Virtual Surrounding Face Geocasting in Wireless Ad Hoc and Sensor Networks

Jie Lian, *Member, IEEE*, Yunhao Liu, *Senior Member, IEEE*, Kshirasagar Naik, *Member, IEEE*, and
Lei Chen, *Member, IEEE*

***Abstract*—Geocasting in wireless sensor and ad hoc networks means delivering a message from a source node to all the nodes in a given geographical region. The objectives of a geocasting protocol are two-fold: guaranteed message delivery and low transmission cost. Most of the existing protocols do not guarantee message delivery, and those that do, incur high transmission costs.**

In this study, we propose the concept of Virtual Surrounding Face (VSF), and design a VSF-based geocasting protocol (VSFG). We also design a SKIP method and a local dominating set (DS) based restricted flooding technique to further reduce the cost of VSFG. Through mathematical analysis and comprehensive simulations, we show that VSFG, together with SKIP and local DS based restricted flooding, guarantees message delivery and has a much lower transmission cost than the previous approaches. The reduction of cost can be up to 65% compared with the most efficient existing approach.

***Index Terms*—Ad hoc networks, geocasting, virtual surrounding face, wireless sensor networks.**

I. INTRODUCTION

GEOCASTING in wireless sensor network is a task to deliver a message from a source node to all nodes located within a given geographic region. An important objective of geocasting is to ensure message delivery while maintaining a low transmission cost (lower number of transmissions). Guaranteed delivery ensures that every sensor in a region receives a copy of the geocasting message. Since sensors are generally powered by batteries, the limited energy of sensors requires geocasting to consume as little energy as possible. Many algorithms have been proposed in the literature [9]–[18] to achieve geocasting. The approaches presented in [9]–[16] do not guarantee message delivery and incur high transmission costs. Of the existing approaches, four algorithms—one in [17] and three in [18]—guarantee message delivery in continuous geocasting regions. Those algorithms, however, incur high transmission costs.

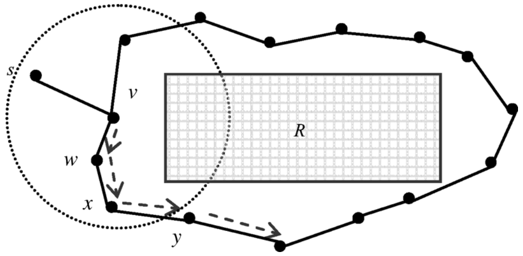


Fig. 1. Inefficiency of face traversal in the existing approaches. In the figure, the dotted circle denotes the transmission radius of v.

In this paper, we propose a geocasting algorithm based on the idea of Virtual Surrounding Face (VSF), and we refer to this algorithm as VSF Geocasting (VSFG). We prove that VSFG guarantees message delivery to the nodes within a geocasting region. In addition, the transmission cost of VSFG is significantly reduced compared with the existing approaches. Guaranteed message delivery in a connected network means that the message can be delivered from any source to any destination with an assumption: for any two neighbor nodes, a MAC layer protocol, such as 802.11a/b and [25], [27], exists to guarantee correct message exchange between them.

In VSFG, a network topology is converted into a planar graph where no two edges cross one another. The network area is partitioned into a set of faces, where a face is a continuous area enclosed by a sequence of edges. In VSFG, all the faces intersecting with a geocasting region R are merged into a unique virtual surrounding face containing R. VSFG includes the following three steps: *VSF forwarding*, *VSF traversal*, and *restricted flooding*. In VSF forwarding, a source delivers a geocasting message to a node on the boundary of VSF, called a VSF node, by using location-based routing [3], [4]. In VSF traversal, the VSF node initiates a face traversal in which all the nodes on the VSF receive a copy of the message. Finally, in restricted flooding, nodes in R that overhear the face traversal message perform restricted flooding within R.

Many approaches [3], [4], [18] can be used for face traversal. Those approaches, however, are not efficient in terms of message complexity. As illustrated in Fig. 1, node v starts a face traversal along the paths $\mathrm{v} \rightarrow \mathrm{w} \rightarrow \mathrm{x} \rightarrow \mathrm{y} \rightarrow \ldots$. In the existing approaches, even though x is a direct neighbor of v, the message is sent from v to w, and then from w to x, introducing an extra transmission. One intuition is that in dense networks, these additional transmissions may be significant compared with the total number of transmissions for face traversal. To reduce the

Manuscript received December 30, 2006; revised August 27, 2007 and November 30, 2007; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor S. Das. First published July 25, 2008; current version published February 19, 2009. This work was supported in part by the NSERC (Canada), the National Basic Research Program of China (973 Program) under Grant 2006CB303000, the National High Technology Research and Development Program of China (863 Program) under Grant 2007AA01Z180, NSFC Key Project Grant 60533110 and Grant 60736016, the Hong Kong RGC Grant HKUST6169/07E and Grant N_HKUST614/07, and the HKUST Nansha Research Fund NRC06/07.EG01. A preliminary, shorter version of this paper was presented at the IEEE ICNP, November 2006.

J. Lian and K. Naik are with the Electrical and Computer Engineering Department, University of Waterloo, Waterloo, ON, N2L 3G1 Canada (e-mail: jlian@swen.uwaterloo.ca; knaik@swen.uwaterloo.ca).

Y. Liu and L. Chen are with the Computer Science Department, Hong Kong University of Science and Technology, Hong Kong, China (e-mail: liu@cse.ust.hk; leichen@cs.ust.hk).

Digital Object Identifier 10.1109/TNET.2008.927251

cost, we propose a SKIP method to allow nodes to skip intermediate nodes during traversal solely based on one-hop neighbors of the nodes.

In the restricted flooding phase in VSFG, every node within R receiving the message for the first time broadcasts the message to its direct neighbors. Neverthelss, restricted flooding has many drawbacks such as high cost and serious contention [19]. We thus design a local DS construction to achieve restricted flooding in VSFG.

The major contributions of this work are as follows.

1) We introduce the concept of VSF, and present an algorithm (VSFG) based on VSF to achieve geocasting with guaranteed message delivery.
2) We propose a SKIP algorithm to let nodes skip some intermediate nodes during face traversal. Few further propose a local DS-based restricted flooding algorithm to reduce transmissions compared with simple restricted flooding. VSFG combined with SKIP is called VSFG_1, and VSFG_1 combined with DS is called VSFG_2.
3) The RFIFT (Restricted Flooding with Intersected Face Traversal) geocasting [18] has the lowest transmission cost among all known existing algorithms. The message complexities of RFIFT is bounded by $3n + k$, where n is the number of nodes on the boundary of the faces intersecting a geocasting region R but not in R, and k is the total number of nodes within R. In our VSFG algorithm, the bound has been reduced to $2n + k$.
4) In VSFG_1 and VSFG_2, each node in a network only needs to maintain the information of its one-hop neighbors. We compare RFIFT, VSFG_1, and VSFG_2 through extensive simulations in different environments. From simulation results, VSFG_2 can reduce up to 65% of the total number of messages required by RFIFT.

The rest of this paper is organized as follows. In Section II, we review related work. We define some terms and describe the concept of VSF in Section III. The VSFG algorithm is discussed in Section IV. We present VSFG_1 and VSFG_2 in Sections V and VI, respectively. The performance of VSFG family is analyzed and evaluated in Sections VII and VIII. We conclude the work in Section IX.

II. RELATED WORK

Generally, geocasting algorithms [4], [17], [18] reduce transmission costs by using location-based routing to deliver a message to a node in a geocasting region R. The node in R then performs restricted flooding within R. Hence, we review three categories of related work: location-based routing, geocasting algorithms, and broadcasting algorithms.

A. Location-Based Routing

Location-based routing has been extensively studied in the literature [1]–[7]. In these techniques, every node in a network knows its geographic location and the locations of all its neighbors. When a source node transmits a message to a destination node with a known location, the source and all intermediate forwarding nodes make their routing decisions based solely on the destination location and the locations of their neighbors. Since the nodes are not required to maintain routing tables, the routing overhead is significantly reduced.

Finn [1] proposed the first formal location-based routing algorithm based on a *greedy principle*, in which each node chooses the neighbor closest to the destination as its next forwarding node. The algorithm fails if a *void* (a large sub-area without nodes) exists in the forwarding direction, that is, the message reaches an intermediate node that is closer to the destination than any of its neighbor nodes.

To ensure message delivery, face routing was introduced in [2]. In face routing, a planar graph derived from the network topology is used, and the network area is partitioned into a set of faces. To transmit a message from a source s to a destination t, the message traverses the face intersecting the line segment st from s to t. If an edge e on the boundary of the traversed face intersects with st and the intersecting point is closer to t than to s, the face, which is next to e and closer to t than the currently traversed face, is traversed. The process is repeated until t is found. Face routing ensures delivery with possible long forwarding paths [3], [4].

To find a routing path close to the optimal path, the Greedy-Face-Greedy (GFG) algorithm, combining greedy routing and face routing, is proposed [3], [4] and its correctness is proved in [30]. In GFG, nodes conduct greedy routing whenever it is possible. In the case when a void exists in the forwarding direction, face routing is used to send the message around the void. Hence, GFG guarantees message delivery and significantly reduces the path lengths. For dense networks, the average length of forwarding paths is approximately equal to that of the shortest hop path. However, both of these algorithms are not asymptotically optimal [6]. Adaptive Face Routing (AFR) [6] is the first GFG-like algorithm achieving the asymptotical optimality of routing path lengths. In a follow up paper [7], GOAFR^+ was proposed to improve average case efficiency.

B. Geocasting Algorithms

Geocasting can be easily achieved by flooding the network, thereby achieving guaranteed message delivery. However, flooding is not energy efficient since it requires at least N transmissions, where N is the total number of nodes in the network. Three classes of geocasting algorithms have been studied in the literature to reduce the flooding cost.

In the first class of algorithms, a restricted forwarding zone, covering both the source node and the geocasting region, is used to limit the scope of flooding [9], [12], [13], [16]. In Location-Based Multicast (LBM) [9], the minimum rectangle containing both the source and the geocasting region is chosen as the forwarding zone. Next, restricted flooding is performed by nodes within the forwarding zone. Two later approaches [12], [13], [16] using different forwarding zones were proposed to reduce the cost. The three algorithms incur high flooding costs since the forwarding zone may be much larger than the geocasting region. Moreover, these algorithms do not guarantee message delivery [18].

The second class of algorithms reduces the high flooding cost by using restricted forwarding zones and intelligent flooding techniques [8], [26]. However, these algorithms do not ensure the delivery of messages as discussed in [18].

In the third class, a geocasting is divided into two phases: location-based unicasting and restricted flooding. In the first phase, location-based routing is used to route a message from a source node to a node in the geocasting region. In the second phase, restricted flooding is performed by the nodes in the region. Generally, this approach reduces the transmission cost. There is, however, no guaranteed message delivery if the topology graph in the geocasting region is not connected.

Various algorithms combining the ideas of location-based unicasting and restricted flooding with face traversal have been proposed with guaranteed message delivery [4], [17], [18].

The first algorithm, called Depth-First Face Tree Traversal (DFFTT), was presented in [4], [18]. In the first phase, DFFTT uses GFG to deliver a geocasting message to a node in a geocasting region R. Then, a face tree covering all the faces that intersect with R is constructed. By traversing every node on the face tree, the message is delivered to all nodes in R.

The second algorithm RFIFT was proposed in [17], [18]. The first phase of RFIFT is identical to DFFTT. In the second phase, RFIFT performs restricted flooding within R and traverses all the faces intersecting R. Each face traversal is determined by a pair of nodes: internal border node and external border node. An *internal border node* is a node in R with a planar neighbor outside of R. Here, two nodes are *planar neighbors* if an edge connecting these two nodes belongs to the planarized network graph. Similarly, an *external border node* is a node outside R, but with a planar neighbor in R. In RFIFT, each internal border node performs traversal by using left-hand rule with respect to all of its planar neighbors that are external border nodes.

The third algorithm [18], namely Entrance Zone Multicasting-based Geocasting (EZMG), sub-divides the surrounding area of a region R into a set of entrance zones. Each source node sends a multicast message to all entrance zones. Each node in entrance zones receiving the message broadcasts the message, and all nodes in R that hear the message perform restricted flooding in R.

The preceding three algorithms guarantee message delivery, but they incur high transmission costs.

C. Broadcasting Algorithms

Broadcasting is a process to send a message to all nodes in a network. Efficient broadcasting algorithms can be modified and applied to reduce the cost of restricted flooding involved in the geocasting algorithms [17], [18]. A straightforward broadcasting can be achieved by using flooding. However, flooding has many drawbacks, such as high cost, contention, and serious message collision [19].

The first type of solutions is clustering-based broadcasting [25], [28], [29]. The algorithms achieve broadcasting by sending messages to voted cluster headers. The second type of solutions is the multipoint relay algorithm [31]. In this algorithm, each node relays the message only to a subset of 1-hop neighbors which cover all its two-hop neighbors. The third type of solutions is dominating set (DS) based algorithms [20], [21]. A DS of a network is defined as a set of nodes such that for any node in the network, the node either belongs to DS or has a direct neighbor in DS. A connected dominating set (CDS) is a DS such that for any two nodes in CDS, there is a path connecting the two nodes and all nodes on the path belong to CDS. By constructing CDS of a network, flooding is performed only by the nodes belongs to CDS.

III. TERMINOLOGY AND VSFG

In this section, we present a network model and propose the concept of Virtual Surrounding Face (VSF).

A. Preliminary

Unit Disk Graph (UDG): UDG is a simplified model of wireless networks in which all nodes have an identical transmission range [4], [6], [7], [17], [18]. Let $UDG(V)$ denote a UDG, where V is a set of *nodes* whose transmission radii are normalized to 1. For a node u, let $\odot(u)$ denote the unit disk centered at u. An edge e_{uv} between u and v exists *if and only if* the Euclidean distance $d(u,v)$ between u and v is not larger than 1. For e_{uv}, u and v are called *UDG neighbors*. We use $N_{UDG}(u)$ to denote the UDG neighbor set of u.

Planar Graph and Gabriel Graph (GG): Face routing plays an important role in routing and geocasting to guarantee message delivery. It can only be applied on a *planar graph* which is a graph with no two edges crossing one another. To planarize a $UDG(V)$, a sub-graph of $UDG(V)$, called a Gabriel graph (GG), is normally employed. Let $\odot(u,v)$ denote a disk with e_{uv} as a diameter. A Gabriel graph $GG(V)$ is: for any two nodes u and v, if $d(u,v) \leq 1$ and $\odot(u,v)$ does not contain any nodes other than u and v, then $e_{uv} \in GG(V)$. For $e_{uv} \in GG(V)$, u and v are called *GG neighbors*. An algorithm to find $GG(V)$ in [4] has a property: if $UDG(V)$ is connected, $GG(V)$ is connected. Let $N_{GG}(u)$ be the GG neighbor set of u.

Border Nodes of Geocasting Regions: For a geocasting region R, let V_R be the set of nodes within R. A node is *an internal node of* R if the node is located in R, and is called *an external node of* R otherwise. For an edge $e_{uv} \in GG(V)$ intersecting the boundary of R, u is called *an internal border node* if u is an internal node, or u is called an *external border node* if u is an external node. An edge e_{uv} is called *a crossing edge of* R if $e_{uv} \in GG(V)$ and e_{uv} intersects R's boundary. A crossing edge connecting two external border nodes of R is called *an external crossing edge*. Let $N_{cross-R}(u) = \{v | e_{uv}\ is\ a\ crossing\ edge\}$ denote a node set with each member and u forms a crossing edges of R.

Faces in Planar Graphs: The edges in a planar graph partition the network area into a set of faces [2], [3]. There are two types of faces: *interior faces* and *exterior faces*. The former is the continuous area bounded by one or more closed sequences of edges. The latter is the unbounded area outside the boundary of a network. In Fig. 2, the network area is partitioned into four faces, F_1, F_2 (dark grey area), F_3 (light grey area), and F_4 (exterior face). Face F_3 is bounded by two sequences of edges: an *outer boundary* and an *inner boundary*. The outer boundary is specified by the sequence of endpoints: $u \rightarrow u_1 \rightarrow u_2 \rightarrow \ldots \rightarrow u_{10} \rightarrow u_{11} \rightarrow u_{12} \rightarrow u_{11} \rightarrow u_{10} \rightarrow u_{13} \rightarrow y \rightarrow z \rightarrow u$. And, the inner boundary is: $v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_4 \rightarrow v_1$.

Face Traversal Rule: We employ *Right-Hand Rule* [1] and *Left-Hand Rule* to traverse a face. In the former, a person explores a face by keeping her right hand on the walls (edges) and she will eventually visit all edges on the face. In the latter, a

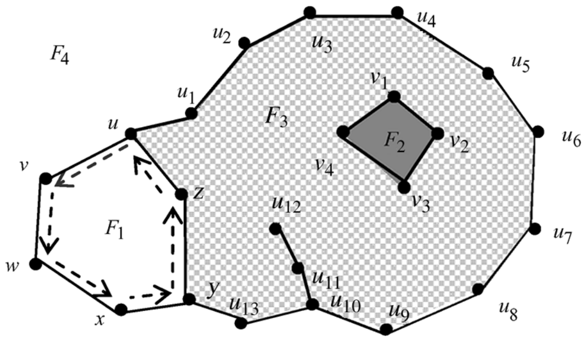


Fig. 2. Face partition and traversal. F_1, F_2, F_3, and F_4 are four faces.

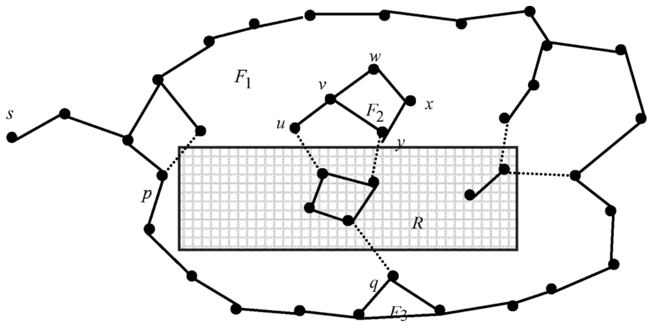


Fig. 3. Illustration of VSF. In the figure, s is the source of geocasting and R (shaded area) is the geocasting region. If all crossing edges (dotted lines) are ignored, all nodes in V_R are disconnected from the rest of nodes in $V - V_R$, where "-" is the set difference. Hence, the area immediately outside R is continuous and is the VSF of R.

person explores a face by keeping her left hand on the walls. We define face traversal illustrated in Fig. 2. Starting from u, to traverse F_1 by the Right-Hand Rule, u will send a message $trav(source, destination, rule)$ to v, where the source is the message sender, the destination is the message recipient, and the rule is either Right- or Left-Hand Rule. For node u, the message is $trav(u, v, Right)$. When v receives this message, v sends the message $trav(v, w, Right)$ to node w. By repeating the step, the message traverses F_1 counterclockwise. Similarly, $trav(u, z, Left)$ can be used to traverse F_1 clockwise.

In face traversal, some nodes may be visited more than once, which occurs when a face contains a dead-end. A *dead-end* of a face is a sub-path such that entering and exiting the sub-path can only be done through the same node. For example in Fig. 2, to traverse face F_3, the traversal path is: $\ldots \rightarrow u_9 \rightarrow u_{10} \rightarrow u_{11} \rightarrow u_{12} \rightarrow u_{11} \rightarrow u_{10} \rightarrow u_{13} \rightarrow \ldots$, in which u_{10} and u_{11} are in a dead end and are visited twice.

B. Basic Idea of Virtual Surrounding Face

For any two faces that share an edge, if the shared edge is ignored, the two faces are merged into one face with a larger area. For a geocasting region R, if we repeatedly merge all faces intersecting with R by ignoring the edges intersecting the boundary of R, we will eventually find a face large enough to contain R. This face is called a *virtual surrounding face* (VSF) of R. An example of VSF is illustrated in Fig. 3. A node on the boundary of a VSF is called a *VSF node*, and an edge on the VSF boundary is called a *VSF edge*. The objective of defining a VSF is as follows. To deliver a message to all the nodes in R, the message can be sent to one node on the boundary of the VSF. The message traverses the boundary of the VSF and each internal border node overhearing the traversal message performs restricted flooding within R. Then all the nodes in R will eventually receive the message.

IV. DISTRIBUTED VSF GEOCASTING ALGORITHM

In this section, we present the design of VSF geocasting (VSFG) which consists of the following three tasks.

- **VSF Forwarding**: A source node s transmits a geocasting message containing the specification of a region R to a node u on the boundary of the VSF by using location-based routing, such as GFG.
- **VSF Traversal**: Node u as chosen above starts VSF traversal. VSF traversal described in this section will be replaced by the SKIP technique given in Section V.
- **VSF Restricted Flooding**: During VSF traversal, each node in R overhearing the traversal message for the first time performs restricted flooding within R. Restricted flooding presented in this section will be replaced by the DS-based restricted flooding given in Section VI.

Let $\mathrm{MSG}(s, R, [option], \ldots)$ be a message containing the source s and a region R. The *option* field contains the task-related information. Each node u knows its own location and the locations of all neighbors in $N_{UDG}(u)$. We assume that all nodes do not change their locations during the geocasting task.

A. VSF Forwarding

VSF forwarding uses location-based routing to deliver a message $\mathrm{MSG}(s, R, [option])$ to a VSF node. Similar to the existing approaches [17], [18], we select a destination reference point r to guide VSF forwarding. The point r is chosen as the geographic point in R with the shortest distance to s. Once a node u receives a message designated for it, u determines if *it* is a VSF node by the following Lemma. All proofs in the paper are ignored due to the space limitation.

Lemma 1: If a node u is an external node of R and an end point of a crossing edge of R, then u is a VSF node.

In Fig. 3, since the up-left corner of R is selected as the reference point, node p receives a forwarding message. By Lemma 1, p finds itself to be a VSF node, and starts the VSF traversal. To achieve VSF forwarding, GFG [3] is modified by checking if there is a crossing edge of each forwarding node with R. The modified GFG guarantees to find a VSF node.

B. VSF Traversal

For a network with a node set V, VSF traversal associated with a geocasting region R is performed on top of $GG(V)$. In other words, each node u involved in face traversal computes u's next traversed node based on $N_{GG}(u)$, and ignores all crossing edges with u as one endpoint.

All the VSF nodes may not be fully connected by VSF edges as discussed in Cases 1 and 2 below. When this situation occurs,

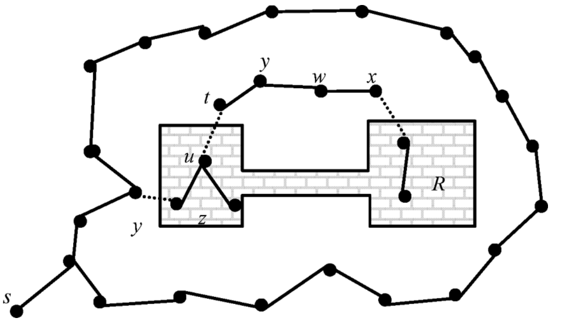


Fig. 4. VSF boundary connected via a node in the geocasting region R.

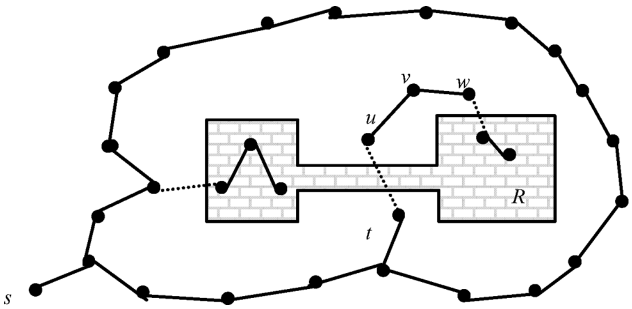


Fig. 5. VSF boundary connected via a crossing edge of the region R.

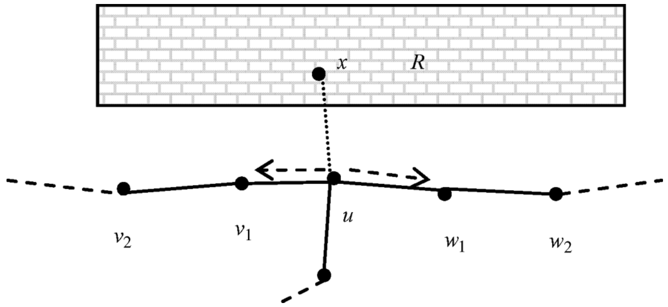


Fig. 6. Case 1: the entrance node u has two VSF neighbor nodes v_1 and w_1.

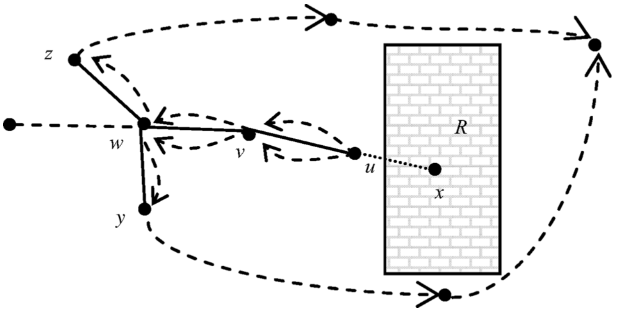


Fig. 7. Case 2: the entrance node u has only one VSF neighbor node v.

the message must go through some nodes in the geocasting region to complete VSF traversal. Two cases are associated with this situation and handled as follows.

Case 1) VSF nodes are connected via a crossing edge that connects two internal and external border nodes. Fig. 4 shows this case, where boundary $tvwx$ is connected to the outer boundary via path $yzut$. When t, which does not receive a traversal message, overhears a flooding message from u, t starts its own face traversal.

Case 2) The VSF nodes are connected via an external crossing edge. Fig. 5 illustrates this case, where VSF boundary uvw is connected with the outer face boundary via e_{tu}, which is ignored during VSF traversal. In this case, when u overhears the traversal message that is sent from node t and is designated for another node for the first time, u starts its own face traversal if e_{tu} intersects R.

The node selected during VSF forwarding starts VSF traversal and is called *an entrance node*. Each entrance node u traverse a face by using a message $\mathrm{MSG}(s, R, u, trav(\ldots))$, where $trav(\ldots)$ is the traversal method in Section III-A.

1) Initiation of VSF Traversal: To reduce traversal time and guarantee delivery, each entrance node simultaneously initiates a VSF traversal in two directions by using the Right-Hand and the Left-Hand Rule. Two possible starting cases are shown in Figs. 6 and 7.

In Fig. 6, entrance node u with two VSF neighbors v_1 and w_1 can find the next traversal node on the VSF by ignoring the crossing edge e_{ux} of R. Then, u sends $MSG(s, R, u, trav(u, v_1, Left))$ to v_1, and $MSG(s, R, u, trav(u, w_1, Right))$ to w_1. When v_1 receives the message designated for itself, v_1 knows itself to be a VSF node and forwards $MSG(s, R, u, trav(v_1, v_2, Left))$ to v_2. Similar steps are repeated until the termination condition, to be given later in this section, is satisfied.

In Fig. 7, entrance node u with only one VSF neighbor v sends v a message $MSG(s, R, u, trav(u, v, Left - Right))$, where $Left - Right$ indicates to apply both Left- and Right-Hand Rules. When v receives MSG, since v has only one traversal node w, v modifies the message to $MSG(s, R, u, trav(v, w, Left - Right))$ and sends it to w. Once w receives the message, due to the Left-Right instruction in the message, and w having two VSF neighbors z and y, w sends $MSG(s, R, u, trav(w, z, Left))$ and $MSG(s, R, u, trav(w, y, Right))$ to z and y, respectively.

2) Termination of VSF Traversal: To prevent from having messages traversing a VSF many times, each VSF node uses a *termination condition* to decide if the received traversal message can be discarded. Let function $next(q, MSG)$ return the node which will be traversed next when node q receives a traversal message MSG. For example in Fig. 7, if w receives $MSG(s, R, u, trav(v, w, Left)$ from v, $next(w, MSG(s, R, u, trav(v, w, Left))) = z$. Then we have:

Termination Condition for VSF: Assume that a VSF node u receives a traversal message $MSG_1(s, R, i_1, trav(v_1, u, Rule_1)$ from node v_1 but u does not forward MSG_1 to other nodes yet. Once u receives another message $MSG_2(s, R, i_2, trav(v_2, u, Rule_2)$ from v_2, u terminates traversal if the following condition is TRUE: $(next(u, MSG_1) = v_2)$ AND $(next(u, MSG_2) = v_1)$ AND $(Rule_1 \neq Rule_2)$.

In the preceding termination condition, v_1 and v_2 may be the same node. One example of this case is shown in Fig. 7, in which node u receives two traversal messages from node v with different traversal rules. The VSF traversal is given in Algorithm 1.

Algorithm 1: VSF Traversal

```
For each node $u$ in $V - V_R$ receiving a MSG from node $v$;
1:  BEGIN
2:    if ( MSG is a face traversal message) {
3:      if ($u$ is the MSG recipient) {
4:        if (Termination condition is satisfied)
5:          if ($u$ has not forwarded any traversing message and $u$ is an external border node) {
6:            $u$ broadcasts the content of MSG;
7:          }
8:          $u$ discards MSG;
9:        } else {            // Termination condition is not satisfied.
10:         $u$ forwards the MSG to the node $v$ = next($u$, MSG);
11:       }
12:     } else {              // $u$ is not the MSG recipient.
13:       if ($u$ receives MSG first time and $e_{uv}$ is an external crossing edge)  {
14:         $u$ starts its face traversal;
15:       } else {  $u$ discards MSG; }
16:     }
17:   } else if (MSG is a broadcasting message) {
18:     if ($u$ receives MSG first time and $e_{uv}$ is an crossing edge) {
19:       $u$ waits for a time period;
20:       if (timeout and $u$ does not receives face traversal MSG) {
21:         $u$ starts its face traversal;
22:       }
23:     } else { $u$ discards MSG; }
24:   }
25: END
```

It is possible that a node u receives two MSGs which satisfy the termination condition before u forwards any traversal message. In this case, u must broadcast the message once to guarantee delivery (Lines 5–8 of Algorithm 1).

C. VSF Restricted Flooding

During VSF traversal, for each node within a region R overhearing a geocasting MSG for the first time, the node performs restricted flooding. To reduce the cost in this phase, the DS-based restricted flooding (Section VI) is developed.

V. SKIPPING TECHNIQUE IN FACE TRAVERSAL

In dense networks, some nodes can skip intermediate nodes to reduce transmission cost during VFS traversal. We propose such an algorithm, namely SKIP, working solely based on the one-hop neighbor knowledge of each node.

A. Single Node Skipping Conditions

We first address the single node skipping by which a node can determine if it can skip one intermediate node. In a given $UDG(V)$, we assume that for an arbitrary node x with the knowledge of its one-hop neighbors, a node u is the next traversed node of x during VSF traversal. From viewpoint of x, whether x can skip u and sends the traversal message to another node v depends on the two conditions as follows:

- **Condition 1**: node x can determine whether u and v are Gabriel neighbors of each other.
- **Condition 2**: node x can determine whether v is the next traversed node with respect of u.

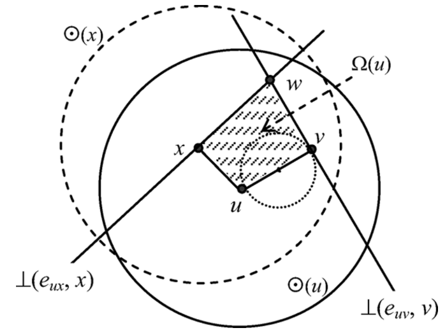


Fig. 8. Case 1: Decision region (shaded area) is a trapezium within $\odot(u)$.

Let $c(u, v)$ denote the center point of an edge e_{uv}. The following lemma addresses Condition 1.

Lemma 2: Let u and v be two neighbors of a node x. If $d(x, c(u, v)) = d(u, v)/2 \leq 1, x$ can determine whether u and v are Gabriel neighbors of each other.

The intuition behind Lemma 2 is that the condition $d(x, c(u, v)) = d(u, v)/2 \leq 1$ implies that disk $\odot(u, v)$ is fully contained by the unit disk $\odot(x)$ centered at x. Hence, x has sufficient knowledge to find if there is a node in $\odot(u, v)$.

Then we derive Condition 2. For an edge e_{uv} in $GG(V)$, let $\perp (e_{uv}, v)$ denote the perpendicular line of e_{uv} through v. As the example shown in Fig. 8, we assume that x is the node receiving a face traversal message. In a UDG, x knows the locations of all nodes in the unit disk $\odot(x)$. Furthermore, we assume that u is the next visited node with respect to x. Let $\overrightarrow{ux}$ denote a ray starting at node u through node x. To determine the next visited node after u, x scans an area by rotating $\overrightarrow{ux}$ clockwise (keeping u stationary) until find the first encountered node v not in the geocasting region R such that v is a Gabriel neighbor of u (based on Lemma 2). This step is called *GG neighbor scan process* of $\overrightarrow{ux}$ performed by x (denoted by $scan(\overrightarrow{ux}, x)$), and the angle $\angle xuv$ is called *scan angle*. Then we need to find a condition by which x can determine if v is the next visited node with respect to u based solely on x's local knowledge. Assume that v is the node obtained by using $scan(\overrightarrow{ux}, x)$ in Fig. 8. We draw two lines $\perp (e_{uv}, v)$ and $\perp (e_{ux}, x)$, and define *decision region of* u (denoted by $\Omega(u)$) by the two cases as follows.

- **Case 1 of $\boldsymbol{\Omega(u)}$**: If the two lines intersect at a point w within the scan angle and located in $\odot(u)$, the decision region $\Omega(u)$ is defined as the trapezium $wxuv$ (the shaded area in Fig. 8. Otherwise, $\Omega(u)$ is defined in Case 2.
- **Case 2 of $\boldsymbol{\Omega(u)}$**: Let w_1 be the intersection of $\perp (e_{uv}, v)$ and $\odot(u)$ in the scan angle $\angle xuv$, and w_2 the intersection of $\perp (e_{ux}, x)$ and $\odot(u)$ in $\angle xuv$. The decision region $\Omega(u)$ is defined as the area enclosed by line segments w_1v, vu, ux, xw_2, and arc w_2w_1 (the shaded area in Fig. 9).

Let nodes x, u and v be the respective nodes shown in Figs. 8 and 9, and R is the geocasting region. We give the single node skipping condition in Lemma 3 as follows.

Lemma 3: If decision region $\Omega(u)$ is fully contained in $\odot(x)$ and there is no node in $\Omega(u) - R, x$ can determine by using its local knowledge that v is the next visited node with respect to u, where $\Omega(u) - R$ is a sub-area of $\Omega(u)$ not in R.

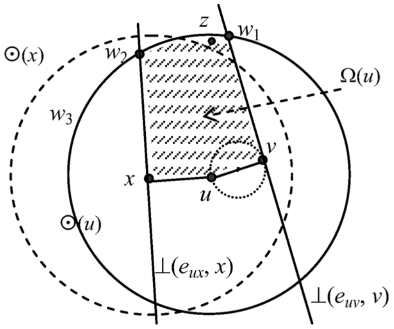


Fig. 9. Case 2: Decision region is bounded by an arc of $\odot(u)$.

Algorithm 2: SKIP

```
Input: A node $s$ which is the recipient of a face traversal message.
Output: A list $L$ of skipped nodes in the traversed order.
BEGIN
    $u \leftarrow$ the next visited node of $s$;    $x \leftarrow s$;
    for (true) {
        append $u$ at the end of $L$;
        $v \leftarrow$ the node found by the scan process $scan(\overrightarrow{ux}, s)$;
        if (($\Omega(u) \subset \odot(s)$) && (($\Omega(u) - R$) contains no node)) {
            $x \leftarrow u$;    $u \leftarrow v$;
        } else { return L; }
    }
END
```

B. Multiple Nodes Skipping Conditions

In many applications, nodes are densely deployed which makes multiple nodes skipping possible. Hence, we derive the multiple-node skipping conditions as follows. Assume that a node s holds a message and determines that s can skip $(k-1)$ nodes with current destination u. Then assume that the traversal sequence, which contains the $(k-1)$ nodes without skipping, is $s \to s_1 \to \ldots \to s_{k-2} \to x \to u$, where $x = s_{k-1}$. We have:

Lemma 4: For the sequence of nodes $s \to s_1 \to s_2 \to \ldots \to s_{k-2} \to x \to u$ given above, let v be the node found by the scan process $scan(\overrightarrow{ux}, s)$. Then if the decision region $\Omega(u)$ is fully located in $\odot(s)$ and there is no node located in $\Omega(u) - R$, s can determine skipping based on its local knowledge that v is the next visited node with respect to u.

Lemmas 3 and 4 lead to SKIP method in Algorithm 2.

C. Applying SKIP in VSFG

To apply SKIP in VSFG, we consider the following cases and modify Algorithm 1 accordingly.

Case 1) For a VSF node u receiving a traversal MSG, u executes Algorithm 2 to obtain the skipping list L. Then u forwards the MSG along with L to the node at the end of L.

Case 2) It is possible that some internal nodes are only connected to a VSF node which is skipped during traversal. The solution of the problem is as follows. Assume VSF node x sends MSG containing the skipping list L to v. For each node u in L, when u overhears MSG, u can find itself in L. Then u computes a set N' of nodes such that each node in N' is either a GG neighbor of u located in R or forms an external crossing edge with u. If all nodes in N' are UDG neighbors of x or v, u discards MSG. Otherwise, u broadcasts the MSG.

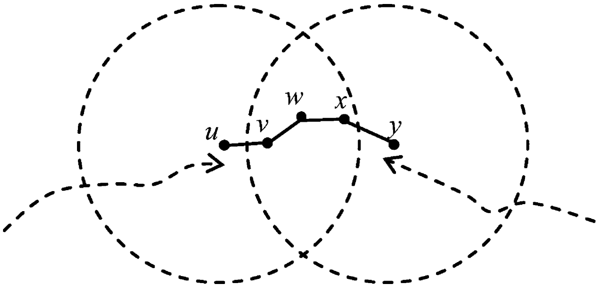


Fig. 10. Termination condition in double directional traversal with SKIP. In this figure, the dashed circles denote the transmission ranges of u and y, and the dashed arrows denote the traversal direction.

Case 3) Due to SKIP and double directional traversal, two traversal messages may not meet at the same node. This situation is illustrated in Fig. 10, in which u sends MSG_1 to x (skips v and w), and y sends MSG_2 to v (skip x and w). If we use the termination condition in Section IV, it is possible that face traversal will last forever. We remedy this problem by using the skipping list L contained in the MSGs. When v, w, and x overhear MSG_1 and MSG_2, they can extract L from the MSGs. By applying termination condition for nodes in L, v, w, and x can determine if they stop. For example, by using L in MSG_1, v finds that the next visited node is w without applying SKIP. Similarly, from MSG_2, v finds that the next visited node is u. Since traversal rules in the two messages are not same, v concludes that the termination condition holds.

By adding the results of Cases 1, 2, and 3 into Algorithm 1, VSFG is combined with SKIP and is called VSFG_1.

VI. DOMINATING-SET-BASED RESTRICTED FLOODING

The last step of VSFG is restricted flooding which, however, has significant drawbacks: high cost, contention, and message collision. We propose an algorithm to replace restricted flooding in VSFG to overcome the drawbacks.

When each node knows the locations of its neighbors, the algorithm in [20] and [21] allows the node to determine if it is in the DS without extra message exchange. This algorithm is designed to build a DS for the entire network, called a *global DS*. We can directly apply the algorithm in VSFG. However, for networks with stationary nodes, global DS incurs a load balancing problem. This is because the global DS is fixed and all broadcasts are only performed by the nodes in the DS. These nodes deplete energy much faster than the others. Instead of global DS, we construct local DS which varies for different geocasting regions and broadcasting orders of nodes.

Local DS construction is performed by each internal node u which overhears a geocasting MSG. Assume that each node has a unique ID. Once u overhears an MSG with a geocasting region R and u has not broadcasted the MSG yet, u computes and maintains four neighbor sets below:

- $N_{tx}(u)$: the set of UDG neighbors of u which have already transmitted the MSG (the MSG can be either a VSF traversal message or a broadcasting message with R).
- $N_{rx}(u)$: the set of UDG neighbors of u such that for $\forall v \in N_{rx}(u)$, v is an internal node of R and v has not transmitted the MSG yet.

Algorithm 3: Local DS-based Restricted Flooding

```
Input: for each node $u$ in $R$ overhearing MSG (either VSF traversal message or restricted flooding message) at the first time;
1:  BEGIN
2:    while ($u$ overhears MSG from $x$ with $R$) {
3:      if (($u$ is in $R$) == false) { $u$ ignores MSG and go to line 2; }
4:      $N_{tx}(u) \leftarrow N_{tx}(u) \cup \{x\}$; $u$ computes $N_{rx}(u)$, $N_{ext}(u)$, and $N_{int}(u)$;
5:      if ($N_{ext}(u) \cup N_{int}(u) == \emptyset$) { $u$ discards the MSG; exit; }
6:      for (every node $v$ in $N_{ext}(u) \cup N_{int}(u)$) {
7:        need-Broadcast ← true;
8:        for (every node $w$ in $N_{rx}(u)$) {
9:          if ($ID(w) > ID(u)$ && $v$ is a UDG neighbor of $w$) {
10:           need-Broadcast ← false; break;
11:         }
12:       }
13:       if (need-Broadcast = = true) { break; }
14:     }
15:     if (need-Broadcast == false) { $u$ discards the MSG; exit; }
16:     $u$ chooses a random back-off time $\Delta t$ and sets a timer;
17:     while ( the timer is not timeout ) {
18:       if ( $u$ receives a new MSG from a node not in $N_{tx}(u)$ {
19:         $u$ cancels the timer; goto Line 3;
20:       }
21:     } // timeout occurs
22:     $u$ broadcasts the MSG; exit;
23:   }
24: END
```

- $N_{ext}(u)$: the set of GG neighbors of u such that $\forall v \in N_{ext}(u), v$ has no UDG neighbor in $N_{tx}(u), e_{vu}$ forms a crossing edge, and v has not transmitted the MSG yet.
- $N_{int}(u)$: the set of UDG neighbors of u such that for each node v in $N_{int}(u), v$ is an internal node of R, v is not in $N_{tx}(u) \cup N_{rx}(u)$, and v has not transmitted the MSG yet.

Let $\mathrm{ID}(x)$ denote the ID of node x. Then u determines if it broadcasts the MSG or not by using the following two rules.

Rule 1) if $N_{ext}(u) \cup N_{int}(u)$ is empty, u discards the MSG. Otherwise, u performs Rule 2 as follows.

Rule 2) for every node $v \in N_{ext}(u) \cup N_{int}(u)$, if there exists a node w in $N_{rx}(u)$ such that $\mathrm{ID}(w) > \mathrm{ID}(u)$ and v is a UDG neighbor of w, then u discards the MSG. Otherwise, u belongs to the local DS and broadcasts the MSG.

Local DS-based restricted flooding is given in Algorithm 3. The intuitions behind Algorithm 3 are as follows. For each internal node u overhearing a MSG but not transmitting the MSG yet, u has a responsibility to transmit the MSG to the set of its neighbors which do not receive the MSG, and this set is called *a responsible set of* u. Obviously, the neighbors of u in $N_{tx}(u) \cup N_{rx}(u)$ have received the MSG. Hence, the responsible set of u is $N_{ext}(u) \cup N_{int}(u)$ which consists of neighbors of u without receiving the MSG based on u's local knowledge. However, if u can pass its responsibilities for all the nodes in $N_{ext}(u) \cup N_{int}(u)$ to other nodes in $N_{rx}(u), u$ needs not transmit the MSG. Rules 1 and 2 depict these situations.

Rule 1 shows a simple situation in which if all u's neighbors have received the MSG, u can simply discard the MSG. In Rule 2, for node v in $N_{ext}(u) \cup N_{int}(u)$, if u finds a neighbor w which has received the MSG, it is possible that u passes the responsibility for v to w. To avoid that u passes to w while w passes back to u, the IDs of u and w are used to break the loop. If the responsibilities of all nodes in u's responsible set can be passed to other nodes in $N_{rx}(u), u$ drops the MSG.

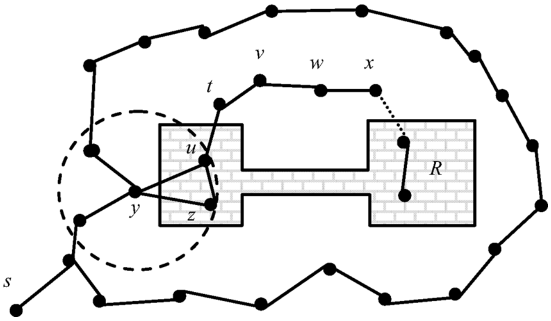


Fig. 11. Illustration of connection between outer and inner VSF pieces. In the figure, the dashed circle denotes the transmission range of node y.

We use nodes in $N_{ext}(u) \cup N_{int}(u)$ as the responsible set instead of $N_{int}(u)$ is because VSF of a region R may consist of several pieces which are connected only via nodes in R. In this case, message delivery is guaranteed only if we consider $N_{ext}(u) \cup N_{int}(u)$. Fig. 11 shows an example in which VSF piece $tvwx$ is connected via nodes u in R to the outer VSF piece. Since t is in $N_{ext}(u)$, if we only consider $N_{int}(u)$ without including $N_{ext}(u)$, u will not broadcast the MSG (based on Rule 1), resulting in that t, v, w, and x are not visited and all nodes in the right part of R do not receive the MSG.

VSFG$_1$ embedded with Algorithm 3 is named as VSFG$_2$. In VSFG$_2$, different local DSs are constructed for different geocasting regions. In addition, different transmitting orders of nodes result in different local DSs. Hence, VSFG$_2$ does not have the load balancing problem incurred in global DSs [21].

VII. PROPERTY AND PERFORMANCE ANALYSIS

In this section, we prove guaranteed message delivery for VSFG family, analyze asymptotic bounds, and compare performance of VSFG family with existing approaches.

A. Guaranteed Message Delivery in VSFG

Guaranteed message delivery is justified due to three properties below. In the discussions, assume that a network $UDG(V)$ is connected, and $GG(V)$ is the Gabriel graph.

Property 1: For every node outside a VSF transmitting a message to a node in a geocasting region R on top of $GG(V)$, the message must pass through at least one VSF node.

Let $\Psi = \{F_1, F_2, \ldots, F_k\}$ be the set of faces that intersect R on $GG(V)$, and V_Ψ be the set of nodes not in R but on the boundary of F_i for an arbitrary $i (1 \leq i \leq k)$. Let $V'_\Psi \subset V_\Psi$ be the node set in which each node is an end point of a crossing edge of R. Then we have Properties 2 and 3 as follows.

Property 2: Visiting all nodes in V'_Ψ and performing restricted flooding within R is sufficient to ensure delivery.

Property 3: Each node in V'_Ψ is located on the boundary of the VSF formed by VSFG.

By using the preceding three Properties, Theorem 1 can be obtained immediately as follows.

Theorem 1: VSFG guarantees message delivery.

B. Guaranteed Message Delivery in VSFG$_1$

Guaranteed message delivery of VSFG$_1$ is justified due to two VSF properties as follows.

Property 4: VSFG_1 guarantees delivery of the geocasting message to all VSF nodes of the geocasting region R.

Property 5: VSFG_1 ensures delivery of geocasting message to all internal border nodes of the geocasting region R.

From Properties 4 and 5, Theorem 2 is obtained as follows.

Theorem 2: VSFG_1 guarantees message delivery.

C. Guaranteed Message Delivery in VSFG₂

The difference of VSFG_1 and VSFG_2 is that VSFG_2 uses DS-based restricted flooding. By Property 4, VSFG_1 guarantees to visit all VSF nodes. We prove that DS-based restricted flooding can deliver messages to all nodes in R as the simple restricted flooding does in Theorem 3 as follows.

Theorem 3: VSFG_2 guarantees message delivery.

D. Performance Analysis of VSFG

Similar to the existing approaches [4], [17], [18], the total transmission cost C of VSFG is subdivided into three parts associated to the three phases as follows:

- VSF forwarding: Let C_u be the forwarding cost measured by the transmissions required in forwarding.
- VSF traversal: Let C_f be the traversal cost measured by the number of transmissions required to traverse VSF.
- Restricted flooding: Let C_r denote the total cost (number of transmissions) in restricted flooding.

Obviously, $C = C_u + C_f + C_r$. In the first phase, GFG is modified to find the entrance node. In the second phase, we give C_f in Theorem 4.

Theorem 4: The total number of transmissions C_f required in VSF traversal is bounded by $C_f \leq 2n$ in VSFG, VSFG_1, and VSFG_2, where n is the total number of VSF nodes.

In the restricted flooding phase, the worst case in VSFG family is that each node in the region broadcasts once. Let k be the number of nodes in the region. Then we have $C_r \leq k$.

E. Analytical Performance Comparison

RFIFT [17], [18] is the known most efficient algorithm with guaranteed message delivery, so we compare VSFG with RFIFT. RFIFT and VSFG have three similar phases, so we discuss the costs involved in these three phases separately.

In the forwarding phase, RFIFT chooses the center point of a geocasting region R as the destination reference point. In contrast, VSFG uses a point in R with the shortest distance to the geocasting source as the reference point. Hence, the path discovered in VSFG is slightly shorter than that in RFIFT.

In the face traversal phase, referring to the results shown in [18], the total number of transmissions in this phase is constrained by $3n'$, where n' is the number of nodes that are on the faces intersecting R. From Theorem 4, it is easy to show that $n \leq n'$, where n is the total number of VSF nodes in VSFG (also VSFG_1 and VSFG_2). Therefore, VSFG reduces the upper bound of the traversal cost from $3n$ in RFIFT to $2n$.

In the restricted flooding phase, since VSFG_2 uses local DS-based restricted flooding, its cost is much smaller than the cost in RFIFT for dense networks.

VIII. PERFORMANCE EVALUATION AND DISCUSSION

In this section, we compare the performance of RFIFT, VSFG_1, and VSFG_2. Due to the approximately identical unicasting costs C_u in the three algorithms, we do not show C_u individually. Instead, we use the traversal cost C_f, the flooding cost C_r, and the total cost C as performance metrics. Two sets of experimental results are presented in various network topologies with stationary nodes. First, we compare the three algorithms in networks with randomly distributed nodes. Second, we compare the algorithms in networks with randomly inserted voids, which represent some practical network topologies due to the existence of obstacles.

A. Simulation Results for Base Networks

The first experiment is done by using a routing-level simulator on randomly generated networks. In each sample network, nodes are randomly distributed in a 20×20 area such that the average degree is g. We vary the value of g to observe the impact of the network density on the number of transmissions. All nodes have an identical transmission radius of 1 unit. These sample networks are called *base networks*. For each g, 10 base networks are generated in the simulation. For each base network, we randomly generate 10 $W \times H$ rectangular geocasting regions. We also vary the values of W and H to observe the impact of sizes of regions on transmission costs.

Fig. 12(a)–(c) shows C_f, C_r, and C for geocasting regions with $W = 3$ and $H = 1.5$. The x axis denotes the average degree g of networks. The vertical bars in Fig. 12(a) correspond to 95% confidence interval of the mean value. To make the figure clear, we did not include confidence intervals in other figures and they are in a small range around the sample mean. Similarly, Fig. 13(a)–(c) shows C_f, C_r, and C for regions with $W = 5$ and $H = 2.5$. From these figures, we have the following observations.

First, from Figs. 12(a) and 13(a), C_f of VSFG_1 is identical to that of VSFG_2 and is much smaller than that of RFIFT. The higher the network density is, the higher the reduction percentage of C_f comparing with RFIFT. Reader may note that in Fig. 12(a), C_f in a network with $g = 10$ is higher than C_f in a network with $g = 15$ for the same algorithm. This is because in sparse networks, the probability of traversed faces containing the outer boundary of the entire network is higher, resulting in the high transmission cost.

Second, according to Figs. 12(b) and 13(b), VSFG_1 and RFIFT have identical costs of restricted flooding. On the other hand, VSFG_2 significantly reduces the flooding cost C_r of RFIFT. From the figures, we can observe that VSFG_2 has almost an identical C_r in networks with different densities. In RFIFT, C_r is proportional to the densities of networks.

Third, according to Figs. 12(c) and 13(c), VSFG_1 reduces the total cost C of RFIFT by 20%, and VSFG_2 reduces the total cost C of RFIFT by 30% to 65%.

For fixed geocasting regions, when the network density increases, the reduction percentage of C in VSFG_1 remains approximately unchanged comparing with RFIFT. In the same situations, the reduction percentage of C in VSFG_2 increases

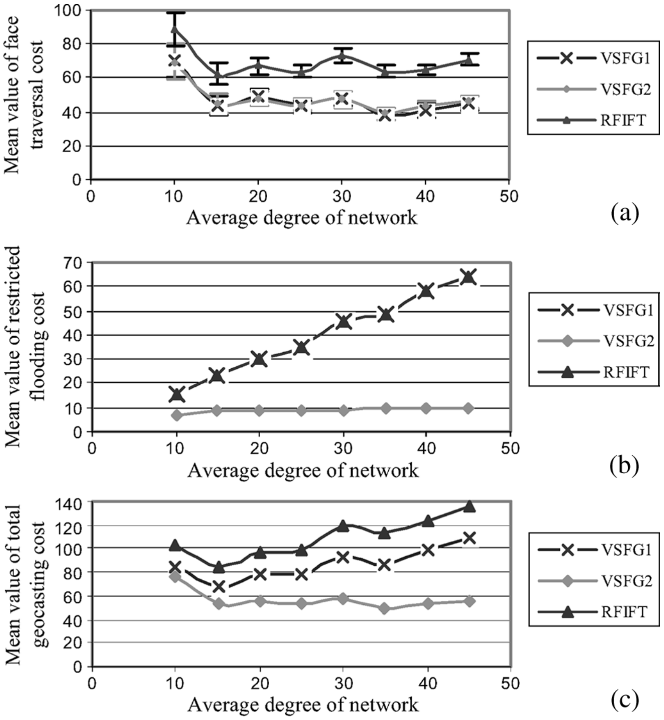


Fig. 12. Three costs for base networks with 3 × 1.5 geocasting regions.

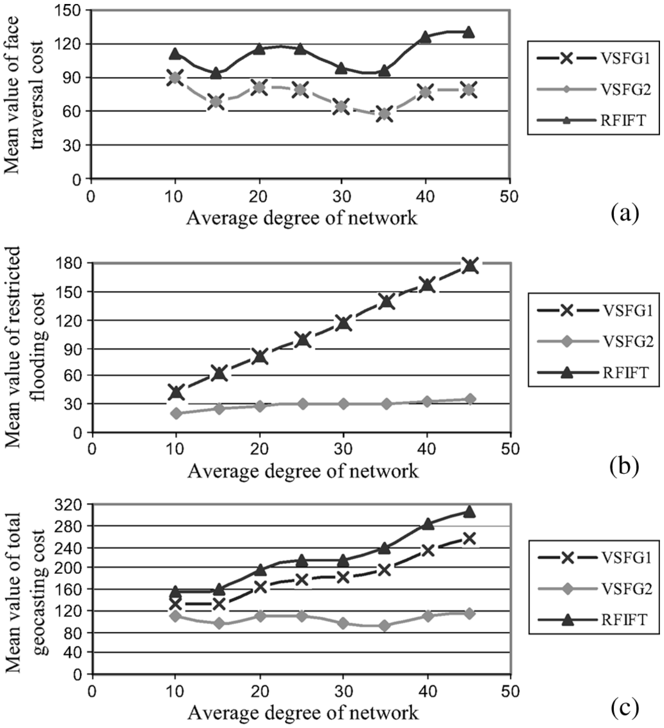


Fig. 13. Three costs for base networks with 5 × 2.5 geocasting regions.

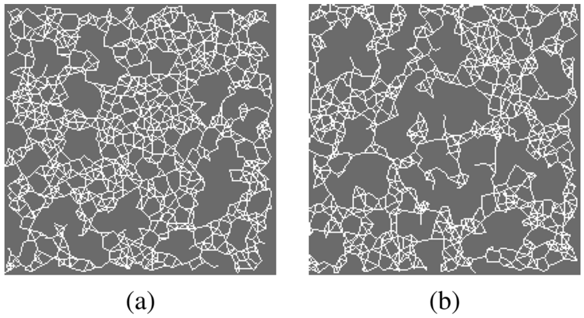


Fig. 14. Void networks generated from base network with $g = 10$. (a) Void network with 15 voids. (b) Void network with 30 voids.

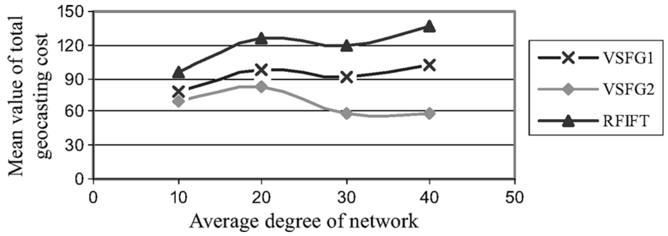


Fig. 15. Costs for void networks with 15 voids and 3 × 1.5 regions.

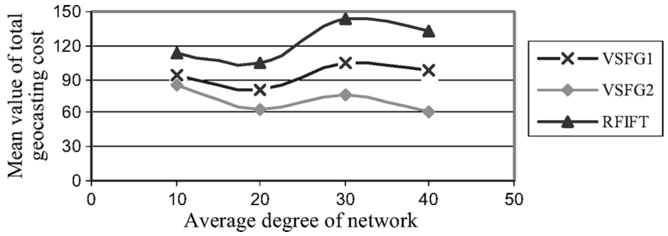


Fig. 16. Costs for void networks with 30 voids and 3 × 1.5 regions.

comparing with RFIFT. This is because VSFG_2 has an identical C_r in networks with different densities. C_r has higher impact on C than C_f for large regions, and C_r has less impact on C than C_f for small regions. When the size of region increases, the reduction percentage of C in VSFG_1 decreases slightly comparing with RFIFT. In the same situations, the reduction percentage of C in VSFG_2 increases. This is because for large regions, the cost C_r has a higher impact on C.

B. Simulation Results for Void Networks

In the second experiment, we evaluate the performance in networks with voids. Simulation is performed on a set of sample void networks generated from the base networks. For each base network, we randomly place a number of 1.5 × 1.5 square voids within the network area, and all the nodes in the voids are removed. The value of the void number is varied from 15 to 30. Fig. 14 shows two void networks with 15 and 30 voids. We use void networks because they represents some realistic networks due to node mobility and obstacles.

Figs. 15–18 show the simulation results in void networks. We only show the total costs of three algorithms since the face traversal cost and the restricted flooding cost follow the similar distributions shown in Figs. 12 and 13. From Figs. 15–18, we have following observations.

First, VSFG_1 reduces 20% to 25% of the total cost C involved in RFIFT, and VSFG_2 reduces 30% to 65% of C in RFIFT. For a fixed R and a fixed void number, the reduction percentage of C in VSFG_1 approximately remains unchanged with the increase of network densities comparing with RFIFT. In the same situations, VSFG_2 performs slightly better with the

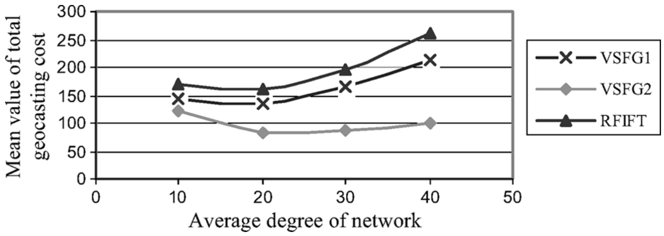


Fig. 17. Costs for void networks with 15 voids and 5 × 2.5 regions.

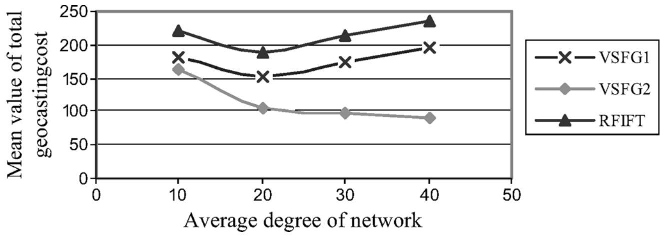


Fig. 18. Costs for void networks with 30 voids and 5 × 2.5 regions.

increase of network densities. This result is similar to that obtained in base networks, and the reason is similar as well.

Second, for a fixed geocasting region and a fixed network density, comparing with RFIFT, the reduction percentages of C for VSFG_1 and VSFG_2 decrease slightly when the number of voids in networks increases. Third, comparing with RFIFT, VSFG_1 and VSFG_2 can achieve a slightly higher performance gain in base networks than in void networks.

C. Applying VSFG in Practical Wireless Networks

VSFG fails in realistic wireless networks as it requires topology graphs to be planar [32]. In practical, the constructed Gabriel graphs by VSFG may not be planar. Several realistic test-beds [32] show such observations. However, VSFG is still useful and applicable in realistic environments.

First, many applications are deployed in environments with small obstacles. One realistic application is the contour map monitoring project deployed above the surface of sea [24]. In such applications, signal coverage areas are close to unit disk and VSFG can be directly applied.

In other situations in which VSFG fails frequently, one solution is possible to overcome the problem. The basic idea is to construct an *overlay unit disk graph* (OUDG) above the realistic sensor network satisfying the properties of UDG. Construction of OUDG is based on the two following rules.

- *Rule 1*: if two nodes are within unit distance D but not direct neighbors, we build a virtual path between the two nodes and treat them as neighbors of each other in OUDG.
- *Rule 2*: if two nodes are not within distance D but they are direct neighbors, this link is removed from the OUDG.

VSFG can be directly applied over OUDGs. However, OUDG construction requires that each node knows the locations of all nodes in the network, which is not practical. A realistic way is that let each node u know the locations of nodes within m hops to u. With m-hop neighbor information, an intermediate graph can be constructed based on Rules 1 and 2. The constructed graph may not be an OUDG. The Gabriel graph built above an intermediate graph may contain intersected edges. However, according to experiments results not shown here, intermediate graphs constructed with $m = 2$ or $m = 3$ have very high probabilities to support VSFG without failure in most of situations.

IX. CONCLUDING REMARK

In this paper, we propose a geocasting algorithm VSFG with guaranteed message delivery and a low transmission cost. In VSFG, a virtual surrounding face (VSF) of a geocasting region is constructed by ignoring edges intersecting the region. By traversing all the boundary nodes of VSF and performing restricted flooding within the geocasting region, all nodes are guaranteed to receive the message. In addition, we propose a SKIP algorithm and a DS-based restricted flooding algorithm to further reduce the transmission cost. The VSFG_2 algorithm, combining these two algorithms, significantly reduces the cost.

Among the existing algorithms, RFIFT has the lowest transmission cost. In RFIFT, the cost for face traversal is limited to $3n$, where n is the number of nodes on the boundaries of faces intersecting R. In the algorithms of VSFG family, this bound is reduced to $2n$. In addition, by applying SKIP and local DS-based restricted flooding, VSFG significantly improves the performance on average cases for dense networks. From the simulation results, VSFG_2 reduces up to 65% of the total cost required in RFIFT.

REFERENCES

[1] G. G. Finn, "Routing and addressing problems in large metropolitan internetworks," ISI Research Report ISU/RR-87-180, 1987.

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

[3] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks," in *Proc. ACM Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, 1999, pp. 48–55.

[4] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks," *Wireless Networks*, vol. 7, no. 6, pp. 609–616, 2001.

[5] H. Gupta, Z. Zhou, S. R. Das, and Q. Gu, "Connected sensor cover: Self-organization of sensor networks for efficient query execution," *IEEE/ACM Trans. Networking*, vol. 14, no. 1, pp. 55–67, Feb. 2006.

[6] F. Kuhn, R. Wattenhofer, and A. Zollinger, "Asymptotically optimal geometric mobile ad hoc routing," in *Proc. ACM Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, 2002, pp. 24–33.

[7] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger, "Geometric ad hoc routing: Of theory and practice," in *Proc. PODC*, 2003, pp. 63–72.

[8] A. Qayyum, L. Viennot, and A. Laouiti, "Multipoint relaying for flooding broadcast messages in mobile wireless networks," in *Proc. Hawaii Int. Conf. System Sciences*, 2002, pp. 3898–3907.

[9] Y. B. Ko and N. H. Vaidya, "Geocasting in mobile ad hoc networks: Location-based multicast algorithms," in *Proc. Workshop on Mobile Computer System and Applications*, 1999, pp. 101–110.

[10] S. Basagni, I. Chlamtac, and V. R. Syrotiuk, "Geographic messaging in wireless ad hoc networks," in *Proc. VTC*, 1999, pp. 1957–1961.

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

[12] I. Stojmenovic, A. P. Ruhil, and D. K. Lobiyal, "Voronoi diagram and convex hull based geocasting and routing in wireless networks," in *Proc. IEEE Symp. Computers and Communication (ISCC'03)*, 2003, vol. 1, pp. 51–56.

[13] Y. B. Ko and N. H. Vaidya, "Flooding-based geocasting protocols for mobile ad hoc networks," *Mobile Networks and Applications*, vol. 7, no. 6, pp. 471–480, 2002.

[14] C. Schwingenschlogl and T. Kosch, "Geocast enhancements of AODV for vehicular networks," *ACM Mobile Computing and Communication*, vol. 6, no. 3, pp. 96–97, 2002.

[15] B. An and S. Papavassiliou, "Geomulticasting: Architectures and protocols for mobile ad hoc networks," *J. Parallel Distrib. Comput.*, vol. 63, pp. 182–195, 2003.

[16] T. Camp and Y. Liu, "An adaptive mesh-based protocol for geocast routing," *J. Parallel Distrib. Comput.*, vol. 62, no. 2, pp. 196–213, 2003.

[17] K. Seada and A. Helmy, "Efficient geocasting with perfect delivery in wireless networks," in *Proc. IEEE Wireless Communications and Networking Conf. (WCNC 2004)*, Mar. 2004, vol. 4, pp. 2551–2556.

[18] I. Stojmenovic, "Geocasting with guaranteed delivery in sensor networks," *IEEE Wireless Commun.*, vol. 11, no. 6, pp. 29–37, Dec. 2004.

[19] S. Ni, Y. Tseng, Y. Chen, and J. Sheu, "The broadcast storm problem in mobile ad hoc network," in *Proc. ACM MOBICOM*, 1999, pp. 152–162.

[20] J. Wu and H. Li, "A dominating-set-based routing scheme in ad hoc wireless networks," *Telecommun. Syst. J.*, vol. 3, Special Issue on Wireless Networks, pp. 63–84, 2001.

[21] I. Stojmenovic, M. Seddigh, and J. Zunic, "Dominating sets and neighbor elimination based broadcasting algorithms in wireless networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 13, no. 1, pp. 14–25, Jan. 2002.

[22] J. Wu and F. Dai, "Mobility management and its applications in efficient broadcasting in mobile ad hoc networks," in *Proc. IEEE INFOCOM*, 2004, pp. 339–350.

[23] J. Lian, S. Naik, Y. Liu, and L. Chen, "Virtual surrounding face based geocasting with guaranteed message delivery for ad hoc and sensor networks," in *Proc. IEEE Int. Conf. Network Protocols (ICNP'06)*, Nov. 2006, pp. 198–207.

[24] Y. Liu and M. Li, "Iso-Map: Energy-efficient contour mapping in wireless sensor networks," in *Proc. IEEE ICDCS*, Jun. 2007, p. 36.

[25] A. Muqattash and M. Krunz, "A distributed transmission power control protocol for mobile ad hoc networks," *IEEE Trans. Mobile Comput.*, vol. 3, no. 2, pp. 113–128, Apr./Jun. 2004.

[26] W. H. Liao *et al.*, "Geogrid: A geocasting protocol for mobile ad hoc networks based on grid," *J. Internet Technol.*, vol. 1, no. 2, pp. 23–32, 2000.

[27] M. Garetto, J. Shi, and E. Knightly, "Modeling media access in embedded two-flow topologies of multi-hop wireless networks," in *Proc. ACM MOBICOM*, Aug. 2005, pp. 200–214.

[28] R. Purtoosi, H. Taheri, A. Mohammadi, and F. Foroozan, "A high performance cluster-based flooding algorithm for wireless ad hoc networks," in *Proc. IEEE Int. Conf. Computer and Information Technology*, 2005, pp. 417–423.

[29] F. Foroozan and K. Tepe, "A high performance cluster-based broadcasting algorithm for wireless ad hoc networks based on a novel gateway selection approach," in *Proc. ACM Int. Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks*, 2005, pp. 65–70.

[30] H. Frey and I. Stojmenovic, "On delivery guarantees of face and combined greedy-face routing algorithms in ad hoc and sensor networks," in *Proc. ACM MOBICOM*, 2006, pp. 390–401.

[31] A. Qayyum, L. Viennot, and A. Laouiti, "Multipoint relaying: An efficient technique for flooding in mobile wireless networks," in *Proc. 35th Annu. Hawaii Int. Conf. System Sciences (HICSS-35)*, 2002, pp. 298–307.

[32] Y.-J. Kim, R. Govindan, B. Karp, and S. Shenker, "Geographic routing made practical," presented at the 2nd USENIX/ACM Symp. Networked System Design and Implementation (NSDI'05), Boston, MA, May 2005.

Jie Lian (M'05) received the B.S. degree in computer science from Inner Mongolia University, China, in 1992, the M.A. degree from Asian Institute of Technology, Thailand, in 1998, and the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2006.

He is now a research assistant in the Department of Electrical and Computer Engineering at the University of Waterloo. His research interests are in the areas of ad hoc sensor networks, P2P system, power-saving techniques for portable wireless devices, and RFID.

Yunhao Liu (SM'06) received the B.S. degree in automation from Tsinghua University, China, in 1995, and the M.A. degree from the Beijing Foreign Studies University, China, in 1997, and the M.S. and Ph.D. degrees in computer science and engineering from Michigan State University in 2003 and 2004, respectively.

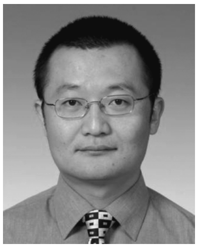
He is now with the Department of Computer Science and Engineering at the Hong Kong University of Science and Technology. He is also an Adjunct Professor of Xi'an Jiaotong University, Jilin University, and Ocean University of China. His research interests include wireless sensor network, peer-to-peer computing, and pervasive computing.

Dr. Liu and his student Li Mo received the Grand Award of Hong Kong ICT Best Innovation and Research Award 2007. He is a member of the ACM.

Kshirasagar Naik (M'94) received the B.S. degree from Sambalpur Unversity, India, the M.Tech. degree from the Indian Institute of Technology, Kharagpur, India, the M.Math. degree in computer science from the University of Waterloo, Waterloo, ON, Canada, and the Ph.D. degree in electrical and computer engineering from Concordia University, Montreal, QC, Canada.

He is an Associate Professor in the Department of Electrical and Computer Engineering at the University of Waterloo. His research areas are wireless communication, resource allocation in 3G systems, ad hoc sensor networks, MAC protocols for wireless LAN, Bluetooth, mobile computing, and P2P systems.

Dr. Naik is a member of the ACM.



Lei Chen (M'02) received the B.S. degree in computer science and engineering from Tianjin University, China, in 1994, the M.A. degree from Asian Institute of Technology, Thailand, in 1997, and the Ph.D. degree in computer science from the University of Waterloo, Waterloo, ON, Canada, in 2005.

He is now an Assistant Professor in the Department of Computer Science and Engineering at the Hong Kong University of Science and Technology. His research interests include multimedia databases, sensor databases, peer-to-peer databases, and probabilistic databases.

Dr. Chen is a member of the ACM.