Surviving Holes and Barriers in Geographic Data Reporting for Wireless Sensor Networks

Yangfan Zhou* Michael R. Lyu* Jiangchuan Liu†
*Dept. of Computer Science & Engineering, The Chinese Univ. of Hong Kong, Shatin, Hong Kong
†School of Computing Science, Simon Fraser Univ., Burnaby, BC, Canada

Abstract

Geographic forwarding is a favorable scheme for data reporting in Wireless Sensor Networks (WSNs) due to its simplicity and low-overhead. However, WSNs are usually subject to complicated environmental factors. Network holes (i.e., the areas where no nodes inside) and barriers (i.e., those blocking the communication between two close nodes) are inevitable in practical deploying environments. These issues pose an obstacle to adopting geographic forwarding in WSNs, while current approaches lack an efficient method to tolerate such negative factors. In this paper we specifically tailor a waypoint-based Geographic Data Reporting Protocol (GDRP) for WSNs. Inherited from geographic forwarding, GDRP is light-weighted and hence well-suits WSNs. But unlike current approaches that often find suboptimal paths, GDRP adopts an intelligent strategy to select a best set of waypoints via which packets can efficiently circumvent holes and barriers, and it can thus find better paths. Extensive simulations are conducted to verify the advantages of GDRP in tolerating network holes and obstacles in WSNs.

1. Introduction

In-situ sensing with wireless sensor networks (WSNs) is a promising technique for environmental monitoring applications [2, 16, 17]. In these applications, a crucial task for a WSN is to convey the sensor data to a sink so that the sink can obtain the information of interest. *Geographic forwarding* [4, 12, 13] is favorable for this sensor-to-sink data reporting task. Its basic idea, namely, *greedy forwarding*, is that a packet is forwarded to a neighbor which is geographically closer to the intended destination till the packet eventually reaches the destination. In geographic forwarding, each node just needs to maintain the information of its neighborhood to select its next-hop neighbor. This provides it nice scalability and low overhead merits, which are specifically desirable for WSNs due to their large-scale feature and the energy constraints suffered by sensor nodes [6].

However, network holes (*i.e.*, the areas where no nodes inside) and barriers (*i.e.*, those blocking the communication between two close nodes) are inevitable in practice [1]. Various real-world geographical environments, *e.g.*, the existence of puddles or buildings where sensors cannot be deployed causes holes. Hills, walls, or even trees may cut wireless link of two nodes even if they are in the theoretical communication range of each other, and thus form barriers. All these can make greedy forwarding infeasible as a node may not find any neighbor nearer to a packet's intended destination than itself.

To tolerate the failure of greedy forwarding, traditional geographic forwarding schemes enter a *detour mode* in which a packet is sent to a neighbor farther to the destination but potential in bypassing a hole [4, 13]. The detour mode tends to forward data packets along the boundaries of holes. It usually turns out that the path from the source to the destination is much longer than the optimum [3]. Longer path incurs more energy consumption for packet transportation. It is therefore critical to enhance the survivability of geographic forwarding in practical deployment environments of WSNs, where there are many network holes and barriers.

In this paper, we specifically tailor a waypoint-based Geographic Data Reporting Protocol (GDRP) for WSNs to address this challenging problem. The purpose of waypoint-based geographic forwarding is to minimize the unnecessary detours by forwarding packets along a sequence of waypoints so that the path can bypass holes and barriers. Lengthy routes due to the detour mode can thus be avoided. GDRP adopts a trial-and-error approach: The information of holes and barriers is accumulated during the runtime of GDRP. Based on such information, better and better waypoint sequences can then be designed. Unlike current state-of-the-art approaches [3, 11, 21] which may find waypoints that result in suboptimal path, GDRP can find an optimal path from the source to the sink. We formulate how to select the waypoints as a tractable problem and provide its solution with details in handling realistic network situations.

978-1-4244-5113-5/09/$25.00 ©2009 IEEE

In contrast to the existing work [3, 11, 21] that provides heuristics in finding waypoints, we prove the performance guarantee of GDRP.

The rest of this paper is organized as follows. Section 2 provides the motivations of this paper and sketches GDRP. Section 3 elaborates GDRP and illustrates how it can survive network holes and barriers energy-efficiently. We study the performance of GDRP with extensive simulations in Section 4. Section 5 presents the related work and Section 6 concludes this paper.

2. Waypoint-Based Geographic Forwarding

We consider a WSN consisting of a sink d and N stationary sensor nodes randomly deployed in a 2-dimensional network area ϕ (*i.e.*, $\phi \subset \mathbb{R}^2$). Since location-awareness is a basic requirement for geographic forwarding algorithms [4, 13], we consider that each node is aware of its own location, which can be obtained by GPS or a localization approach (*e.g.*, [5]). Let r denote the communication range of a node. Each node u can then know the locations of its neighbors, *i.e.*, those nodes that have a wireless link with u. Note that due to the existence of barriers in practical working environments, two nodes may not have a wireless link even when their distance is less than r. In this case, they are not neighbors.

2.1. Geographic forwarding

Geographic forwarding schemes [4, 13] firstly planarize a network into a planar graph G_P (*e.g.* Gabriel Graph or Relative Neighborhood Graph) with a distributed and localized algorithm as shown in Figure 1. They usually contain two working modes: *the greedy mode* and *the detour mode*. In the greedy mode, a node u selects a node v as its next hop among u's neighbors that are nearer to the sink than itself when v is the nearest one to the sink (*e.g.*, in Figure 1, the source selects u_1 and u_1 selects u_2). When holes and barriers exist, the greedy mode may not be feasible, *i.e.*, no neighbor is nearer to the sink (*e.g.*, for u_3). To handle this case, geographic forwarding schemes send packets in the detour mode: Packets are sent clockwise (or counter-clockwise) along the face of G_P which are closer to the sink (*e.g.*, u_3 selects u_4 as its next-hop and u_4 selects u_5 as its next-hop), until greedy forwarding is feasible again at a node (u_8) which is also closer to the sink than the node where greedy forwarding first got stuck (u_3) [13].

2.2. Design considerations of waypoint-based geographic forwarding

The design objective of a data reporting scheme is to minimize the energy consumptions of the packet transportation from the source to the destination. Since hop number of a path shows how many nodes are involved in the packet transportation, it is a natural indicator to such energy consumptions. We define:

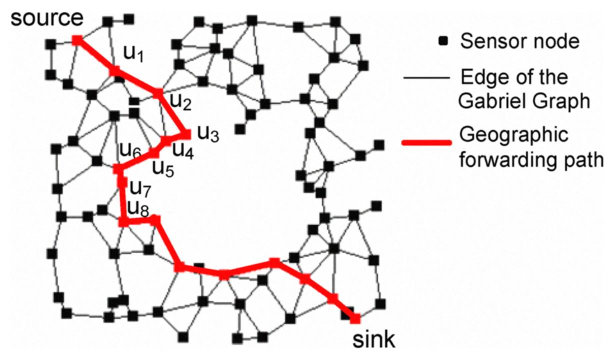


Figure 1. A geographic forwarding example

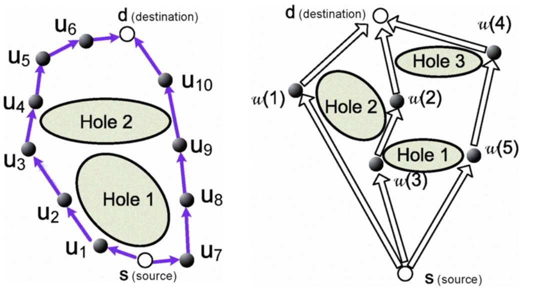
Definition 1 *The topological length of a path is the total number of hops between the source and the destination of the path.*□

A *waypoint sequence* is a sequence of sensor nodes that serve as intermediate relays for the packet transportation from the source to the destination. Let W=[$w(1)$, w(2),..., $w(M)$] denote a waypoint sequence with size M where each element denotes a waypoint node. Note that we deem the source as the first waypoint and the destination as the last waypoint. We define:

Definition 2 *The Euclidean length of a waypoint sequence* $[w(1), ..., w(M)]$ *is* $\sum_{i=2}^{M} \mathbb{D}(w(i), w(i-1))$*, where* $\mathbb{D}(\cdot,\cdot)$ *denotes the Euclidean distance between two nodes.*□

In *waypoint-based geographic forwarding*, packets from the first waypoint, *i.e.*, the source, will be sent to the second waypoint. A waypoint, when receiving a packet, will send the packet to its next waypoint until the last waypoint (*i.e.*, the destination) is reached. Packets are transported between two adjacent waypoints with a geographic forwarding scheme.

The aim of a waypoint-based geographic forwarding scheme is to find a waypoint sequence $[w(1), ..., w(M)]$ given a source $w(1)$ and destination $w(M)$ so as to minimize topological length of the path from $w(1)$ to $w(M)$. To achieve this, two issues are generally considered. First, greedy forwarding should always be feasible between two adjacent waypoints based on the notion that the greedy mode is better than the detour mode in terms of topological length. Second, the Euclidean length of the waypoint sequence should be minimized since longer Euclidean length usually incurs larger topological length. However, these two considerations in the existing approaches are not yet adequate.



(a). A network scenario where greedy forwarding results in suboptimal path.

(b). A network scenario where routing along the convex hull causes suboptimal results.

Figure 2. Examples of suboptimal results

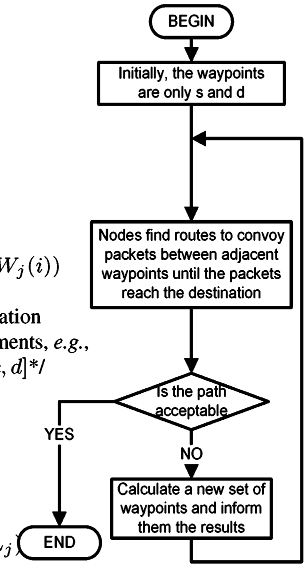
First, even if greedy forwarding is always feasible, geographic routing cannot always achieve optimal routes [11]. Figure 2(a) shows an example where a greedy forwarding path (*i.e.*, the left-hand side path) is just a suboptimal path. In fact, the topological length of greedy forwarding between s and d is tightly-bounded by $O(\mathbb{D}^2(s,d))$ [10]. Such a *quadratic* relation makes the theoretical performance of the greedy mode only comparable to the detour mode. We are motivated to enhance waypoint-based geographic forwarding so that the topological length between two adjacent waypoints is *linearly* related to their distance.

Second, current approaches lack a strategy that can calculate a waypoint sequence with minimum Euclidean length. Huang [11] has recently suggested that packets should be forwarded along the convex hull of the geometry set Z which consists of the source, the destination, and the holes and barriers between them. Supposing there is a 2-dimensional Cartesian coordinate system with its x-axis passing the source and the destination, this approach finds a waypoint sequence where the waypoints form the $y > 0$ or $y < 0$ half of the convex hull of Z. However, this approach may not find a good waypoint sequence, since a shorter sequence can penetrate the set Z rather than going around it. Figure 2(b) shows an example where the waypoint sequence $[s, w(3), w(2), d]$ is shorter in Euclidean length than the waypoint sequences $[s, w(1), d]$ and $[s, w(5), w(4), d]$ that can be found by this approach.

How can we find a shorter path than those that go around Z while the path can still bypass the holes and the barriers in Z? A brute-force strategy is that for each hole or barrier in front, packets are forwarded to both left-hand side and right-hand side of the hole or barrier. Euclidean lengths of all the possible resulting waypoint sequences are compared and the minimum one is selected. This strategy can find a best waypoint sequence since it tries all options. However, the number of options is 2^m where m is the number of holes and barriers between the source and the destination. When there are many holes and barriers in between, the number of options is huge, which is surely unacceptable. Hence, a better approach that can converge in polynomial time is desired.

Procedure 1 The mechanism of GDRP

```
EndFlag ⇐ false
L_0 ⇐ ∅
j ⇐ 0
W_1 ⇐ [s, d]
repeat
    j ⇐ j + 1
    P_j ⇐ ∅
    k ⇐ |W_j|
    i ⇐ k
    while i ≠ 1 do
        P′ ⇐ route(W_j(i − 1), W_j(i))
        P_j ⇐ P′ ⊕ P_j
        /* ⊕ denotes the concatenation
        operation of two path segments, e.g.,
        [a, b, c] ⊕ [c, d] = [a, b, c, d]*/
        i ⇐ i − 1
    end while
    if accept(P_j) then
        EndFlag ⇐ true
    else
        L_j ⇐ L_{j−1} ∪ P_j
        W_{j+1} ⇐ cal_waypoint(L_j)
    end if
until EndFlag is true
```



2.3. Overview of GDRP

In order to address the above two issues, we specifically design GDRP, a waypoint-based geographic data reporting protocol for WSNs. GDRP employs a trial-and-error approach to find an optimal path. Round by round, based on its current knowledge on the holes and barriers between the source and destination, it tries a so-far-the-best waypoint sequence to bypass the holes and barriers. Gradually it reinforces its knowledge on the holes and barriers, until eventually an optimal path is found to bypass the holes and barriers. Procedure 1 illustrates its major mechanism.

Let j denote the round index. P_j denotes a path from the source to the destination found in round j and L_j denotes all the paths that have been found in round j and before, *i.e.*, $L_j = P_1 \cup P_2 \cup ... \cup P_j$. In the first round, since no knowledge on the in-network holes and barriers is available, the waypoints are only the source and the destination. The path discovery scheme of GDRP finds the first path P_1 from the source to the destination. In round j ($j > 1$), the waypoint calculation scheme selects a sequence of waypoints W_j=$[w_j(1), \ldots, w_j(M_j)]$ based on L_{j-1}. And the path discovery scheme forwards packets between each adjacent waypoint pair $w_j(i-1)$ and $w_j(i)$ ($\forall i = 2, ..., M_j$) so that packets are conveyed from the source to the destination. A new path P_j is thus found. The above procedure is conducted iteratively until the resulting path is *acceptable*.

GDRP adopts a tidy framework where there are essentially only three components:

- Waypoint calculation, denoted by *cal_waypoint*$(\cdot)$ in line 19 of Procedure 1;
- Path discovery between adjacent waypoints, denoted by *route*$(\cdot,\cdot)$ of line 11 in Procedure 1;
- Examination of whether an existing path is acceptable, denoted by *accept*$(\cdot)$ in line 15 of Procedure 1.

The first and the third components are conducted by only the sink, while the in-network nodes are responsible for the second component. The sink (*e.g.*, a laptop computer, a PDA, or a robot in a mobility-assisted WSN) is usually with more powerful computational capability and less energy constraint than the in-network sensor nodes. We hence put the major computational loads in the first and the third components, while the second component is made light-weighted and carefully tailored for energy-constraint sensor nodes. These components will be illustrated in Section 3.

2.4. GDRP preliminaries

2.4.1 Packet format

The header of a GDRP packet contains three fields: the geographic location of the intended destination, the location of the waypoint that the packet is currently heading for, and the locations of nodes the packet has been visited. The packet format is shown in Figure 3.

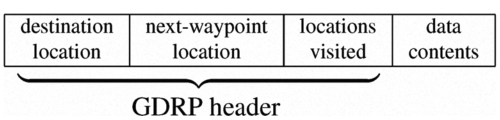


Figure 3. GDRP Packet Format

The *destination location* field save the sink location, which is unchanged during the packet's trip to the sink. The *locations visited* field is incrementally updated by each node that forwards the packet. A record of such locations is generally required by a waypoint-based geographic forwarding approach for calculating a waypoint sequence [3, 11, 21]. In round 1 of Procedure 1, as only the source and the sink are waypoints, the source fills the location of the sink into the *next-waypoint location* field, meaning that the packet is heading for the sink. Then, new waypoint sequences can be calculated in round j ($j > 1$). When a packet reaches each designated waypoint, this field would be updated by the waypoint to its next waypoint.

Note that such a packet header is small in size. Most of the overhead is in the *locations visited* field, which, however, is only applied before Procedure 1 converges. After GDRP finds a waypoint sequence that results in an acceptable path, such a field is then unnecessary. Hence, the packet overhead caused by the header is inconsiderable, which sticks to our objective of energy efficiency.

2.4.2 Waypoint table

For a node selected as waypoint, the sink would inform it and let it know the location of its next adjacent waypoint. A *waypoint table* is designed to save such information. Each record of the waypoint table contains only two fields: The *destination location* field and the *next waypoint location* field. The *next waypoint location* field tells where the next waypoint is for those packets targeting at what the corresponding *destination location* field indicates. Finally, note that waypoint tables are quite small in size (tens of bytes for typical WSNs) for the state-of-the-art sensor platforms [6].

3. Surviving Holes and Barriers with GDRP

3.1. When a path is acceptable

Let us firstly formulate *perfect sequence in geographic forwarding* (in short, perfect sequence) and *strongly* perfect sequence as follows.

Definition 3 *A sequence of nodes* $[u_0, u_1, ..., u_n, w]$ *is a perfect sequence in geographic forwarding if it satisfies:*

- *The distance between any two adjacent nodes in the subsequence* $[u_0, ..., u_n]$ *is less than or equal to* r*;*
- u_i *is geographically nearer to* w *than node* u_k *if* $i > k$.
- *The distance between* u_i ($i > 0$) *and any other nodes except* u_{i-1} *and* u_{i+1} *in* $[u_1, ..., u_n]$ *is larger than* r*;*
- *Given a* x-y *coordinate system with its* x*-axis passing* u_0 *and* w*, the maximum difference of the* y*-coordinates between any two nodes are no more than* $d = \alpha \cdot r$*, where* α *is a constant.*

A perfect sequence is a strongly perfect sequence if:

- *The distance between* w *and* u_n *is less than or equal to* r*, while the distance between* w *and any other nodes in the sequence is larger than* r.□

The first three requirements guarantee that the nodes except the last in a perfect sequence form a *path segment* with the last node being the destination, where greedy forwarding is always feasible. A strongly perfect sequence further ensures all nodes form a greedy forwarding *path*. Moreover, the nodes in a perfect sequence are confined in a rectangular area based on the fourth requirement. As demonstrated in Figure 4, the whole path [u_0, u_1, ..., u_6, destination] is a strongly perfect sequence, while [u_0, u_1, u_2, u_3, destination] is a perfect sequence since the distance between u_3 and destination is larger than r. A good property of strongly perfect sequence is as follows.

Lemma 1 *The topological length of a path is linearly related to the Euclidean distance between the source and the destination if the path is a strongly perfect sequence.*□

Proof: Let us denote the path as [u_0, u_1, ..., destination]. Suppose the distant between the source to the destination

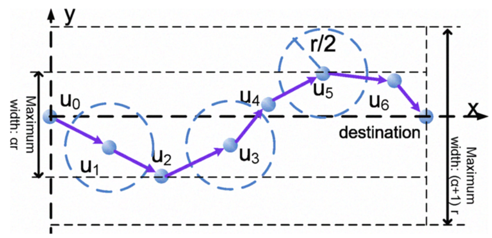


Figure 4. A strongly perfect sequence

is l. As shown in Figure 4, draw the circles centered at u_i where i is odd with their radii equal to $\frac{r}{2}$. All these circles must be in a rectangular area with length l and width $(\alpha + 1)r$ because all nodes are in a rectangular area with length l and width αr. Moreover, these circles should not intersect each other. Otherwise, the distance between two non-adjacent nodes is less than r which contradicts the requirements of strongly perfect sequence. Therefore, the maximum number of such circles $n(l)$ are bounded by:

$$n(l) \leq \frac{l \cdot (\alpha+1)r}{\pi(\frac{1}{2}r)^2} = \frac{4(\alpha+1)}{\pi r} \cdot l. \quad (1)$$

The total number of nodes (*i.e.*, u_i with i being odd or even) in the path is bounded by twice of the circle numbers. The topological length of a strongly perfect sequence, denoted by $H_{SPS}(l)$, is hence bounded by:

$$H_{SPS}(l) \leq \frac{8(\alpha+1)}{\pi r} \cdot l, \quad (2)$$

which is linearly related to l and thus Lemma 1 is proved.

Corollary 1 *If an algorithm find a path which is a strongly perfect sequence, the algorithm is a linear approximation to the shortest path algorithm in terms of topological length.*

Proof: Let us again suppose the distance between the source and the destination is l. The topological length of the shortest path $H_S(l)$ between them is lower-bounded by $\lceil \frac{l}{r} \rceil$. Therefore according to Lemma 1, we get:

$$H_{SPS}(l) \leq \frac{8(\alpha+1)}{\pi} \lceil \frac{l}{r} \rceil \leq \frac{8(\alpha+1)}{\pi} H_{SP}(l). \quad (3)$$

It shows that the topological length of a strongly perfect sequence will not be worse than $\frac{8(\alpha+1)}{\pi}$ times of that of the shortest path, which proves Corollary 1.

Therefore, we deem that given a path segment from waypoint $w(i\text{-}1)$ to $w(i)$, if the path segment is a strongly perfect sequence, the in-network holes and barriers do not considerably influence the data transportation between $w(i\text{-}1)$ and $w(i)$. Hence, GDRP can accept a path and stop Procedure 1 if the path is an acceptable path defined as follows.

Definition 4 *A path P_j is an acceptable path for Procedure 1 if the path segments between any two adjacent waypoints, i.e., $w_j(i\text{-}1)$ and $w_j(i)$ ($\forall i = 2, ..., M$), are strongly perfect sequences.*□

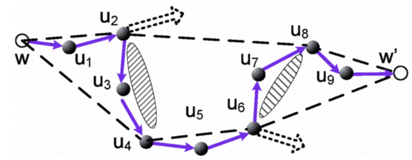


Figure 5. An example on modeling the influences of holes/barriers and calculating waypoint sequence

3.2. Calculating a new waypoint sequence

In order to effectively bypass holes and barriers, the waypoints should be determined based on the locations of the holes and barriers. However, since the sink does not have a global picture of the network, holes and barriers cannot be known beforehand. GDRP has to learn the information of holes and barriers with the paths found in previous rounds. With a trial-and-error approach, the knowledge on the holes and barriers can be accumulated. Consequently, better and better waypoint sequences can be designed. In what follows, we first discuss how GDRP learns the knowledge of holes and barriers with the paths found in previous rounds. And then, we provide how GDRP calculates a waypoint sequence based on such knowledge.

3.2.1 Modeling the influences of holes and barriers

When a path is not an acceptable path, there exists at least one path segment between a pair of adjacent waypoints which is not a strongly perfect sequence. Let $[w, u_1, u_2, ..., w']$ denote such a path segment where u_1, ..., and u_i are the intermediate nodes, and w, w' are the adjacent waypoints. In this case, there are some holes or barriers that influence the packet transportation from w to w'.

Since the impact of holes or barriers can be considered as how they make the path segment "unperfect", they can be inferred by finding which parts of the segment make it fail to be a strongly perfect sequence. Thus, we can model the impact of the holes and barriers using these parts. Figure 5 demonstrates an example. In this example, $[w, u_1, u_2, w']$, $[u_4, u_5, u_6, w']$, and $[u_8, u_9, w']$ are all perfect sequences; whereas $[u_2, u_3, u_4]$ and $[u_6, u_7, u_8]$ make the whole path segment fail to be a strongly perfect sequence. Therefore, it can be inferred that there are some holes or barriers lying in the shaded areas shown in Figure 5.

Specifically, for $[w, u_1, u_2, ..., w']$, GDRP first finds those parts that form perfect sequences together with w'. And then it considers the rest parts are resulting from holes or barriers between the two waypoints. We call such a part a *detour part*, where the first node is called the *starting node* of the detour part and the last one is its *ending node*. Consider the example case shown in Figure 5: $[u_2, u_3, u_4]$ and $[u_6, u_7, u_8]$ are two detour parts with starting nodes being u_2 and u_6, and ending nodes being u_4 and u_8, respectively.

Thus, for each path P_j found in round j of Procedure 1, if the path is not an acceptable path, a set of detour parts in P_j, denoted by B_j, can be found. And then the collection $\{B_k\}_{k=1}^{j}$ can be regarded as the current knowledge of the in-network holes and barriers by the end of round j of Procedure 1. Note that since the impacts of holes and barriers are the same with the notion that they both make a path contain detour parts, formulating their impacts as the detour parts actually provide GDRP a generic treatment to both holes and barriers.

3.2.2 Obtaining the best waypoint sequence

Based on the current knowledge $\{B_k\}_{k=1}^{j}$, our objective is to find a best set of waypoints W_{j+1} for round j+1 so as to make the packets bypass the known holes and barriers in round j+1 and minimize the topological length of the path found in round j+1.

We consider the starting nodes and the ending nodes of the detour parts as the *potential waypoints*, *i.e.*, waypoints in W_{j+1} are selected from the set of the starting nodes and the ending nodes of all detour parts in $\{B_k\}_{k=1}^{j}$. In the example shown in Figure 5, u_2, u_4, u_6, and u_8 are hence the potential waypoints. The reason is justified as follows.

We select an ending node of a detour part as a potential waypoint because from this node on the path is clear of the hole or barrier modeled by the detour part, since a new perfect sequence starts from this node. In other words, the known hole or barrier modeled by a detour part does not influence the path any longer from its ending node on.

We select a starting node as potential waypoint as we take into accounts the possibility that the node can avoid the corresponding detour part by just forwarding packets to another direction. Again consider the example in Figure 5. If u_2 and u_6 are waypoints, they can attempt to explore the network in the other directions indicated by the dotted arrows in the figure to avoid the known holes or barriers modeled by their corresponding detour parts. Finally, note that when a node is the starting node of more than one detour part, it means that both directions have been tried. Consequently, it would not be considered as a potential waypoint.

Given the potential waypoints composed by the starting nodes and the ending nodes of $\{B_k\}_{k=1}^{j}$, we expect in round j+1 GDRP can find an acceptable path, *i.e.*, GDRP can find a strongly perfect sequence between each pair of the adjacent waypoints. Therefore, draw a line segment between two adjacent waypoints in W_{j+1}, it cannot intersects a known detour part. Otherwise, the path segment found in round j+1 between the waypoints cannot be a strongly perfect sequence as it will detour when facing the hole or obstacle modeled by the detour part. Based on this consideration, we calculate the waypoint sequence W_{j+1} as follows.

First, construct a subgraph G_W of the network in which the vertex set includes the potential waypoints, the source s, and the sink d. Two vertices in G_W share an edge if and only if the line segment between the two vertices does not intersect any of the detour parts[1]. Figure 5 shows an example where the dotted line segments denote the edges of G_W. Then let each edge in graph G_W be weighted by its Euclidean distance and find a shortest path in G_W from the source s and the destination d. The potential waypoints along such a shortest path is hence considered as the resulting waypoint sequence. In the example shown in Figure 5, the shortest path is $w \rightarrow u_2 \rightarrow u_8 \rightarrow w'$ and hence the waypoint sequence is $[w, u_2, u_8, w']$. Lastly, note that no additional overhead is introduced in this procedure since G_W is constructed merely with L_j=$\{P_k\}_{k=1}^{j}$ collected in each round of Procedure 1. Based on how G_W is constructed and the property of shortest path, we can obtain the following lemma.

Lemma 2 *The waypoint sequence found by GDRP is with minimum Euclidean length according to the current knowledge of the holes and barriers.*

Suppose in round k, GDRP eventually converges. The topological length of P_k is then linearly related to the Euclidean length of W_k. Since the Euclidean length of W_k is the minimum, we can hence guarantee the resulting path is an optimal path between the source and the destination.

3.3. Geographic forwarding between adjacent waypoints

Now we discuss how we tailor the path discovery scheme of a sensor node so that it can conduct geographic forwarding task between the adjacent waypoints. First of all, such a path discovery scheme should always ensure successful packet delivery between two waypoints. Hence we base the path discovery scheme adopted in traditional geographic forwarding approaches (see Section 2.1) [13]. But two changes are made. The first change is that the right-hand rule or left-hand rule may be adopted even if greedy forwarding is feasible. This is due to the fact that even greedy forwarding is feasible, there may be a hole or barrier ahead. GDRP should try to forward packet to another direction to bypass the hole or barrier.

The second change is that in GDRP, the staring node of a detour part (*i.e.* a node which faces a hole or barrier ahead) can choose the opposite direction to what is chosen in the previous round of Procedure 1, so as to explore a new path

[1]The starting node and the ending node of the same detour part will not share an edge.

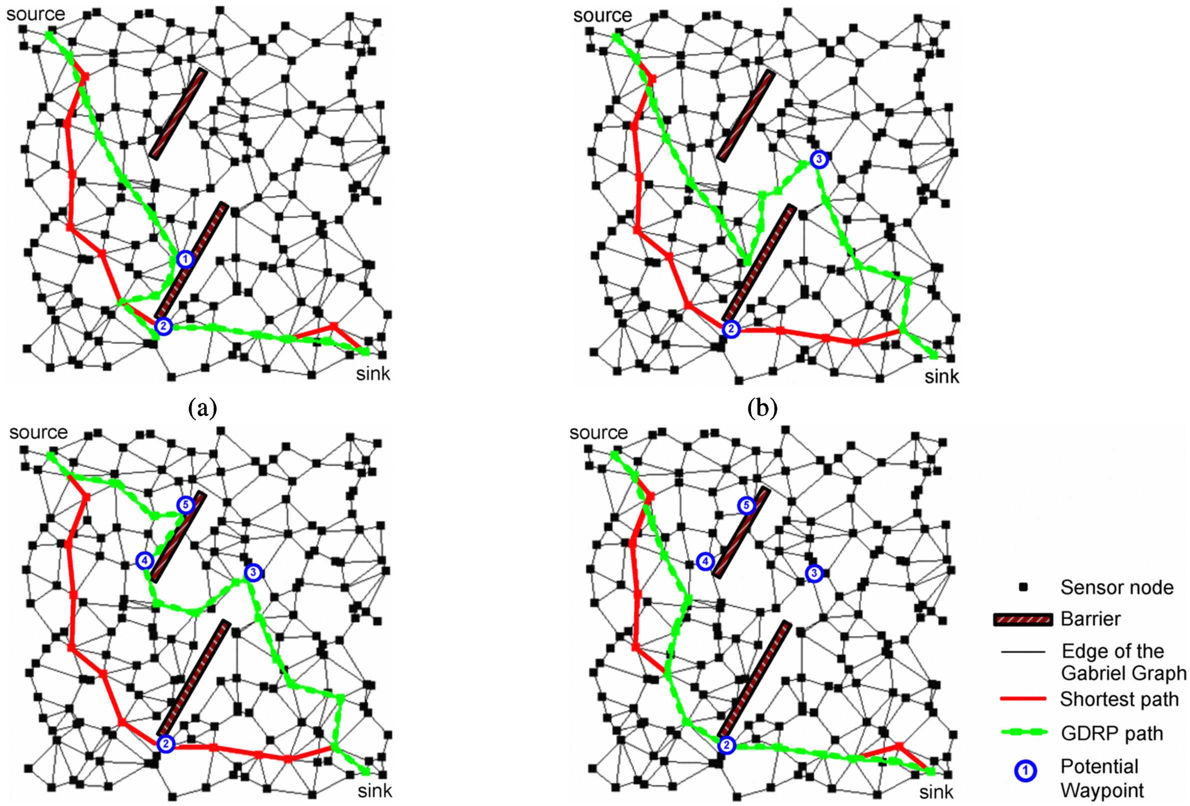


Figure 6. A case demonstration of GDRP

which is potentially better than what is found in the previous round. For conducting this task, we design a *direction table* for each starting node. Each record of a direction table contains two field: the *waypoint* field and the *direction* field. The *direction* field saves the rule (left-hand or right-hand) adopted in the previous round of Procedure 1 when forwarding packets to the waypoint recorded in the corresponding *waypoint* field. After each round of Procedure 1, if GDRP finds out that a node is a starting node of an newly-found detour part, it would inform the node. Suppose in the previous round the packets forwarded by this node is heading for a waypoint w. The node would then check its routing direction table and find the record of w and change the direction in its *direction* field accordingly[2].

We now elaborate the behavior of a node when it runs GDRP. First, when receiving a packet, a node reads the packet's *next-waypoint location* field. If the node is the intended waypoint, it would update the *next-waypoint location* field of the packet according to the node's waypoint table before it sends the packet. Otherwise, it just proceeds to send the packet.

When sending a packet, if no record of the corresponding waypoint can be found in a node's direction table or there is no direction table, the node would simply adopts the same strategy as traditional geographic forwarding to send the packet: Employ greedy forwarding if it is feasible, or otherwise take predetermined left-hand rule or right-hand rule and enter the detour mode to forward the packet. But if the record can be found, it will choose a routing direction according to the *direction* field of the record and adopt left-hand rule or right-hand rule accordingly .

3.4. A case demonstration of GDRP

Figure 6 demonstrates the resulting paths found in each round of GDRP in an example network. In the first round, since no network barriers and holes are known, the sink considers [source, sink] as the waypoint sequence. The path in Figure 6(a) is found. the sink finds two potential waypoints w_1 and w_2 shown in the figure. It then informs w_1 to change its forwarding direction since it is the starting node of the detour part from w_1 to w_2.

Because the line segment from the source to the sink

[2]If there is no record of w, the node would create one for w. The location of w is saved in its *waypoint* field. The *direction* field is set to the opposite direction of the default one.

does not intersect the known detour part from w_1 to w_2, in round 2, the sink again sets [source, sink] as the waypoint sequence since it is the shortest path in G_W from the source to the sink. Now when a packet reaches w_1, w_1 would forward the packet to anther direction. The resulting path is shown in Figure 6(b). Another detour part from w_1 to w_3 is found and w_3 is then a potential waypoint. Since w_1 is the starting node of two detour parts, the sink knows w_1 fails to bypass a hole or barrier and hence w_1 will not be selected as a potential waypoint any more.

Now the sink finds out that the waypoint sequence [source, w_3, sink] is the shortest path in G_W from the source to the sink. It selects w_3 as a waypoint and informs it in round 3. Packets are now first sent to w_3 and then to the sink. Figure 6(c) shows the resulting path. The sink finds the third detour part from w_5 to w_4. w_4 and w_5 are then the new potential waypoints and G_W is updated.

Now the waypoint sequence [source, w_2, sink] is the shortest path in G_W from the source to the sink. In round 4, the sink sets w_2 as a waypoint and informs it. In this round, the path segments from the source to w_2 and from w_2 to the sink are both strongly perfect sequences as shown in Figure 6(d). GDRP completes its task in finding a best set of waypoint. Packets, from then on, can be forwarding along the path found in round 4.

For comparison purpose, we also plot the shortest path in Figure 6. It shows the final resulting path of GDRP is comparable to the shortest path in terms of topological length.

4. Simulation Study

In order to study the effectiveness of GDRP in surviving network holes and barriers, we simulate a WSN. Energy-efficiency is studied in terms of the topological length of the resulting paths. We compare our GDRP protocol with GPSR (a geographic forwarding protocol proposed in [13]) and a waypoint-based geographic forwarding protocol like that proposed in [11], which is named CONVEX-W in this paper. GPSR tolerates holes and barriers by entering the detour mode instead of relying on waypoints. CONVEX-W always find waypoints in one side of the line segment from the source to the sink, which forms the half convex hull of the source, the sink, and the known holes and barriers in between. In our simulations, we also find shortest paths with global network information for comparison purpose.

All these geographic forwarding protocols planarize the network based on its Gabrial Graph for the detour mode in our simulation studies. The default forwarding direction in the detour mode takes the right-hand rule. The constant α for GDRP in determining perfect sequences is set to 2 empirically. the sink is located at one corner of the network area ($\frac{r}{2}$ away from two boundaries), while the source is located at its opposite corner (also $\frac{r}{2}$ away from two boundaries). Network holes and barriers are simulated by inserting ellipses and line segments into the network. The line segments cut the intersecting wireless links, while the ellipses are the areas in which there are no sensor nodes. The ellipses and the line segments do not intersect the network boundary to avoid geographic forwarding routes packets along network perimeters. Finally, the details of the simulation network settings are shown in Table 1, which are typical WSN settings. For each setting in our following performance studies, we adopt 60 different random seeds in every runs and the results are averaged. We do not consider the cases that geographic forwarding fails to deliver a packet, which is generally caused by the inserted holes and barriers that result in an unconnected network.

Table 1. Simulation Settings

Area of sensor field	$400m \times 400m$
Node deployment scheme	Randomly deployed in a uniform manner
Node communication range r	$[40m, 70m]$
Sensor node number	$[100, 500]$
Holes and barrier number	$[2, 8]$

We first study how the number of network holes and barriers influences the topological lengths of paths. Figure 7 demonstrates the results where the communication range is $60m$ and the number of sensor nodes is 500. It shows that as the number of holes and barriers increases, the paths found by all the protocols result in larger topological length. This is because with more holes and barriers in between, greedy forwarding faces higher chances to fail, which results in longer paths for bypassing more holes and barriers. We can see that our GDRP always outperforms GPSR and CONVEX-W. This verifies the advantage of the waypoint selection algorithm adopted in GDRP. Note that even when the hole and barrier number is large, unlike that of the other two protocols, the performance of GDRP does not dramatically deviate from the optimum shortest path. This shows the effectiveness of GDRP in surviving network holes and barriers.

We also study how GDRP performs with respect to the node density and node communication range. Six holes and barriers are injected into the network. Figure 8 shows the results where we let the communication range be $60m$ and change the node number from 100 to 500, and Figure 9 provides the results where we set the node number to 400 and change the communication range from 30 to 60.

It can be seen that the larger the node density or communication range is, the better all these protocols perform. This is quite natural: Larger node density or communication range result in larger per-hop progress in geographic forwarding [23]. Moreover, larger node density or communication range also decreases the chance that greedy for-

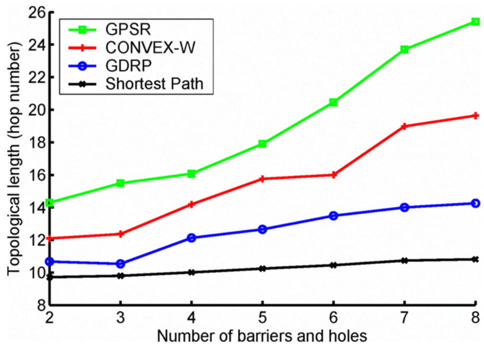


Figure 7. Topological lengths with different numbers of holes and barriers

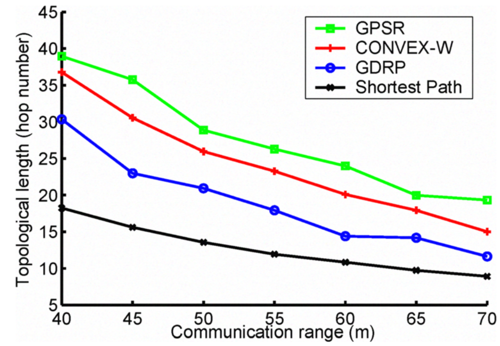


Figure 9. Topological lengths as a function of communication range

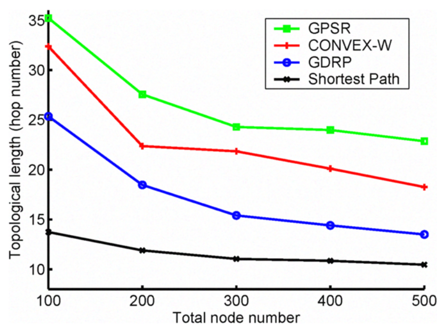


Figure 8. Topological lengths as a function of in-network node number

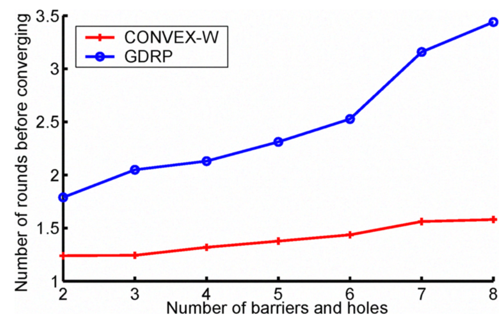


Figure 10. The average number of rounds needed before converging

warding fails since more neighbors are available as potential greedy forwarding candidates. These results further confirms that GDRP always outperforms GPSR and CONVEX-W in different network settings.

Finally, although GDRP performs well in terms of the topological lengths of the paths it finds, an important concern is the number of rounds it requires in finding an acceptable path. Figure 10 compares GDRP and CONVEX-W since CONVEX-W is also an iterative protocol. The communication range is $60m$ and the number of sensor nodes is 500 in this study. It shows that although GDRP needs more rounds to converge than CONVEX-W does, the round number is still small even when the hole and barrier number is high. Moreover, it do not grow dramatically with the hole and barrier number. This shows the nice tractability of the waypoint selection problem modeled in GDRP.

5. Related Work

Geographic routing is first proposed by Karnakis *et al* in [12]. Greedy Perimeter Stateless Routing (GPSR) and several other algorithms are subsequently proposed [4, 13]. Frey and Stojmenovic further show that recovery from a greedy forwarding failure is always possible without changing a face in the detour mode [9].

Face routing incurs a longer path [3]. Many schemes are proposed to improve face routing. Fang *et al* study how to locate network holes and propose to route packet along the boundary of a hole [7]. Leong *et al* present a geographic routing mechanism without planarizing the network [15]. These approaches generally need a protocol to obtain the information of the network holes and aim to minimize its overhead, resulting a more complicated implementation. They largely focus on holes and lack a scheme to handle network barriers. Furthermore, related work also includes those focusing on finding a geometric embedding of the network where greedy forwarding is always feasible [8, 14]); and those assigning the nodes virtual coordinates, via which data forwarding is conducted [18, 19].

Waypoint-based geographic forwarding is proposed in [3] and Huang [11] study how to select a set of waypoints adaptively. But the waypoint selection scheme may be trapped to suboptimal results. Moreover, Zhao *et al* [21] propose to conduct random shift to the locations of the waypoints to avoid some nodes are always selected.

Theoretical performances of greedy forwarding is widely studied. For example, Wan *et al* provide the asymptotic bounds of transmission range to ensure greedy forwarding [20]. Zorzi *et al* provide the bounds on hop-count and latency performance of greedy forwarding [22, 23].

6. Conclusion

Geographic forwarding has long been advocated as a promising technique in transporting sensor data in WSNs. However, in practical WSN deployments, network holes and barriers are inevitable. This poses a critical challenge to traditional geographic forwarding and consequently makes it very inefficient in terms of energy consumption. In this paper we aim to improve the energy efficiency of geographic forwarding, so as to enhance the survivability of the network in practical deployment environments. We address this problem by proposing a waypoint-based geographic forwarding approach called GDRP. We prove the performance guarantee of GDRP and verify its effectiveness in tolerating network holes and barriers with extensive simulation studies.

There are, however, many open problems for enhancing the energy efficiency of geographic forwarding. For example, a protocol that can tune the transmission power of sensor nodes to bypass holes and barriers are of great interest.

Acknowledgements

This work was substantially supported by a grant from the Research Grants Council of the Hong Kong S.A.R., China (Project No. CUHK4158/08E). J. Liu's work was supported in part by a Canadian NSERC Discovery Grant and an NSERC Strategic Project Grant.

References

[1] N. Ahmed, S. S. Kanhere, and S. Jha. The holes problem in wireless sensor networks: a survey. *ACM Mobile Computing and Communications Review*, 9(2):4–18, April 2005.

[2] A. Ailamaki, C. Faloutos, P. S. Fischbeck, M. J. Small, and J. VanBriesen. An environmental sensor network to determine drinking water quality and security. *Sigmod Rec.*, 32(4):47–52, 2003.

[3] L. Blazevic, J.-Y. L. Boudec, and S. Giordano. A location-based routing method for mobile ad hoc networks. *IEEE Transactions on Mobile Computing*, 4(2):97–110, March-April 2005.

[4] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. *Kluwer Wireless Networks*, 7(6):609–616, 2001.

[5] N. Bulusu, J. Heidemann, and D. Estrin. GPS-less low-cost outdoor localization for very small devices. *IEEE Personal Communication*, October 2000.

[6] D. Culler, D. Estrin, and M. Srivastava. Overview of sensor networks. *IEEE Computer Magazine*, 37(8):41–49, August 2004.

[7] Q. Fang, J. Gao, and L. J. Guibas. Locating and bypassing holes in sensor networks. In *Proc. of INFOCOM*, 2004.

[8] R. Flury, S. V. Pemmaraju, and R. Wattenhofer. Greedy routing with bounded stretch. In *Proc. of INFOCOM*, 2009.

[9] H. Frey and I. Stojmenovic. On delivery guarantees of face and combined greedy-face routing algorithms in ad hoc and sensor networks. In *Proc. of MOBICOM*, September 2006.

[10] J. Gao, L. J. Guibas, J. Hershberger, L. Zhang, and A. Zhu. Geometric spanner for routing in mobile networks. In *Proc. of MOBIHOC*, October 2001.

[11] H. Huang. Adaptive algorithms to mitigate inefficiency in greedy geographical routing. *IEEE Communication Letters*, 10(3):150–152, March 2006.

[12] E. Karnakis, H. Singh, and J. Urrutia. Compass routing on geometric networks. In *Proc. of the CCCG*, Aug 1999.

[13] B. Karp and H. T. Kung. GPSR: greedy perimeter stateless routing for wireless networks. In *Proc. of MOBICOM*, 2000.

[14] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger. Geometric ad hoc routing: of theory and practice. In *Proc. of PODC*, 2003.

[15] B. Leong, B. Liskov, and R. Morris. Geographic routing without planarization. In *Proc. of NSDI*, 2006.

[16] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson. Wireless sensor networks for habitat monitoring. In *Proc. of WSNA*, 2002.

[17] L. Selavo, A. Wood, Q. Cao, T. Sookoor, H. Liu, A. Srinivasan, Y. Wu, W. Kang, J. Stankovic, D. Young, and J. Porter. Luster: Wireless sensor network for environmental research. In *Proc. of the ACM Sensys*, November 2007.

[18] G. Tan, M. Bertier, and A.-M. Kermarrec. Convex partition of sensor networks and its use in virtual coordinate geographic routing. In *Proc. of INFOCOM*, 2009.

[19] M.-J. Tsai, F.-R. Wang, H.-Y. Yang, and Y.-P. Cheng. Virtualface: An algorithm to guarantee packet delivery of virtual-coordinate-based routing protocols in wireless sensor networks. In *Proc. of INFOCOM*, 2009.

[20] P.-J. Wan, C.-W. Yi, F. Yao, and X. Jia. Asymptotic critical transmission radius for greedy forwarding routing in wireless ad hoc networks. In *Proc. of MobiHoc*, May 2006.

[21] G. Zhao, X. Liu, and M.-T. Sun. Anchor-based geographic routing for sensor networks using projection distance. In *Proc. of ISWPC*, Febrary 2007.

[22] M. Zorzi and R. R. Rao. Geographic random forwarding (GeRaF) for ad hoc and sensor networks: Energy and latency performance. *IEEE Transactions on Mobile Computing*, 2(4):349–365, Oct.-Dec. 2003.

[23] M. Zorzi and R. R. Rao. Geographic random forwarding (GeRaF) for ad hoc and sensor networks: Multihop performance. *IEEE Transactions on Mobile Computing*, 2(4):337–347, Oct.-Dec. 2003.