HOSTED BY

ELSEVIER

Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/dcan

DCN Digital Communications and Networks
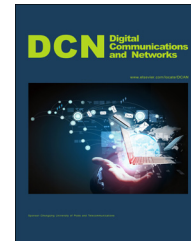
Improving routing efficiency through intermediate target based geographic routing

CrossMark

Zongming Fei[a,*], Jianjun Yang[b], Hui Lu[c]

[a]*Department of Computer Science, University of Kentucky, Lexington, KY 40506-0495, USA*
[b]*Department of Computer Science and Information Systems, University of North Georgia, Oakwood, GA 30566, USA*
[c]*School of Electronic and Information Engineering, Beihang University, Beijing 100191, China*

Abstract

The greedy strategy of geographical routing may cause the local minimum problem when there is a hole in the routing area. It depends on other strategies such as perimeter routing to find a detour path, which can be long and result in inefficiency of the routing protocol. In this paper, we propose a new approach called Intermediate Target based Geographic Routing (ITGR) to solve the long detour path problem. The basic idea is to use previous experience to determine the destination areas that are shaded by the holes. The novelty of the approach is that a single forwarding path can be used to determine a shaded area that may cover many destination nodes. We design an efficient method for the source to find out whether a destination node belongs to a shaded area. The source then selects an intermediate node as the tentative target and greedily forwards packets to it, which in turn forwards the packet to the final destination by greedy routing. ITGR can combine multiple shaded areas to improve the efficiency of representation and routing. We perform simulations and demonstrate that ITGR significantly reduces the routing path length, compared with existing geographic routing protocols.

1. Introduction

In wireless networks, a node can communicate with a nearby neighbor node directly. However, it is much more complicated when it needs to send messages to a destination node farther away out of the range of its wireless signal. In this situation, it relies on other nodes to relay its packets step by step until they reach the destination. Routing protocols [1-6] have been proposed to find a routing path from a source node to a destination node. They can be classified into proactive routing protocols and on-demand routing protocols depending on when the paths are determined. Proactive protocols, such as DSDV [1], TBRPF [2],

*Corresponding author.

E-mail addresses: fei@netlab.uky.edu (Z. Fei),
jianjun.yang@ung.edu (J. Yang), luhui.buaa@gmail.com (H. Lu).

and OLSR [3], exchange routing information periodically between hosts, and constantly maintain a set of available routes for all nodes on the network. In contrast, on-demand (or reactive) routing protocols, such as AODV [4], DSR [5], and TORA [6], delay route discovery until a particular route is required, and propagate routing information only on demand. There are also a few hybrid protocols, such as ZRP [7], HARP [8], and ZHLS [9], which combine proactive and reactive routing strategies. Most of these protocols involve broadcasting link state messages or request messages in order to find a path. The flooding of information can cause the scalability issue with these routing protocols.

Location information is used to simplify the routing process on wireless networks. Previous work has demonstrated that the location information can be obtained either through GPS or by using virtual coordinates [10-12]. Geographic routing exploits the location information and makes the routing in ad hoc networks scalable. The source node first acquires the location of the destination node it wants to communicate with, then forwards the packet to one of its neighbors that is closest to the destination. This process is repeated until the packet reaches the destination. A path is found via a series of independent local decisions rather than flooding. Each node only maintains information about its neighbors. However, geographic routing has to deal with the local minimum phenomenon, in which a packet may get stuck at a node that does not have a closer neighbor to the destination, even though there is a path from the source to the destination on the network. This typically happens when there is a void area (or hole) that has no active nodes. In wireless ad hoc networks, the holes are caused by various reasons [13]. For instance, malicious nodes can jam the communication to form jamming holes. If the signal of the nodes is not strong enough to cover everywhere on the network plane, coverage holes exist. Moreover, routing holes can be formed either due to voids in node deployment or because of failure of nodes due to various reasons such as a malfunction or battery depletion.

Many solutions have been proposed to deal with the local minimum problem. Karp and Kung proposed the Greedy Perimeter Stateless Routing (GPSR) protocol, which guarantees the delivery of the packet if a path exists [14]. When a packet is stuck at a node, the protocol will route the packet around the faces of the graph to get out of the local minimum. Several approaches were proposed that are originated from the face routing. Although they find the available routing paths, they often cause long detour paths. It is a hot topic to avoid the long detour path problem in the research community [15,16] and it is useful for a variety of applications [17].

To avoid such long detour paths, this paper proposes a new approach called Intermediate Target Based Geographic Routing (ITGR). The source determines destination areas which are shaded by the holes based on previous forwarding experience. It also records one or more intermediate nodes called landmark nodes and uses them as tentative targets. The routing path from the source node to the next tentative target is greedy. The routing paths from one tentative target to another and finally to the destination are greedy as well. Hence the total routing path is constructed by a series of greedy routing paths. The novelty of the approach is that a single forwarding path can be used to determine an

area that may cover many destination nodes. We design an efficient method for the source to find out whether a destination node belongs to a shaded area. Using intermediate nodes as tentative targets and greedily forwarding packets to them can avoid the original long detour paths. To further improve the efficiency of representation and routing, we design the mechanism for ITGR to combine multiple shaded areas. Simulations show that ITGR reduces routing path length by 17% and the number of forwarding hops by 15%, compared with GPSR.

The rest of the paper is organized as follows. Section 2 discusses related work on geographical routing and how the local minimum problem is dealt with. Section 3 proposes a novel method for detecting shaded areas and presents a new Intermediate Target based Geographic Routing protocol. It also presents the method for combining multiple cache entries to save the state information and reduce the search time. Section 4 evaluates the proposed schemes by simulations and describes performance results. Section 5 concludes the paper.

2. Related work

Many geographic routing protocols have been developed for ad hoc networks. In early protocols, each intermediate node on the network forwards packets to its neighbor closest to the destination, till the destination is reached. Packets are simply dropped when greedy forwarding causes them to end up at a local minimum node.

To solve the local minimum problem, geometric face routing algorithm (called Compass routing) [18] was proposed that guarantees packet delivery in most (but not all) networks. Several practical algorithms, which are variations of face routing, have since been developed. By combining greedy and face routing, Karp and Kung proposed the Greedy Perimeter Stateless Routing (GPSR) algorithm [14]. It consists of the greedy forwarding mode and the perimeter forwarding mode, which is applied in the regions where the greedy forwarding does not work. An enhanced algorithm, called Adaptive Face Routing (AFR), uses an ellipse to restrict the search area during routing so that in the worst case, the total routing cost is no worse than a constant factor of the cost for the optimal route [19]. The latest addition to the face routing related family is GPVFR, which improves routing efficiency by exploiting local face information [20].

To support geometric routing better in large wireless networks, several schemes were proposed to maintain geographic information on planar faces [21]. Gabriel Graph [22] and Relative Neighborhood Graph [23] are earlier sparse planar graphs constructed by planarization algorithms, with the assumption that the original graph is a unit-disk graph (UNG) [24]. Dense planar graphs are constructed from UNGs based on Delaunay triangulation [25]. The Cross-Link Detection Protocol (CLDP) [21] produces a subgraph on which face-routing-based algorithms are guaranteed to work correctly without making a unit-disk graph assumption. The key insight is that starting from a connected graph, nodes can independently probe each of their links using a right-hand rule to determine if the link crosses some other link on the network.

More recently, an idea based on the method of figuring out the void areas in advance was explored. A node keeps the coordinates of key nodes as well as the locations of its neighbors. The forwarding nodes will use the information to avoid approaching the holes [26-28]. Also related is GLR, a geographic routing scheme for large wireless ad hoc networks [29]. In the algorithm, once a source node sends packets to a destination node and meets a hole, the source node saves the location of the landmark node to its local cache. If any packet is to be forwarded to the *same* destination, the source node will forward the packet through the landmark. So each entry in the cache can only be used for a single destination node. In contrast, our approach learns from previous experience and generalizes it to cover an area of destination nodes. The number of nodes that can benefit from one cache entry can be orders of magnitude larger. Yet we design a simple way to represent the area and an efficient algorithm to decide whether a destination node is in the area.

3. Intermediate target based routing

3.1. The basic idea

We use a simple example to illustrate the basic idea of our approach. We assume that all nodes are static and distributed in a two dimensional space. As shown in Fig. 1, we assume that S is the source node and D_1, D_2 and D_3 are three different destination nodes. When S wants to send packets to D_1, it can find an efficient path by greedy forwarding.

However, when S wants to send a packet to D_2, it uses the greedy forwarding and the packet will reach node P. Because of the existence of the void area, P is closer to D_2 than all of the P's neighbors. So P cannot reach D_2 by greedy forwarding and is called a *local minimum node*. Fortunately, we have various routing algorithms [14] to let P change from the greedy mode to the perimeter routing mode. The packet will be forwarded along a detour path until it arrives at node B, where the forwarding mode is changed from the perimeter routing mode to greedy forwarding. Node B is called a *landmark node*. After node B, the packet can be forwarded to destination D_2 by greedy forwarding. Because D_2 is shaded by the hole, the original
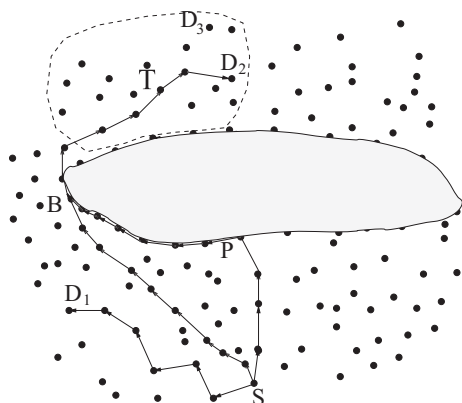
simple greedy forwarding has to take a detour. This detour path can be long.

To deal with routing inefficiency caused by the detour, we can let either destination node D_2 or landmark node B inform source S that such a detour occurred. After receiving the message, S keeps a record that associates D_2 with B, meaning that if the destination is D_2, forward through intermediate node B. After that, if S later needs to send packets to D_2, it can send them to B first (using B as an intermediate target) by greedy forwarding. The path will be from S to B and then to D_2, instead of from S to P, to B, and then to D_2. This new path can be much shorter and may be the best path to get to D_2 from S. The significance of the technique depends on how likely S needs to send packets to D_2 again.

Now consider that S needs to send a packet to D_3. Most likely, it will be forwarded to P by greedy forwarding, then go through a detour using perimeter routing to B, and finally reach D_3. The question we are interested in is whether the detour information about D_2 can be used to guide the forwarding by S for packets to D_3. In other words, can we generalize the strategy of using the intermediate node B for packet forwarding from the single destination node D_2 to multiple nodes?

The basic idea of this paper is to find a shaded area T such that for any destination node $D \in T$, source node S can benefit from using B as an intermediate target. Packets will be forwarded from S to B using greedy forwarding and then B will relay the packets to the final destination using greedy forwarding. The challenge is to find a simple representation of shaded area T and an efficient algorithm to determine whether a target node D is in the shaded area.

3.2. Shaded area

The shaded area for source node S can be determined by the locations of the local minimum node P, the landmark node B and the source node S. This is a learning process for S when it finds out that its packets are sent over a detour path. When a packet arrives at a node in perimeter mode, this node will determine whether it is a landmark node by checking whether it should change the forwarding mode to Greedy. If it is a landmark node, it will inform the source node of its own location (B) and the location of the local minimum node P (recorded in the packet).

When S learns the locations of B and P, we define a shaded area as shown in Fig. 2. We connect S with P using a
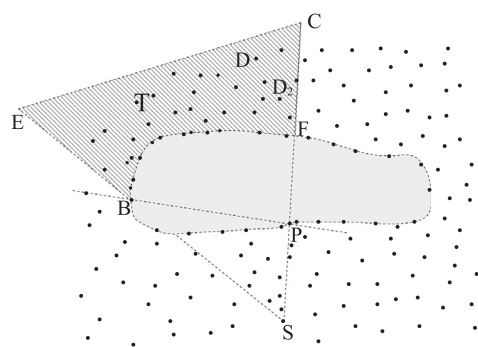

Fig. 1 Greedy path and detour path.

Fig. 2 The shaded area.

straight line and extend it to intersect with the hole at another point F. Ray SF further extends to some point C. We connect S with B using a straight line and extend it to some point E. Then the area semi-enclosed by EB, the perimeter from B to F and FC is the shaded destination area T. Hence, if S needs to send packets to any destination node D in T, the destination is hidden behind the hole. To avoid a detour path, S sends the packets to B first, and B will then relay them to D. Both paths can be greedy paths. We observe that for some destination node $D' \in T$, the greedy forwarding from S to D' may be stuck at a different local minimum node (other than P). However, forwarding to B first can still benefit by having a shorter path than going through the local minimum node.

Given a destination node D, we need to determine whether it is in the destination area T. As shown in Fig. 2, the area is enclosed by rays and partial edges of the hole polygon. To simplify the calculation, our first step is to extend the shaded area to include the area enclosed by line BP, line PF and arc BF, since it is in the void area and has no active nodes. The new destination area becomes the area semi-enclosed by $EBPC$.

After this extension, the determination of a destination node D in the shaded area becomes simple. If a destination node D satisfies the following conditions, it must be located in the shaded area.

(1) D and P are located on the same side of line SB;
(2) D and B are located on the same side of line SP; and
(3) D and S are located on the opposite sides of line BP.

Suppose that the coordinates of nodes S, B and P are $S(x_s, y_s)$, $B(x_b, y_b)$ and $P(x_p, y_p)$, respectively. Line SB can be described by the following equation:

$$\frac{y - y_s}{y_b - y_s} = \frac{x - x_s}{x_b - x_s}.$$

It can be written as

$$(y_b - y_s)x - (x_b - x_s)y + (x_b y_s - x_s y_b) = 0. \tag{1}$$

Let $f_1(x, y) = (y_b - y_s)x - (x_b - x_s)y + (x_b y_s - x_s y_b)$. Suppose D's coordinates are $D(x_d, y_d)$. D and P are located on the same side of line SB if and only if $f_1(x_d, y_d) * f_1(x_p, y_p) > 0$. To include the case of D being on line SB, we use

$$f_1(x_d, y_d) * f_1(x_p, y_p) \geq 0. \tag{2}$$

Similarly, we find the equation for line SP as

$$f_2(x, y) = (y_p - y_s)x - (x_p - x_s)y + (x_b y_s - x_s y_p) = 0, \tag{3}$$

and the equation for line BP as

$$f_3(x, y) = (y_b - y_p)x - (x_b - x_p)y + (x_b y_p - x_p y_b) = 0. \tag{4}$$

Nodes D and B are located on the same side of line SP if

$$f_2(x_d, y_d) * f_2(x_b, y_b) \geq 0. \tag{5}$$

Nodes D and S are located on the opposite sides of line SP if

$$f_3(x_d, y_d) * f_3(x_s, y_s) \leq 0. \tag{6}$$

If all three conditions Eqs. (2), (5) and (6) are met, node D is in the shaded area.

3.3. ITGR routing scheme

In ITGR routing, besides the source address S and the destination address D, a packet may contain a list of intermediate targets $\langle l_1, l_2, \ldots, l_k \rangle$, which will be called the *ITGR list* for the rest of the paper. We define *the target* $\mathcal{T}$ of a packet as either the first element on the ITGR list if the list exists, or the destination address D if the list does not exist. Similar to other geographic routing schemes, a packet forwarded in ITGR routing is either in Greedy mode or perimeter mode. Theoretically it can use any perimeter routing algorithm. However, for simplicity of presentation, we assume that GPSR is used. Therefore, perimeter mode will also be called *GPSR mode*. As stated in GPSR routing, packets in GPSR mode will contain the location of the local minimum node P, when forwarding is changed from Greedy mode to GPSR mode.

In ITGR routing, nodes have a local cache with entries representing shaded areas. Each shaded area is in the form of $\langle P_i, B_i \rangle$, where P_i is the location of the local minimum node and B_i is the location of the landmark node.

When source S needs to send a packet to destination D, it calls function ITGR_send(). As described in Fig. 3, ITGR_-send() first gets the target $\mathcal{T}$ of the packet. Next, it searches its local cache to see whether the target $\mathcal{T}$ is in any of the shaded areas. If yes, it extracts the landmark node (B_1). Then it uses this landmark node B_1 as the destination and searches whether it is in any shaded area. If it is, we get the landmark node B_2. This process will continue until we have a landmark node B_k not in any shaded area. Assume the list of landmark nodes we get is $B_1, B_2, \ldots, B_k$. If the packet does not contain an ITGR list, it creates one with elements $B_k, B_{k-1}, \ldots, B_1$. If the packet has an ITGR list, these elements are added in the front. We expect that in most cases, this list contains only one element B_1. After that, we need to set $\mathcal{T}$ to the value of the first element of the ITGR list.

As a last step, it forwards the packet to the neighbor that is closest to $\mathcal{T}$. If no neighbor is closer to $\mathcal{T}$ than the current node, it changes the packet to GPSR mode and follows the GPSR rules for forwarding (including putting the address of the current node as the local minimum node in

ITGR send()
    if the packet contains the ITGR list
        $\mathcal{T}$ = first element of the ITGR list;
    else $\mathcal{T} = D$;
    Search local cache;
    if $\mathcal{T}$ is in a shaded area
        Extract the list of landmark nodes $< B_k, \cdots, B_1 >$;
        if ITGR list exists
            Prepend $< B_k, \cdots, B_1 >$ to the list;
        else Create ITGR list $< B_k, \cdots, B_1 >$;
        $\mathcal{T} = B_k$;
    if there is a neighbor closer to $\mathcal{T}$;
        Greedy mode forwarding to the neighbor closest to $\mathcal{T}$;
    else
        Record the current node as the local minimum node in the packet;
        GPSR mode forwarding;

Fig. 3 ITGR Sending Algorithm.

```
ITGR process()
    if its address is equal to destination D
        Forwarding is finished and exit;
    if ITGR list exists and its address is equal
        to the first element of ITGR list
        Remove its address from the list;
        Call ITGR_send() to send the packet to next hop;
    elseif the packet is in Greedy mode forwarding
        Call ITGR_send() to send the packet to next hop;
    elseif the packet is in GPSR mode forwarding
        Set the value of 𝒯 as the target of the packet;
        if the current node has a neighbor closer to 𝒯
            Send a landmark_exist_msg to source S with
                local minimum node P and its own address
                as the landmark node;
            Change to Greedy mode forwarding and call ITGR send();
        else Continue GPSR forwarding;
```

Fig. 4   ITGR Processing Algorithm.

the packet). Note that *ITGR_send*() is not only used by the original source node, but will be used by other intermediate nodes along the path. In that case, it is called by the *ITGR_process*() function described in Fig. 4. The change from Greedy to GPSR mode is more likely to happen at those intermediate nodes than the original source node.

After a node receives a packet from a neighbor, it will process the packet. The node has to deal with several cases. It will be either the final destination node, the intermediate target node, the local minimum node, the landmark node, or other forwarding nodes on the path. In most cases, the node will call *ITGR_send*() to forward the packet to the next hop.

Fig. 4 describes processing algorithm *ITGR_process*() that a node will run after receiving a packet. It first checks whether its address is equal to destination *D*. If it is, the forwarding process is finished. Otherwise, it checks whether there is an ITGR list and whether its address is equal to the first element on the list. If that is the case, it is the intermediate target. Thus it removes itself from the list and then calls *ITGR_send*() to send the packet to the next hop.

Next, depending on the forwarding mode of the packet, *ITGR_process*() processes the packet differently. If the packet is in Greedy mode, the algorithm calls *ITGR_send*() to forward the packet to the next hop. If the packet is in GPSR mode, the algorithm will do GPSR processing. Specifically, if the condition of changing to Greedy mode is satisfied according to GPSR routing,[1] it will change the forwarding mode to Greedy. In addition to forwarding the packet by calling *ITGR_send*(), it sends a *landmark_exist_msg* to source *S* with the locations of the local minimum node *P* and its own (as the landmark). Otherwise, it continues GPSR forwarding.

When the source receives *landmark_exist_msg*, it will put the local minimum node *P* and the landmark node *B* as an entry in its local cache.

3.4.   Combining entries about shaded areas

If node *S* sends many packets to different destinations, several detour paths will be generated by the GPSR routing strategy. In this way, multiple entries with the format ⟨*LocalMinimum, Landmark*⟩ might be generated and saved in the cache of node *S*. Among these entries, some shaded areas may overlap with each other. They can have the same or different landmark nodes. Though these cache entries can be used in their original form, merging them saves space and facilitates efficient entry lookup. In this section, we investigate how multiple entries in the cache are combined.

Once node *S* receives a *landmark_exist_msg*⟨*P, B*⟩ from a landmark node, instead of inserting the new entry into the cache directly, it first looks up the entries in its local cache and possibly combines the new entry with an existing entry. There are two situations *S* needs to handle. One is that *S* finds an existing entry in its cache with the same landmark *B*. The other is that *S* finds an existing entry in its cache whose landmark is not *B*, but the corresponding shaded area overlaps with the shaded area of ⟨*P, B*⟩.

In the first situation, suppose that *S* finds the entry ⟨*P′, B*⟩ exists in its cache. *S* then updates its entries as follows[2]:

Case 1: ⟨*P′, B*⟩ ⊂ ⟨*P, B*⟩. This is the case in which *B* and *P* are on the opposite sides of *SP′* (Fig. 5). This scenario can be determined by the coordinates of these points as follows. Suppose the coordinates of points *S, B, P* and *P′ are* $S(x_s, y_s)$, $B(x_b, y_b)$, $P(x_p, y_p)$ and $P′(x_{p′}, y_{p′})$, respectively. Then the equation of line *SP′* is

$$(y_{p′} - y_s)x - (x_{p′} - x_s)y + (x_{p′}y_s - x_s y_{p′}) = 0.$$

Let $g_1(x, y) = (y_{p′} - y_s)x - (x_{p′} - x_s)y + (x_{p′}y_s - x_s y_{p′})$. Nodes *B* and *P* are located on the opposite sides of line *SP′* if

$$g_1(x_b, y_b) * g_1(x_p, y_p) \leq 0. \tag{7}$$

S updates the entries by removing ⟨*P′, B*⟩ and inserting ⟨*P, B*⟩.

Case 2: ⟨*P, B*⟩ ⊂ ⟨*P′, B*⟩. This is the case in which *B* and *P′* are on the opposite sides of *SP*. (Fig. 6). We also use the coordinates of the points and the equation of line *SP* to determine their relative locations. Because the existing entry ⟨*P′, B*⟩ covers the new entry ⟨*P, B*⟩, *S* simply discards ⟨*P, B*⟩.

The second situation is that *S* finds a new entry ⟨*P′, B′*⟩ related with ⟨*P, B*⟩, but they have two different landmarks *B* and *B′*. We discuss different scenarios in which their corresponding shaded areas overlap with each other. Otherwise, *S* can simply insert the new entry.

Case 1. ⟨*P′, B′*⟩ ⊂ ⟨*P, B*⟩. This is the case in which *B* and *P* are on the opposite sides of *SB′*, and *B* and *P* are on the opposite sides of *SP′* (Fig. 7). The update is that *S removes* ⟨*P′, B′*⟩ and then inserts ⟨*P, B*⟩. *S* does this update because ⟨*P, B*⟩ fully covers ⟨*P′, B′*⟩.

Case 2. ⟨*P, B*⟩ ⊂ ⟨*P′, B′*⟩. This is the case in which *B* and *P* are on the same side of *SB′*, and also on the same side of *SP′* (Fig. 8). In this scenario, *S discards* ⟨*P, B*⟩ because the area determined by the new entry ⟨*P, B*⟩ is covered by the existing entry ⟨*P′, B′*⟩.

Case 3. ⟨*P, B*⟩ and ⟨*P′, B′*⟩ are overlapped as follows. *B* and *P* are on opposite sides of *SB′*, and *B* and *P* are on the same side of *SP′* (Fig. 9). The update is that *S* keeps the entry ⟨*P′, B′*⟩ and inserts a new entry ⟨*B′, B*⟩. S does this

---

[1]One such condition is that the the forwarding node finds out that one of its neighbors is closer to D than itself.

[2]Note that ⟨*P, B*⟩ also represents the area determined by the entry ⟨*P, B*⟩.
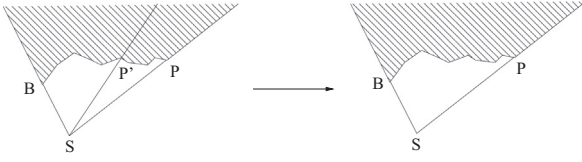
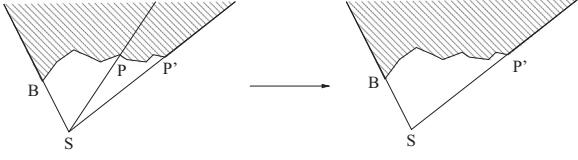**Fig. 5** Entry update: Case 1.

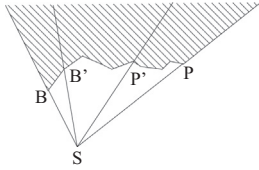**Fig. 6** Entry update: Case 2.

Fig. 7 Combining entries: Case 1.
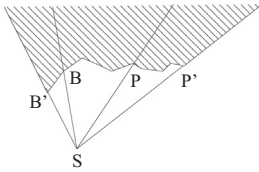
Fig. 8 Combining entries: Case 2.
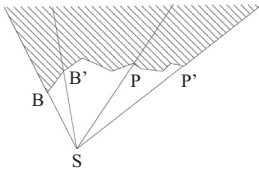
Fig. 9 Combining entries: Case 3.
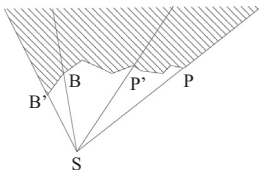
Fig. 10 Combining entries: Case 4.

Fig. 11 The average path length.

Fig. 12 The average number of hops.

4. Performance evaluation

We use the easim3D wireless network simulator [30] to evaluate the performance of the proposed mechanism. We use a noiseless immobile radio network environment with an area of 400 m × 400 m. Nodes distributed in the area have a transmission radius of 40 m.

We implemented both the GPSR routing protocol and our ITGR routing protocol using this simulation model. Two metrics, the length of routing path and the number of hops, are used. The number of nodes (density) varies from 50 to 300 with an increment of 50. For each case, 10 connected networks are generated with void areas set inside the network.

Fig. 11 shows the average path length when the number of nodes changes from 50 to 300. The average path length in ITGR is 17.52% shorter than that of GPSR when there are 50 nodes on the network. When the density of networks increases, the ITGR performs a little bit better. Fig. 12 shows the average number of hops with the number of nodes changing from 50 to 300. Similarly, the average number of hops in ITGR is 14.97% less than that of GPSR in the 50 node case. In both Figs. 11 and 12, the path length and the hop count with 50 nodes (both GPSR and ITGR) are much smaller than the other cases. This is because on the network plane, to guarantee the network's connectivity, 50 nodes have to

because the new entry $\langle P, B \rangle$ is considered as two areas: BSB' and $B'SP$. $B'SP$ is included in $B'SP'$, so only BSB' is inserted.

Case 4. $\langle P, B \rangle$ and $\langle P', B' \rangle$ are overlapped as follows. B and P are on the same side of SB', and B and P are on opposite sides of SP' (Fig. 10). The update is that S removes the entry $\langle P', B' \rangle$ and then inserts two new entries $\langle B, B' \rangle$ and $\langle P, B \rangle$.
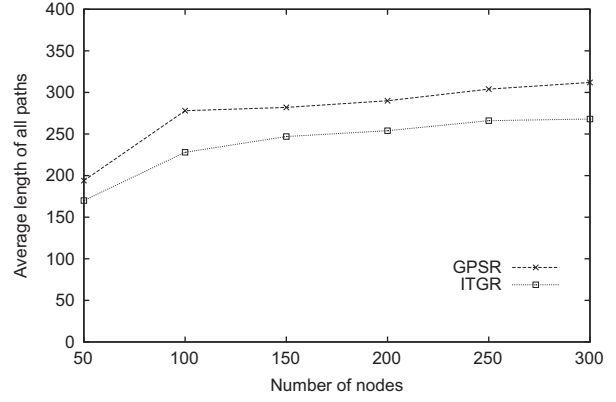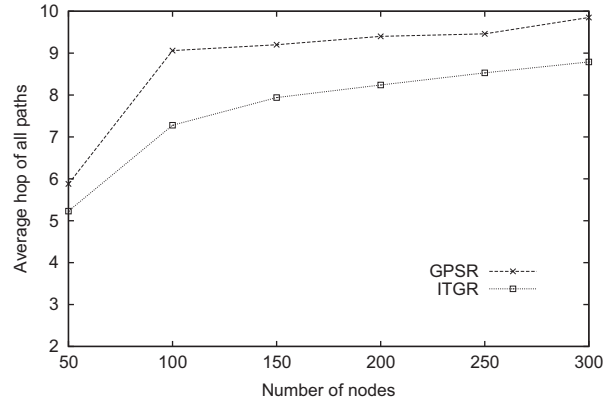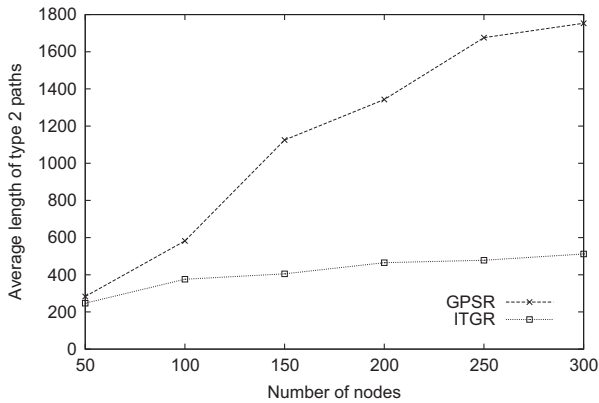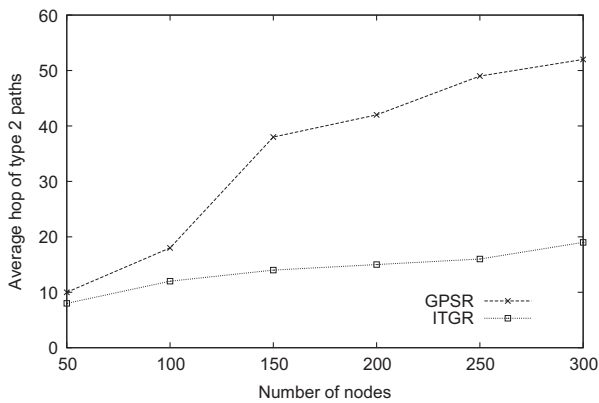
Table 1 The average percentage of type 2 paths over all paths.

The number of nodes	50	100	150	200	250	300
Percentage		23.2	21.1	18.7	17.6	16.4	16.2

Fig. 13 The average length of type 2 paths.

Fig. 14 The average number of hops of type 2 paths.

Fig. 15 The length of longest paths.

Fig. 16 The maximal number of hops of the paths.

Table 2 The average number of nodes with cache entries and percentages over all nodes.

Network size	50	100	150	200	250	300
Number of nodes with entries	4	9	16	18	21	25
Percentage (%)	8.0	9.0	10.67	9.0	8.4	8.3

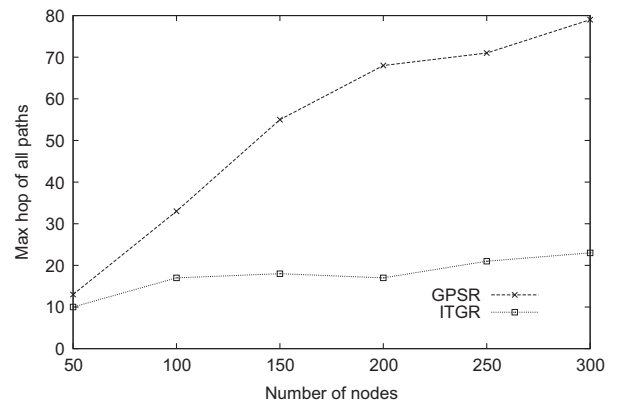be distributed in a relatively small area. This results in the shorter length and smaller number of hops.

To further illustrate the effect of ITGR on path length and hop count, we divide the tested paths into two types. For a routing path in ITGR routing, if no node in this path uses ITGR list for routing, we call this path a type 1 path. Otherwise the path is a type 2 path. We collect the data for the paths when GPSR routing is used.

The percentage of type 2 paths over all paths are shown in Table 1. It ranges from 23.2% for 50 node networks to 16.2% for 300 node networks. The larger the number of nodes on the network, the smaller the percentage. This is explained as follows. In the simulations, the nodes are distributed in a plane with a fixed size. The sizes of holes in sparse networks are larger than those in dense networks. Therefore, more paths are affected by void areas when the number of nodes is small.

Figs. 13 and 14 compare the performance of type 2 paths only. Compared with GPSR, ITGR has much shorter paths and

fewer hops. The gap between ITGR and GPSR increases when the number of nodes on the network increases. This is because when the number of nodes is larger, detour paths generated by GPSR are longer. For type 2 paths, the average length of ITGR is only 29.5% that of GPSR and the number of hops is only 27.3% for 300 node networks. From these two figures, we see that ITGR shortens the long paths significantly.

One benefit from ITGR is the reduction of the long detour path. To see the effect more clearly, we are interested in observing the longest paths (measured either in length or in number of hops) in ITGR and GPSR. We compare the length of the longest paths generated by ITGR and GPSR in Fig. 15. When there are 50 nodes on the networks, we do not see much difference. However, when the number of nodes increases from 100 to 300, the length of the longest path generated by GPSR also increases from 2 times to almost 5 times the length of the longest path generated by ITGR. In Fig. 16, we compare the maximal number of hops of the
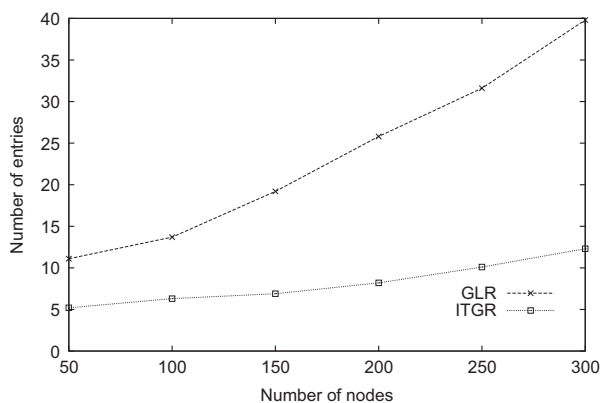
Fig. 17 The overheads of ITGR and GLR.

Fig. 18 The average length of paths of ITGR and GLR.

paths in ITGR and GPSR. We see a similar pattern. When the number of nodes increases, the difference between GPSR and ITGR becomes larger. Hence ITGR avoids most of the long detour paths resulting from GPSR.

ITGR is a hybrid protocol containing both proactive and reactive aspects. The proactive operation is to save the $\langle LocalMinimum, \ Landmark \rangle$ entries to a local cache. From our experiments, we find that the number of nodes that save the entries is not large, relative to the number of all nodes in the networks. Table 2 shows that the number of nodes with cache entries when the number of nodes on the networks changes from 50 to 300. It shows that the number of nodes with entries is about 10% of the total number of nodes on the networks.

Finally, we examine the control overhead of ITGR, by comparing it with GLR. The control overhead is measured in terms of the number of cache entries saved in the nodes. We calculate the number of cache entries stored at each node. The overall overhead is the summation of these numbers. Fig. 17 shows that the overhead of ITGR is much smaller than that of GLR. When the number of nodes on the network increases from 50 to 300, the difference in number of entries between the two schemes becomes larger. Since the entry of GLR is in the format of $\langle B_i, \ D_i \rangle$, GLR has to save an entry for almost every destination node hidden behind a hole. On the contrary, the entry of ITGR is in the form of $\langle P_i, \ B_i \rangle$, which can cover an area containing many destination nodes.

To compare the performance of ITGR and GLR in terms of path length, we randomly generate 100 networks with 150 nodes each. In each network, 100 pairs of source and destination nodes are randomly selected. Since both schemes use previous experience to improve the performance of future transmissions, sending to the same destination multiple times will get better results. Therefore, for each pair of nodes, we present the results when the source repeatedly sends a packet to the destination from once to 128 times. The average length of paths generated by all the 100 pairs of nodes on all the 100 networks are reported in Fig. 18. The average length of paths of GLR is a little shorter than that of ITGR only when the number of repeatedly sending times is larger than 16, but not significantly. When the number of the repeatedly sending times is less than 16, ITGR generates shorter paths than GLR. This is because ITGR improves the routing
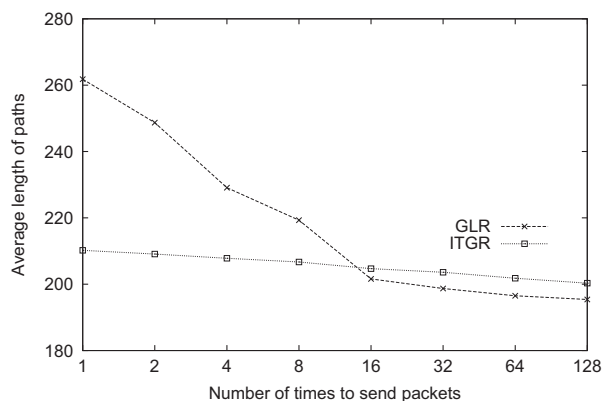
performance even if the source node has not sent a packet to the same destination before.

5. Conclusion

In this paper, we presented a new geographic routing approach called ITGR in order to avoid the long detour path. It detects the destination areas that might be shaded by the holes from previous routing experience. Then it selects the landmarks as tentative targets to construct greedy sub-paths. The approach can be used to avoid local minimum nodes. We design the scheme in such a way that a single detour path to a given destination can be used to avoid the detour path to many destinations in the future. We demonstrated a simple representation used for determining whether a node is in the shaded area. We also developed a method to reduce the overhead at nodes by combining multiple entries into one. The simulations demonstrate that our approach can result in significant shorter routing path and fewer hops than an existing geographic routing algorithm.

Acknowledgment

References

[1] C. Perkins, P. Bhagwat, Highly dynamic destination sequenced distance vector routing for mobile computers, in: ACM SIG-COMM, 1994.

[2] M. Lewis, F. Templin, B. Bellur, R. Ogier, Topology broadcast based on reverse-path forwarding, IETF MANET Internet Draft, Nov. 2002.

[3] P. Jacquet, P. Muhlethaler, A. Qayyam, Optimized link-state routing protocol, IETF MANET Internet Draft, March 2002.

[4] E. Royer, C. Perkins, Ad-hoc on-demand distance vector routing, in: The Second IEEE Workshop on Mobile Computing Systems and Applications, 1999.

[5] D. Johnson, D. Maltz, Y. Hu, J. Jetcheva, The dynamic source routing protocol for mobile ad hoc networks, IETF MANET Internet Draft, Feb. 2002.

[6] V.D. Park, M.S. Corson, A highly adaptive distributed routing algorithm for mobile wireless networks, in: IEEE INFOCOM, 1997.

[7] Z.J. Haas, M.R. Pearlman, Performance of query control schemes for the zone routing protocol, IEEE/ACM Trans. Netw. 9 (4) (2001) 427–438.

[8] N. Navid, S. Wu, C. Bonnet, HARP: Hybrid ad hoc routing protocol, in: International Symposium on Telecommunications, 2001.

[9] M. Joa-Ng, I.-T. Lu, Peer-to-peer zone-based two-level link state routing for mobile ad hoc networks, IEEE J. Sel. Areas Commun. 17 (8) (1999) 1415–1425.

[10] S. Basagni, I. Chlamtac, V.R. Syrotiuk, B.A. Woodward, A distance routing effect algorithm for mobility (DREAM), in: ACM/IEEE MobiCom, 1998.

[11] Y.-B. Ko, N.H. Vaidya, Location-aided routing (LAR) in mobile ad hoc networks, in: ACM/IEEE MobiCom, 1998.

[12] J. Li, J. Jannotti, D. DeCouto, D. Karger, R. Morris, A scalable location service for geographic ad-hoc routing, in: The Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking, 2000.

[13] N. Ahmed, S. Kanhere, S. Jha, The holes problem in wireless sensor networks: a survey, SIGMOBILE Mob. Comput. Commun. Rev. 9, April 2005.

[14] B. Karp, H. Kung, GPSR: Greedy perimeter stateless routing for wireless networks, in: ACM/IEEE International Conference on Mobile Computing and Networking, 2000.

[15] J. Yang, Z. Fei, ITGR: Intermediate target based geographic routing, in: Proceedings of the 19th International Conference on Computer Communications and Networks (ICCCN 2010), Zurich, Switzerland, 2010, pp. 1–6.

[16] J. Yang, Z. Fei, HDAR: hole detection and adaptive geographic routing for ad hoc networks, in: The Fourth IEEE International Workshop on Wireless Mesh and Ad Hoc Networks (WiMAN 2010), Zurich, Switzerland, 2010.

[17] J. Yang, Z. Fei, Broadcasting with prediction and selective forwarding in vehicular networks, Int. J. Distrib. Sensor Netw. 2013, http://dx.doi.org/10.1155/2013/309041.

[18] E. Kranakis, H. Singh, J. Urrutia, Compass routing on geometric networks, in: 11-th Canadian Conference on Computational Geometry, 1999.

[19] F. Kuhn, R. Wattenhofer, A. Zollinger, Asympotically optimal geometirc mobile ad-hoc routing, in: The Sixth international Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, 2002.

[20] B. Leong, S. Mitra, B. Liskov, Path vector face routing: geographic routing with local face information, in: The 13th IEEE International Conference on Network Protocols, Boston, MA, 2005.

[21] Y.-J. Kim, R. Govindan, B. Karp, S. Shenker, Geographic routing made practical, in: NSDI, 2005.

[22] K.R. Gabriel, R.R. Sokal, A new statistical approach to geographic variation analysis, Syst. Zool. 18 (3) (1969) 259–270.

[23] G.T. Toussaint, The relative neighborhood graph of finite planar set, Pattern Recognit. 4 (12) (1980) 261–268.

[24] B.N. Clark, C.J. Colbourn, Unit disk graphs, Discrete Math. 86 (1–3) (1990) 165–177.

[25] X.-Y. Li, G. Calinescu, P.-J. Wan, Distributed construction of a planar spanner and routing for ad hoc wireless networks, in: Proceedings of IEEE INFOCOM, New York, NY, 2002, pp. 1268–1277.

[26] F. Xing, Y. Xu, M. Zhao, K. Harfoush, HAGR: Hole aware greedy routing for geometric ad hoc networks, in: IEEE/AFCEA Military Communications Conference, 2007.

[27] Z. Jiang, J. Ma, W. Lou, J. Wu, An information model for geographic greedy forwarding in wireless ad-hoc sensor networks, in: INFOCOM. The 27th Conference on Computer Communications, IEEE, Phoenix, AZ, 2008, pp. 1499-1507.

[28] Q. Fang, J. Gao, L. Guibas, Locating and bypassing holes in sensor networks, Mobile Netw. Appl. 11 (2) (2006) 187–200.

[29] J. Na, C.-K. Kim, GLR: A novel geographic routing scheme for large wireless ad-hoc networks, Comput. Netw. 50 (2006) 3434-3448.

[30] C. Liu, EASIM 3D. ⟨http://www.oocities.org/gzcliu_proj/index.htm⟩, 2008.