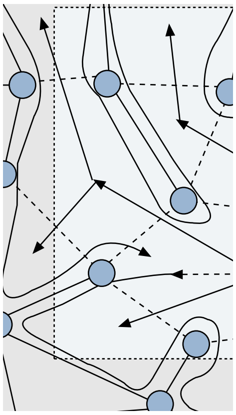
GEOCASTING WITH GUARANTEED DELIVERY IN SENSOR NETWORKS

IVAN STOJMENOVIC, SITE, UNIVERSITY OF OTTAWA



Recent technological advances have enabled the development of low-cost, low-power, and multifunctional sensor devices. Sensor networks consist of a large number of sensor nodes that collaborate together using wireless communication and asymmetric many-to-one data.

ABSTRACT

In a geocasting problem, a message is sent from one node to all the nodes located in a designated region. For example, a monitoring center needs to contact all active sensors within a monitored area to either gather data from them periodically or provide its location to sensors covering a certain area for event reporting. Intelligent flooding methods exist for this task when all active sensors belong to the monitored area. However, when a particular area containing only a small subset of active sensors needs to be monitored, the problem reduces to geocasting. Most existing geocasting solutions are shown not to guarantee delivery. We describe three approaches to guarantee delivery. Two of them are face traversal schemes, based on depth-first search of the face tree and traversal of all faces that intersect the border of the geocasting region, respectively. In the entrance zone multicasting-based approach, the monitoring center divides the entrance ring of a geocast region into zones of diameter equal to the transmission radius. The problem is decomposed into multicasting toward the center of each zone, and flooding from these nodes. Improvements to all methods can be made by applying neighbor or area dominating sets and coverage, and converting nodes that are not selected to sleep mode. All solutions that guarantee delivery are reported here for the first time (except a message inefficient version of the face tree traversal scheme).

INTRODUCTION

Recent technological advances have enabled the development of low-cost, low-power, multifunctional sensor devices. These nodes are autonomous devices with integrated sensing, processing, and communication capabilities. Sensor networks consist of a large number of sensor nodes that collaborate using wireless communication and asymmetric many-to-one data. Indeed, sensor nodes usually send their data to a specific node called the *sink node* or *monitoring station*, which collects the requested information. All nodes cannot communicate directly with the monitoring station, since such communication may be over long distances that will drain power quickly. Hence, sensors operate in a self-organized and decentralized manner, and message communication takes place via multihop spreading. To enable this, the network must maintain the best connectivity as long as possible. A sensor's battery is not replaceable, and sensors may operate in hostile or remote environments. Therefore, energy consumption is considered the most important resource, and the network must be self-configured and self-organized. The best energy conservation method is to put as many sensors as possible to sleep. The network must be connected to remain functional so that the monitoring station may receive messages sent by any of the active sensors. An intelligent strategy for selecting and updating a set of active sensors that are connected is needed in order to extend the network lifetime. This problem is known as the *connected area coverage problem*, which aims to dynamically activate and deactivate sensors while maintaining the full coverage of the monitoring area. Efficient solutions to the connected area coverage problem were discussed recently in [1–3]. When this coverage step is performed first, the large sensor network becomes reasonably sparse but remains connected.

If all active sensors are dedicated to monitoring the same event, the monitoring center may spread the task and establish a reverse broadcast tree using any intelligent flooding protocol [4]. If the network is reasonably sparse, even blind flooding (where each node receiving a message will retransmit it exactly once) is a viable option. However, when the region to be monitored for particular event contains only a small portion of active sensors, flooding the whole network may be an inefficient way to spread the task. This article reviews existing solutions to the geocasting problem. In a multifunctional multi-event sensor environment, a monitoring center may separately handle several geocasting regions and corresponding events. One particular application of geocasting is in tracking mobile objects. A monitoring center may collect reports from sensors in the vicinity of the object, and send periodic signals to sensors adjusting the geocasting region, following the object's movement.

We assume that each sensor is aware of its geographic position with respect to its neighbors and monitoring center. The problem of finding a

1536-1284/04/$20.00 © 2004 IEEE

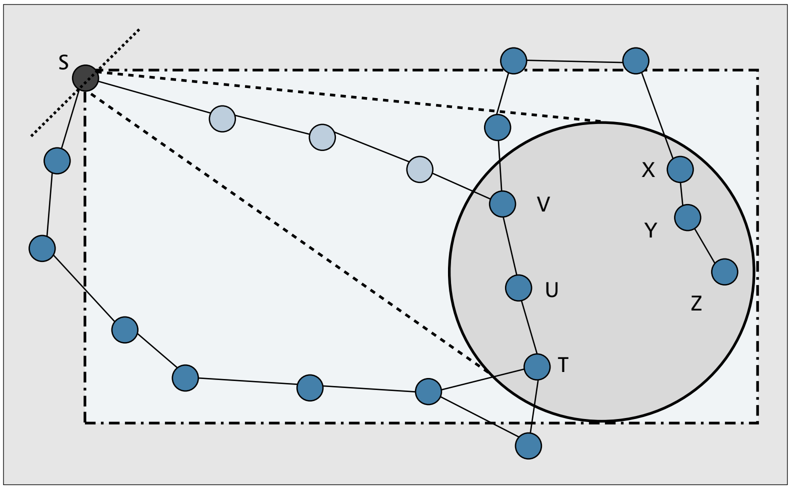


Figure 1. *An example showing that most existing geocasting schemes do not guarantee delivery.*

reasonably accurate sensor location (when sensors are not directly equipped with GPS receivers, which is becoming technologically feasible) was intensively studied recently [5]. We consider only a localized approach in this article. In a localized routing or geocasting algorithm, each node makes a decision on to which neighbor(s) to forward the message based solely on its own location, the locations of its neighboring nodes, and the destination location. In geocasting, the destination is a node approximately in the center of the region, and the information includes the geocast region description. We also assume that a sensor network is static, and the monitoring center is aware of the geocast region to be covered. For simplicity, we assume here that the geocast region is a circle. However, other shapes, such as convex polygons, may similarly be considered.

A number of localized geocasting protocols proposed in the literature [6–13] do not guarantee delivery to all nodes inside a geocasting region. Note that sensors may actually cover the geocast region, but may not be connected inside it because of possible obstacles in the region or differences between communication and sensing radii.

Among localized geocasting algorithms recently discussed in the literature, we show that the only one able to guarantee delivery is a "forgotten" face tree traversal scheme [14, 15]. It is fully memoryless (all information needed for making traffic decisions is carried with the message, and nodes do not need to memorize even very recent traffic). However, it has considerable message overhead. This article assumes that the medium access control (MAC) layer is ideal, that is, each message sent from a node to its neighbor is received properly by that neighbor. The guaranteed delivery property is conditional upon availability of such an ideal MAC layer. A geocasting protocol therefore has a guaranteed delivery property if each node located inside a geocasting region and connected to the source node will receive the packet if an ideal MAC layer is applied. We describe in this article three nontrivial solutions for the geocasting task with guaranteed delivery. One is obtained from [14] by adding a preprocessing step, making the algorithm message efficient afterward (however, it requires a static network after preprocessing and "marking" by one bit some edges with special status). Another is based on multicasting to entrance zones, followed by intelligent flooding. We show that a recently proposed geocasting scheme [16] does not guarantee delivery despite its claim, and then modify it to provide this property. All three algorithms are proven to guarantee delivery. Their expected performance is discussed.

This article is organized as follows. We present localized location-based routing and geocasting algorithms. A geocasting algorithm with guaranteed delivery, based on traversal of faces intersecting a geocast region boundary, is described. Another algorithm [14] based on a traversing face tree is described (a variant of it, with preprocessing and reducing message complexity, is proposed here). We describe a geocasting protocol that guarantees delivery and is efficient compared to the possible alternative. Finally, a conclusion and references complete this article.

Due to space constraints, many details are omitted. The full version of this article can be found in [17]. In particular, existing geocasting techniques that do not guarantee delivery are all described in [17].

POSITION-BASED LOCALIZED ROUTING AND GEOCASTING ALGORITHMS

Finn [18] proposed a *greedy* routing algorithm for ad hoc networks. When node *S* wants to send a message to destination *D*, it uses the location information for *D* and all its one-hop neighbors to determine the neighbor *A* that is closest to *D* among all neighbors of *S*. Figure 2 is an example where *A* is closest to *D* among all neighbors of *S*. The message is forwarded to *A*, and the same procedure is repeated until *D*, if possible, is eventually reached. The greedy route on Fig. 2 would be *SAFVD* (note that *D* is a geocast region center, not a real node).

Almost all existing geocasting algorithms [6–8, 10–13] are based on forwarding messages within a restricted area between a source (or the node currently holding the packet) and a geocasting region, such as between tangents from the current node or source to the geocasting region, a rectangle containing the source and geocasting region, or all nodes closer to the geocasting region than the current node. In Fig. 1 these regions are drawn by dashed, dot-dashed, and dotted lines, respectively. These methods inherently do not guarantee delivery to all nodes connected to the source. If white nodes in Fig. 1 are not in the network, the source is disconnected from the geocasting region, but could be connected via nodes that do not belong to the indicated regions. An additional problem is that these methods do not consider that nodes inside a geocasting region may be disconnected (but could be connected via nodes outside it). Thus, connecting a source to some nodes (e.g., *T*, *U*, *V* in Fig. 1) does not mean that all nodes inside the geocasting region will be reached. For instance, nodes *X*, *Y*, and *Z* in Fig. 1 are disconnected from *T*, *U*, *V* inside the geocasting region. However, they are connected via nodes outside it.

A routing algorithm that guarantees delivery by finding a simple path between source and destination (without any flooding effect) is described in [14]. The Greedy-Face-Greedy (GFG) algorithm [14] applies a greedy method until current node *A* has no neighbor closer to the destination than itself (this is called a *concave* node [14]) or the message is delivered. Concave node *A* switches to recovery mode by applying face routing [19], improved in [14]. Face routing uses only edges of a planar graph. A Gabriel graph (GG) was used in [14], constructed as follows. An edge *AE* belongs to the GG if the disk with diameter *AE* contains no other nodes from the set. This can be tested by verifying whether angles to this edge from all neighbors are acute, and the test does not require any message exchange between neighbors. In Fig. 2 nodes belonging to the GG are marked with bold lines. Face routing guarantees delivery in connected planar graphs, but is followed only until a node closer to the destination than the last concave node is encountered. Such a node switches back to the greedy mode. This mode alteration may repeat a few times, but the message is guaranteed delivery, and the GFG algorithm was shown to be competitive with respect to the shortest path, especially with some improvements given in [20]. These include restricting face routing to nodes in a connected dominating set and applying a shortcut procedure.

Let us illustrate GFG routing [14] through the example in Fig. 2, for a route from *S* to *Y*. Greedy routing *SAFV* is applied until concave node *V* is reached. Node *V* then switches to recovery mode and applies face routing. Face routing follows faces along an imaginary line from source to destination, changing faces at intersections of imaginary line with the faces of the GG. The face route from *V* to *Y* follows an open face as marked with a scribbled line (route *VGHZ*). The return from recovery mode to greedy mode is possible at node *Z*, which is closer to the destination than the previous concave node *V*. *Z* then delivers to *Y*, and the imaginary line *VY* was never crossed in this example.

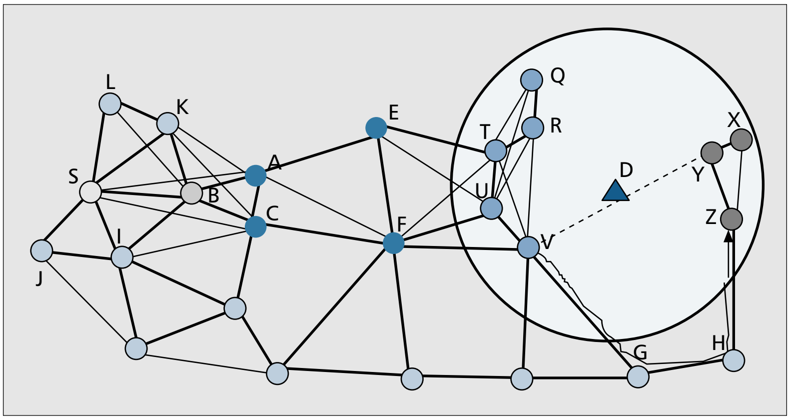


Figure 2. *Gabriel graph, face routing, and GFG.*

A simple geocasting algorithm was proposed in the technical report version of [13]. Source node *S* applies the GFG algorithm [14] to route toward a center *D* of a region until a node inside the region is encountered. In Fig. 2 the route is equal to greedy route *SAFV*. The first node that is inside the region then applies a flooding scheme, restricted to nodes inside the region. This surprisingly simple algorithm has a smaller flooding rate and an increased delivery rate compared to all known methods. However, it also fails to guarantee delivery (Figs. 1–3). Nevertheless, it is used as basic ingredient in the scheme that does guarantee delivery, described later.

GEOCASTING BASED ON TRAVERSING FACES THAT INTERSECT A BOUNDARY

Bose, Morin, Stojmenovic, and Urrutia [14] observed that a geocasting algorithm will guarantee delivery if all faces of a planar graph that are inside or intersect a geocasting region are traversed. The algorithm is based on depth first search of a face tree, and is described in the next section. We first describe a simpler algorithm, where only faces that intersect the region boundary are traversed.

Seada and Helmy [16] observed that it is sufficient to traverse only faces that intersect the boundary of a given geocasting region, and proposed the following algorithm. The source node first uses the GFG algorithm [14] to forward the packet toward the region. Each node that is inside the region will retransmit the packet when receiving it for the first time (regional flooding). "A node is a region border node if it has neighbors outside of the region. By sending perimeter packets to neighbors outside the region (notice that perimeter packets are sent only to neighbors in the planar graph, not to all physical

–Source node *S* sends the message toward the geocasting region, using the GFG algorithm [14].
–Each node inside the region retransmits the message when receiving it for the first time, and ignores it when receiving it again.
–Each internal border node (node inside region having neighbor(s) on a planar graph outside the region) will instruct (together with retransmission) its all perimeter neighbors outside the region to perform right-hand-based face traversals.
–Each external border node (node outside region having neighbor(s) on planar graph inside the region), will initiate right-hand based face traversal(s) with respect to all edges leading to internal perimeter neighbors, after receiving the first copy of the message, and will ignore further received copies unless a packet is received from external neighbor following a different "external" face (in which case it forward it along that face as requested). Each traversal is performed until another node that is inside the region is found.

Algorithm 1. *Geocast traversal, intersecting faces.*

By sending perimeter packets to neighbors outside the region, the faces intersecting the region are traversed. The node outside the region receiving the perimeter mode packet forwards the packet using the right-hand rule to its neighbor and so on.

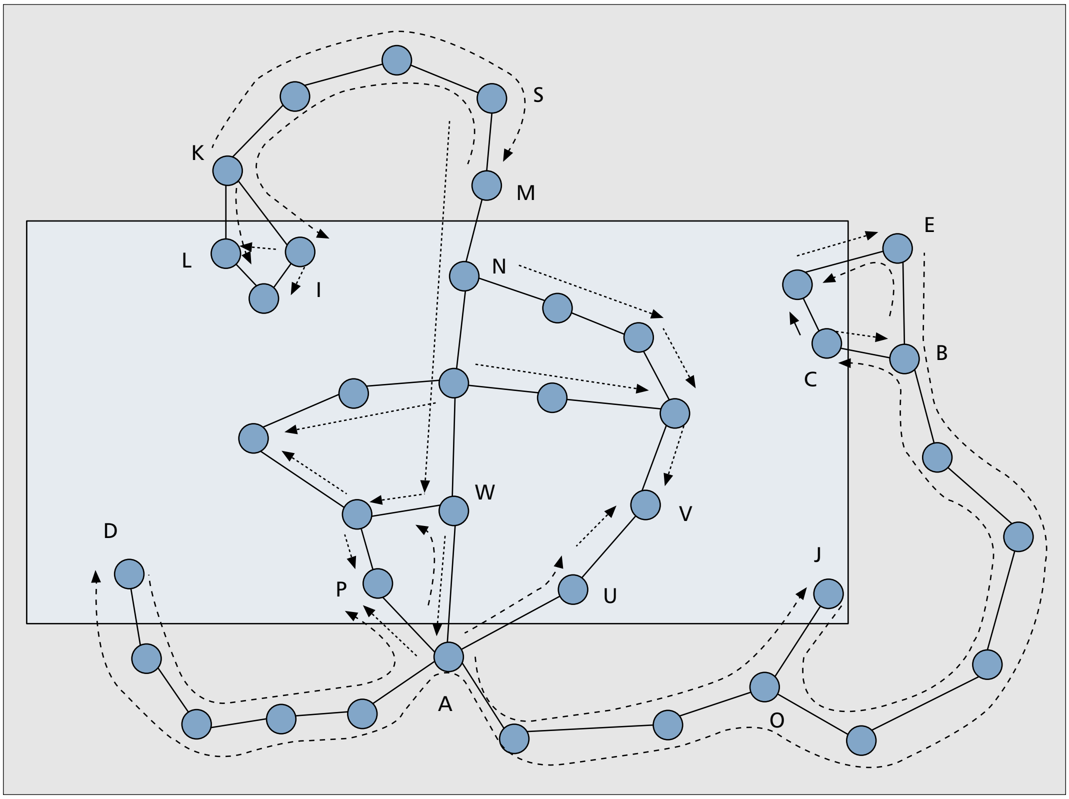


Figure 3. *Face-traversal-based geocasting with guaranteed delivery.*

neighbors), the faces intersecting the region are traversed. The node outside the region receiving the perimeter mode packet forwards the packet using the right-hand rule to its neighbor and so on. The packet goes around the face until it enters the region again. The first node inside the region to receive the perimeter packet floods it inside the region or ignores it if that packet was already received and flooded before" [16]. We showed in [17] that this algorithm [16] does not guarantee delivery, despite the claim. A geocasting algorithm that guarantees delivery is shown in Algorithm 1.

The main difference between our algorithm and the one in [16] is that external border nodes perform right-hand-based face traversals with respect to all corresponding neighboring internal border nodes no matter how the message arrives to them (in [16] it is activated only from internal border neighbor, for one face at a time, as [16, Fig. 8] confirms).

We shall now illustrate in Fig. 3 our algorithm on the same example, starting from source node *S*. Node *M* starts right-hand-based face traversal (finding the first neighbor in the clockwise direction with respect to edge *MN*). Note that right-hand-based face traversal results in counterclockwise face traversal of closed faces and clockwise traversal of the single open face. The face traversal from *M* reached edge *KI* in Fig. 3. Node *I* floods the packet regionally, while node *K* initiates two face traversals with respect to edges *KI* and *KL*. The face traversal with respect to *KI* ends at *L*, while that with respect to *KL* follows the outer boundary until "seeing" edge *MN* again (which then ignores it). Regional flooding reaches node *W*. *W* "alerts" *A* to perform face traversals with respect to *AP*, *AW*, and *AU*. Neighbors *P* (by listening to all traffic from *A*) and *U* (as part of face traversal) receive packets from *A*, and can retransmit as part of regional flooding. One face traversal reaches node *J*. Face traversal from *O* (neighbor of *J*) reaches nodes *B* and *C*. *C* floods to its neighborhood while *B* starts face traversal of the quadrilateral in Fig. 3. Face traversal from *E* bypasses *B*, *O*, and *A*, and reaches node *D* in the other corner.

We shall now prove that the algorithm indeed guarantees delivery to all nodes inside a geocasting region connected to the source node. The proof, in fact, is quite elegant, and is expressed in the following theorem.

Theorem 1 — The described geocasting algorithm, based on traversing faces that intersect a geocast region boundary, guarantees delivery to all nodes inside the geocast region connected to the source.

Proof — We can argue that every face intersecting the geocasting region and connected to the source was fully traversed by the combination of regional flooding and outer face traversals. Consider, for example, the outer boundary in Fig. 3 (the proof is the same for any face). Its traversal started at *MN* and reached *I* (the lower dashed line in Fig. 3). With internal flooding, it reached *LK* from *I*. Then from *K* it reached *SMN* (the upper dashed line). By regional flooding it can reach *UA*. Then face traversals are used to follow the *AJ* piece followed by the *JOBC* piece. Internal flooding then reaches *E*. Face traversal from *E* then bypasses *O* and *A* and reaches *D*, which continues face traversal until AP. Finally,

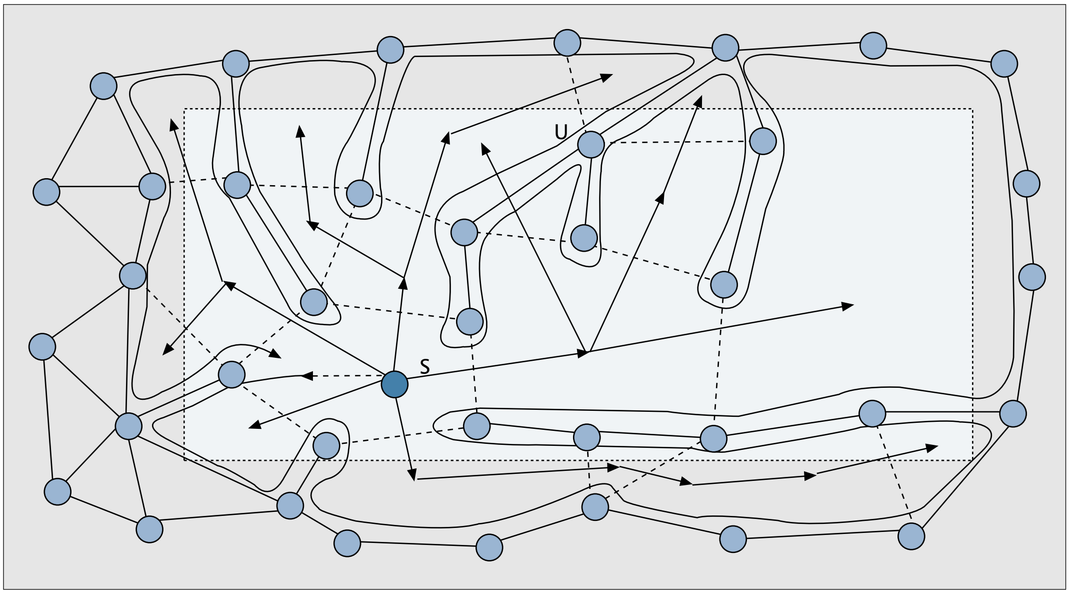


Figure 4. *Face-tree-traversal-based geocasting with guaranteed delivery.*

The face tree is dynamically constructed, during geocasting operation. The geocasting algorithm follows depth first search based traversal of the face tree. For each node in the face tree, it actually traverses the corresponding face. When an entry edge is encountered, the traversal enters a new face.

flooding from P can reach back to MN, and the whole face is traversed. We could make the proof more formal, but believe this informal exemplar explanation suffices. The main argument is that right-hand traversal of any face is composed of pieces containing regional flooding for consecutive face nodes inside a region, and pieces outside the region that are triggered when a packet arrives there. Regional flooding, piecewise face traversal, and connectivity ensure that all possible nodes are reached.

In addition to guaranteeing delivery, the proposed scheme is also close to a message optimal scheme, since each node inside the region retransmits the packet only once. We show in [17] that the total number of messages is limited to $3n' + k < 3N$, where N is the total number of nodes in the network, k is the number of nodes inside the geocasting region, and n' is the number of nodes on faces intersecting the geocasting region and located outside the region. This worst case limit is encouraging and appears smaller than in two other methods that guarantee delivery, described here.

GEOCASTING BASED ON DEPTH-FIRST SEARCH TRAVERSAL OF THE FACE TREE

Bose *et al.* [14, journal version] proposed a geocasting algorithm that guarantees delivery to all nodes connected to the source, in which the packet follows a path from source node (thus single copy of the packet is in the network at any time). To improve latency, parallel paths (and multiple copies of the packet) can be explored at any branches of the face tree being used. The algorithm (its complete description is available in [15]) does not require any memory to be left at nodes, and need only carry some small amount of information with the packet (if entry edges are predetermined for a given source, the message need only contain sender and source information).

The algorithm [14] first applies GFG to route toward a node inside a geocasting region. That node then selects a nearby point S inside the face to act as an artificial source. The face tree from S is constructed in the following way. Given a node S and a face f of a planar graph, the entry edge $\mathit{entry}(f, S)$ is the edge from f that is closest to S. To break the ties, several keys for comparison of edges are used. The primary key is the distance of the edge to S, where the distance is decided by a point C from the edge that is closest to S. If the distances are the same, the secondary key used is the counterclockwise direction of vector SC. In case of further ties (which may occur only when two edges share a common closer endpoint C), consider the size of the angle $\angle SCD$, where D is the other endpoint of the edge. If that still does not resolve it, consider the vector CD, which then must be different. Morin [15] proved that all entry edges are on the boundaries of two faces. In the face tree, the parent of a face f is the face $p(f)$ that contains its entry edge $e(f)$ on its boundary. Obviously, then, $p(f)$ itself has another entry face closer to S, which confirms that a tree of faces is indeed constructed. The face tree is dynamically constructed during geocasting operation. The geocasting algorithm follows depth first search-based traversal of the face tree. For each node in the face tree, it actually traverses the corresponding face. When an entry edge is encountered, the traversal enters a new face. When the traversal (which may recursively go to deeper levels) is completed, it returns to the face. Traversal of each face begins from one end of its entry edge and finishes at the other end of it. Figure 4 illustrates the algorithm. A face tree from S is drawn with directed edges intersecting entry edges (dashed lines). The path taken by the geocasting algorithm is shown by a scribble line, starting from point S. The algorithm visits all edges along the path.

The algorithm [14, 15] is shown in Algorithm 2. In this scheme, opposite(e,f) is the other face

> If the geocasting source is a fixed base station in sensor networks, then entry edges may be determined by flooding the network from base station, and traversing all faces of planar graph to determine and conveniently label entry edges.

```
f ← face containing S
e_start ← e ← an edge of f
repeat
    if e intersects geocast region then
        if e = entry(f, S) {* e is the closest edge to S on f *}
            then f ← ( opposite(e,f) {* return to parent of f, the other face containing e *}
            else if e = entry(opposite(e,f), S) {* e is the closest edge to S on the other face *}
                then f ← opposite(e,f) {* visit child of f, the other face containing e *}
        e ← ( next(e, f)
until e=e_start
```

Algorithm 2. *Geocast face tree traversal.*

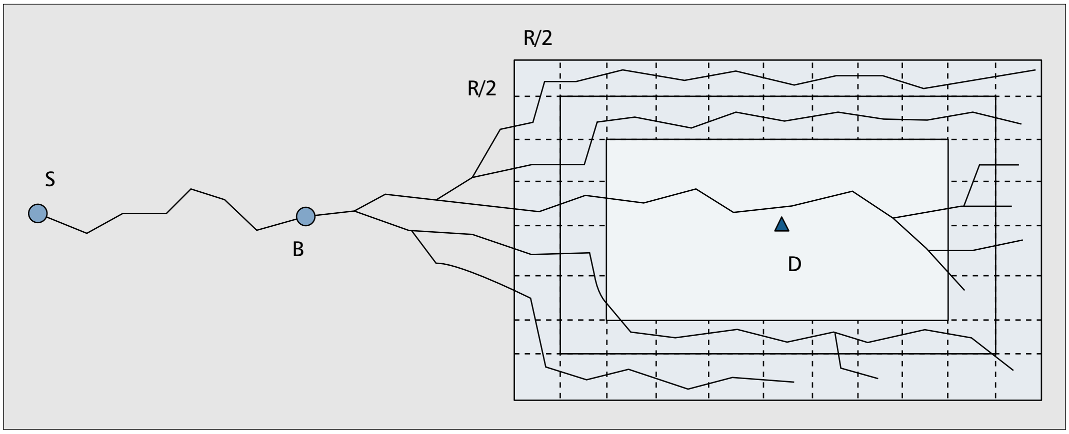


Figure 5. *Covering entrance zones from a remote monitoring center.*

containing the same edge e as the face f currently being traversed. The edge next(e,f) is the next edge being traversed by the right-hand rule from current edge e on face f.

In some applications, entry edges may be determined as the preprocessing step. For example, if the geocasting source is a fixed base station in sensor networks, entry edges may be determined by flooding the network from the base station, and traversing all faces of the planar graph to determine and conveniently label entry edges. Afterward, geocasting regions may be dynamically determined (e.g., to follow a moving object), and geocasting may then proceed as described. It was shown in [17] that in this variant (after preprocessing), the number of messages for geocasting is $< 2(N - 1) < 2(n' + k)$. Communication steps from the source to the geocasting region need to be added, for this and other geocasting algorithms. Therefore, the scheme has reasonable communication overhead under the given assumption. Compared to the previously described scheme, it has less communication overhead when $k < n'$. It does require preprocessing (or a significant number of additional messages at runtime) and offers the additional benefit of providing a single path in the network, which provides time division, which is suitable for applications when sensors networks alternate in reporting to the monitoring center directly (see details in [17]). Lindsey, Raghavendra, and Sivalingam [21] proposed such a framework for energy-efficient data gathering algorithms in sensor networks, but did not describe any localized algorithm for deciding the order of transmission that can be achieved by the described scheme.

For a dynamically selected source of a geocasting message, such preprocessing is not possible. A scheme for testing whether a given edge is an entry edge is described in [14, 15, 17]. The number of messages sent in the scheme overall is $O(N + k \log k)$ [14, 15], where the latter term is due to entry edge tests. The proof that this geocasting algorithm guarantees delivery to all nodes connected to the source is given in [14, 15].

MULTICASTING AND GEOCASTING WITH GUARANTEED DELIVERY

We now describe an algorithm based on the following observation. If a node V inside the geocast region is connected to the source S, the first node U on a route from S to V is no more than transmission radius distance R from the border of the geocasting region. The set of points that are at distance $\leq R$ from the border of the geocasting region is called the *entrance ring*. The entrance ring is subdivided into *entrance zones*. The diameter of each entrance zone must be $\leq R$, and each such division can be used. The geocasting algorithm based on multicasting and entrance zones is shown in Algorithm 3.

In the next three subsections we elaborate on these steps, and prove that guaranteed delivery holds. We also illustrate the algorithm and discuss its message complexity.

DETERMINING ENTRANCE ZONES

The entrance zones should be determined with the following two criteria in mind:

- It is not possible to send a message directly

–Determine entrance zones and their centers. One way of doing so is to draw two perimeters, at distances $R/2$ and R from the perimeter of the geocasting region and inside the region and dividing such entrance ring into zones, each with diameter $\leq R$, in arbitrary fashion.
–Multicast from source S toward centers of each entrance zone until a node inside a zone is reached (these nodes will be called multicast recipients), or a loop in recovery mode of the routing scheme is identified.
–Flood from each multicast recipient. This can be done by blind flooding restricted to nodes inside the region, or by some intelligent flooding scheme [4] that reduces the number of retransmissions. Each node memorizes received packets and ignores repeated copies of the same packet.

■ **Algorithm 3.** *Geocast, entrance zone multicast.*

In the optimal paths method, each node receiving a multicasting message for a group of nodes will forward it to each neighbor that is closest to one of group members. More precisely, each group member is assigned to the neighbor that is closest to it.

from a node outside the geocasting region to a node inside it; this means that the width of all the zones together, measured as the minimum distance between a node outside the geocasting region to a node inside it that does not belong to any entrance ring, must be at least R, the transmission radius of the network.

- The diameter of each entrance zone must be at most R; this means that if a node inside a zone receives the multicast packet, all other nodes in the same zone will receive it after retransmission from that multicast recipient.

The exact construction of entrance zones to satisfy these criteria depends on the shape of the geocasting region. If the geocasting region is a rectangle, for instance, the entrance zones may be composed of two layers of squares of edge length $R/2$, as illustrated in Fig. 5. One dimension (not affecting overall width R) can be increased until the diameter becomes R. Note that some of these regions may be empty, and routing to them will end up in a loop (for clarity, these loops are not drawn in Fig. 5).

POSITION-BASED MULTICASTING

In a multicasting task, the sender node wishes to send the same packet to several other nodes in the network. Routing and broadcasting are all special cases of multicasting. In [22] the authors propose two similar multicasting schemes, with some optimizations. In the *optimal paths* method, each node receiving/multicasting a message for a group of nodes will forward it to each neighbor that is closest to one of the group members. More precisely, each group member is assigned to the neighbor closest to it (provided that neighbor is closer to it than the current node). In the *aggregate paths* method, for each neighbor A, the number of destinations for which A is the closest node is determined. Then a covering algorithm is applied. Basically, a neighbor is chosen that covers the maximum number of destinations, these destinations (and other nodes for which selected node makes some progress) are eliminated from the list, then another neighbor is chosen that covers the maximum number of remaining destinations, and so on. The forwarding list of a multicast group is changed similarly as in the previous algorithm [22]. In both schemes, if no neighbor is closer to one or more destinations, the recovery mode in the GFG algorithm [14] is applied. The virtual destination used for recovery mode is calculated as the position representing the average of the positions of the affected destination nodes. When a node receives a multicast packet in recovery mode, it checks for each destination if it is closer to that destination than the node where the packet entered recovery mode. For all destinations where this is the case, greedy multicast forwarding can be resumed as described in the corresponding scheme. For all other destinations recovery mode is continued, with updated average of positions of affected nodes (those not recovered yet).

Note that the optimal path method (without a recovery scheme) corresponds to the *VD-greedy* scheme [13]. They both use hop count as the metric. Both the optimal and aggregate path methods can be modified by considering metrics other than hop count, such as power, cost, delay, or others. Greedy routing can be replaced by power- and/or cost-aware routing, and forwarding neighbors will be judged based on the metric in question, combined with their coverage ability, for selection.

ENTRANCE ZONE MULTICASTING-BASED GEOCASTING WITH GUARANTEED DELIVERY

This algorithm consists of multicasting toward the centers of all entrance zones, and flooding from the first nodes encountered in each nonempty zone. A zone center is any node inside it (e.g., its center of mass or intersection of zone diagonals). In Fig. 5 the multicasting used in our scheme is illustrated. The source S initiates multicasting, which begins branching at B. This figure applies only greedy forwarding for simplicity. Several entrance zones in Fig. 5 are empty, and the algorithm will make one loop in the GFG algorithm to confirm that (these loops are not drawn). The multicasting scheme can be followed in one of the ways described above. One more specific example for this scheme can be found in the full version of this article [17]. Note that some optimizations can be made here. For example, a few nodes on a path can collectively conclude that some zones are empty and prevent a full loop in GFG. Optimization via merging assigned zones can be made when a few neighbors assign tasks independently; that is, a node can wait for possible new assignment before starting its own forwarding and assignment.

Upon entering any zone, the protocol converts to intelligent flooding inside a geocast region. In all exiting intelligent flooding methods (see a review in [4]), nodes may receive multiple copies of the same message, but forward it once only (or not forward it at all) after a timeout that depends on the protocol selected. Intelligent flooding for geocasting inside the region and existing flooding methods differ in only one sense. Instead of having just one source for flooding, geocast application may have several such sources, one per entrance zone. This difference requires adjusting timeouts to somewhat larger values than in regular flooding tasks, or memorizing past traffic somewhat further, since some messages may be delayed by longer forced routes while being in recovery mode before arriving at an entrance zone. Also, the distances from a given node to entrance zones may be considerably different, adding to the differences in message arrival times.

In sensor networks some nodes are sensing areas, while some other nodes are there to support routing as a basic data communication protocol for data gathering. Some or all sensors can at the same time perform sensing and forwarding traffic tasks.

The monitoring center S may be outside or inside the geocasting region. Although our description implicitly assumes that S is outside the region, the same algorithm also works correctly if S is inside it. We now prove that this geocasting algorithm guarantees delivery.

Theorem 2 — The described geocasting algorithm, based on multicasting to entrance zones and flooding from multicast recipients, guarantees delivery to all nodes inside a geocast region connected to the source.

Proof — The proof that multicasting entrance-zone-based geocasting guarantees delivery is based on two key arguments. First of all, multicasting itself guarantees delivery, based on the guaranteed delivery property of GFG (proven in [14]), which is applied toward every destination. The guaranteed delivery of multicasting is also claimed in [22]. Next, we argue that any node inside a geocasting region connected to the source must be connected to at least one of the mentioned multicasting recipients. Suppose that a node X is inside a geocasting region. Then it is inside an entrance zone, or outside all entrance zones. If it is inside an entrance node, it is at distance $< R$ from a multicasting recipient, and therefore receives a retransmitted message from that recipient. If it is outside all entrance zones and connected to the source S, the path from S to X needs to cross the entrance zones ring somewhere. Since the width of that ring is R, it cannot "jump" over it and cross directly from outside to inside the geocasting region ("escaping" the entrance ring). Therefore, the path contains at least one node in one entrance ring. That node is connected to a multicast recipient, and flooding initiated from that multicast recipient will reach X. Therefore, all nodes connected to the source will receive a geocasting packet, and the algorithm then guarantees delivery.

It appears that (in dense networks) this protocol may have smaller communication overhead with respect to listed methods that do not guarantee delivery [17]. The comparative communication overhead depends on relative distance from the monitoring center to the geocast region. It also depends on the existing coverage of the geocast region by active sensors. Obviously, several empty regions may cause long routes along the network perimeter to recognize them.

Entrance zone multicast-based geocasting is expected to be competitive with face-traversal-based schemes on average. However, in the worst case it can exhibit excessive overhead due to potential face routing along the network perimeter for each empty region. Fortunately, consecutive empty entrance zone do not necessarily require separate face routings, since the multicasting method merges them into a single destination. The worst case appears to be the scenario with every other entrance zone being empty, and thus each requiring separate face routing to be confirmed.

CONCLUSIONS

There are four geocasting algorithms that guarantee delivery. Intelligent flooding delivers to all nodes in the network (solving the broadcasting task), and is best when a geocasting region nearly covers the whole network. The three methods presented here are designed for cases where a geocasting region is relatively small. Among the three proposed schemes, it is expected that (on average) traversing faces that intersect a boundary will perform best when there are many empty entrance zones; otherwise, a multicasting-based solution should be best. Depth-first search-based face tree traversal requires preprocessing for reasonable performance, and has applications for sensor time division when reporting directly to a monitoring center. More reliable conclusions can be made only after performance evaluation.

The performance of the described geocasting protocols need to be evaluated experimentally. These geocasting algorithms assume that nodes have accurate position information about themselves and their neighbors. It is a further interesting problem to study the impact of localization errors on the performance of the proposed geocasting protocols. Note that the effect of localization errors on the performance of the face routing scheme of [14] is presented in [23].

In large and/or dense ad hoc and sensor networks, it is not necessary to use all available nodes to perform data communication tasks. In sensor networks, for example, some nodes are sensing areas, while some others are there to support routing as a basic data communication protocol for data gathering. Some or all sensors can at the same time perform sensing and traffic forwarding tasks. There are several reasons to reduce the number of nodes needed for monitoring or routing. Face routing, for instance, has better performance on a connected dominating set than on a full set [20], since there are fewer nodes, and consequently longer edges to traverse in the considered planar graph. Intelligent flooding is also based on a connected dominating set, where nodes not belonging to it do not need to retransmit the message (see [4] for a survey on dominating-set-based broadcasting). To save energy, sensors may decide between active and sleeping modes, with the goal of providing area coverage for monitoring reasons. Geocasting can be restricted to nodes in a connected dominating set or in an area coverage set. This is applicable to all methods described here.

We assume that all active sensors within a monitored region need to be "alarmed," and that localized algorithms are applied. Geocasting then needs to guarantee delivery, and this article describes all existing methods. The next step is their analysis by means of performance evaluation. The proposed schemes have variants, and allow for optimization with a variety of criteria and a variety of options for their implementation. Comparing geocasting methods depends on the relative size of the geocast region compared to the size of the area containing all sensors or, more precisely, on the ratio of the numbers of active sensors inside the geocast region. When compared to intelligent flooding, the smaller the geocast region, the more advantages our described methods provide. Their mutual comparison depends on parameters such as density, and the existence of "holes" in the network. Performance evaluation may also lead to further improvements of each presented method, or their adjustments to particular scenarios or evaluation criteria. Performance evaluation is left for future work.

The described geocasting protocols may also be used for a few other related applications within sensor networks. One or more monitoring stations may simultaneously geocast to one or more regions. In case of a monitoring region consisting of several disconnected subregions, the same protocol may still be followed. The protocol can also be used for geomulticast applications, such as reporting from one sensor to several monitoring stations.

Ad hoc and sensor networks have recently attracted exponentially increasing interest, including creation of new conferences and journals, as well as a number of books. We envision that this trend will continue in the short term, and that the network-layer problems discussed in this article will continue to be intensively studied. We hope the research efforts will lead to real applications of ad hoc networks, especially sensor networks.

We envision that network-layer problems will continue to be intensively studied. We hope that the research efforts will lead to real applications of ad hoc networks, especially sensor networks.

ACKNOWLEDGMENT

This research is partially supported by an NSERC Discovery grant.

REFERENCES

[1] J. Carle and D. Simplot-Ryl, "Energy Efficient Area Monitoring by Sensor Networks," *IEEE Comp.*, Feb. 2004, pp. 40–47.

[2] X. Wang *et al.*, "Integrated Coverage and Connectivity Configuration in Wireless Sensor Networks," *Proc. ACM SenSys*, Los Angeles, CA, Nov. 2003.

[3] H. Zhang and J. C. Hou, "Maintaining Sensing Coverage and Connectivity in Large Sensor Networks," UIUCDCS-R-2003-2351, June 2003; *Ad Hoc & Sensor Wireless Networks, An Int'l. J.*, vol. 1, 2005.

[4] I. Stojmenovic and J. Wu, "Broadcasting and Activity Scheduling in Ad Hoc Networks," *Ad Hoc Networking* (S. Basagni *et al.*, Eds.), IEEE Press, pp. 205–29.

[5] D. Niculescu, "Positioning in Ad Hoc Sensor Networks," *IEEE Network*, vol. 18, no. 4, July/Aug. 2004, pp. 24–29.

[6] B. An and S. Papavassiliou, "Geomulticasting: Architectures and Protocols for Mobile Ad Hoc Networks," *J. Parallel and Distrib. Comp.*, vol. 63, 2003, pp. 182–95.

[7] S. Basagni, I. Chlamtac, and V. R. Syrotiuk, "Geographic Messaging in Wireless Ad Hoc Networks," *Proc. VTC '99*, Houston, TX, May 1999, 3, 1957–61.

[8] T. Camp and Y. Liu, "An Adaptive Mesh-Based Protocol for Geocast Routing," *J. Parallel Distrib. Comp.*, vol. 63, 2003, pp. 196–213.

[9] Q. Huang, C. Lu, and G. C. Roman, "Mobicast: Just-in-Time Multicast for Sensor Networks under Spatiotemporal Constraints," TR WUCS-02-42, Washington Univ., St. Louis, MO, Dec. 2002.

[10] Y. B. Ko and N. Vaidya, "Flooding-based Geocasting Protocols for Mobile Ad Hoc Networks," *Proc. WMCSA*, 1999, New Orleans, LA, 2002, pp. 471–80.

[11] W. H. Liao *et al.*, "Geogrid: A Geocasting Protocol for Mobile Ad Hoc Networks Based on Grid," *J. Internet Tech.*, vol. 1, 2000, pp. 23–32.

[12] C. Schwingenschlogl and T. Kosch, "Geocast Enhancements of AODV for Vehicular Networks," *ACM Mobile Comp. Commun. Rev.*, vol. 6, no. 3, 2002, pp. 96–97.

[13] I. Stojmenovic, A. P. Ruhil, and D. K. Lobiyal, "Voronoi Diagram and Convex Hull based Geocasting and Routing in Wireless Networks," *Proc. IEEE In'l. Symp. Comp. Commun.*, Kemer-Antalya, Turkey, June 30–July 3, 2003; SITE, Univ. of Ottawa, TR-99-11, Dec. 1999, pp. 51–56.

[14] P. Bose *et al.*, "Routing with Guaranteed Delivery in Ad Hoc Wireless Networks," *ACM DIAL M*, Aug. 1999, pp. 48–55; *Wireless Networks*, vol. 7, no. 6, 2001, pp. 609–16.

[15] P. Morin, "Online Routing in Geometric Graphs," Ph.D. thesis, School of Comp. Scie.., Carleton Univ., Jan. 2001.

[16] K. Seada and A. Helmy, "Efficient Geocasting with Perfect Delivery in Wireless Networks," *Proc. WCNC*, Mar. 2004.

[17] I. Stojmenovic, "Geocasting in ad hoc and Sensor Networks," SITE, Univ. of Ottawa, TR-04-02, Mar. 2004.

[18] G. G. Finn, "Routing and Addressing Problems in Large Metropolitan-Scale Internetworks," ISI res. rep. ISU/RR-87-180, 1987.

[19] E. Kranakis, H. Singh, and J. Urrutia, "Compass Routing on Geometric Networks," *Proc. 11th Canadian Conf. Comp. Geom.*, Vancouver, Canada, Aug. 1999.

[20] S. Datta, I. Stojmenovic, and J. Wu, "'Internal Nodes and Shortcut Based Routing with Guaranteed Delivery in Wireless Networks," *Cluster Comp.*, vol. 5, no. 2, 2002, pp. 169–78.

[21] S. Lindsey, C. Raghavendra, and K. Sivalingam, "Data Gathering Algorithms in Sensor Networks Using Energy Metrics," *IEEE Trans. Parallel and Distrib. Sys.*, vol. 13, no. 9, Sept. 2002, pp. 924–35.

[22] M. Mauve *et al.*, "Position-Based Multicast Routing for Mobile Ad Hoc Networks," TR-03-004, Dept. Comp. Sci., Univ. of Mannheim, 2003; ACM Mobihoc, 2003.

[23] K. Seada, A. Helmy, and R. Govindan, "On the Effect of Localization Errors on Geographic Face Routing in Sensor Networks," *Proc. ACM IPSN*, 2004.

ADDITIONAL READING

[1] F. Kuhn, R. Wattenhoffer, and A. Zollinger, "Worst-Case Optimal and Average-Case Efficient Geometric Ad Hoc Routing," *Proc. ACM MobiHoc*, 2003.

[2] N. Li, J. C. Hou, and L. Sha, "Design and Analysis of an MST-Based Topology Control Algorithm," *Proc. IEEE INFOCOM*, 2003, San Francisco, CA.

BIOGRAPHIES

IVAN STOJMENOVIC (ivan@site.uottawa.ca) received a Ph.D. degree in mathematics. He has held positions in Serbia, Japan, USA, Canada, France and Mexico. He has published over 200 different papers and edited three books on wireless, ad hoc, and sensor networks with Wiley and IEEE Press. He is currently an editor of several journals including IEEE TPDS. He has recently guest edited special issues in several journals including *IEEE Computer* (February 2004), *IEEE Network* (July 2004), and *Wireless Communications and Mobile Computing* (Wiley).