n$). Then, obviously V_i' is the image of I through the homothetic center with the center V_i and the scale factor $\xi - 1$. As I locates inside $\mathscr{C}$, V_i' must stay inside the image of $\mathscr{C}$ through the homothetic transformation with the center V_i and the scale factor $\xi - 1$ (denoted as $\mathscr{C}_i$) (see Fig.4.1). Let $\mathscr{B}$ be the union of all $\mathscr{C}_i$ ($\forall i = 1, ..., n$). Note that when ε is sufficiently small such that $\mathscr{B}$ stays inside the network, then $\mathscr{B}$ is the trajectory of all routing paths from s to t. Accordingly, the trajectory of sensor nodes that participate in forwarding packets are polygons that are limited by the boundaries of $\mathscr{C}_i$ and the base paths. In Fig.4.1, this trajectory are polygons whose boundaries are the green lines and the blue lines.

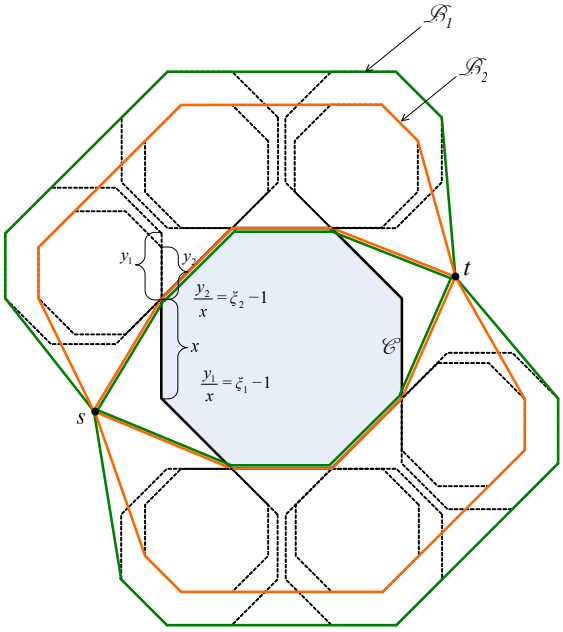


Figure 4.2: The trajectories of Euclidean routing paths with different values of ε
The trajectory of Euclidean routing paths when $\varepsilon = \varepsilon_1$, is bounded by the green line.
The trajectory of Euclidean routing paths when $\varepsilon = \varepsilon_2$, is bounded by the orange line.

Let ε_1 and ε_2 are two positive number where $\varepsilon_1 > \varepsilon_2$. Let $\mathscr{B}_1$ and $\mathscr{B}_2$ be the trajectories of sensor nodes participating in forwarding packets when $\varepsilon = \varepsilon_1$, and $\varepsilon = \varepsilon_2$, respectively. Figure 4.2 illustrates $\mathscr{B}_1$ and $\mathscr{B}_2$, in which the boundary of $\mathscr{B}_1$ is colored orange and the boundary of $\mathscr{B}_1$ is colored green. In the following we will prove the following statement: *"If $\mathscr{B}_1$ and $\mathscr{B}_2$ stay inside the network, then the probability for a node to forward a packet in the case $\varepsilon = \varepsilon_1$ does not exceed that in the case $\varepsilon = \varepsilon_2$"*.

Denote ξ_1 and ξ_2 the scale factors in the homothetic transformations in the cases $\varepsilon = \varepsilon_1$ and $\varepsilon = \varepsilon_2$, respectively. According to formula (4.1), ξ_1 and ξ_2 is proportional to ε_1 and ε_2, respectively. Let $\mathscr{R}_1$ be a Euclidean routing path obtained in the case $\varepsilon = \varepsilon_1$. Then, obviously, there must exist such a Euclidean routing path $\mathscr{R}_2$ obtained in the case $\varepsilon = \varepsilon_2$ that $\mathscr{R}_1$ is an image of $\mathscr{R}_2$ through a homothetic transformation with the scale factor $\frac{\xi_1}{\xi_2}$ (because they are image of a base path through homothetic transformations with the scale factors of ξ_1 and ξ_2, respectively). Therefore, according to Lemma 10 the length of $\mathscr{R}_1$ does not exceed $\frac{\xi_1}{\xi_2}$ times that of $\mathscr{R}_2$. Consequently, let T_1 and T_2 be the total hopcounts of all packets from s to t in the cases $\varepsilon = \varepsilon_1$ and $\varepsilon = \varepsilon_2$, respectively.

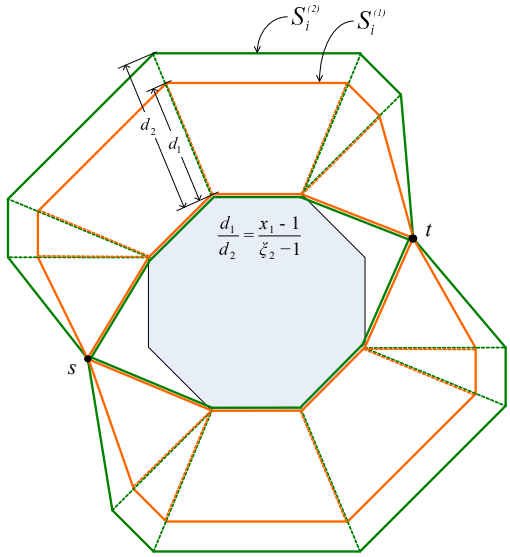


Figure 4.3: Comparison of the areas of $\mathscr{B}_1$ and $\mathscr{B}_2$.
In order to compare $\mathscr{B}_1$ and $\mathscr{B}_2$, we divide them into sub-regions by dotted lines.

Then, we can deduce that:

$$L_1 \leq \frac{\xi_1}{\xi_2} L_2 \tag{4.2}$$

Moreover, let us divide $\mathscr{B}_1$ and $\mathscr{B}_2$ into sub-regions by the segments connecting their vertices as shown in Fig.4.3. Denote the i-th sub-region of $\mathscr{B}_1$ as $S_i^{(1)}$ and the corresponding sub-region of $\mathscr{B}_2$ as $S_i^{(2)}$ (Fig.4.3). Obviously, the area of $S_i^{(1)}$ is $\frac{\xi_1 - 1}{\xi_2 - 2}$ times that of the area of $S_i^{(2)}$. Let n_1 and n_2 be the number of nodes participating in forwarding packets in the case $\varepsilon = \varepsilon_1$ and $\varepsilon = \varepsilon_2$, respectively. If we assume that the nodes are evenly distributed in the network, then n_1 and n_2 are proportional to the areas of $\mathscr{B}_1$ and $\mathscr{B}_2$, respectively. Accordingly, we have the following:

$$n_1 = \frac{\xi_1 - 1}{\xi_2 - 1} n_2 \tag{4.3}$$

Denote by p_1 and p_2 the probabilities for a node to forward a packet in the cases $\varepsilon = \varepsilon_1$ and $\varepsilon = \varepsilon_2$, respectively. Then, we have:

$$p_1 = \frac{L_1}{n_1} \tag{4.4}$$

$$p_1 = \frac{L_2}{n_2} \tag{4.5}$$

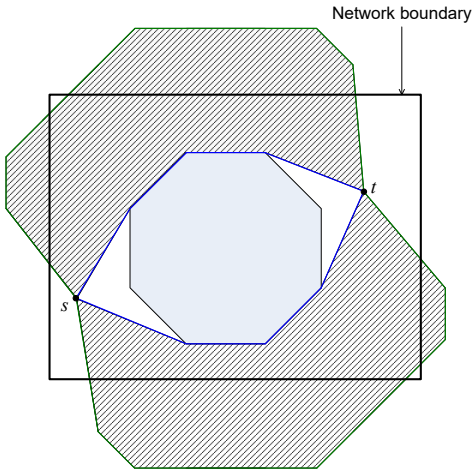


Figure 4.4: When ε is too large, the trajectory of the Euclidean routing paths may exceed the network boundary.

From (4.2), (4.3) and (4.4), it can be deduced that $p_1 \leq p_2$. It means that when ε is sufficiently small such that the trajectory of all routing paths from s to t stays inside the network, then increasing ε tends to improve the load balance.

However, when ε is too large, the trajectory of all routing paths from s to t may exceed the network boundary as shown in Fig. 4.4. In this case, the area of the trajectory becomes almost stable when increasing ε. It means that increasing ε cannot increase the number of sensors participating in the routing process. In the meanwhile, the routing path stretch still gradually increases when increasing ε. Consequently, increasing ε tends to decrease the load balance.

4.3 Routing path stretch

Let s,t be an arbitrary source-destination pair, in the following we will prove that the stretch of a routing path from s to t is upper bounded by $1+\varepsilon$.

Lemma 9

Let s and t are two points on the plane. Let $l_{\mathbb{C}}(s,t)$ be the shortest Euclidean path from s to t that bypasses $\mathscr{H}(s)$, $\mathscr{H}(t)$ and $\mathbb{C}(\overline{s},\overline{t})$. Let $l_{\mathbb{H}}(s,t)$ be the shortest Euclidean path from s to t that bypasses $\mathbb{H}$. Then, $\frac{|l_{\mathbb{C}}(s,t)|}{|l_{\mathbb{H}}(s,t)|} \leq \frac{1}{\sin\frac{(n-2)\pi}{2n}}$.

Proof

We will prove by showing that there exists a path from s to t that bypasses $\mathscr{H}(s)$, $\mathscr{H}(t)$ and $\mathbb{C}(\overline{s},\overline{t})$ and that has a length less than $\frac{1}{\sin\frac{(n-2)\pi}{2n}}|L_{\mathbb{H}}(s,t)|$ (*).

Let us divide $L_{\mathbb{H}}(s,t)$ into three parts: the first one is the path from s to a point on the boundary of $\mathscr{C}(s)$ (named as *go-in path*), the second one is the path from the point on the boundary of $\mathscr{C}(s)$ (named as *go-between path*) to a point on the boundary of $\mathscr{C}(t)$, and the last one if the path from the point on the boundary of $\mathscr{C}(t)$ to t (named as *go-in path*). We denote these three parts as l_o, l_b and l_i, and denote the two end points of l_b as B_s and B_t (see Fig.4.5). Note that if s does not stay inside a core polygon then $l_o \equiv s$, and if t does not stay inside a core polygon then $l_i \equiv t$. Then, (*) can be proved by showing that there exists a path from B_s to B_t that bypasses $\mathscr{H}(s)$, $\mathscr{H}(t)$ and $\mathbb{C}(\overline{s},\overline{t})$ and that has a length less than $\frac{1}{\sin\frac{(n-2)\pi}{2n}}|l_b|$. Obviously, l_b is comprised of the vertices of convex hull of the holes. Suppose $l_b = B_s H_1^1 ... H_1^{n_1} ... H_m^1 ... H_m^{n_m} B_t$, where $H_j^1 ... H_j^{n_j}$ is a boundary segment of convex hull of hole $\mathscr{H}_j$ ($\forall j = \overline{1,m}$) (see Fig. 3.20). Let us denote $\mathscr{C}_j$ as the core polygon of $\mathscr{H}_j$. Note that, for each $j \in \{1,2,...,m-1\}$, segment $H_j^{n_j} H_{j+1}^1$ intersect $\mathscr{C}_j$ at at most two points and one of them is $H_j^{n_j}$. We denote the second intersection point as I_j^2, where $I_j^2 \equiv H_j^{n_j}$ if $H_j^{n_j} H_{j+1}^1$ intersects $\mathscr{C}_j$ at only one point. Similarly, $H_j^{n_j} H_{j+1}^1$ also intersects $\mathscr{C}_{j+1}$ at at most two points, one of them is H_{j+1}^1. We denote the other intersection point as I_{j+1}^1 which may coincides with C_{j+1}^1 if $H_j^{n_j} H_{j+1}^1$ intersects $\mathscr{C}_{j+1}$ at only one point. We also denote the intersection point of $B_s H_1^1$ with $\mathscr{C}_1$ as I_1^1 and the intersection point of $H_m^{n_m} B_t$ with $\mathscr{C}_m$ as I_m^2. We will prove that, the path $\mathfrak{L} = sI_1^1 \left\{I_1^1 \sim I_1^2\right\}_{\mathscr{C}_1} ... I_j^1 \left\{I_j^1 \sim I_j^2\right\}_{\mathscr{C}_j} I_j^2 .... I_1^{i_m} \left\{I_m^1 \sim I_m^2\right\}_{\mathscr{C}_m} I_m^2 t$ (the direction of $\left\{I_j^1 \sim I_j^2\right\}_{\mathscr{C}_j}$ is the same as that of $\left\{H_j^1 \sim H_j^{n_j}\right\}_{\mathscr{H}_j}$) satisfies condition (*). Note that, $I_j^2 I_{j+1}^1$ is a part of $H_j^{n_j} H_{j+1}^1$ ($\forall j = \overline{1,m}$), thus they do not intersect any core polygon. $\mathfrak{L}$ is comprised of core polygons' boundary segments and the set of $I_j^2 I_{j+1}^1$ ($j = \overline{1,m}$), thus $\mathfrak{L}$ does not intersect any core polygon. Since all core polygons are disjoint, $\mathfrak{L}$ stays outside of all holes.

Now we will prove that $|\mathfrak{L}| \leq \frac{1}{\sin\frac{(n-2)\pi}{2n}}|L_{\mathbb{H}}(s,t)|$. Note that, the length of $\mathfrak{L}$ can be decomposed as:

$$|\mathfrak{L}| = \left|B_s I_1^1\right| + \sum_{j=1}^{m} \left|\left\{I_j^1 \sim I_j^2\right\}_{\mathscr{C}_j}\right| + \sum_{j=1}^{m-1} \left|I_j^2 I_{j+1}^1\right| + \left|I_m^2 B_t\right| \tag{4.6}$$

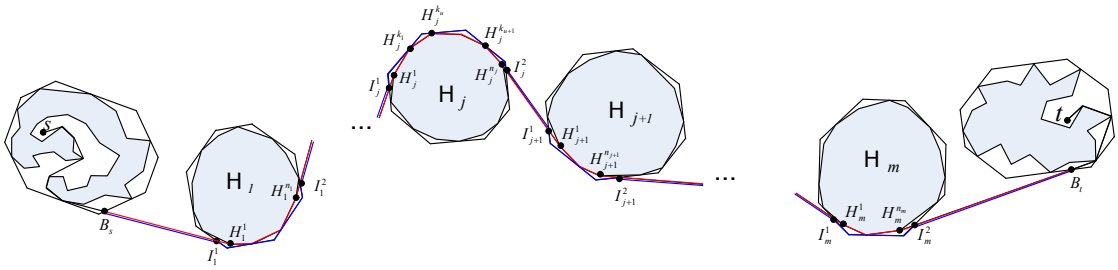


Figure 4.5: Illustration of $L_{\mathbb{H}}(s,t)$, l_b and $\mathfrak{L}$. The red line is l_b the blue line is $\mathfrak{L}$.

The length of $L_{\mathbb{H}}(s,t)$ can be decomposed as:

$$|L_{\mathbb{H}}(s,t)| = |B_s I_1^1| + \sum_{j=1}^{m} \left| I_j^1 \left\{ H_j^1 \sim H_j^{n_j} \right\}_{\mathscr{H}_j} I_j^2 \right| + \sum_{j=1}^{m-1} |I_j^2 I_{j+1}^1| + |I_m^2 B_t| \tag{4.7}$$

From (4.6) and (4.7), it is obviously that we can prove the lemma by showing that:

$$\left| \{ I_j^1 \sim I_j^2 \}_{\mathscr{C}_j} \right| \leq \frac{1}{\sin \frac{(n-2)\pi}{2n}} \left| I_j^1 \left\{ H_j^1 \sim H_j^{n_j} \right\}_{\mathscr{H}_j} I_j^2 \right| \forall j \in \{1,...,m\} \tag{4.8}$$

We denote $H_j^{k_1},...,H_j^{k_v}$ as the $\mathscr{H}_j$'s vertices that belong to $\left\{ H_1^1 \sim H_j^{n_j} \right\}_{\mathscr{H}_j}$ and stay on the boundary of $\mathscr{C}_j$, and denote $C_j^u C_j^{u+1}$ as the edge of $\mathscr{C}_j$ that contains $H_j^{k_u}$ $(\forall u = \overline{1,v})$. Using triangular cosin formula, we have:

$$\begin{aligned} \left| H_j^{k_u} C_j^{u+1} \right| + \left| C_j^{u+1} H_j^{k_{u+1}} \right| &\leq \frac{1}{\sin \frac{\angle H_j^{k_u} C_j^{u+1} H_j^{k_{u+1}}}{2}} \left| H_j^{k_u} H_j^{k_{u+1}} \right| \\ &\leq \frac{1}{\sin \frac{(n-2)\pi}{2n}} \left| \left\{ H_j^{k_u} \sim H_j^{k_{u+1}} \right\}_{\mathscr{H}_j} \right| \quad \forall j = \overline{1,m} \end{aligned} \tag{4.9}$$

Similarly, we have:

$$\left| I_j^1 C_j^1 \right| + \left| C_j^1 H_j^{k_1} \right| \leq \frac{1}{\sin \frac{(n-2)\pi}{2n}} \left(\left| I_j^1 H_j^1 \right| + \left| \left\{ H_j^1 \sim H_j^{k_1} \right\}_{\mathscr{H}_j} \right| \right) \tag{4.10}$$

$$\left| I_j^2 C_j^{v+1} \right| + \left| C_j^{v+1} H_j^{k_v} \right| \leq \frac{1}{\sin \frac{(n-2)\pi}{2n}} \left(\left| I_j^2 H_j^{n_j} \right| + \left| \left\{ H_j^{k_v} \sim H_j^{n_j} \right\}_{\mathscr{H}_j} \right| \right) \tag{4.11}$$

By summing up (4.9), (4.10) and (4.11) we obtain (4.8)■

Lemma 10

Let l' be the image of l through the transformation $F(l,\mathbb{P},\alpha)$, then $\frac{|l'|}{|l|} \leq \alpha$.

Proof

According to the definition of F in Section 3.7.3.2, l' is comprised of $L_1^{'},...,L_k^{'}$ where $L_i^{'}$ is the image of L_i through a homothetic transformation F_i. Suppose L_i consists of n_i vertices $L_i^1,...,L_i^{n_i}$, then according to (3.77) the scale factor of F_i is defined as follows.

$$\xi_i = 1 + (\alpha - 1)\frac{z_i + a_i}{x_i + y_i} \tag{4.12}$$

where $x_i = |L_i|$, $y_i = \left|O_i L_i^1\right| + \left|O_i L_i^{n_i}\right|$,

$$z_i = \begin{cases} \left|sL_i^1\right| + \frac{1}{2}\left|L_i^{n_i} L_{i+1}^1\right|, \text{ if } i = 1 & (4.13) \\ \frac{\left|L_i^1 L_{i-1}^{n_{i-1}}\right| + \left|L_i^{n_i} L_{i+1}^1\right|}{2}, \text{ if } k = \overline{2, k-1} & (4.14) \\ \left|L_i^{n_i} t\right| + \frac{1}{2}\left|L_{i-1}^{n_{i-1}} L_i^1\right|, \text{ if } i = k & (4.15) \end{cases}$$

Suppose $L_i^{'} = L_i^{'1},...,L_i^{'n_i}$ where $L_i^{'j} = F_i(L_i^j)$, then we have:

$$\left|l'\right| = \left|sL_1^{'1}\right| + \sum_{i=1}^{k}\left|L_i^{'}\right| + \sum_{i=1}^{k-1}\left|L_i^{'n_i} L_{i+1}^{'1}\right| + \left|L_k^{'n_k} t\right| \tag{4.16}$$

$$|l| = \left|sL_1^1\right| + \sum_{i=1}^{k}|L_i| + \sum_{i=1}^{k-1}\left|L_i^{n_i} L_{i+1}^1\right| + \left|L_k^{n_k} t\right| \tag{4.17}$$

Since $L_i^{'}$ is the image of L_i ($\forall j = \overline{1,m}$) through a homothetic transformation with the scale factor ξ_j, we have:

$$\left|L_i^{'}\right| = \xi_i |L_i| \tag{4.18}$$

Using triangular inequality, we have:

$$\left|sL_1^{'1}\right| \leq |sL_1| + \left|L_1 L_1^{'1}\right| = |sL_1| + (\alpha_1 - 1)|I_1 L_1| \tag{4.19}$$

$$\left|L_k^{'n_k} t\right| \leq \left|L_k^{n_k} t\right| + \left|L_k^{n_k} L_k^{'n_k}\right| = \left|L_k^{n_k} t\right| + (\xi_k - 1)\left|I_k L_k^{n_k}\right| \tag{4.20}$$

$$\begin{aligned} \left|L_i^{'n_i} L_{i+1}^{'1}\right| &\leq \left|L_i^{'n_i} L_i^{n_i}\right| + \left|L_i^{n_i} L_{i+1}^1\right| + \left|L_{i+1}^1 L_{i+1}^{'1}\right| \\ \Rightarrow \left|L_i^{'n_i} L_{i+1}^{'1}\right| &\leq \left|L_i^{n_i} L_{i+1}^1\right| + (\xi_j - 1)\left(\left|I_i L_i^1\right| + \left|I_i L_i^{n_i}\right|\right) \end{aligned} \tag{4.21}$$

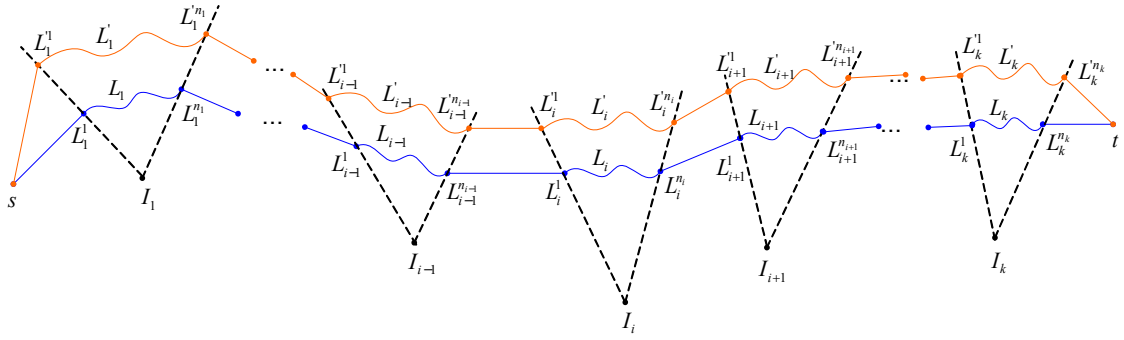


Figure 4.6: Illustration of l and l'. The blue line is l, the orange line is l'.

Summing up (4.18), (4.19), (4.20) and (4.21), we have:

$$\left|l'\right| \leq |sL_1| + \left|L_k^{n_k} t\right| + \sum_{i=1}^{k-1} \left|L_i^{n_i} L_{i+1}^1\right| + \sum_{i=1}^{k} \xi_i |L_i| + \sum_{i=1}^{k-1} (\xi_j - 1) \left(\left|I_i L_i^1\right| + |I_i L_i^{n_i}|\right) \tag{4.22}$$

The right side of (4.22) equals to:

$$\sum_{i=1}^{k} \xi_i x_i + (\xi_i - 1) y_j + z_i$$

By substituting value of ξ_i from (4.12), we have $\xi_i x_i + (\xi_i - 1) y_i + z_i = \alpha (x_i + z_i)$. Consequently, $|l'| \leq \xi \Gamma_1 \sum_{i=1}^{k} x_i + z_i = \alpha |l|$■

Theorem 3

The stretch of every routing path is upper bounded by $1 + \varepsilon$.

Let us denote by $\mathscr{L}(s,t)$ the theoretical shortest hole-bypassing Euclidean path from s to t, and $\mathscr{R}(s,t)$ the Euclidean routing path from s to t determined by our protocol. According to the network model assumed in Section 3.1, the length of the theoretical shortest routing path and the length of the real routing path determined by our protocol can be approximated by $|\mathscr{L}(s,t)|$ and $|\mathscr{R}(s,t)|$, respectively. Therefore, the routing path stretch can be approximated by $\frac{|\mathscr{R}(s,t)|}{|\mathscr{L}(s,t)|}$. In the following, we will prove that $\frac{|\mathscr{R}(s,t)|}{|\mathscr{L}(s,t)|} \leq 1 + \varepsilon$.

Case 1: s is a t-aware node

On the one hand, $\mathscr{R}(s,t)$ is the image of the base path (i.e., $\mathscr{B}(s)$) through the transformation $F(\mathscr{B}(s),\mathbb{A}(s),\alpha_1)$, thus according to Lemma 10, we have:

$$\frac{|\mathscr{R}(s,t)|}{|\mathscr{B}(s)|} \leq \alpha_1 \tag{4.23}$$

On the other hand, according to Lemma 9, we have:

$$\frac{|l_{\mathbb{C}}(s,t)|}{|\mathscr{L}(s,t)|} \leq \frac{1}{\sin\frac{(n-2)\pi}{2n}} \tag{4.24}$$

Moreover, as $|\mathscr{B}(s)| = \theta_1 |l_{\mathbb{C}}(s,t)|$, we have:

$$|\mathscr{B}(s)| \leq \frac{1}{\sin\frac{(n-2)\pi}{2n}} \theta_1 |\mathscr{L}(s,t)| \tag{4.25}$$

From (4.23) and (4.25), it can be deduced that $|\mathscr{R}(s,t)| \leq \alpha_1 \frac{1}{\sin\frac{(n-2)\pi}{2n}} \theta_1 |\mathscr{L}(s,t)|$. Since $\alpha_1 = \frac{1+\varepsilon}{\theta_1} \sin\frac{(n-2)\pi}{2n}$ (according to (3.83)), we have $|\mathscr{R}(s,t)| \leq (1+\varepsilon)|\mathscr{L}(s,t)|$.

Case 2: s is a t-blind node

Let us denote $\mathscr{R}(t',t)$ as the Euclidean routing path determined by t'. Suppose $l_{\mathbb{C}}(s,v)$ is the shortest path from s to a vertex of $\mathscr{C}(t)$ that bypasses $\mathscr{H}(s)$ and $\mathbb{C}(\overline{s})$ (i.e., v is a vertex of $\mathscr{C}(t)$). Let u denote the intersection of $\mathscr{L}(s,t)$ with the boundary of $\mathscr{C}(t)$, and V_1V_2 denote the edge of $\mathscr{C}(t)$ containing u. Denote by $\mathscr{L}(s,u), \mathscr{L}(s,V_1), \mathscr{L}(s,V_2)$ the shortest Euclidean paths bypassing $\mathbb{H}(s)$ from s to u, V_1, V_2, respectively. Denote by $l_{\mathbb{C}}(s,V_1), l_{\mathbb{C}}(s,V_2)$ the shortest Euclidean paths bypassing $\mathscr{H}(s)$ and $\mathbb{C}(\overline{s})$ from s to V_1, V_2, respectively. Since $l_{\mathbb{C}}(s,v)$ is the shortest Euclidean path from s to a vertex of $\mathscr{H}(t)$ that bypasses $\mathscr{H}(s)$ and $\mathbb{C}(\overline{s})$, we have:

$$|l_{\mathbb{C}}(s,v)| \leq Min\{|l_{\mathbb{C}}(s,V_1)|, |l_{\mathbb{C}}(s,V_2)|\} \tag{4.26}$$

Moreover, according to Lemma 9, we have:

$$|l_{\mathbb{C}}(s,V_1)| \leq \frac{1}{\sin\frac{(n-2)\pi}{2n}} |\mathscr{L}(s,V_1)| \tag{4.27}$$

$$|l_{\mathbb{C}}(s,V_2)| \leq \frac{1}{\sin\frac{(n-2)\pi}{2n}} |\mathscr{L}(s,V_2)| \tag{4.28}$$

Therefore, from (4.26), (4.27) and (4.28), it can be deduced that:

$$|l_{\mathbb{C}}(s,v)| \leq \frac{1}{\sin\frac{(n-2)\pi}{2n}} Min\left\{|\mathscr{L}(s,V_1)|, |\mathscr{L}(s,V_2)|\right\} \tag{4.29}$$

Note that

$$|\mathscr{L}(s,V_1)| \leq |\mathscr{L}(s,u)| + uV_1 \tag{4.30}$$
$$|\mathscr{L}(s,V_2)| \leq |\mathscr{L}(s,u)| + uV_2 \tag{4.31}$$

Consequently, from (4.26)(4.27)(4.28)(4.44)(4.30)(4.31), we deduce that:

$$\begin{aligned} |l_{\mathbb{C}}(s,v)| &\leq \frac{1}{\sin\frac{(n-2)\pi}{2n}} \left\{ |\mathscr{L}(s,u)| + \frac{|V_1V_2|}{2} \right\} \\ &\leq \frac{1}{\sin\frac{(n-2)\pi}{2n}} \left\{ |\mathscr{L}(s,u)| + \frac{d_{\mathscr{C}(t)}}{2} \right\} \end{aligned} \tag{4.32}$$

This is equivalent to

$$|\mathscr{L}(s,u)| \geq \sin\frac{(n-2)\pi}{2n} |l_{\mathbb{C}}(s,v)| - \frac{e_{\mathscr{C}(t)}}{2} \tag{4.33}$$

As l_G is the shortest Euclidean path from a vertex of $\mathscr{C}(t)$ to t, we have:

$$|l_G| \leq |\mathscr{L}(u,t)| + \frac{e_{\mathscr{C}(t)}}{2} \tag{4.34}$$

Consequently, from (4.33) and (4.34), we obtain:

$$\begin{aligned} |\mathscr{L}(s,t)| \geq Max \Bigg\{ &\sin\frac{(n-2)\pi}{2n} |l_{\mathbb{C}}(s,v)| - \frac{e_{\mathscr{C}(t)}}{2}, \\ &\sin\frac{(n-2)\pi}{2n} |l_{\mathbb{C}}(s,v)| + |l_G| - e_{\mathscr{C}(t)} \Bigg\} \end{aligned} \tag{4.35}$$

The routing path length from s to t can be decomposed into two parts: the first one is the path from s to t' (denoted by $l_{t'}$) and the second one is the path from t' to t (i.e., that can be approximated by $\mathscr{R}(t',t)$). Therefore, we have:

$$|\mathscr{R}(s,t)| = |l_{t'}| + \left|\mathscr{R}(t',t)\right| \tag{4.36}$$

Since $\mathscr{R}(t',t)$ is the image of $\mathscr{B}(t')$ through the homothetic transformation with the scale factor of α_2, according to Lemma 10, we have:

$$\left|\mathscr{R}(t',t)\right| \leq \alpha_2 \left|\mathscr{B}(t')\right| \tag{4.37}$$

Therefore,

$$|\mathscr{R}(s,t)| \leq |l_{t'}| + \alpha_2 \left|\mathscr{B}(t')\right| \tag{4.38}$$

According to (3.85), α_2 can be either 1, or $\frac{(1+\varepsilon)L_0 - |l_{t'}|}{|\mathscr{B}(t')|}$.

1) If $\alpha_2 = \frac{(1+\varepsilon)L_0 - |l_{t'}|}{|\mathscr{B}(t')|}$

Substituting the value of α_2 into (4.38), we obtain: $|\mathscr{R}(s,t)| \leq (1+\varepsilon)L_0$. According to (4.35), $|\mathscr{L}(s,t)| \geq L_0$, thus $|\mathscr{R}(s,t)| \leq (1+\varepsilon)|\mathscr{L}(s,t)|$ ■

2) If $\alpha_2 = 1$

Suppose w is the vertex of $\mathscr{C}(t)$ which is the end point of $\mathscr{B}(s)$. Let us denote θ_2 as the ratio of $\mathscr{B}(t')$ to the length of the shortest path from t' to t. We have the following:

$$|\mathscr{R}(s,t)| \leq |l_{t'}| + \left|\mathscr{B}(t')\right| \leq \theta_2 \{|\mathscr{R}(s,w)| + |\mathscr{L}(w,t)|\}$$

Note that $\mathscr{R}(s,w)$ is the image of $\mathscr{B}(s)$ through the homothetic transformation with the scale factor of α_1, thus $|\mathscr{R}(s,w) \leq \alpha_1 |\mathscr{B}(s)|$ (according to Lemma 10). Moreover, as $|\mathscr{B}(s)| = \theta_1 |l_{\mathbb{C}}(s,v)| \leq \Gamma_1 |l_{\mathbb{C}}(s,v)|$ and $|l_{\mathbb{C}}(s,v)| \leq \frac{1}{\sin\frac{(n-2)\pi}{2n}} \left(|\mathscr{L}(s,u)| + \frac{e_{\mathscr{C}(t)}}{2}\right)$ (see (4.47)), consequently we have:

$$|\mathscr{R}(s,w)| \leq \frac{\alpha_1 \theta_1}{\sin\frac{(n-2)\pi}{2n}} \left(|\mathscr{L}(s,u)| + \frac{e_{\mathscr{C}(t)}}{2}\right) \tag{4.39}$$

We also have:

$$|\mathscr{L}(w,t)| \leq |\mathscr{L}(u,t)| + \frac{p_{\mathscr{C}(t)}}{2} \tag{4.40}$$

From (4.39), (4.39) and (4.40), we obtain:

$$|\mathscr{R}(s,t)| \leq \Gamma_2 \left[\frac{\alpha_1 \theta_1}{\sin\frac{(n-2)\pi}{2n}} \left(|\mathscr{L}(s,u)| + \frac{e_{\mathscr{C}(t)}}{2}\right) + |\mathscr{L}(u,t)| + \frac{p_{\mathscr{C}(t)}}{2}\right] \tag{4.41}$$

$$|\mathscr{R}(s,t)| \leq \theta_2 \left[\frac{\alpha_1 \theta_1}{\sin\frac{(n-2)\pi}{2n}} \left(|\mathscr{L}(s,t)| + \frac{e_{\mathscr{C}(t)}}{2}\right) + \frac{p_{\mathscr{C}(t)}}{2}\right] \tag{4.42}$$

According to (3.85), $\alpha_1 = Max\left\{1, \frac{\sin\frac{(n-2)\pi}{2n}\left\{\frac{(1+\varepsilon)L_0}{\theta_2} - \frac{p_{\mathscr{C}(t)}}{2}\right\}}{\theta_1\left(L_0 + \frac{e_{\mathscr{C}(t)}}{2}\right)}\right\}$.

*) If $\alpha_1 = \frac{\sin\frac{(n-2)\pi}{2n}\left\{\frac{(1+\varepsilon)L_0}{\theta_2} - \frac{p_{\mathscr{C}(t)}}{2}\right\}}{\theta_1\left(L_0 + \frac{e_{\mathscr{C}(t)}}{2}\right)}$:

By substituting the value of α_1 into (4.41), we obtain:

$$|\mathscr{R}(s,t)| \leq (1+\varepsilon)L_0 \tag{4.43}$$

Since $|\mathscr{L}(s,t)| \geq L_0$ (see (4.35)), consequently $|\mathscr{R}(s,t)| \leq (1+\varepsilon)|\mathscr{L}(s,t)|$.

*) If $\alpha_1 = 1$:

(4.42) is equivalent to the following:

$$|\mathscr{R}(s,t)| \leq \theta_2\left[\frac{\theta_1}{\sin\frac{(n-2)\pi}{2n}}\left(|\mathscr{L}(s,t)| + \frac{e_{\mathscr{C}(t)}}{2}\right) + \frac{p_{\mathscr{C}(t)}}{2}\right] \tag{4.44}$$

According to the definition, we have: $\theta_2 \leq \Gamma_2$, thus $\theta_2 \leq Max\left\{1, \frac{L_0(1+\varepsilon)}{\frac{\theta_1}{\sin\frac{(n-2)\pi}{2n}}\left(L_0 + \frac{e_{\mathscr{C}(t)}}{2}\right) + \frac{p_{\mathscr{C}(t)}}{2}}\right\}$.

If $\theta_2 \leq \frac{L_0(1+\varepsilon)}{\frac{\theta_1}{\sin\frac{(n-2)\pi}{2n}}\left(L_0 + \frac{e_{\mathscr{C}(t)}}{2}\right) + \frac{p_{\mathscr{C}(t)}}{2}}$, then $\theta_2 \leq \frac{|\mathscr{L}(s,t)|(1+\varepsilon)}{\frac{\theta_1}{\sin\frac{(n-2)\pi}{2n}}\left(|\mathscr{L}(s,t)| + \frac{e_{\mathscr{C}(t)}}{2}\right) + \frac{p_{\mathscr{C}(t)}}{2}}$ (because $|\mathscr{L}(s,t)| \geq L_0$, according to (4.35)). Therefore, $|\mathscr{R}(s,t)| \leq (1+\varepsilon)\,|\mathscr{L}(s,t)|$.

If $\theta_2 = 1$, then (4.44) is equivalent to the following:

$$|\mathscr{R}(s,t)| \leq \frac{\theta_1}{\sin\frac{(n-2)\pi}{2n}}\left(|\mathscr{L}(s,t)| + \frac{e_{\mathscr{C}(t)}}{2}\right) + \frac{p_{\mathscr{C}(t)}}{2} \tag{4.45}$$

Now, note that $\theta_1 \leq Max\left\{1, \frac{\sin\frac{(n-2)\pi}{2n}\left((1+\varepsilon)d_{\mathscr{C}(t)}(s) - \frac{p_{\mathscr{C}(t)}}{2}\right)}{d_{\mathscr{C}(t)}(s) + \frac{e_{\mathscr{C}(t)}}{2}}\right\}$. Moreover, since s is a t-blind node, it must satisfies condition (3.76), thus:

$$\frac{e_{\mathscr{C}(t)}}{2\sin\frac{(n-2)\pi}{2n}} + \frac{p_{\mathscr{C}(t)}}{2} \leq \left(1+\varepsilon - \frac{1}{\sin\frac{(n-2)\pi}{2n}}\right)d_{\mathscr{C}(t)}(s) \leq \left(1+\varepsilon - \frac{1}{\sin\frac{(n-2)\pi}{2n}}\right)|\mathscr{L}(s,t)| \tag{4.46}$$

Consequently, if $\theta_1 = 1$ then by substituting (4.46) into (4.44) we obtain that $|\mathscr{R}(s,t)| \leq (1+\varepsilon)|\mathscr{L}(s,t)|$.

Otherwise, if $\theta_1 = \frac{\sin\frac{(n-2)\pi}{2n}\left((1+\varepsilon)d_{\mathscr{C}(t)}(s)-\frac{p_{\mathscr{C}(t)}}{2}\right)}{d_{\mathscr{C}(t)}(s)+\frac{e_{\mathscr{C}(t)}}{2}}$, then note that $\frac{\sin\frac{(n-2)\pi}{2n}\left((1+\varepsilon)d_{\mathscr{C}(t)}(s)-\frac{p_{\mathscr{C}(t)}}{2}\right)}{d_{\mathscr{C}(t)}(s)+\frac{e_{\mathscr{C}(t)}}{2}} \leq \frac{\sin\frac{(n-2)\pi}{2n}\left((1+\varepsilon)|\mathscr{L}(s,t)|-\frac{p_{\mathscr{C}(t)}}{2}\right)}{d_{\mathscr{C}(t)}(s)+\frac{|\mathscr{L}(s,t)|}{2}}$. Therefore,

$$\theta_1 \leq \frac{\sin\frac{(n-2)\pi}{2n}\left((1+\varepsilon)\,|\mathscr{L}(s,t)|-\frac{p_{\mathscr{C}(t)}}{2}\right)}{d_{\mathscr{C}(t)}(s)+\frac{|\mathscr{L}(s,t)|}{2}} \tag{4.47}$$

Substituting (4.47) into (4.44), we obtain $|\mathscr{R}(s,t)| \leq (1+\varepsilon)|\mathscr{L}(s,t)|$■

5 Numerical results

5.1 Evaluation metrics

We evaluate and compare the performance of BSMH with three existing protocols, namely GPSR [21] and LVGR [29] and EDGR [134]. As mentioned in Section 1.2, we focus on three design factors: routing path length minimization, control overhead minimization and load balance maximization. In this section, we investigate how does our proposed protocol satisfy these design factors by using the following evaluation metrics:

1. **Metrics regarding routing path length minimization**

   - *Average routing path stretch:* The routing path stretch of a routing path is defined by the ratio of its hop count to that of the theoretical shortest path. The average routing path stretch of a routing protocol is the average value of routing path stretches of all the routing paths determined by the protocol.
   - *Average delay:* We evaluate the average end-to-end delay of all data packets that successfully arrive at the destinations.

2. **Metrics regarding control overhead minimization**

   - *Total amount of control packets:* Control packets are defined as packets that are not data packets. In BSHM and LVGR, control packets consist of packets for exchanging node information (named as HELLO packets), packets for determining hole boundaries (named as HBA packets), and packets for broadcasting hole information (named as HCI packets). In GPSR, the control packet is only HELLO packet. In EDGR, control packets include beacons that periodically broadcast node information (i.e., energy, location, ...), and burst packets that determine the anchors. We measure the total amount (in bytes) of all the control packets which has been transmitted from when the simulation start till the first node dies.

3. **Metrics regarding load balance maximization**

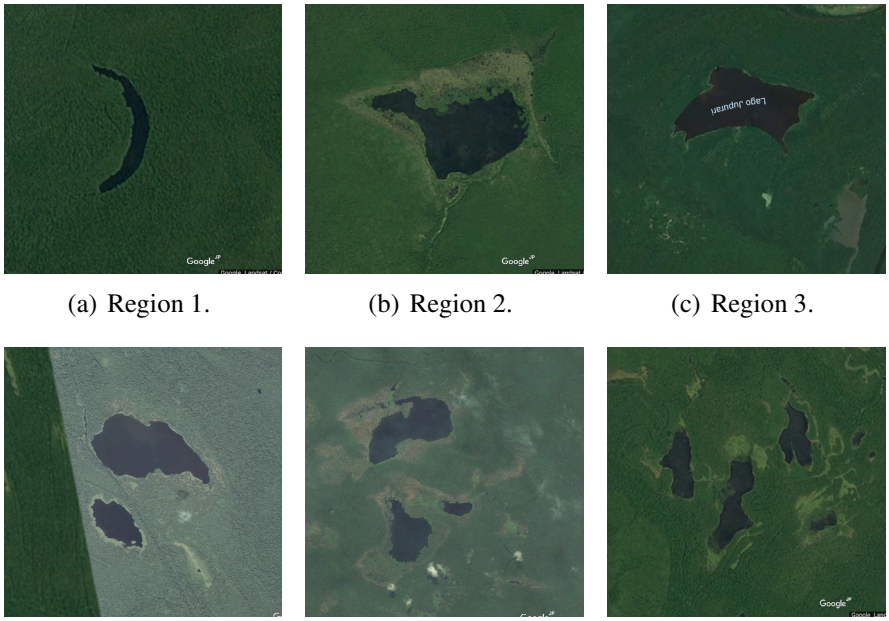
   - *Maximum packet forwarding ratio:* Which indicates the maximum ratio of the number of packets forwarded by a node to the total number of packets sent. Specifically, let p_i be the number of packets forwarded by the i-th node, and p be the total number of packet sent by all source nodes, then the maximum forwarding ratio is defined by $Max\left\{\frac{p_i}{p}\right\}$. In general, the smaller maximum packet forwarding ratio reflects the better load balance.
   - *Network lifetime:* As described in Section 1, in large-scale wireless sensor networks, the death of even only one node may affect the operation of the whole network. Accordingly, the network lifetime should be defined as the time period until the first node dies. Indeed, according to our survey presented in Section 2.3, this definition is the one used the most in the literature. In our experiments, all protocols spend the first $400s$ for the setup phase, and the first data packet is sent after that. Thus, we define the network lifetime as the time period from the first data packet was sent until the first node dies.

Besides the above metrics, the following metrics are also used because they are the common metrics which have been used frequently in evaluating routing protocols in WSNs, according to our survey in Section 2.3.

5. **Other metrics**

- *Delivery ratio:* Which is defined by the ratio of the number of data packets successfully arriving at the destinations to the total number of data packets sent by the sources.
- *Energy consumption per packet:* Which is the ratio of the total energy consumption of all the nodes to the total number of packets successfully delivered.

5.2 Simulation settings

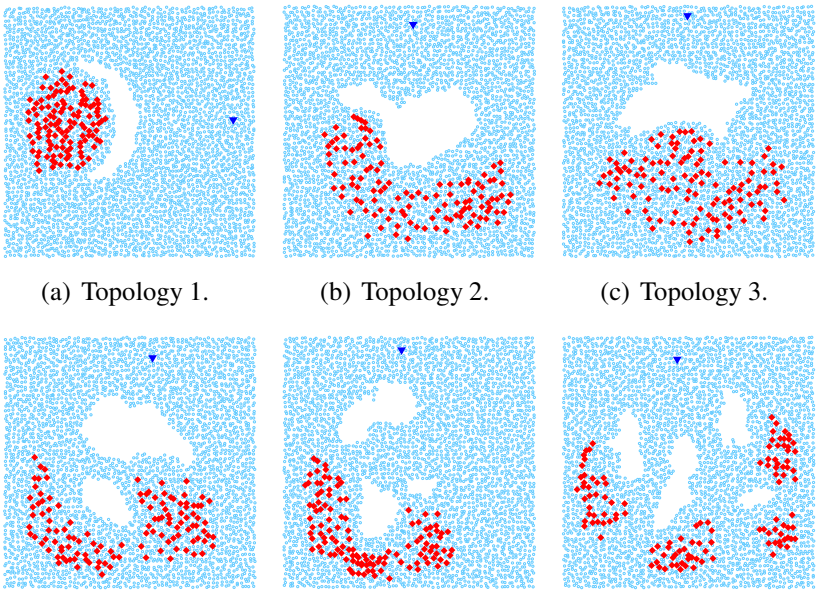


(a) Region 1. (b) Region 2. (c) Region 3.

(d) Region 4. (e) Region 5. (f) Region 6.

Figure 5.1: Real maps obtained from the Google Earth.
The black regions represent the lakes.

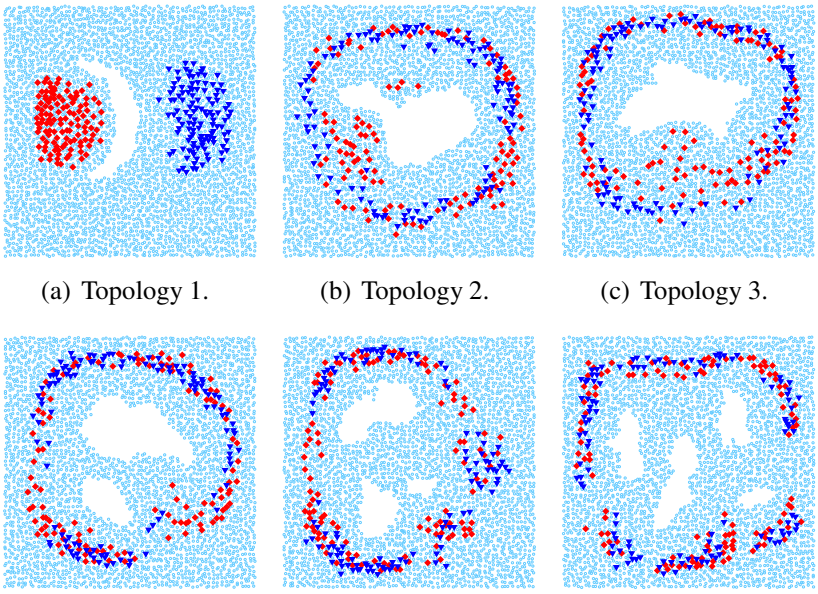
We use various network topologies that are generated based on the real map obtained from the Google Earth as follows. First, we extract from the maps around the Amazon ten regions which contain obstacles, and embed each them into a $1000 \times 1000 m^2$ network area. Then, for each network area, we randomly scatter about 4000 nodes by dividing the network into 63×63 square grid and in each square, we put one sensor in a random position. Finally, we remove all sensor nodes that stay inside the obstacles. Figure 5.1 shows the real images obtained from the Google Earth. From these



(a) Topology 1. (b) Topology 2. (c) Topology 3.

(d) Topology 4. (e) Topology 5. (f) Topology 6.

Figure 5.2: Network topologies for 1-n communication.



(a) Topology 1. (b) Topology 2. (c) Topology 3.

(d) Topology 4. (e) Topology 5. (f) Topology 6.

Figure 5.3: Network topologies for n-n communication.
The blue circles represent the sensors. The red diamonds represent the sources and the blue triangle represents the destination.

Factor	*Value*
MAC type	CSMA/CA
Interface queue model	DropTail
Transmission of radio	TwoRayGround
Antenna type	OmniAntenna
Queue length	50 packets
Transmission range	40 m
Node initial energy	30 J
Node idle power	9.6 mW
Node receive power	45 mW
Node transmit power	88.5 mW
Packet sending interval	10 s
Data packet size	50 bytes

Table 5.1: Parameters of a sensor node.

images, we create six network topologies for 1-n communication (multiple sources vs one destination) (Fig.5.2) and other sixes topologies for n-n communication (multiple sources vs multiple destinations) (Fig.5.3). There are total 120 source-destination pairs which are randomly chosen to intersect with the holes. The simulated time is 2000s. The first 400s is for network setup and the remaining 1600s is for data forwarding. In the network setup phase, the nodes broadcast HELLO packets which contain their own location information. In LVGR and BSMH, the network setup phase is also for locating and disseminating information of the holes and the forbidden areas. The plotted values are the average of 10 simulation runs along with 95% confidence interval. The experiments are conducted using the NS-2 simulator and on a computer with an Intel Core i5-4570 3.2 GHz x 4 CPU and 8 GB of RAM, and running Ubuntu 14.04 64-bit. The MAC protocol, interface queue model, radio model, antenna type, queue length, transmission range are set to the default values of NS2. To study the energy consumption of the protocols, we used the energy model suggested by Shnayder *et al.* [175]. Specifically, the power supply is set to 3V, the currents regrading idle state, receiving state and transmitting state are set to 3.2mA, 15mA and 21.5mA, respectively. Accordingly, the per second energy consumption with respect to idle state, receiving state and transmitting state are 9.6mW, 45mW and 88.5mW, respectively. The initial energy of all node is set to 30J to ensure that the network lifetimes achieved by all protocols are smaller than the simulated time. As the maximum size of a packet following 802.15.4 standard

is 127 bytes, if the packet size exceeds 127 bytes, there will be multiple packets sent. Therefore, the data packet size should be set to less than 127 bytes. We have conducted the experiments with the data packets size of 50 bytes and 100 bytes and found that the trend of the results does not depend on the data packet size. Therefore, in the following, we set the size of all data packets to 50 bytes. The sizes of control packets depend on their types as specified in Table 3.2 and Table 3.3. The fragmentation threshold of the control packets is set to the default value in NS2, i.e., 1000 bytes. When the size of a control packet exceeds the fragmentation threshold, it will be fragmented into multiple packets. Table 5.1 summarizes the parameters used for the sensor nodes.

5.3 Impact of the stretch factor ε

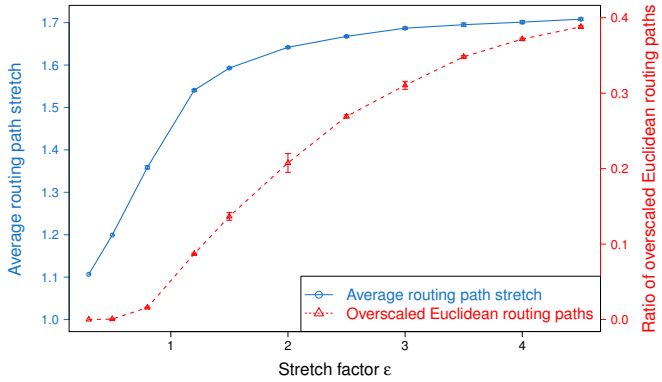


Figure 5.4: Impact of ε on routing path stretch.
The blue lines represent the average stretch of all routing paths, the red lines represent the percentage of Euclidean routing paths that exceed the network boundary.

In this section, we study the impact of the stretch factor, ε, on the performance of BSMH. To do so, we vary the value of ε from 0.3 to 4.5 and observe its impact on the routing path stretch and load balance achieved by BSMH. Although the experiments are conducted on all topologies shown in Fig. 5.2 and 5.3, to facilitate the readability, in what follows, we show only the results on one topology with single hole, i.e., Topology 2 (Fig. 5.2(b)).

Figure 5.4 depicts the impact of ε on the routing path stretch. As shown, the average routing path stretch increases gradually when ε increases. This phenomenon can be

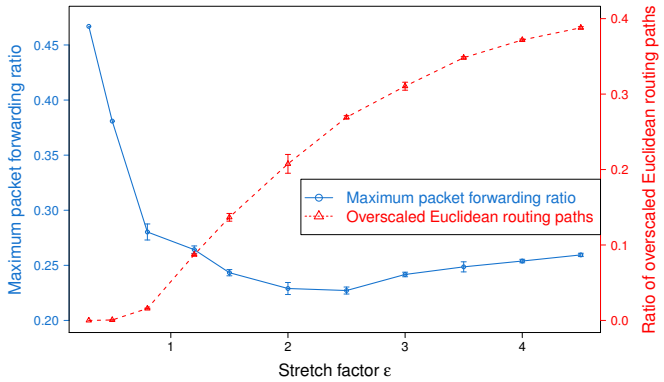


Figure 5.5: Impact of ε on load balance.
The blue lines represent the maximum packet forwarding ratio of sensor nodes, the red lines represent the percentage of Euclidean routing paths that exceed the network boundary.

explained as follows. In BSMH, packets are forwarded along the Euclidean paths that are the image of the base paths through homothetic transformations. The scale factor of the homothetic transformations is proportional to ε. Therefore, the increase of ε leads to the enlargement of the Euclidean paths, and increase the routing path stretch, consequently.

The impact of ε on the maximum packet forwarding ratio is shown in Fig.5.5. As shown, when ε increases from 0.3 to 0.8, almost all of the Euclidean routing paths stay inside the network. Therefore, increasing ε helps to enlarge the number of sensors participating in forwarding packets, thereby reduce the maximum packet forwarding ratio. When ε varies from 0.8 to 2.0, a number of Euclidean routing paths may exceed the network boundary, thus the increase of ε almost can't help to improve the load balance (i.e., the maximum forwarding ratio is almost stable when ε varies in this range). The greater ε, the more Euclidean routing paths exceeding the network boundary. Specifically, when ε reaches 2.5, more than 25% of Euclidean routing paths staying outside of the network boundary. Hence, increasing ε can't enlarge the number of sensors participating in forwarding packets. In the meanwhile, the increase of ε results in the increase of routing path length (see Fig.5.4), thus imposes more traffic load on the sensor nodes. Consequently, when $\varepsilon \geq 2.5$, the increase of ε worsen the load balance.

Figure 5.6 illustrates the impact of ε on the network lifetime. The trend of the network lifetime is similar to that of the load balance but they are not exactly the same.

As shown, the network lifetime gets the peak at $\varepsilon = 1.5$. Before the peak, the increase of ε extends the network lifetime significantly, but beyond the peak, the network lifetime decreases fast when increasing ε. This phenomenon can be explained as follows. When $\varepsilon \leq 1.5$, the increase of ε improve the load balance significantly (this can be seen through the big slope in Fig.5.5), thus the network lifetime is extended. When ε varies from 1.5 to 2.5, the load balance stays almost the same, but the average routing path stretch of all routing paths increases (as shown in Fig.5.4). Note that besides the energy consumed for sending packets, nodes have to spend energy for receiving packets from its neighbors. The increase of average routing path stretch results in the enlargement of the total traffic in the network, which may increases the energy consumed for packet receiving. In consequence, the network lifetime decreases when ε varies from 1.5 to 2.5. Beyond 2.5, increasing of ε not only increases the routing path stretch, but also worsens the load balance, thus the network lifetime is shortened.

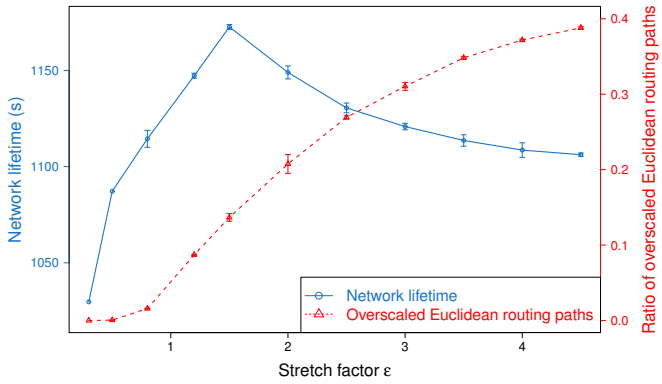


Figure 5.6: Impact of ε on the network lifetime.
The blue lines represent the network lifetime, the red lines represent the percentage of Euclidean routing paths that exceed the network boundary.

Finally, it can be seen that the experimental results show the consistency with the theoretical analysis described in Section 4.2.

5.4 Comparison of BSMH and other benchmarks

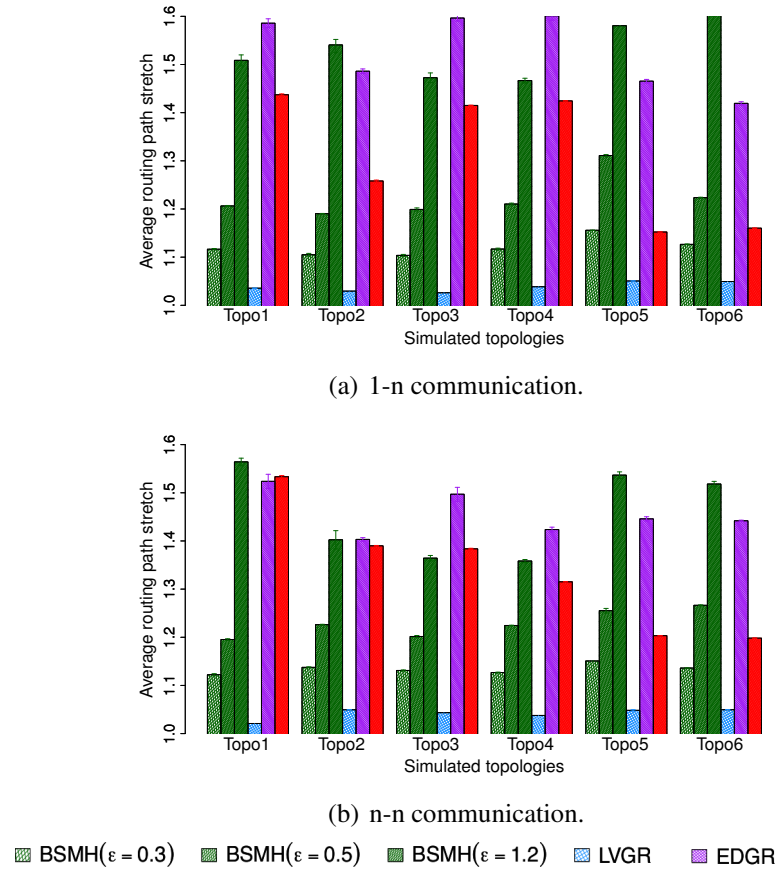
In the following sections, we will compare the performance of our protocol with those of three existing protocols, namely GPSR [21] and LVGR [29] and EDGR [134]. GPSR

uses greedy forwarding as the default mode. When a packet encounters the local minimum phenomenon, it switches from greedy to perimeter mode whereby the packet is forwarded along the hole boundary. In LVGR, the packet is forwarded along a hole-bypassing Euclidean path from the source to the destination. This path is determined by using a visibility graph whose vertices are the vertices of the convex hulls of the holes. In EDGR, the source nodes randomly choose an anchor lists from two candidates. The packet then is forwarded towards the chosen anchor lists, gradually. Moreover, the next hop is heuristically chosen based on not only location information but also the residual energy.

The stretch factor (i.e., ε) of BSMH is set to 0.3, 0.5 and 1.2, the number of vertices of the core polygons is set to 8.

5.4.1 Average routing path stretch

The comparison in terms of the average routing path stretch is shown in Fig.5.7. As shown, for all network topologies, LVGR attains the best performance and followed by BSMH with $\varepsilon = 0.3$. However, as will be shown in Section 5.4.7 and 5.4.6, the shortest routing path stretch of LVGR has to compensate by a poor performance regarding the load balance and the network lifetime. In general, BSMH with small values of ε (i.e., $\varepsilon = 0.3; 0.5$) attains smaller routing path stretch than EDGR and GPSR. However, the increase of ε will increases the routing path stretch, and when $\varepsilon = 1.2$, the routing path stretch of BSMH is higher than that of EDGR and GPSR in 8/12 cases. More specifically, we have the following detail observations. Regarding to the 1-n communication, the average routing path stretch attained by BSMH with $\varepsilon = 0.3$ is always less than 79% that of EDGR in all topologies, and 97% that of GPSR in 5/6 topologies (excepting Topology 5). When ε reaches 0.5, BSMH results in the average routing path stretch which is smaller than 89% that of EDGR in all cases, and smaller than 97% that of GPSR in 4/6 cases (in the other cases, the average routing path stretch caused by BSMH and GPSR are almost similar). Regarding the n-n communication, the average routing path stretch attained by BSMH with $\varepsilon = 0.3$ is less than 81% that of EDGR and 96% that of GPSR in all topologies. In the best cases, the BSMH gains the average routing path stretch begin smaller than 75% that of EDGR and 72% that of GPSR. When $\varepsilon = 0.5$, the average routing path stretch caused by BSMH is always less than 89% that of EDGR, and less than 93% that of GPSR in 4/6 cases (in the other cases, the average



(a) 1-n communication.

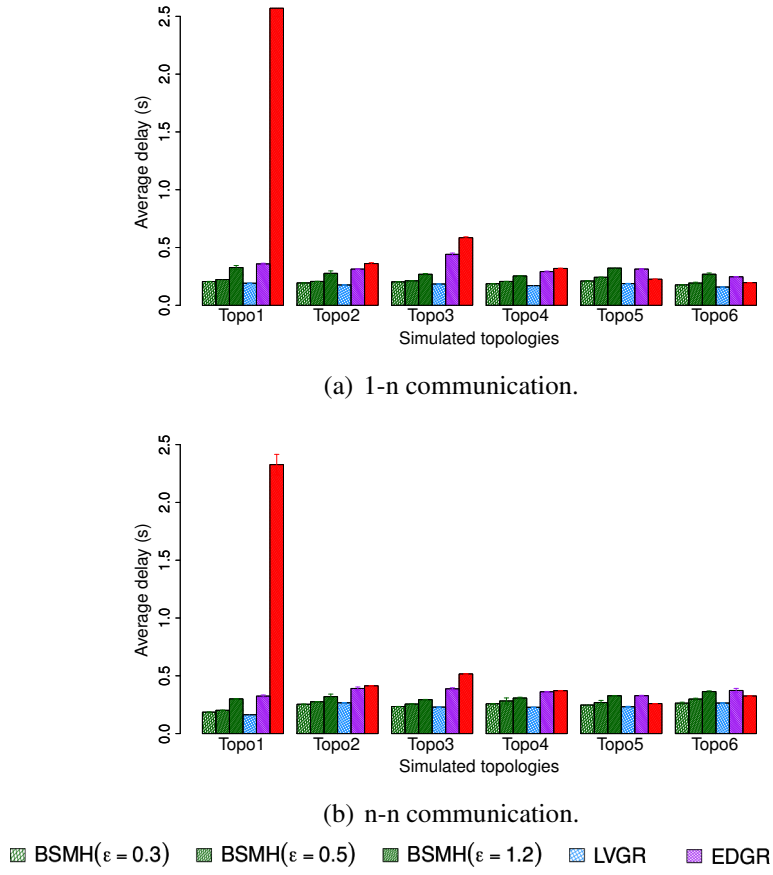
(b) n-n communication.

Figure 5.7: The average routing path stretch of successfully delivered packets.

routing path stretch caused by BSMH and GPSR are almost similar).

5.4.2 Average end-to-end delay

Figure 5.8 depicts the average end-to-end delay of all successfully delivered packets. Since LVGR attains the best performance in terms of the average routing path stretch, it also results in the smallest average end-to-end delay. Following LVGR, BSMH with small values of ε (i.e., $\varepsilon = 0.3$ and 0.5) gains the second-best performance. Moreover, the performance gap between LVGR and BSMH is insignificant when ε is small. For examples, when $\varepsilon = 0.3$, the average end-to-end delay resulted by BSMH is smaller than 1.1 times that caused by LVGR in both 1-n and n-n communication. Similar to the routing path stretch, the end-to-end delay caused by BSMH increases with the increase



(a) 1-n communication.

(b) n-n communication.

Figure 5.8: Average end-to-end delay of successfully delivered packets.

of ε. When ε reaches 1.2, BSMH may cause an end-to-end delay higher than 1.7 time that of LVGR in the worst cases (i.e, Topo 3, 4 in 1-n communication). In comparison with EDGR, it can be seen that the end-to-end delay caused by BSMH with $\varepsilon = 0.3$ and 0.5 are smaller than that of EDGR in all topologies. Specifically, BSMH with $\varepsilon = 0.3$ results in an average routing path stretch smaller than 71% and 75% regrading 1-n and n-n communications, respectively. When ε increases to 0.5, the ratio between the routing path stretch of BSMH to that of EDGR is smaller than 0.78 and 0.81 with respect to 1-n and n-n communications, respectively. The performance of GPSR strongly depends on the network topology. Specifically, its end to end delay is extremely higher than those of the other protocols in some cases such as Topologies 1 and 3 despite of the fact that its average routing path stretch is not much higher than that of the other protocols . We observe that the end-to-end delay is affected not only by the transmission time

between the hops, but also is contributed by the processing time in the hops. While the former one closely related to the routing path routing path stretch, the later one has a strong relationship with the load balance. Hence, there are two main reasons causing the worse performance of GPSR in terms of end-to-end delay. The first reason is due to the long route path (as described in Section 5.4.1), and the second reason is due to the traffic concentration around the hole boundaries (which has been shown in Section 5.4.6).

5.4.3 Control overhead

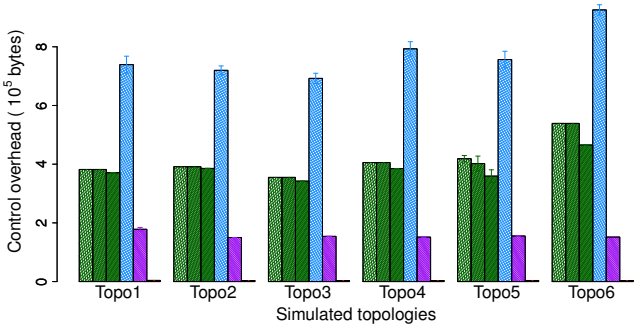


Figure 5.9: Total amount of control packets.

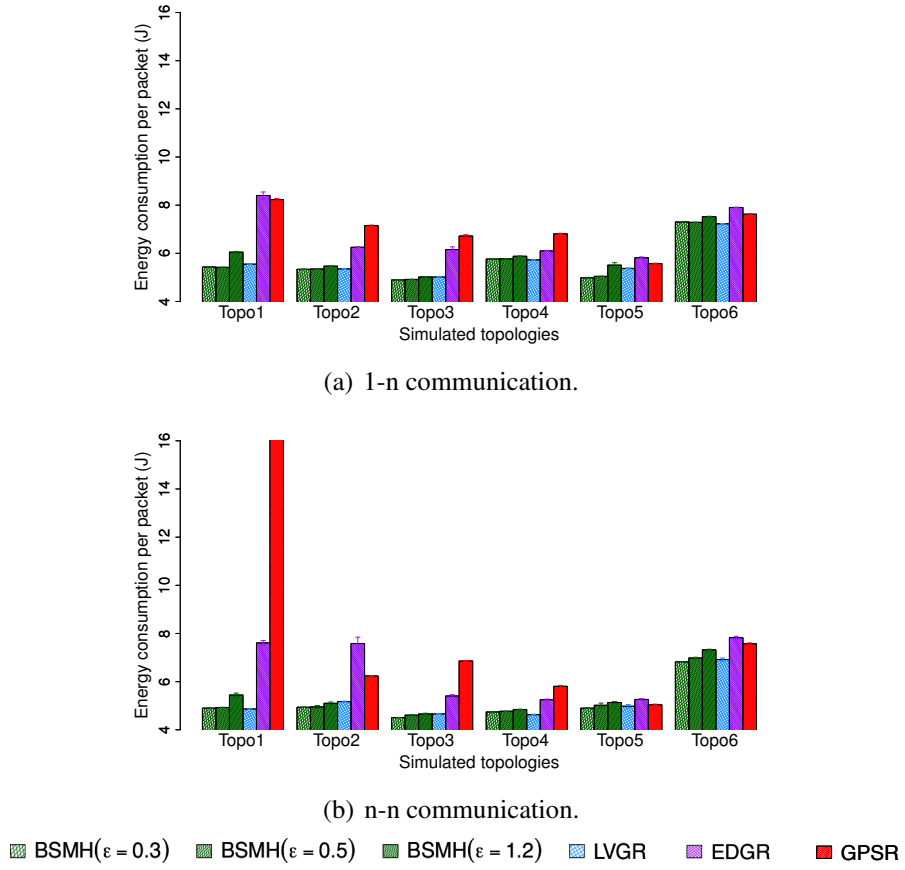
The total amount of control packets required by the protocols is presented in Figure 5.9. As the control overhead does not depend on the communication type, we plot the results regarding 1-n communication.

As expected, GPSR achieves the best performance and its overhead is extremely smaller than those of the others. Although EDGR requires to send the beacons periodically, these beacons are quite lightweight, therefore the overhead caused by EDGR is still smaller than those of BSMH and LVGR. Both BSMH and LVGR need to determine the hole boundaries and disseminate hole information. Moreover, since the control packets used in these two phases are quite heavy, BSMH and LVGR result in the worst performance. However, note that the hole boundary determination and hole information dissemination phases of BSMH and LVGR are conducted only one time while the beaconing of EDGR is conducted periodically, the control overhead of EDGR may higher than those of BSMH and LVGR when increasing the simulation time. By using core polygon instead of the exact hole, BSMH reduces overhead significantly compared with

LVGR. Specifically, the overhead caused by BSMH is always smaller than 53% that caused LVGR.

Another observation is that the overhead of BSMH slightly decreases when increasing the ε. This is due to the fact that the increase of ε will shrink the region for disseminating hole information.

5.4.4 Energy consumption per packet



(a) 1-n communication.

(b) n-n communication.

Figure 5.10: Average energy consumption per packet.

The average energy consumed to delivery one packet is shown in Figure 5.10. In general, BSMH and LVGR outperform EDGR and GPSR. Moreover, the performance of BSMH and LVGR are almost similar. GPSR shows the worst performance and its energy consumption per packet is much higher than those caused by other protocols in

most the cases. EDGR results in a poor performance regarding Topologies 1, 2, 3, 4. This is due to its high delivery ratios and large routing path stretch (see Section 5.4.5 and 5.4.1). In the other topologies, EDGR attains a similar performance with BSMH and LVGR. This phenomenon can be explained as followed: As BSMH and LVGR use shorter routing paths (reflected by better performances in terms of routing path stretch as shown in Section 5.4.1), they may consume less energy than EDGR regarding data transmission. However, as will be shown in Section 5.4.3, the control overhead caused by EDGR is much smaller than those of BSMH and LVGR, thus the energy consumed by transmitting the control packets of EDGR is smaller than those of BSMH and LVGR.

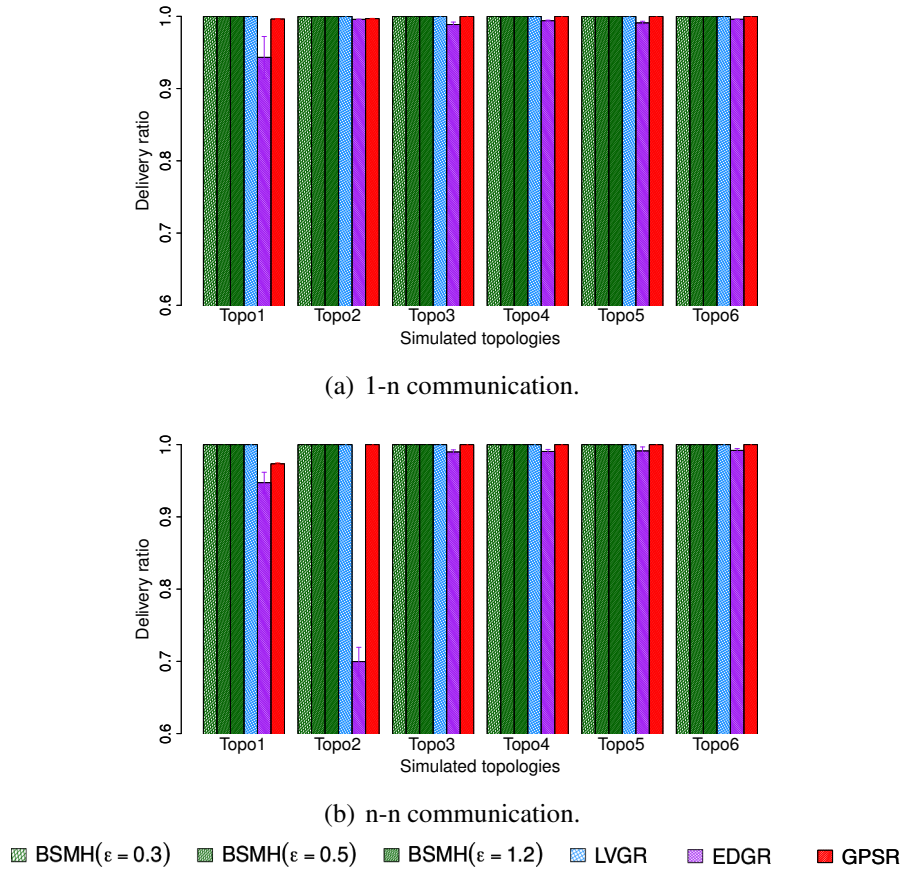
5.4.5 Packet delivery ratio

Figure 5.11 depicts the delivery ratio of the protocols. As shown, BSMH and LVGR achieve the best performance with the delivery ratio approximately equals to 1 in both 1-n and n-n communication types. The reason is because BSMH and LVGR forward all packets along hole bypassing paths. As hole bypassing paths don't intersect the holes' interior, the local minimum problem is solved thoroughly, and all packets are delivered to the destinations. GPSR achieves a good performance in topologies where the sources stay outside of concave areas of the holes, e.g., Topologies 4, 5, 6. When the sources stay inside of concave areas of the holes, e.g., Topologies 1, 2, GPSR suffer from local minimum problem and some packets are dropped. In all the cases, the delivery ratio of GPSR is higher than 97%.

EDGR attains a high delivery ratios in Topologies 3, 4, 5, 6, but its performance degrades severely in Topologies 1, 2. In the worst case, i.e., Topo 2 in n-n communication, EDGR can delivery only less than 70% of packets to the destinations. We observe that the main reason for packet dropping of EDGR is because the burst packets (which are used to determine the anchor list) could not come back to the source before sending data packets.

5.4.6 Maximum packet forwarding ratio

The maximum packet ratios caused by the protocols are shown in Fig.5.12. As shown, our protocol (especially with high values of ε) strongly outperforms the other ones. Specifically, the maximum packet forwarding ratio caused by BSMH for all settings of ε is smaller than those caused by LVGR, EDGR and GPSR in all topologies except-

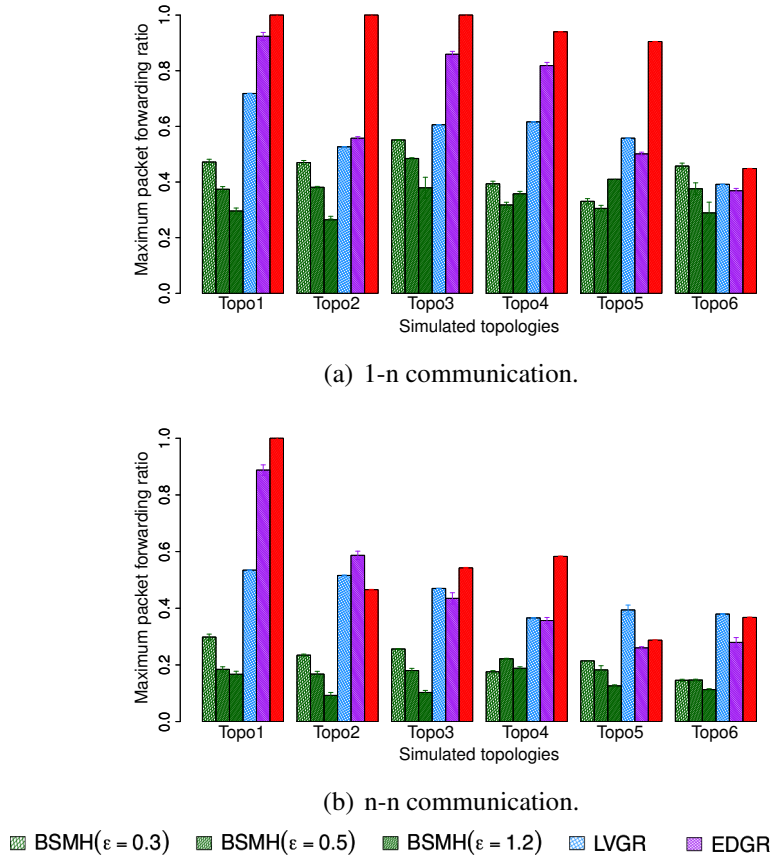


(a) 1-n communication.

(b) n-n communication.

Figure 5.11: The ratio of the number of packets successfully delivered to the total number of packets sent.

ing topology 6 in 1-n communication. EDGR and LVGR attain the second-best and third-best performances in most of the cases, and GPSR shows the highest maximum packet forwarding due to the perimeter routing nature. With $\varepsilon = 1.2$, BSMH results in the maximum packet forwarding ratio which is less than 73% that of LVGR, 81% and EDGR and 64% that of GPSR, in all 1-n communication topologies. In the best cases, the maximum packet forwarding ratio caused by BSMH is smaller 41%, 32% and 27% those of LVGR, EDGR and GPSR, respectively. Regarding the n-n communication, BSMH with ε reduces the maximum packet forwarding ratio by the factors of 0.51, 0.52 and 0.44 compared to LVGR, EDGR, and GPSR, respectively. In the best cases, the maximum packet forwarding ratio resulted by BSMH is smaller than 0.18, 0.16 and 0.17 times those of LVGR, EDGR, and GPSR, respectively. The maximum packet for-



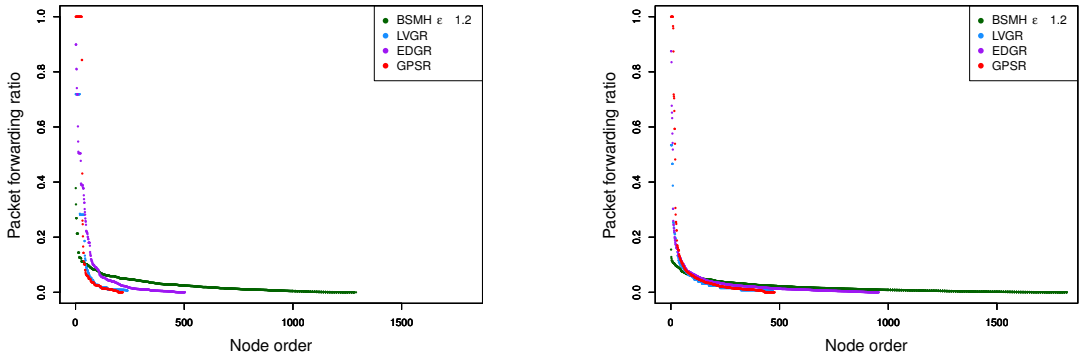
(a) 1-n communication.

(b) n-n communication.

Figure 5.12: The maximum percentage of packets forwarded by a sensor node.

warding ratio caused by BSMH with $\varepsilon = 0.5$ is always smaller than 96% that of LVGR, 68% that of EDGR (in 5/6 cases), and 84% that of GPSR in the 1-n communication. With respect to the n-n communication, the maximum packet forwarding ratio caused by BSMH with $\varepsilon = 0.5$ is always smaller than 60% that of LVGR, 70% that of EDGR and 63% that of GPSR. When reducing ε to 0.3, BSMH outperforms all of the existing protocols regarding all topologies in the n-n communication and 5/6 cases regarding the 1-n communication.

Regarding Topology 6 in 1-n communication, it can be seen that BSMH with $\varepsilon = 0.3$ shows the worst performance and BSMH with $\varepsilon = 0.5$ shows the third-worst performance among all of the protocols. This phenomenon can be explained as follows. Firstly, as this topology consists of many small holes, the diversity of routing paths determined by the other protocols (i.e., LVGR, EDGR, GPSR) is increased. Therefore,



(a) 1-n communication.

(b) n-n communication.

Figure 5.13: Packet forwarding ratio distribution regarding Topology 4.
The x-axis represent the node orders that are sorted by the descending order of the packet forwarding ratio. The y-axis represents the packet forwarding ratios and the x-axis represent the orders of the nodes.

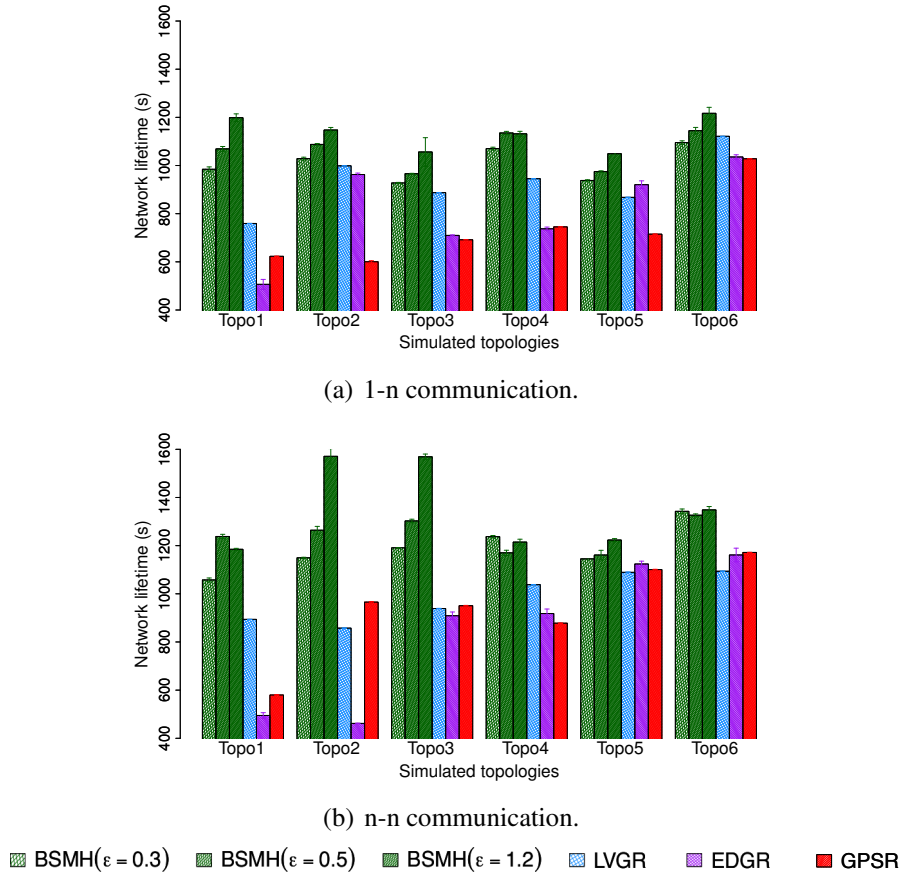
the performance attained by the other protocols in this topology is better than those attained by the same protocols in other topologies. Secondly, as the spaces between the holes in this topology are small, the scale factors used in the homothetic transformations for determining the Euclidean routing paths in BSMH is limited. In other words, the diversity of routing paths determined by BSMH is degraded in Topology 6. Consequently, BSMH with small values of ε can not improve load balance in comparison with other protocols.

Another observation is that the maximum packet forwarding ratios of all protocols in n-n communication are smaller than those in 1-n communication. This can be seen by comparing Fig.5.12(a) and Fig.5.12(b), where the values plotted in Fig.5.12(b) are much smaller than those plotted in Fig.5.12(a). Moreover, the performance gaps between BSMH and the other protocols regarding n-n communication is larger than those regarding 1-n communication. This phenomenon can be explained by the so-called hotspot problem caused by the high traffic load around the destination.

To facilitate the understanding, we also plot the distribution of packet forwarding ratios of all nodes in Fig.5.13. It can be seen that the span of the distributions caused by BSMH in both 1-n and n-n communication are much larger than those caused by the other protocols. This result means that the number of sensors participating in data forwarding in BSMH is much more than those in the other protocols. Moreover, the

height of the distribution attained by BSMH is much smaller than those of the other protocols. This result implies that the maximum number of packets forwarded by a node in BSMH is much smaller than those in the other protocols.

5.4.7 Network lifetime



(a) 1-n communication.

(b) n-n communication.

Figure 5.14: The network lifetime.

The network lifetime achieved by the protocols are shown in Fig.5.14. As shown, the network lifetime achieved by BSMH (especially with high values of ε) is much higher than that of the other protocols. LVGR attains the second-best performance in most of the cases. GPSR and EDGR show the smallest network lifetime. With $\varepsilon = 1.2$, BSMH results in the network lifetime which is higher than 1.1 times that of LVGR, 1.1 times that of EDGR and 1.2 times that of GPSR in 1-n communication, for all

network topologies. Specifically, in the best cases, BSMH with $\varepsilon = 1.2$ can extend the network lifetime by the factors of 1.5, 2.4 an 1.9 compared to LVGR, EDGR and GPSR. Regarding the n-n communication, BSMH with $\varepsilon = 1.2$ achieved the network lifetime higher than 1.1 times those of LVGR, EDGR and GPSR in all topologies. In the best cases, the ratios of the network lifetime achieved by BSMH to those achieved by LVGR, EDGR and GPSR are 1.8, 3.4 and 2.0, respectively.

Generally, the decrease of ε results in the decrease of the network lifetime of BSMH. When reducing ε to 0.3, the network lifetime attained by BSMH is still higher than LVGR, EDGR and GPSR in all topologies excepting Topology 6 in 1-n communication. Specifically, in the best cases, the ratios of the network lifetime of BSHM with $\varepsilon = 0.3$ to those of LVGR, EDGR and GPSR are 1.3, 1.9 and 1.7 in the 1-n communication, and 1.3, 2.5 and 1.8 in the n-n communication, respectively.

Regarding Topology 6 in 1-n communication, it can be seen that BSMH with $\varepsilon = 0.3$ attains a worse performance than LVGR. This phenomenon can be explained similarly as in Section 5.4.6, i.e., the diversity of routing paths determined by LVGR is increased thanks to the presence of many small holes, while the diversity of routing paths determined by BSMH is reduced due to the small spaces between the holes.

Another observation is that the network lifetimes achieved by 1-n communications are much smaller than those achieved by n-n communications. This can be seen by comparing Fig.5.14(a) and Fig.5.14(b). This phenomenon can be explained by the so-called hotspot problem caused by the high traffic load around the destination.

5.4.8 Section summary

In this section, we evaluated the impact of ε on the performance of BSMH. The evaluation results showed the consistency with the theoretical analysis presented in Section 4.2. Specifically, the greater the ε, the higher the routing path stretch. Regarding the load balance and network lifetime, when ε is small, increase of ε improves the load balance and extends the network lifetime. However, when ε is significantly large, as the possibility for a route to reach the network boundary increases, the impact of ε on load balance and network lifetime becomes inversely. The value of ε where the network lifetime attains the peak depends on the network topologies (i.e., the location and the size of the holes, the location of the sources and destinations, etc). According to our experimental results, in general, the peak of network lifetime is attained with ε rang-

ing from 1 to 2.5. In practice, the value of ε should be chosen in accordance with the application requirement. For examples, for delay sensitive applications (e.g., disaster forecast, battlefield surveillance, ...), one should choose small values of ε (e.g., 0.1 to 0.5) to assure the delay constraint. In contrary, for applications that can tolerate latency (e.g., agricultural applications), one should choose large values of ε (e.g. 1.0 to 2.0) to prolong the network lifetime.

We also conducted extensive experiments to compare the performance of BSMH and three benchmarks, GPSR, LVGR and EDGR. The experimental results can be summarized as follows. Regarding the routing path length, BSMH with small values of ε (i.e., $\varepsilon \leq 0.5$) gains a better performance than EDGR and GPSR, but worse than LVGR. Moreover, with $\varepsilon = 0.3$ BSMH attains a similar performance compared with LVGR. With large values of BSMH, the routing path stretch achieved by BSMH may become greater than those of EDGR and GPSR. Considering the control overhead, BSMH reduces the overhead significantly compared to LVGR thanks to the use of core polygons, but its overhead is higher than those of EDGR and GPSR. Regarding the delivery ratio, BSMH can successfully deliver 100% packets. Moreover, BSMH gains the best performance in terms of energy consumption. Specifically, BSMH's performance is similar to that of LVGR and better than EDGR and GPSR. Finally, in terms of load balance, BSMH strongly outperforms all other protocols. Especially, BSMH extends the network lifetime significantly compared to the other protocols.

6 Conclusion

6.1 Dissertation summary

In this dissertation, we studied the routing problem in wireless sensor networks with multiple holes. We aimed at designing a hole bypassing routing protocol, named BSMH, that targets three factors: routing path length, control overhead, and load balance.

Our key ideas are to use elastic forbidden areas and dynamic routing paths. Specifically, for each hole, we create a forbidden area from which all packets are kept to stay away. This forbidden area is an equiangular polygon circumscribing the hole. The number of the vertices of the forbidden area is determined to guarantee the stretch upper bound. For each packet, the source node determines the so-called base path which is a Euclidean path bypassing all the holes. The base path then is magnified using homothetic transformations to obtain the so-called Euclidean routing path. The homothetic transformation centers are chosen randomly to provide the diversity of the routing path, thereby balance the traffic. In the meanwhile, the scale factors are controlled to conserve the stretch upper bound. The vertices of the Euclidean path are inserted into the packet header and act as virtual anchors guiding the packet. The packet then is for-

warded towards the virtual anchors gradually until reaching the destination or reaching an intermediate node having more detail information about the holes. In the latter case, the intermediate node will redetermine the routing path before forwarding the packet to the destination.

We conducted rigorous theoretical analysis on the stretch of BSMH. The analysis results proved that the stretch of all routing paths determined by BSMH does not exceed $(1+\varepsilon)$, where ε is an arbitrary positive number. We also performed a preliminary and intuitive theoretical analysis of the impact of ε on the performance of BSMH. The analysis results revealed that increasing ε will enlarge the routing path length. Regarding the load balance feature, the increase of ε when ε is small can help to balance traffic better. However, when ε is significantly large, increasing ε will worsen the load balance. We also performed extensive simulation experiments to compare the performance of BSMH with state-of-the-art benchmarks. The experimental results showed that BSMH balances the traffic much better than the other protocols. The experimental results also reconfirmed the correctness of our theoretical analysis.

6.2 Discussion

6.2.1 Dynamic hole scenario

In this dissertation, we considered only network topologies with static holes. Our proposed protocol then is designed given an assumption that the holes do not change while the packets are forwarded. However, in practice, the new holes may be created as well as the existing holes may be updated frequently due to external destroy or the energy depletion. To deal with this problem, our proposed protocol can be customized as follows.

First, the change of the holes can be easily detected by periodically conducting the first two phases in our proposed protocol, or by using some existing protocols such as [176]. Specifically, when a node detects the dead of its neighbors, it triggers the hole determination phase to identify whether new holes appear or existing holes change.

Second, when a new hole is detected, or an existing hole is updated, the hole boundary nodes broadcast the updated hole information using the second phase of our proposed protocol. All nodes in the network then update the hole information in their local memory.

Finally, the nodes exploit the updated information to forward data packets using the

third phase in our proposed protocol.

6.2.2 Energy consumption model

In this research, we focused on the theoretical aspect rather than practical deployment. Therefore, in the experiments, we used the default energy model provided by NS2. By which, the energy consumption of a sensor node is comprised of three parts: transmitting energy, receiving energy, and idling energy. The transmitting/receiving energy is the energy consumed for transmitting/receiving packets. This energy is calculated by multiplying the transmission/reception time period and the transmission/reception power. The idling energy is the energy spent while the node is idle (i.e., without transmitting and receiving packets). The idling energy is calculated by multiplying the idling time period with the idle power. The transmission power, reception power and idle power are set to constants for all nodes. Moreover, the initial energy of all nodes is assumed to be identical.

Although the energy consumption model provided in NS2 does not consider the processing energy (i.e., the energy caused by computation), surveys in [19, 20] have shown that processing energy is insignificant compared to transmitting energy. Specifically, the energy for transmitting a single bit is approximately the same as that needed for processing a thousand operations. In the case of BSMH, as shown in Section 4.1.3, the computational complexity for determining the base path candidates, for choosing a base path from the base path, and for determining the next hop are $O\left((2v+(h-2)n)^3\right)$, $O(c+2v+(h-2)n)$, and $O(b)$, respectively, where v is the maximum number of the vertices of a hole, h is the number of the holes, n is the number of the vertices of a core polygons, c is the number of the base path candidates and b is the maximum number of 1-hop neighbors of a sensor node. Normally, v is smaller than hundreds; h, n, c, b are smaller than tens. For examples, let us consider a network consisting of 10 holes, each of which has less than 100 vertices, the number of the vertices of a core polygon is 8, the maximum neighbors of a sensor is 10, and the number of base path candidates is 10, then the processing energy for determining base path candidates is approximately the same as the energy consumption for transmitting 2KB; the processing energy for choosing base path from candidate list and determining the next hop are only less than energy consumption for transmitting 0.03 bit and 0.001 bit, respectively. Note that the base path candidate determination and base path selection are conducted only at the

source node. Moreover, to conserve the energy consumption, the source node may determine the base path candidates only one time and stores the information in the local memory. Accordingly, it can be seen that the processing energy is minor compared to the transmission energy, and thus it can be ignored.

In practice, the energy consumption model of sensor node may be more complicated. For examples, the transmission/reception power may depend on the receiver, and the initial energy may differs among nodes. Although the network lifetime achieved by a routing protocol has a strong relationship with the nodes' energy consumption model, and although we have not conducted experiments under complicated energy consumption models, we performed extensive experiments about energy model-independent factors such as routing path stretch, delay, load balance. Note that these factors have strong a relationship with the energy consumption. Therefore, our experimental results regarding these factors can give us a suggestion on the energy consumption property of our proposed protocol, which is independent of the used energy consumption model.

6.3 Open issues and future work

6.3.1 Beyond the geographical information

In this dissertation, we exploited only the geographical information in making the routing decision. We observe that besides the geographical information, other information including the residual energy, the link quality, the packet drop ratio, ... can also be used to enhance the performance of BSMH. For examples, the next hop may be the neighbor which is not necessarily closest to the destination, but rather it is the one with either the most residual energy, the best link quality or the lowest drop packet ratio. We believe that exploiting more information in making the routing decision can somehow improve the routing performance (i.e., reducing the routing path stretch and increasing the load balance). However, acquiring this extra information may cause a large overhead, instead. Therefore, one must consider the trade-off between the gain and the lost carefully.

6.3.2 Energy consumption optimization in WSNs

Due to the resource limitation of sensor nodes, energy conserving is one of the most critical problem when handling wireless sensor networks. Besides optimizing energy

expenditure in data forwarding, reducing energy can be achieved by jointly using other techniques such as optimizing sensor placement, scheduling sensors to alternatively going to sleep state.

In the literature, many efforts have been devoted to addressing the issue of maximizing the network lifetime while guaranteeing the target coverage and connectivity [177, 178, 179]. A common approach is to determine a family of connected cover sets (i.e., a set of sensors that can monitor all targets and transfer the sensed data to the sink) and the corresponding active time periods so as to maximize their total active time.

Although many works have been done, the algorithms proposed so far suffer from some critical problems including the lack of consideration on the bounded relay hop constraint, the scarce in guaranteeing the performance ratio, and the oversimplification in modeling the network (i.e., almost all of the existing works consider only homogeneous static networks with only one sink, the energy consumption model of sensors is usually oversimplified). Motivated by this insufficiency, one of our future direction will focus on the sensor scheduling problem in general network scenarios such as mobile WSNs, heterogeneous WSNs, ... Moreover, besides the coverage and connectivity, we also consider other constraints such as bounded relay hop.

6.3.3 Rechargeable WSNs

In this research, we targeted un-rechargeable WSNs, where the nodes' batteries cannot be replenished once depleted. Under this assumption, we focused on reducing energy consumption and improving load balance which are key factors in prolonging the network lifetime.

Nowadays, with advances in energy harvesting methods such as solar energy, wind energy, vibration energy, thermal energy, wireless transfered energy, etc, rechargeable WSNs have emerged [180, 181]. In a rechargeable WSN, sensor nodes can replenish their energy by converting ambient energy (solar, wind, etc) from the environment into electricity [182, 183], or by receiving energy through wireless transmission from a special mobile charger [184, 185]. In such scenarios, the network lifetime maximization problem becomes much more complicated. Specifically, besides minimizing nodes' energy consumption and maximizing load balance, there are many other important problems that should be concerned. For examples, in the context of wireless rechargeable WSNs which exploit a mobile charger to power the sensor nodes, one has to consider

the problems of optimizing the traversal strategy of the mobile charger, scheduling the charging tasks.

Although the rechargeable wireless sensor networks are out of this thesis' scope, it could be an interesting direction for future exploration.

6.3.4 Other issues

Besides the issues described above, there are some other open issues that can be studied in the future. For examples, applying reinforcement learning in making the routing decision as well as scheduling the sensor activities; exploiting other routing techniques such as clustered routing, Genetic Algorithm-based routing.

6.4 Concluding remarks

In this dissertation, we have studied two critical problems of large-scale wireless sensors networks with holes, i.e., energy conservation and network lifetime extension, in the context of routing protocol. To conserve the energy and prolong the network lifetime, we aimed at designing a hole bypassing protocol which optimizes three factors: routing path length, control overhead and load balance. To the best of our knowledge, our proposed protocol is the first one addressing and solving thoroughly at the same time all of these three factors. Our key ideas are to use elastic forbidden areas and dynamic routing paths. We believe that these ideas can be used to improve the performance of routing protocols as well as solve other traffic control problems in not only WSNs but also other networks.

Bibliography

[1] S. H. Lee, S. Lee, H. Song, and H. S. Lee, "Wireless sensor network design for tactical military applications : Remote large-scale environments," in *MILCOM 2009 - 2009 IEEE Military Communications Conference*, pp. 1–7, Oct 2009.

[2] E. Onur, C. Ersoy, H. Delic, and L. Akarun, "Surveillance wireless sensor networks: Deployment quality analysis," *IEEE Network*, vol. 21, pp. 48–53, November 2007.

[3] D. Chen, Z. Liu, L. Wang, M. Dou, J. Chen, and H. Li, "Natural disaster monitoring with wireless sensor networks: A case study of data-intensive applications upon low-cost scalable systems," *Mob. Netw. Appl.*, vol. 18, pp. 651–663, Oct. 2013.

[4] M. Erdelj, M. Król, and E. Natalizio, "Wireless sensor networks and multi-uav systems for natural disaster management," *Computer Networks*, vol. 124, pp. 72 – 86, 2017.

[5] G. Han, X. Yang, L. Liu, M. Guizani, and W. Zhang, "A disaster management-oriented path planning for mobile anchor node-based localization in wireless sensor networks," *IEEE Transactions on Emerging Topics in Computing*, pp. 1–1, 2017.

[6] A. S. Bhosle and L. M. Gavhane, "Forest disaster management with wireless sensor network," in *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, pp. 287–289, March 2016.

[7] E. Cayirci and T. Coplu, "Sendrom: Sensor networks for disaster relief operations management," *Wireless Networks*, vol. 13, pp. 409–423, Jun 2007.

[8] A. ur Rehman, A. Z. Abbasi, N. Islam, and Z. A. Shaikh, "A review of wireless sensors and networks' applications in agriculture," *Computer Standards & Interfaces*, vol. 36, no. 2, pp. 263 – 270, 2014.

[9] F. Ingelrest, G. Barrenetxea, G. Schaefer, M. Vetterli, O. Couach, and M. Parlange, "Sensorscope: Application-specific sensor network for environmental monitoring," *ACM Transaction on Sensor Network*, vol. 6, pp. 17:1–17:32, Mar. 2010.

[10] T. Ojha, S. Misra, and N. S. Raghuwanshi, "Wireless sensor networks for agriculture: The state-of-the-art in practice and future challenges," *Computers and Electronics in Agriculture*, vol. 118, pp. 66 – 84, 2015.

[11] X. Shi, L. Fan, J. Xia, Z. Tang, and H. Li, "An environment monitoring system for precise agriculture based on wireless sensor networks," in *2011 Seventh International Conference on Mobile Ad-hoc and Sensor Networks(MSN)*, vol. 00, pp. 28–35, 12 2011.

[12] Y. Zhu, J. Song, and F. Dong, "Applications of wireless sensor network in the agriculture environment monitoring," *Procedia Engineering*, vol. 16, pp. 608 – 614, 2011.

[13] M. M. Baig and H. Gholamhosseini, "Smart health monitoring systems: An overview of design and modeling," *Journal of Medical Systems*, vol. 37, pp. 98 – 98, Jan 2013.

[14] M. R. Yuce, "Implementation of wireless body area networks for healthcare systems," *Sensors and Actuators A: Physical*, vol. 162, no. 1, pp. 116 – 129, 2010.

[15] H. Alemdar and C. Ersoy, "Wireless sensor networks for healthcare: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2688 – 2710, 2010.

[16] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, pp. 393–422, 2002.

[17] N. Ahmed, S. S. Kanhere, and S. Jha, "The holes problem in wireless sensor networks: A survey," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 9, pp. 4–18, Apr. 2005.

[18] Q. Fang, J. Gao, and L. J. Guibas, "Locating and bypassing routing holes in sensor networks," in *Proc. of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM'04*, vol. 4, pp. 2458–2468 vol.4, March 2004.

[19] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava, "Energy-aware wireless microsensor networks," *IEEE Signal Processing Magazine*, vol. 19, pp. 40–50, March 2002.

[20] G. J. Pottie and W. J. Kaiser, "Wireless integrated network sensors," *Magazine Communications of the ACM*, vol. 43, pp. 51–58, May 2000.

[21] B. Karp and H. T. Kung, "Gpsr: Greedy perimeter stateless routing for wireless networks," in *Proc. of the 6th Annual International Conference on Mobile Computing and Networking*, MobiCom '00, (New York, NY, USA), pp. 243–254, ACM, 2000.

[22] F. Kuhn, R. Wattenhofer, and A. Zollinger, "Worst-case optimal and average-case efficient geometric ad-hoc routing," in *Proc. of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '03, (New York, NY, USA), pp. 267–278, 2003.

[23] F. Kuhn, R. Wattenhofer, and A. Zollinger, "Asymptotically optimal geometric mobile ad-hoc routing," in *Proc. of the 6th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, DIALM '02, (New York, NY, USA), pp. 24–33, 2002.

[24] F. Yu *et al.*, "Efficient Hole Detour Scheme for Geographic Routing in Wireless Sensor Networks," in *Proc. of the 67th IEEE Vehicular Technology Conference, VTC'08*, pp. 153–157, 2008.

[25] H.Choo, M.Choi, M.Shon, and D.S.Kim, "Efficient hole bypass routing scheme using observer packets for geographic routing in wireless sensor networks," *ACM SIGAPP Applied Computing Review*, vol. 11, pp. 7–16, 2011.

[26] Y. Tian *et al.*, "Energy-Efficient Data Dissemination Protocol for Detouring Routing Holes in Wireless Sensor Networks," in *Proc. of IEEE International Conference on Communications, ICC'08*, pp. 2322–2326, 2008.

[27] G. Y. Chang, J. P. Sheu, C. W. Chen, S. Y. Wang, and J. F. Huang, "A hole avoiding routing protocol in wireless sensor networks," in *Proc. of 2014 IEEE iThings 2014, GreenCom 2014, and CPSCom 2014*, pp. 75–79, Sept 2014.

[28] M. Won, W. Zhang, and R. Stoleru, "Goal: A parsimonious geographic routing protocol for large scale sensor networks," *Ad Hoc Netw.*, vol. 11, pp. 453–472, Jan. 2013.

[29] M. Won and R. Stoleru, "A low-stretch-guaranteed and lightweight geographic routing protocol for large-scale wireless sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 11, pp. 18:1–18:22, Aug. 2014.

[30] T. Gaber, S. Abdelwahab, M. Elhoseny, and A. E. Hassanien, "Trust-based secure clustering in wsn-based intelligent transportation systems," *Computer Networks*, pp. 151 – 158, 2018.

[31] C. Peckens, C. Porter, and T. Rink, "Wireless sensor networks for long-term monitoring of urban noise," *Sensors*, vol. 18, no. 9, 2018.

[32] Y. Miao, G. Wu, C. Liu, M. S. Hossain, and G. Muhammad, "Green cognitive body sensor network: Architecture, energy harvesting and smart clothing based applications," *IEEE Sensors Journal*, pp. 1–1, 2018.

[33] J. M. Barcelo-Ordinas, J. P. Chanet, K. M. Hou, and J. García-Vidal, "A survey of wireless sensor technologies applied to precision agriculture," in *Precision agriculture '13* (J. V. Stafford, ed.), pp. 801–808, Wageningen Academic Publishers, 2013.

[34] B. Rashid and M. H. Rehmani, "Applications of wireless sensor networks for urban areas: A survey," *Journal of Network and Computer Applications*, vol. 60, pp. 192 – 219, 2016.

[35] A. Sarkar and T. S. Murugan, "Routing protocols for wireless sensor networks: What the literature says?," *Alexandria Engineering Journal*, vol. 55, no. 4, pp. 3173 – 3183, 2016.

[36] Y. Ko and N. H. Vaidya, "Location-aided routing (lar) in mobile ad hoc networks," *Wireless Networks*, vol. 6, pp. 307–321, Sep 2000.

[37] N. Ahmed, S. S. Kanhere, and S. Jha, "The holes problem in wireless sensor networks: A survey," *Mobile Computing and Communications Review*, vol. 9, pp. 4–18, Apr. 2005.

[38] H.-W. Ferng, R. Tendean, and A. Kurniawan, "Energy-efficient routing protocol for wireless sensor networks with static clustering and dynamic structure," *Wireless Personal Communications*, vol. 65, pp. 347–367, Jul 2012.

[39] F. Yu, S. Park, E. Lee, and S. . Kim, "Elastic routing: a novel geographic routing for mobile sinks in wireless sensor networks," *IET Communications*, vol. 4, pp. 716–727, April 2010.

[40] J. Ben-Othman and B. Yahya, "Energy efficient and qos based routing protocol for wireless sensor networks," *Journal of Parallel and Distributed Computing*, vol. 70, no. 8, pp. 849 – 857, 2010.

[41] F. Ren, J. Zhang, T. He, C. Lin, and S. K. D. Ren, "Ebrp: Energy-balanced routing protocol for data gathering in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, pp. 2108–2125, Dec 2011.

[42] L. A. Villas, A. Boukerche, H. S. Ramos, H. A. B. F. de Oliveira, R. B. de Araujo, and A. A. F. Loureiro, "Drina: A lightweight and reliable routing approach for in-network aggregation in wireless sensor networks," *IEEE Transactions on Computers*, vol. 62, pp. 676–689, April 2013.

[43] H. Zhang and H. Shen, "Energy-efficient beaconless geographic routing in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, pp. 881–896, June 2010.

[44] J. Luo and J.-P. Hubaux, "Joint sink mobility and routing to maximize the lifetime of wireless sensor networks: The case of constrained mobility," *IEEE/ACM Trans. Netw.*, vol. 18, pp. 871–884, June 2010.

[45] J. Luo, J. Hu, D. Wu, and R. Li, "Opportunistic routing algorithm for relay node selection in wireless sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 11, pp. 112–121, Feb 2015.

[46] C. Tunca, S. Isik, M. Y. Donmez, and C. Ersoy, "Ring routing: An energy-efficient routing protocol for wireless sensor networks with a mobile sink," *IEEE Transactions on Mobile Computing*, vol. 14, pp. 1947–1960, Sep. 2015.

[47] D. Djenouri and I. Balasingham, "Traffic-differentiation-based modular qos localized routing for wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 10, pp. 797–809, June 2011.

[48] L. Cheng, J. Niu, J. Cao, S. K. Das, and Y. Gu, "Qos aware geographic opportunistic routing in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, pp. 1864–1875, July 2014.

[49] A. E. Abdulla, H. Nishiyama, and N. Kato, "Extending the lifetime of wireless sensor networks: A hybrid routing algorithm," *Computer Communications*, vol. 35, no. 9, pp. 1056 – 1063, 2012.

[50] T. Hu and Y. Fei, "Qelar: A machine-learning-based adaptive routing protocol for energy-efficient and lifetime-extended underwater sensor networks," *IEEE Transactions on Mobile Computing*, vol. 9, pp. 796–809, June 2010.

[51] "Qosnet: An integrated qos network for routing protocols in large scale wireless sensor networks," *Computer Communications*, vol. 33, no. 11, pp. 1334 – 1342, 2010.

[52] C. Petrioli, M. Nati, P. Casari, M. Zorzi, and S. Basagni, "Alba-r: Load-balancing geographic routing around connectivity holes in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, pp. 529–539, March 2014.

[53] F. Liu, C. Tsui, and Y. J. Zhang, "Joint routing and sleep scheduling for lifetime maximization of wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 9, pp. 2258–2267, July 2010.

[54] S. Ruhrup, H. Kalosha, A. Nayak, and I. Stojmenovic, "Message-efficient beaconless georouting with guaranteed delivery in wireless sensor, ad hoc, and actuator networks," *IEEE/ACM Transactions on Networking*, vol. 18, pp. 95–108, Feb 2010.

[55] J. Zhang, Q. Wu, F. Ren, T. He, and C. Lin, "Effective data aggregation supported by dynamic routing in wireless sensor networks," in *2010 IEEE International Conference on Communications*, pp. 1–6, May 2010.

[56] V. N. Soares, J. J. Rodrigues, and F. Farahmand, "Geospray: A geographic routing protocol for vehicular delay-tolerant networks," *Information Fusion*, vol. 15, pp. 102 – 113, 2014. Special Issue: Resource Constrained Networks.

[57] X. Li, J. Yang, A. Nayak, and I. Stojmenovic, "Localized geographic routing to a mobile sink with guaranteed delivery in sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 30, pp. 1719–1729, October 2012.

[58] M. Liu, S. Xu, and S. Sun, "An agent-assisted qos-based routing algorithm for wireless sensor networks," *Journal of Network and Computer Applications*, vol. 35, no. 1, pp. 29 – 36, 2012. Collaborative Computing and Applications.

[59] T. Amgoth and P. K. Jana, "Energy-aware routing algorithm for wireless sensor networks," *Computers & Electrical Engineering*, vol. 41, pp. 357 – 367, 2015.

[60] U. Lee, P. Wang, Y. Noh, L. F. M. Vieira, M. Gerla, and J. Cui, "Pressure routing for underwater sensor networks," in *2010 Proceedings IEEE INFOCOM*, pp. 1–9, March 2010.

[61] Y. Chen and Y. Lin, "Mobicast routing protocol for underwater sensor networks," *IEEE Sensors Journal*, vol. 13, pp. 737–749, Feb 2013.

[62] Y. Yang, C. Zhong, Y. Sun, and J. Yang, "Network coding based reliable disjoint and braided multipath routing for sensor networks," *Journal of Network and Computer Applications*, vol. 33, no. 4, pp. 422 – 432, 2010.

[63] F. Ren, J. Zhang, Y. Wu, T. He, C. Chen, and C. Lin, "Attribute-aware data aggregation using potential-based dynamic routing in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, pp. 881–892, May 2013.

[64] B. Zeng and Y. Dong, "An improved harmony search based energy-efficient routing algorithm for wireless sensor networks," *Applied Soft Computing*, vol. 41, pp. 135 – 147, 2016.

[65] C. Hung, K. C. Lin, C. Hsu, C. Chou, and C. Tu, "On enhancing network-lifetime using opportunistic routing in wireless sensor networks," in *2010 Proceedings of 19th International Conference on Computer Communications and Networks*, pp. 1–6, Aug 2010.

[66] C. Ok, S. Lee, P. Mitra, and S. Kumara, "Distributed routing in wireless sensor networks using energy welfare metric," *Information Sciences*, vol. 180, no. 9, pp. 1656 – 1670, 2010.

[67] Y. Noh, U. Lee, P. Wang, B. S. C. Choi, and M. Gerla, "Vapr: Void-aware pressure routing for underwater sensor networks," *IEEE Transactions on Mobile Computing*, vol. 12, pp. 895–908, May 2013.

[68] M. Hammoudeh and R. Newman, "Adaptive routing in wireless sensor networks: Qos optimisation for enhanced application performance," *Information Fusion*, vol. 22, pp. 3 – 15, 2015.

[69] O. Banimelhem and S. Khasawneh, "Gmcar: Grid-based multipath with congestion avoidance routing protocol in wireless sensor networks," *Ad Hoc Networks*, vol. 10, no. 7, pp. 1346 – 1361, 2012.

[70] A. Ahmad, N. Javaid, Z. A. Khan, U. Qasim, and T. A. Alghamdi, "$(ach)^2$: Routing scheme to maximize lifetime and throughput of wireless sensor networks," *IEEE Sensors Journal*, vol. 14, pp. 3516–3532, Oct 2014.

[71] J. Chen, Z. Li, and Y.-H. Kuo, "A centralized balance clustering routing protocol for wireless sensor network," *Wireless Personal Communications*, vol. 72, pp. 623–634, Sep 2013.

[72] S. Sharma and S. K. Jena, "Cluster based multipath routing protocol for wireless sensor networks," *SIGCOMM Comput. Commun. Rev.*, vol. 45, pp. 14–20, Apr. 2015.

[73] J. A. Sanchez, R. Marin-Perez, and P. M. Ruiz, "Beacon-less geographic multicast routing in a real-world wireless sensor network testbed," *Wireless Networks*, vol. 18, pp. 565–578, Jul 2012.

[74] R. W. L. Coutinho, A. Boukerche, L. F. M. Vieira, and A. A. F. Loureiro, "Geographic and opportunistic routing for underwater sensor networks," *IEEE Transactions on Computers*, vol. 65, pp. 548–561, Feb 2016.

[75] M. Tao, D. Lu, and J. Yang, "An adaptive energy-aware multi-path routing protocol with load balance for wireless sensor networks," *Wireless Personal Communications*, vol. 63, pp. 823–846, Apr 2012.

[76] T. Hayes and F. H. Ali, "Location aware sensor routing protocol for mobile wireless sensor networks," *IET Wireless Sensor Systems*, vol. 6, no. 2, pp. 49–57, 2016.

[77] Y. Li and J. Ren, "Source-location privacy through dynamic routing in wireless sensor networks," in *2010 Proceedings IEEE INFOCOM*, pp. 1–9, March 2010.

[78] M. Azharuddin, P. Kuila, and P. K. Jana, "Energy efficient fault tolerant clustering and routing algorithms for wireless sensor networks," *Computers & Electrical Engineering*, vol. 41, pp. 177 – 190, 2015.

[79] A. Awad, R. German, and F. Dressler, "Exploiting virtual coordinates for improved routing performance in sensor networks," *IEEE Transactions on Mobile Computing*, vol. 10, pp. 1214–1226, Sep. 2011.

[80] M. Ayaz, A. Abdullah, I. Faye, and Y. Batira, "An efficient dynamic addressing based routing protocol for underwater wireless sensor networks," *Computer Communications*, vol. 35, no. 4, pp. 475 – 486, 2012.

[81] E. Zeydan, D. Kivanc, C. Comaniciu, and U. Tureli, "Energy-efficient routing for correlated data in wireless sensor networks," *Ad Hoc Networks*, vol. 10, no. 6, pp. 962 – 975, 2012.

[82] P. Huang, C. Wang, and L. Xiao, "Improving end-to-end routing performance of greedy forwarding in sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, pp. 556–563, March 2012.

[83] X. Chen, Z. Dai, W. Li, Y. Hu, J. Wu, H. Shi, and S. Lu, "Prohet: A probabilistic routing protocol with assured delivery rate in wireless heterogeneous sensor networks," *IEEE Transactions on Wireless Communications*, vol. 12, pp. 1524–1531, April 2013.

[84] Y.-F. Hu, Y.-S. Ding, L.-H. Ren, K.-R. Hao, and H. Han, "An endocrine cooperative particle swarm optimization algorithm for routing recovery problem of wireless sensor networks with multiple mobile sinks," *Information Sciences*, vol. 300, pp. 100 – 113, 2015.

[85] T. Hayes and F. Ali, "Robust ad-hoc sensor routing (raser) protocol for mobile wireless sensor networks," *Ad Hoc Networks*, vol. 50, pp. 128 – 144, 2016.

[86] X. Wu, Y. Xiong, W. Huang, H. Shen, and M. Li, "An efficient compressive data gathering routing scheme for large-scale wireless sensor networks," *Computers & Electrical Engineering*, vol. 39, no. 6, pp. 1935 – 1946, 2013. Special Issue on Wireless Systems: Modeling, Monitoring, Transmission, Performance Evaluation and Optimization.

[87] A. Mostefaoui, M. Melkemi, and A. Boukerche, "Localized routing approach to bypass holes in wireless sensor networks," *IEEE Transactions on Computers*, vol. 63, pp. 3053–3065, Dec 2014.

[88] G. Tan and A. Kermarrec, "Greedy geographic routing in large-scale sensor networks: A minimum network decomposition approach," *IEEE/ACM Transactions on Networking*, vol. 20, pp. 864–877, June 2012.

[89] J. Jung, S. Park, E. Lee, S. Oh, and S. Kim, "Omlrp: Multi-hop information based real-time routing protocol in wireless sensor networks," in *2010 IEEE Wireless Communication and Networking Conference*, pp. 1–6, April 2010.

[90] H. Kumar Deva Sarma, K. Avijit, and R. Mall, "e^2r^2: Energy-efficient and reliable routing for mobile wireless sensor networks," vol. 10, pp. 1–6, 06 2010.

[91] A. S. K. Mammu, A. Sharma, U. Hernandez-Jayo, and N. Sainz, "A novel cluster-based energy efficient routing in wireless sensor networks," in *2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA)*, pp. 41–47, March 2013.

[92] H. Alwan and A. Agarwal, "Multi-objective qos routing for wireless sensor networks," in *2013 International Conference on Computing, Networking and Communications (ICNC)*, pp. 1074–1079, Jan 2013.

[93] J. Wang, Y. Zhang, J. Wang, Y. Ma, and M. Chen, "Pwdgr: Pair-wise directional geographical routing based on wireless sensor network," *IEEE Internet of Things Journal*, vol. 2, pp. 14–22, Feb 2015.

[94] T. Lee, C. Qiao, M. Demirbas, and J. Xu, "Abc: A simple geographic forwarding scheme capable of bypassing routing holes in sensor networks," *Ad Hoc Networks*, vol. 8, no. 4, pp. 361 – 377, 2010.

[95] H. Huang, G. Hu, F. Yu, and Z. Zhang, "Energy-aware interference-sensitive geographic routing in wireless sensor networks," *IET Communications*, vol. 5, pp. 2692–2702, Dec 2011.

[96] Y. Xu, F. Ren, T. He, C. Lin, C. Chen, and S. K. Das, "Real-time routing in wireless sensor networks: A potential field approach," *ACM Trans. Sen. Netw.*, vol. 9, pp. 35:1–35:24, June 2013.

[97] K. Saleem and N. Fisal, "Enhanced ant colony algorithm for self-optimized data assured routing in wireless sensor networks," in *2012 18th IEEE International Conference on Networks (ICON)*, pp. 422–427, Dec 2012.

[98] M. A. Razzaque, M. H. U. Ahmed, C. S. Hong, and S. Lee, "Qos-aware distributed adaptive cooperative routing in wireless sensor networks," *Ad Hoc Networks*, vol. 19, pp. 28 – 42, 2014.

[99] A. Laouid, A. Dahmani, A. Bounceur, R. Euler, F. Lalem, and A. Tari, "A distributed multi-path routing algorithm to balance energy consumption in wireless sensor networks," *Ad Hoc Networks*, vol. 64, pp. 53 – 64, 2017.

[100] T. Amgoth and P. K. Jana, "Energy and coverage-aware routing algorithm for wireless sensor networks," *Wireless Personal Communications*, vol. 81, pp. 531–545, Mar 2015.

[101] R. E. Mohemed, A. I. Saleh, M. Abdelrazzak, and A. S. Samra, "Energy-efficient routing protocols for solving energy hole problem in wireless sensor networks," *Computer Networks*, vol. 114, pp. 51 – 66, 2017.

[102] A. Liu, Z. Zheng, C. Zhang, Z. Chen, and X. Shen, "Secure and energy-efficient disjoint multipath routing for wsns," *IEEE Transactions on Vehicular Technology*, vol. 61, pp. 3255–3265, Sep. 2012.

[103] D. Zhang and E. Dong, "A virtual coordinate-based bypassing void routing for wireless sensor networks," *IEEE Sensors Journal*, vol. 15, pp. 3853–3862, July 2015.

[104] H. Alwan and A. Agarwal, "Multi-objective reliable multipath routing for wireless sensor networks," in *2010 IEEE Globecom Workshops*, pp. 1227–1231, Dec 2010.

[105] Y. Yao, Q. Cao, and A. V. Vasilakos, "Edal: An energy-efficient, delay-aware, and lifetime-balancing data collection protocol for heterogeneous wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 23, pp. 810–823, June 2015.

[106] S. Yessad, L. Bouallouche-Medjkoune, and D. Aïssani, "A cross-layer routing protocol for balancing energy consumption in wireless sensor networks," *Wireless Personal Communications*, vol. 81, pp. 1303–1320, Apr 2015.

[107] L. B. Ribeiro and M. F. de Castro, "Bio4sel: A bio-inspired routing algorithm for sensor network lifetime optimization," in *2010 17th International Conference on Telecommunications*, pp. 728–734, April 2010.

[108] E. A. Khalil and B. A. Attea, "Stable-aware evolutionary routing protocol for wireless sensor networks," *Wireless Personal Communications*, vol. 69, pp. 1799–1817, Apr 2013.

[109] B. Yahya and J. Ben-Othman, "Relax: An energy efficient multipath routing protocol for wireless sensor networks," in *2010 IEEE International Conference on Communications*, pp. 1–6, May 2010.

[110] X. Meng, X. Shi, Z. Wang, S. Wu, and C. Li, "A grid-based reliable routing protocol for wireless sensor networks with randomly distributed clusters," *Ad Hoc Networks*, vol. 51, pp. 47 – 61, 2016.

[111] R. W. Coutinho, A. Boukerche, L. F. Vieira, and A. A. Loureiro, "Transmission power control-based opportunistic routing for wireless sensor networks," in *Proceedings of the 17th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, MSWiM '14, pp. 219–226, 2014.

[112] Y. Sun, Q. Jiang, and M. Singhal, "An edge-constrained localized delaunay graph for geographic routing in mobile ad hoc and sensor networks," *IEEE Transactions on Mobile Computing*, vol. 9, pp. 479–490, April 2010.

[113] J. Rao and A. O. Fapojuwo, "A battery aware distributed clustering and routing protocol for wireless sensor networks," in *2012 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1538–1543, April 2012.

[114] S. Cakici, I. Erturk, S. Atmaca, and A. Karahan, "A novel cross-layer routing protocol for increasing packet transfer reliability in mobile sensor networks," *Wireless Personal Communications*, vol. 77, pp. 2235–2254, Aug 2014.

[115] J. Carnley, B. Sun, and S. K. Makki, "Torp: Tinyos opportunistic routing protocol for wireless sensor networks," in *2011 IEEE Consumer Communications and Networking Conference (CCNC)*, pp. 111–115, Jan 2011.

[116] L. Pradittasnee, S. Camtepe, and Y. Tian, "Efficient route update and maintenance for reliable routing in large-scale sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 13, pp. 144–156, Feb 2017.

[117] M. Koulali, A. Kobbane, M. E. Koutbi, and M. Azizi, "Qdgrp : A hybrid qos distributed genetic routing protocol for wireless sensor networks," in *2012 International Conference on Multimedia Computing and Systems*, pp. 47–52, May 2012.

[118] A. Pal and A. Nasipuri, "Pcor: A joint power control and routing scheme for rechargeable sensor networks," in *2014 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 2230–2235, April 2014.

[119] Y. Jiang, W. Shi, X. Wang, and H. Li, "A distributed routing for wireless sensor networks with mobile sink based on the greedy embedding," *Ad Hoc Networks*, vol. 20, pp. 150 – 162, 2014.

[120] M.-T. Sun, K. Sakai, B. R. Hamilton, W.-S. Ku, and X. Ma, "G-star: Geometric stateless routing for 3-d wireless sensor networks," *Ad Hoc Networks*, vol. 9, no. 3, pp. 341 – 354, 2011.

[121] S. P. Fucai Yu and G. Hu, "Hole plastic scheme for geographic routing in wireless sensor networks," in *Proc. of the 15th IEEE International Conference on Communications, ICC'15, London, United Kingdom, June 8-12, 2015*, pp. 6444–6449, 2015.

[122] M. I. Akbas and D. Turgut, "Lightweight routing with qos support in wireless sensor and actor networks," in *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*, pp. 1–5, Dec 2010.

[123] R. W. Pazzi, D. Zhang, A. Boukerche, and L. Mokdad, "E-trail: Energy-efficient trail-based data dissemination protocol for wireless sensor networks with mobile sinks," in *2011 IEEE International Conference on Communications (ICC)*, pp. 1–5, June 2011.

[124] A. E. Kostin, Y. Fanaeian, and H. Al-Wattar, "Anycast tree-based routing in mobile wireless sensor networks with multiple sinks," *Wireless Networks*, vol. 22, pp. 579–598, Feb 2016.

[125] N. Chakchouk, B. Hamdaoui, and M. Frikha, "Wcds-dcr: An energy-efficient data-centric routing scheme for wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 12, pp. 195–205, 02 2012.

[126] J. Hao, Z. Yao, K. Huang, B. Zhang, and C. Li, "An energy-efficient routing protocol with controllable expected delay in duty-cycled wireless sensor networks," in *2013 IEEE International Conference on Communications (ICC)*, pp. 6215–6219, June 2013.

[127] Y. Song, C. Gui, X. Lu, H. Chen, and B. Sun, "A genetic algorithm for energy-efficient based multipath routing in wireless sensor networks," *Wireless Personal Communications*, vol. 85, pp. 2055–2066, Dec 2015.

[128] R.-S. Ko, "A distributed routing algorithm for sensor networks derived from macroscopic models," *Comput. Netw.*, vol. 55, pp. 314–329, Jan. 2011.

[129] X. Fan, F. Du, and W. Wei, "A geography-based void-bypassing routing protocol for wireless sensor network," *Wireless Personal Communications*, vol. 90, pp. 259–280, Sep 2016.

[130] M. Won and R. Stoleru, "A hybrid multicast routing for large scale sensor networks with holes," *IEEE Transactions on Computers*, vol. 64, pp. 3362–3375, Dec 2015.

[131] H. A. de Oliveira, A. Boukerche, D. L. Guidoni, E. F. Nakamura, R. A. Mini, and A. A. Loureiro, "An enhanced location-free greedy forward algorithm with hole bypass capability in wireless sensor networks," *Journal of Parallel and Distributed Computing*, vol. 77, pp. 1 – 10, 2015.

[132] F. Zhou, G. Trajcevski, R. Tamassia, B. Avci, A. Khokhar, and P. Scheuermann, "Bypassing holes in sensor networks: Load-balance vs. latency," *Ad Hoc Networks*, vol. 61, pp. 16 – 32, 2017.

[133] H. Huang, H. Yin, G. Min, X. Zhang, W. Zhu, and Y. Wu, "Coordinate-assisted routing approach to bypass routing holes in wireless sensor networks," *IEEE Communications Magazine*, vol. 55, pp. 180–185, July 2017.

[134] H. Huang, H. Yin, G. Min, J. Zhang, Y. Wu, and X. Zhang, "Energy-aware dual-path geographic routing to bypass routing holes in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 17, pp. 1339–1352, June 2018.

[135] "Geographic routing and hole bypass using long range sinks for wireless sensor networks," *Ad Hoc Networks*, vol. 67, pp. 1–10, Dec. 2017.

[136] S. Kim, T. Yang, C. Kim, H. Cho, and S. Kim, "Dynamic anchors based void avoidance scheme for real-time application in wsns," in *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*, pp. 1–5, Sep. 2017.

[137] J. Yang and Z. Fei, "Hdar: Hole detection and adaptive geographic routing for ad hoc networks," in *2010 Proceedings of 19th International Conference on Computer Communications and Networks*, pp. 1–6, Aug 2010.

[138] J. Yang, Z. Fei, and J. Shen, "Hole detection and shape-free representation and double landmarks based geographic routing in wireless sensor networks," *Digital Communications and Networks*, vol. 1, no. 1, pp. 75 – 83, 2015.

[139] O. Yilmaz, O. Dagdeviren, and K. Erciyes, "Localization-free and energy-efficient hole bypassing techniques for fault-tolerant sensor networks," *Journal of Network and Computer Applications*, vol. 40, pp. 164 – 178, 2014.

[140] J. Gao, F. Li, and Y. Wang, "Distributed load balancing mechanism for detouring routing holes in sensor networks," in *2012 IEEE Vehicular Technology Conference (VTC Fall)*, pp. 1–5, Sep. 2012.

[141] P. Kar and S. Misra, "Detouring dynamic routing holes in stationary wireless sensor networks in the presence of temporarily misbehaving nodes," *International Journal of Communication Systems*, vol. 30, no. 4, 2017.

[142] S. Kim, S. Oh, H. Park, J. Lee, and S. Kim, "Disjoint multipath scheme with hole detouring strategy in wireless sensor networks," in *2011 IEEE Vehicular Technology Conference (VTC Fall)*, pp. 1–5, Sep. 2011.

[143] H. P. Gupta, S. V. Rao, A. K. Yadav, and T. Dutta, "Geographic routing in clustered wireless sensor networks among obstacles," *IEEE Sensors Journal*, vol. 15, pp. 2984–2992, May 2015.

[144] D. Zhang, E. Dong, Z. Cao, P. Qian, and J. Xu, "An efficient bypassing void routing protocol based on virtual coordinate mapping for wsn," in *The 20th Asia-Pacific Conference on Communication (APCC2014)*, pp. 94–99, Oct 2014.

[145] K. Jung, E. Lee, S. Oh, Y. Yim, and S. Kim, "Localized disjoint multipath routing protocol in irregular wireless sensor networks," in *2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pp. 2454–2458, Sep. 2013.

[146] S. Kim, C. Kim, H. Cho, Y. Yim, and S. Kim, "Void avoidance scheme for real-time data dissemination in irregular wireless sensor networks," in *2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)*, pp. 438–443, March 2016.

[147] H. Huang, J. Zhang, X. Zhang, B. Yi, Q. Fan, and F. Li, "Emgr: Energy-efficient multicast geographic routing in wireless sensor networks," *Computer Networks*, vol. 129, pp. 51 – 63, 2017.

[148] D. Zhang and E. Dong, "A bypassing void routing combining of geographic and virtual coordinate information for wsn," in *2015 22nd International Conference on Telecommunications (ICT)*, pp. 118–122, April 2015.

[149] D. Zhang and E. Dong, "An efficient bypassing void routing protocol based on virtual coordinate for wsns," *IEEE Communications Letters*, vol. 19, pp. 653–656, April 2015.

[150] C. Chang, C. Chang, Y. Chen, and S. Lee, "Active route-guiding protocols for resisting obstacles in wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 59, pp. 4425–4442, Nov 2010.

[151] J. You, Q. Han, D. Lieckfeldt, J. Salzmann, and D. Timmermann, "Virtual position based geographic routing for wireless sensor networks," *Computer Communications*, vol. 33, no. 11, pp. 1255 – 1265, 2010.

[152] G. W. Denardin, C. H. Barriquello, A. Campos, and R. N. do Prado, "A geographic routing hybrid approach for void resolution in wireless sensor networks," *Journal of Systems and Software*, vol. 84, no. 10, pp. 1577 – 1590, 2011.

[153] F. Li, B. Zhang, and J. Zheng, "Geographic hole-bypassing forwarding protocol for wireless sensor networks," *IET Communications*, vol. 5, pp. 737–744, April 2011.

[154] M. Goswami, C.-C. Ni, X. Ban, J. Gao, X. D. Gu, and V. Pingali, "Load balanced short path routing in large-scale wireless networks using area-preserving maps," in *Proceedings of the 15th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '14, pp. 63–72, 2014.

[155] Y. C. Hwa-Chun Ma, Prasan Kumar Sahoo, "Computational geometry based distributed coverage hole detection protocol for the wireless sensor networks," *Journal of Network and Computer Applications*, vol. 34, no. 5, pp. 1743–1756, 2011.

[156] E. Zhao, J. Yao, H. Wang, and Y. Lv, "A coverage hole detection method and improvement scheme in wsns," in *Proceedings of 2011 International Conference on Electric Information and Control Engineering*, pp. 985–988, 2011.

[157] Y. Zhang, X. Zhang, Z. Wang, and H. Liu, "Virtual edge based coverage hole detection algorithm in wireless sensor networks," in *2013 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1488–1492, April 2013.

[158] W.-C. Chu and K.-F. Ssu, "Location-free boundary detection in mobile wireless sensor networks with a distributed approach," *Computer Networks*, vol. 70, pp. 96 – 112, 2014.

[159] R. Ghrist and A. Muhammad, "Coverage and hole-detection in sensor networks via homology," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN '05)*, 2005.

[160] F. Yan, P. Martins, and L. Decreusefond, "Connectivity-based distributed coverage hole detection in wireless sensor networks," in *Proceedings of the 54th IEEE Global Telecommunications Conference (GLOBECOM '11)*, pp. 1–6, 2011.

[161] H. Chintakunta and H. Krim, "Divide and conquer: Localizing coverage holes in sensor networks," in *Proceedings of the 7th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON '10)*, pp. 1– 8, 2010.

[162] F. Yan, A. Vergne, P. Martins, and L. Decreusefond, "Homology-based distributed coverage hole detection in wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 23, no. 6, pp. 1705–1718, 2015.

[163] R. Beghdad and A. Lamraoui, "Boundary and holes recognition in wireless sensor networks," *Journal of Innovation in Digital Ecosystems*, vol. 3, no. 1, pp. 1 – 14, 2016.

[164] L. Yi, X. Deng, M. Wang, D. Ding, and Y. Wang, "Localized confident information coverage hole detection in internet of things for radioactive pollution monitoring," *IEEE Access*, vol. 5, pp. 18665–18674, 2017.

[165] T. Amgoth and P. K. Jana, "Coverage hole detection and restoration algorithm for wireless sensor networks," *Peer-to-Peer Networking and Applications*, vol. 10, pp. 66–78, Jan 2017.

[166] P. Bose, P. Morin, I. Stojmenovir, and J. Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks," in *Proc. of the 5th IEEE International Conference*, pp. 347–352, 2008.

[167] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger, "Geometric ad-hoc routing: Of theory and practice," in *Proc. of the 22nd Annual Symposium on Principles of Distributed Computing*, PODC '03, (New York, NY, USA), pp. 63–72, 2003.

[168] S. Subramanian, S. Shakkottai, and P. Gupta, "On optimal geographic routing in wireless networks with holes and non-uniform traffic," in *Proc. of the 26th IEEE International Conference on Computer Communications, INFOCOM'07*, pp. 1019–1027, May 2007.

[169] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388–404, 2000.

[170] N. Jan, A. R. Hameed, B. Ali, R. Ullah, K. Ullah, and N. Javaid, "A balanced energy consuming and hole alleviating algorithm for wireless sensor networks," in *2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, pp. 231–237, March 2017.

[171] M. Choi and H. Choo, "Bypassing Hole Scheme Using Observer Packets for Geographic Routing in WSNs," in *Proc. of International Conference on Information Networking, ICOIN'11*, pp. 435–440, 2011.

[172] N. Bulusu, J. Heidemann, and D. Estrin, "Gps-less low-cost outdoor localization for very small devices," *IEEE Personal Communications*, vol. 7, pp. 28–34, Oct 2000.

[173] D. T. Lee and F. P. Preparata, "Euclidean shortest paths in the presence of rectilinear barriers," *Networks*, vol. 14, no. 3, pp. 393–410, 1984.

[174] H. Alt and E. Welzl, "Visibility graphs and obstacle-avoiding shortest paths," *Zeitschrift für Operations-Research*, vol. 32, pp. 145–164, May 1988.

[175] V. Shnayder, M. Hempstead, B. rong Chen, G. Werner-Allen, and M. Welsh, "Simulating the power consumption of large-scale sensor network applications," in *Proc. of the 2nd ACM Conference on Embedded Networked Sensor Systems, SenSys 2004*, pp. 188–200, ACM, 2004.

[176] K.-V. Nguyen, P. L. Nguyen, H. Phan, and T. D. Nguyen, "A distributed algorithm for monitoring an expanding hole in wireless sensor networks," *Informatica*, vol. 40, no. 2, pp. 181–195, 2016.

[177] H. Yetgin, K. T. K. Cheung, M. El-Hajjar, and L. H. Hanzo, "A survey of network lifetime maximization techniques in wireless sensor networks," *IEEE Communications Surveys Tutorials*, vol. 19, no. 2, pp. 828–854, 2017.

[178] Z. Zhang, J. Willson, Z. Lu, W. Wu, X. Zhu, and D. Du, "Approximating maximum lifetime k -coverage through minimizing weighted k -cover in homogeneous wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 24, pp. 3620–3633, December 2016.

[179] A. Pananjady, V. K. Bagaria, and R. Vaze, "Optimally approximating the coverage lifetime of wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 25, pp. 98–111, Feb 2017.

[180] G. Tuna and V. Gungor, "2 - energy harvesting and battery technologies for powering wireless sensor networks," in *Industrial Wireless Sensor Networks* (R. Budampati and S. Kolavennu, eds.), pp. 25 – 38, Woodhead Publishing, 2016.

[181] F. Akhtar and M. H. Rehmani, "Energy replenishment using renewable and traditional energy resources for sustainable wireless sensor networks: A review," *Renewable and Sustainable Energy Reviews*, vol. 45, pp. 769 – 784, 2015.

[182] C. Wang, J. Li, Y. Yang, and F. Ye, "A hybrid framework combining solar energy harvesting and wireless charging for wireless sensor networks," in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, pp. 1–9, April 2016.

[183] A. H. Dehwah, S. B. Taieb, J. S. Shamma, and C. G. Claudel, "Decentralized energy and power estimation in solar-powered wireless sensor networks," in *2015 International Conference on Distributed Computing in Sensor Systems*, pp. 199–200, June 2015.

[184] L. Xie, Y. Shi, Y. T. Hou, and H. D. Sherali, "Making sensor networks immortal: An energy-renewal approach with wireless power transfer," *IEEE/ACM Transactions on Networking*, vol. 20, pp. 1748–1761, Dec 2012.

[185] L. Fu, L. He, P. Cheng, Y. Gu, J. Pan, and J. Chen, "Esync: Energy synchronized mobile charging in rechargeable wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 65, pp. 7415–7431, Sep. 2016.

List of publications

Journal papers

1 **Phi Le Nguyen**, Yusheng Ji, Huy Vu, Zhi Liu, Khanh-Van Nguyen, "Distributed Hole-Bypassing Protocol in WSNs with Constant Stretch and Load Balancing", Computer Networks journal, vol.129 (2017), pp.232-250. (Research article, 19 pages)

Top conference proceedings

2 **Phi Le Nguyen**, Yusheng Ji, Trung Thanh Nguyen, Thanh-Hung Nguyen, "Constant Stretch and Load Balanced Routing Protocol for Bypassing Multiple Holes in Wireless Sensor Networks", IEEE 16th International Symposium on Network Computing and Applications (NCA), Oct. 2017. (Full paper, 9 pages). Top conference list: No.157

Other referred proceedings

3 **Phi Le Nguyen**, Khanh-Van Nguyen, Huy Vu, Yusheng Ji, "A Time and Energy Efficient Protocol for Locating Coverage Holes in WSNs", IEEE 41st Conference on Local Computer Networks (LCN), Nov. 2016, pp.180-183. (Short paper, 4 pages). Top conference list: No.138

4 **Phi Le Nguyen**, Yusheng Ji, Trung Thanh Nguyen, Thanh-Hung Nguyen, "A Delay-Guaranteed Geographic Routing Protocol with Hole Avoidance in WSNs",

IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), Oct. 2017, pp. 135-143. (Full paper, 9 pages)

5 **Phi Le Nguyen**, Yusheng Ji, Khanh Le, Thanh-Hung Nguyen, "Load Balanced and Constant Stretch Routing in the Vicinity of Holes in WSNs", 15th IEEE Annual Consumer Communications Networking Conference (CCNC), Jan. 2018. (Full paper, 6 pages)

6 **Phi Le Nguyen**, Yusheng Ji, Khanh Le, Thanh-Hung Nguyen, "Routing in the Vicinity of Multiple Holes in WSNs", accepted by 5th International Conference on Information and Communication Technologies for Disaster Management (ICT-DM 2018). (Full paper, 9 pages) (Best student paper award).

7 Nguyen Thi Hanh, **Phi Le Nguyen**, Phan Thanh Tuyen, Huynh Thi Thanh Binh, Ernest Kurniawan, Yusheng Ji, "Node Placement for Target Coverage and Network Connectivity in WSNs with Multiple Sinks", 15th IEEE Annual Consumer Communications Networking Conference (CCNC), Jan. 2018. (Full paper, 6 pages)

8 Van An Le, **Phi Le Nguyen**, Yusheng Ji, "Deep Convolutional LSTM Network-based Traffic Matrix Prediction with Partial Information", accepted by IFIP/IEEE International Symposium on Integrated Network Management (IM 2019). (Full paper, 8 pages), Top conference list: No.179.

Technical reports

9 **Phi Le Nguyen**, Vu Quoc Huy, Yusheng Ji, "Hole Aware Real-time Routing Protocol in WSNs", IEICE Technical Report, 2017-01-CQ, Jan. 2017.

10 **Phi Le Nguyen**, Yusheng Ji, Thanh Trung Nguyen, Thanh-Hung Nguyen, "Hole Bypassing Geographic Routing Protocol with Delay Guarantee in WSNs", Technical Report of IEICE, IN2016-163, March 2017.

11 **Phi Le Nguyen**, Yusheng Ji, "Load Balanced Geographic Routing in the Vicinity of Holes in WSNs", Technical Report of IEICE, September 2017.

12 **Phi Le Nguyen**, Yusheng Ji, Le Hieu, Nguyen Duc Anh, Van An Le, Thanh Hung Nguyen, "Network Lifetime Maximization with Full Area Coverage in WSNs", IEICE Technical Report, Vol. 118, No. 192, CQ2018-54, pp. 47-52, Aug. 2018.

13 Van An Le, **Phi Le Nguyen**, Yusheng Ji, "Traffic Matrix Prediction based on Bidirectional Recurrent Neural Network and Long Short-Term Memory", IEICE Technical Report, Vol. 118, No. 140, CQ2018-40, pp. 51-56, July 2018.