Geocasting with guaranteed delivery in sensor networks

Ivan Stojmenović

SITE, University of Ottawa, Ottawa, Ont K1N6N5, Canada www.site.uottawa.ca/~ivan
ivan@site.uottawa.ca

Abstract

In a geocasting problem, a message is sent from one node to all the nodes located in a designated region. For example, monitoring center needs to contact all active sensors within a monitored area to either gather data from them periodically, or to provide its location to sensors covering 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 then describe three approaches to guarantee delivery. Two of them are face traversal schemes and are based on depth-first search of the face tree and traversal of all faces that intersect the border of geocasting region, respectively. In the entrance zone multicasting based approach, the monitoring center divides entrance ring of geocast region into zones of diameter equal to the transmission radius. The problem is decomposed into multicasting toward centers 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 face tree traversal scheme).

1. Introduction

Recent technological advances have enabled the development of low-cost, low-power, and 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 together 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 distance that will drain the power quickly. Hence, sensors operate in a self-organized and decentralized manner and message communication takes place via multi-hop spreading. To enable this, the network must maintain the best connectivity as long as it is possible. Sensor's battery is not replaceable, and sensors may operate in hostile or remote environments. Therefore energy consumption is considered as 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 message 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 are discussed recently in [CS, WXZLP, ZH]. 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 then monitoring center may spread the task and establish reverse broadcast tree using any intelligent flooding protocol (see [SW]). If the network is reasonably sparse, even blind flooding (where each node receiving message will retransmit it exactly once) is a viable option. Otherwise, that is when the region to be monitored for particular event contains only a small portion of active sensors, flooding the whole network may become inefficient way for spreading the task. This article reviews existing solutions to the geocasting problem. In a multifunctional multi-event sensor environment, monitoring center may separately handle several geocasting regions and corresponding events. One particular application of geocasting is in tracking mobile objects. Monitoring center may collect reports from sensors in the vicinity of the object, and send periodic signals to sensors adjusting geocasting region, following 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 reasonably accurate sensor location (when sensors are not directly equipped with GPS receiver, which becomes technologically feasible) is intensively studied recently (see a survey [N]). We shall consider only localized approach in this article. In a localized routing or geocasting algorithm, each node makes decision to which neighbor(s) to forward the message based solely on the location of itself, its neighboring nodes, and destination. In case of geocasting, *D* is a node approximately in the center of the region, this includes geocast region description. We also assume that sensor network is static, and the monitoring center is aware of 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 literature [AP, BCS, CL, HLR, KV1, LTLS, SK, SRL], does not guarantee delivery to all the nodes inside 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 claimed in literature, we show that the only one that is able to guarantee delivery is a ‘forgotten’ face tree traversal scheme in [BMSU, M]. It is fully memoryless scheme (all the information needed for making traffic decisions are 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 medium access 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 ideal MAC layer. A geocasting protocol therefore has guaranteed delivery property if each node, located inside geocasting region, and connected to the source node, will receive the packet if ideal MAC layer is applied. We describe in this article three nontrivial solutions for the geocasting task with guaranteed delivery. One is obtained from [BMSU] by adding a preprocessing step, making algorithm message efficient afterwards (it, however, requires 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 recently proposed geocasting scheme [SH] does not guarantee delivery, despite the 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. Section 2 presents localized location based routing and geocasting algorithms. A geocasting algorithm with guaranteed delivery, based on traversal of faces intersecting geocast region boundary, is described in Section 3. Another algorithm [BMSU] based on traversing face tree, is described in section 4 (a variant of it, with preprocessing, and reducing message complexity, is proposed here). Section 5 describes a geocasting protocol that guarantees delivery and is efficient compared to possible alternative. Finally, conclusion and references complete this article.

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

2. Position based localized routing and geocasting algorithms

Finn [F] proposed *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 for all its one hop neighbors to determine the neighbor *A* which is closest to *D* among all neighbors of *S*. Fig. 2 is an example where *A* is the 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 geocast region center, not a real node).

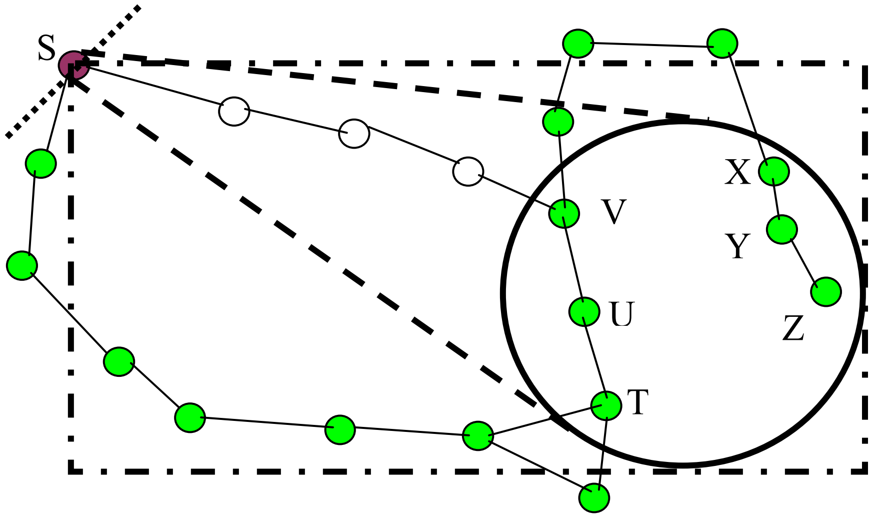

Figure 1. Example where most existing geocasting schemes do not guarantee delivery

Almost all existing geocasting algorithms [BCS, KV1, AP, SRL, LTLS, SK, CL] are based on forwarding messages within a restricted area between source (or node currently holding the packet) and geocasting region, such as between tangents from current node or source to geocasting region, rectangle containing source and geocasting region, or all nodes closer to geocasting region than 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 source. If white nodes in Fig. 1 are not in the network, the source is disconnected from geocasting region, but could be connected via nodes that do not belong to indicated regions. Additional problem is that these methods do not consider that nodes inside geocasting region may be disconnected (but could be connected via nodes outside it). Thus connecting 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 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 [BMSU]. The *GFG* (Greedy-Face-Greedy) algorithm [BMSU] applies greedy method until current node *A* has no neighbor closer to destination than itself (such node is called *concave* node [BMSU]), or the message is delivered. Concave node *A* switches to the recovery mode, by applying face routing [KSU], improved in [BMSU]. Face routing uses only edges of a planar graph. Gabriel graph (*GG*) was used in [BMSU], constructed as follows. An edge *AE* belongs to *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 *GG* are marked with bold lines. Face routing guarantees delivery in connected planar graphs, but is followed only until a node closer to destination than the last concave node is encountered. Such a node switches back to the greedy mode. This mode alteration may repeat few times, but the message is guaranteed the delivery, and *GFG* algorithm was shown to be competitive with respect to the shortest path, especially with some improvements given in [DSW]. The improvements in [DSW] include restricting face routing to nodes in a connected dominating set, and applying a shortcut procedure, explained in a later section.

Let us illustrate the *GFG* routing [BMSU] on example in Fig. 2, for 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 imaginary line from source to destination, changing faces at intersections of imaginary line with the faces of *GG*. Face route from *V* to *Y* follows open face as marked with scribble line (route *VGHZ*). The return from recovery mode to greedy mode is possible at node *Z* which is closer to destination than the previous concave node *V*. *Z* then delivers to *Y,* and the imaginary line *VY* was never crossed in this example.

A simple geocasting algorithm was proposed in technical report version of [SRL]. Source node *S* applies *GFG* algorithm [BMSU] to route toward center *D* of 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 smaller flooding rate and increased delivery rate compared to all known methods. However, it also fails to guarantee delivery (see Figs. 1, 2, 3). Nevertheless it is used as basic ingredient in the scheme that does guarantee delivery, described in section 4.

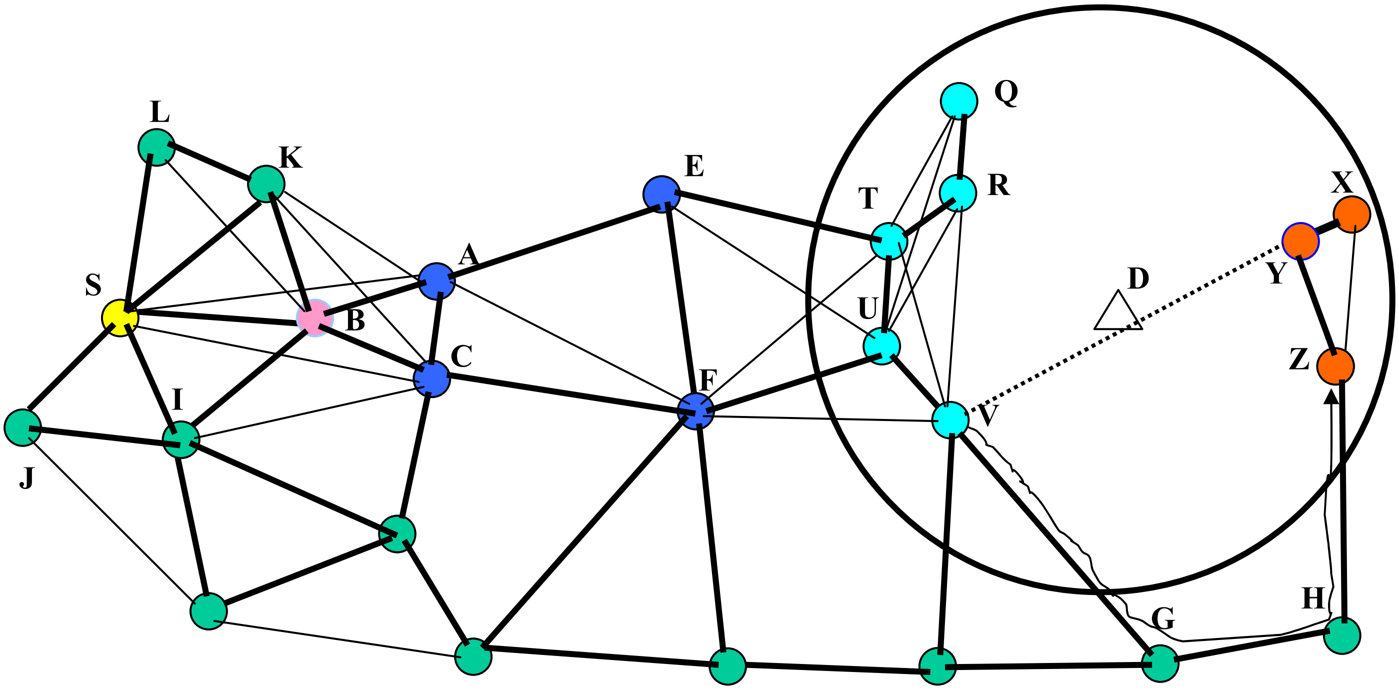

Figure 2. Gabriel graph, Face routing and *GFG*

3. Geocasting based on traversing faces that intersect boundary

Bose, Morin, Stojmenovic and Urrutia [BMSU] observed that a geocasting algorithm will guarantee delivery if all faces of a planar graph which are inside or intersect geocasting region are traversed. The algorithm is based on depth first search of 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.

Saeda and Helmy [SH] observed that it is sufficient to traverse only faces that intersect the boundary of a given geocasting region, and proposed the following algorithm. Source node first uses GFG algorithm [BMSU] to forward the packet toward the region. Each node that is inside 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 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' [SH]. We showed in [S] that this algorithm [SH] does not guarantee delivery, despite the claim. A geocasting algorithm that guarantees delivery can be described as follows.

Algorithm **Geocast _traversal_intersecting_faces**

- Source node *S* a sends the message toward geocasting region, using *GFG* algorithm [BMSU].
- 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 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.

The main difference between this algorithm and one in [SH] 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 arrived to it (in [SH], it is activated only from internal border neighbor, for one face at a time).

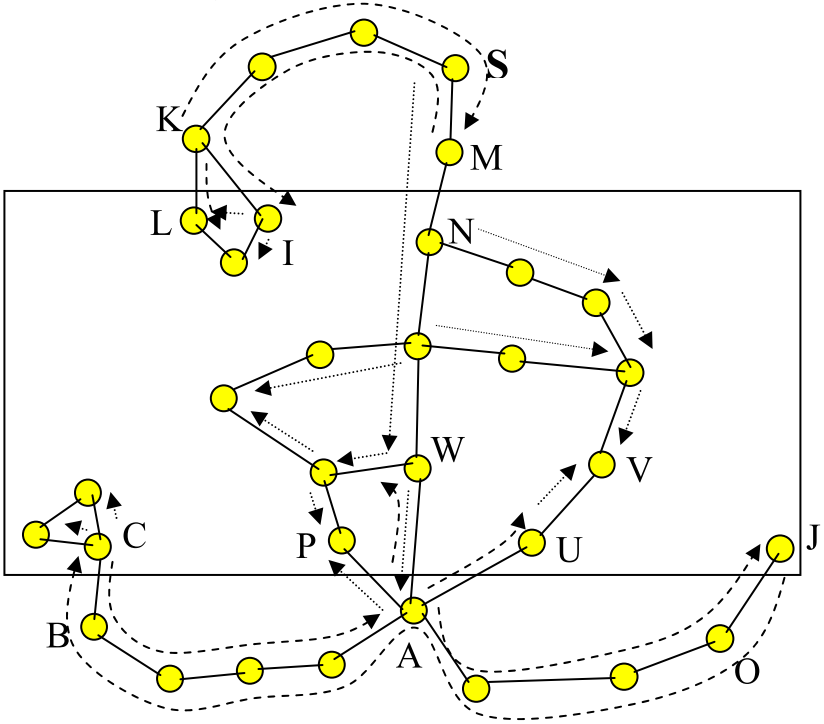

Figure 3. Face traversal based geocasting with guaranteed delivery

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 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*. Face traversal with respect to *KI* ends at *L*, while face traversal with respect to *KL* follows 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*) bypasses *A* and reaches nodes *B* and *C*. *C* floods to its neighborhood while *B* starts face traversal that reaches *A* again.

We shall now *prove* that the algorithm indeed guarantees delivery to all nodes inside 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 geocast region boundary, guarantees delivery to all nodes inside geocast region, connected to the source.

Proof. We can argue that every face, intersecting geocasting region and connected to the source, was fully traversed by combination of regional flooding and outer face traversals. Consider, for example, outer boundary in Fig. 3 (the proof is same for any face). Its traversal started at *MN* and reached I (by lower dashed line in Fig. 3). With internal flooding, it reached *LK* from *I*. Then from *K* it reached *SMN* (upper dashed line) . By regional flooding it can reach *UA*. Then face traversals are used to follow *AJ* 'piece', followed by *JABC* 'piece'. Face traversal from *C* then reaches *A* again, and with regional flooding it reaches *D*. Then face traversal *DEFGH*. Finally, flooding from *H* can reach back *MN*, and the whole face is traversed. We could make the proof more formal, but believe that 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 region, and pieces outside region that are triggered when packet arrives there. Regional flooding, piecewise face traversal and connectivity assure reaching all possible nodes.♦

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

4. Geocasting based on depth-first search traversal of face tree

Bose et all (journal version of [BMSU]) 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 [BMSU] (whose complete description is available in [M]) does not require any memory to be left at nodes, and needs only to carry some small amount of information with the packet (if entry edges are predetermined for a given source, then the message only needs to contain sender and source information).

The algorithm [BMSU] first applies *GFG* to route toward a node inside geocasting region. That node then selects a nearby point *S* inside the face to act as 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 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 edge to *S,* where the distance is the decided by a point *C* from the edge that is closest to *S.* If distances are same, secondary keys used is the counterclockwise direction of vector *SC.* In case of further ties (which may occur only when two edges share common closer endpoint *C*), consider the size of the angle *∠SCD,* where *D* is the other endpoint of the edge. If that still does not resolve then consider the vector *CD* which then must be different. Morin [M] 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 other entry face closer to *S,* which confirms that a tree of faces is indeed constructed. Face tree is dynamically constructed, during geocasting operation. The geocasting algorithm follows depth first search based traversal of face tree. For each node in face tree, it actually traverses the corresponding face. When an entry edge is encountered, the traversal enters new face. When traversal (which may recursively go to deeper levels) is completed, the traversal 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. Face tree from *S* is drawn with directed edges intersecting entry edges (which are drawn with dashed lines). The path taken by geocasting algorithm is shown with dotted scribble line, starting from point *S*, which is near real node *U* acts on behalf of *S*. The algorithm visits all edges along the path.

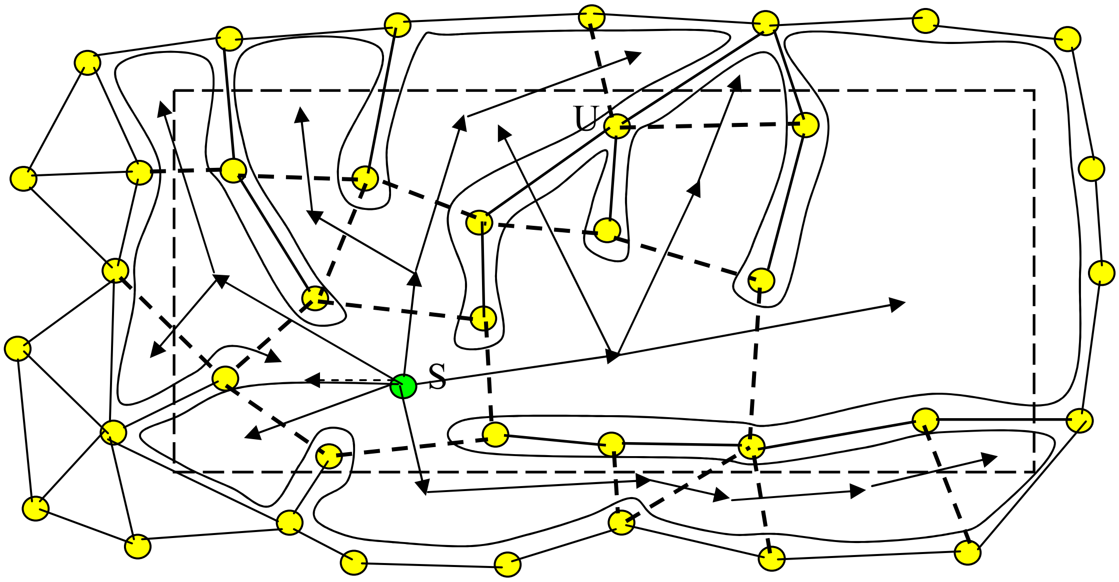

Figure 4. Face tree traversal based geocasting with guaranteed delivery

The algorithm [BMSU, M] can be described as follows. In this scheme, opposite(*e*,*f*) is the other face containing the same edge *e* as the face f currently being traversed. The edge next(*e*,*f*) is the next edge being traversed by right-hand rule from current edge *e* on face *f*.

Algorithm **Geocast_Face_Tree_Traversal**

```
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
```

In some applications (see section 4.3), entry edges may be determined as the preprocessing step. For example, 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. Afterwards, geocasting regions may be dynamically determined (e.g. to follow a moving object) and geocasting may then proceed as described. It was shown in [S] that, in this variant (after preprocessing), the number of messages for geocasting is $< 2(N-1) < 2(n'+k)$. Communication steps from source to the geocasting region need to be added, for this and other geocasting algorithms. Therefore the scheme has reasonable communication overhead under given assumption. Compared to the previously described scheme, it has less communication overhead when $k<n'$. It does require preprocessing (or significant additional number of messages at run time) and offers 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 [S]). Lindsey, Raghavendra, and Sivalingam [LRS] proposed such framework for energy efficient data gathering algorithms in sensor networks, but do not describe any localized algorithm for deciding the order of transmission, which can be achieved by described scheme.

For dynamically selected source of geocasting message, such preprocessing is not possible. A scheme for testing whether a given edge is an entry edge is described in [BMSU, M, S]. The number of messages sent in the scheme overall is $O(N + k \log k)$ [BMSU, M], where the later 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 [BMSU, M].

5. Multicasting and geocasting with guaranteed delivery

We now describe an algorithm based on the following observation. If a node V inside geocast region is connected to the source S then the first node U on a route from S to V is no more than transmission radius distance R from the border of 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, can be described as follows.

Algorithm **Geocast_Entrance_Zones_Multicast**

- 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 geocasting region, and inside the region, and dividing such entrance ring into zones, each with diameter ≤*R,* in arbitrary fashion,
- Multicast from source *S* toward centers of each entrance zone, until a node inside zone is reached (these nodes will be called multicast recipients), or a loop in recovery mode of routing scheme is identified,
- Flood from each multicast recipients. This can be done by blind flooding restricted to nodes inside the region, or by some intelligent flooding scheme [SW] which reduces the number of retransmissions. Each node memorizes received packets and ignores repeated copies of the same packet.

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

5.1. Determining entrance zones

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

- it is not possible to send directly a message from a node outside the geocasting region to a node inside it; this means that the width of the all zones together, measured as minimum distance between a node outside geocasting region to a node inside it which 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 then all other nodes in the same zone will receive it after retransmission from that multicast recipient.

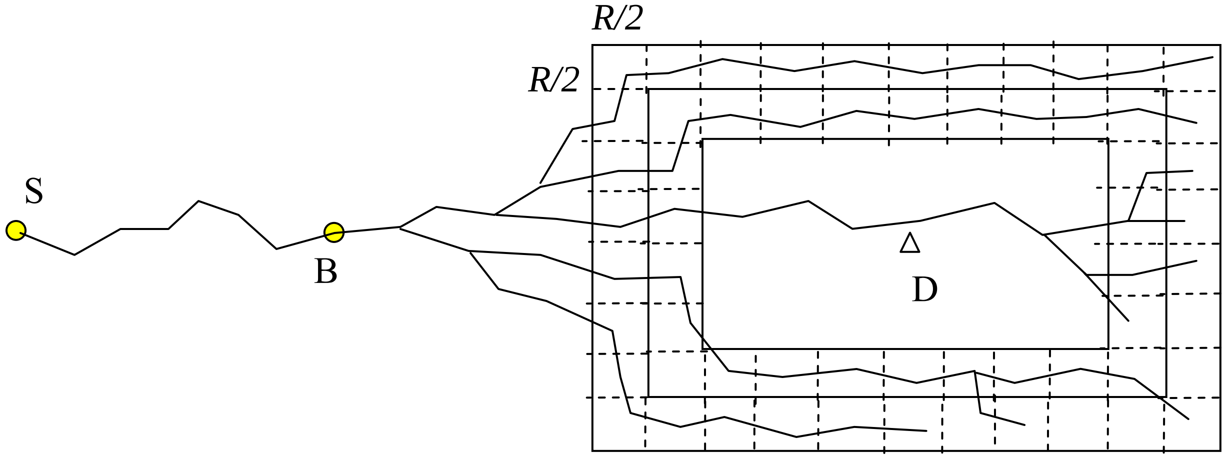

Figure 5. Covering entrance zones from a remote monitoring center

The exact construction of entrance zones to satisfy these criteria depends on the shape of geocasting region. If the geocasting region is a rectangle then, 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 route to them will end up in a loop (for clarity, these loops are not drawn in Fig. 5).

5.2. Position based multicasting

In a multicasting task, 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 [MFWL], authors propose two similar multicasting schemes, with some optimizations. In the *optimal paths* method, each node receiving 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 neighbor that is closest to it (provided that neighbor is closer to it than 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 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 maximal number of remaining destinations etc. The forwarding list of multicast group similarly is changed as in the previous algorithm [MFWL]. In both schemes, if no neighbor is closer to one or more destinations then the recovery mode in *GFG* algorithm [BMSU] is applied. The virtual destination used for the 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 recovery scheme) corresponds to the *VD-greedy* scheme [SRL]. They both use hop count as the metric. Both optimal and aggregate paths methods can be modified by considering metrics different from hop count, such as power, cost, delay or other. Greedy routing can be replaced by power and/or cost aware routing, and forwarding neighbors will be judged based on metric in question, combined with their coverage ability, for their selection.

5.3. Entrance zone multicasting based geocasting with guaranteed delivery

The algorithm consists of multicasting toward ‘centers’ of all entrance zones, and flooding from the first nodes encountered in each non-empty zone. Zone center is any node inside it, for example its center of mass or intersection of zone diagonals). In Fig. 5, 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 *GFG* algorithm to confirm that (these loops are not drawn). Multicasting scheme can be followed in one of ways described above. One more specific example for this scheme can be found in accompanied full version of this article [S]. Note that some optimizations can be made here. For example, few nodes on a path can collectively conclude that some zones are empty, and prevent full loop in *GFG*. Optimization via merging assigned zones can be made when 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 an intelligent flooding inside geocast region. In all exiting intelligent flooding methods (see a review in [SW]), nodes may receive multiple copies of the same message, but forward it once only (or not forward at all), after a timeout which depends on protocol selected. The 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 longer, 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.

The monitoring center S may be outside or inside geocasting region. Although our description implicitly assumes that S is outside the region, the same algorithm works correctly if S is also inside it. We will 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 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 guaranteeing delivery property of *GFG* (proven in [BMSU]), which is applied towards every destination. The guaranteed delivery of multicasting is also claimed in [MFWL]. Next, we argue that any node inside 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 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 retransmitted message from that recipient. If it is outside all entrance zones and connected to the source S, then the path from S to X needs to cross entrance zones ring somewhere. Since the width of that ring is R, it cannot ‘jump’ over it and cross directly from outside geocasting region to inside geocasting region (‘escaping’ 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 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 which do not guarantee delivery [S]. The comparative communication overhead depends on relative distance from monitoring center to geocast region. It also depends on the existing coverage of geocast region by active sensors. Obviously, several empty regions may cause long routes along network perimeter to recognize them.

The 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 network perimeter for each empty region. Consecutive empty entrance zones, fortunately, do not necessarily require separate face routings, since multicasting method merges them into a single destination. The worst case appears to be the scenario with every other entrance zone being empty, thus each of them 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 (solves broadcasting task), and is best when geocasting region is nearly covering the whole network. It is also competitive when geocasting region covers large portion of the network. The three methods presented here are designed for cases where geocasting region is relatively small. Among three proposed schemes, it is expected that (on average) traversing faces that intersect boundary will perform best when there are many empty entrance zones, otherwise 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 monitoring center. More reliable conclusions can be made only after their performance evaluation.

The performance of described geocasting protocols need to be evaluated experimentally. The described 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 proposed geocasting protocols. Note that the effect of localization errors on the performance of face routing scheme [BMSU] is presented in [SHG].

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

We assume that all active sensors within monitored region need to be ‘alarmed’, and assume the application of localized algorithms. Geocasting then needs to guarantee delivery, and this article describes all such existing methods. The next step is their analysis by means of performance evaluation. The proposed schemes have its variants, and allow for optimization with variety of criteria and variety of options for their implementation. Comparison between geocasting methods depends on relative size of 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 size of geocast region, the more advantages our described methods provide. Their mutual comparison depends on parameters such as density, and existence of ‘holes’ in the network. Performance evaluation may reveal also lead to further improvements of each presented method, or their adjustments to particular scenarios or evaluation criteria. The performance evaluation is left for future work.

The described geocasting protocols may also be used for few other related applications within sensor networks. One or more monitoring stations may simultaneously geocast to one of more regions. In case of monitoring region consisting of several disconnected sub-regions, 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 network have recently attracted exponentially increasing interest, including creation of new conferences, new journals, publishing a number of books. We envision that this trend will continue in the short term, and we envision that network layer problems, discussed in this article, will continue to be intensively studied. We hope that the research efforts will lead toward real applications of ad hoc network, especially sensor networks.

Acknowledgement

This research is partially supported by NSERC Discovery grant.

References

[AP] B. An and S. Papavassiliou, Geomulticasting: architectures and protocols for mobile ad hoc networks, J. Parallel and Distributed Computing, 63, 2003, 182-195.

[BCS] S. Basagni, I. Chlamtac, V.R. Syrotiuk, Geographic messaging in wireless ad hoc networks, Proc. 49th IEEE Int. Vehicular Techn. Conf. VTC'99, Houston, TX, May 1999, 3, 1957-1961.

[BMSU] P. Bose, P. Morin, I. Stojmenović, and J. Urrutia, Routing with guaranteed delivery in ad hoc wireless networks, ACM DIAL M, Aug. 1999, 48-55; Wireless Networks, 7, 6, 2001, 609-616.

[CL] T. Camp and Y. Liu, An adaptive mesh-based protocol for geocast routing, J. Parallel Distributed Computing, 63, 2003, 196-213.

[CS] J. Carle and D. Simplot-Ryl, Energy efficient area monitoring by sensor networks, IEEE Computer, February 2004, 40-47.

[DSW] S. Datta, I. Stojmenović, J. Wu, Internal nodes and shortcut based routing with guaranteed delivery in wireless networks, Cluster Computing, 5, 2, 2002, 169-178.

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

[HLR] Q. Huang, C. Lu, G.C. Roman, Mobicast: Just-in-time multicast for sensor networks under spatiotemporal constraints, TR WUCS-02-42, Washington Univ., St. Louis, USA, Dec. 2002.

[KSU] E. Kranakis, H. Singh and J. Urrutia, Compass routing on geometric networks, Proc. 11th Canadian Conf. on Comp. Geom., Vancouver, August, 1999.

[KV1] Y.B. Ko and N. Vaidya, Flooding-based geocasting protocols for mobile ad hoc networks, Proc. WMCSA, 1999, New Orleans; Mobile Networks and Applications 7, 471-480, 2002.

[KWZ] F. Kuhn, R. Wattenhoffer, A. Zollinger, Worst-case optimal and average-case efficient geometric ad hoc routing, Proc. ACM MobiHoc, 2003.

[LHS] N. Li, J.C. Hou, L. Sha, Design and analysis of an MST-based topology control algorithm, Proc. IEEE INFOCOM, 2003, San Francisco, USA.

[LRS] S. Lindsey, C. Raghavendra, K. Sivalingam, Data gathering algorithms in sensor networks using energy metrics, IEEE Trans. Parallel & Distrib. Systems, 13, 9, Sept. 2002, 924-935.

[LTLS] W.H. Liao, Y.C. Tseng, K.L. Lo, J.P. Sheu, Geogrid: A geocasting protocol for mobile ad hoc networks based on grid, Journal of Internet Technology, 1, 2000, 23-32.

[MFWL] M. Mauve, H. Fusler, J. Widmer, T. Lang, Position-based multicast routing for mobile ad hoc networks, TR-03-004, Dept. Comp. Sci., Univ. of Mannheim, 2003; ACM Mobihoc, 2003.

[M] P. Morin, Online routing in geometric graphs, PhD thesis, School of Computer Science, Carleton University, January 2001.

[N] D. Niculescu, Positioning in ad hoc sensor networks, IEEE Networks, 2004, to appear.

[SH] K. Saeda, A. Helmy, Efficient geocasting with perfect delivery in wireless networks, Proc. WCNC, March 2004.

[SHG] K. Saeda, A. Helmy, R. Govindan, On the effect of localization errors on geographic face routing in sensor networks, Proc. ACM IPSN, 2004.

[SK] C. Schwingenschlogl and T. Kosch, Geocast enhancements of AODV for vehicular networks, ACM Mobile Computing and Communications Review, 6, 3, 2002, 96-97.

[SRL] Stojmenović I., A.P. Ruhil and D.K. Lobiyal, Voronoi diagram and convex hull based geocasting and routing in wireless networks, Proc. IEEE Int. Symp. on Computers and Communications ISCC, Kemer-Antalya, Turkey, June 30 – July 3, 2003, 51-56; SITE, Univ. of Ottawa, TR-99-11, Dec. 1999.

[S] I. Stojmenović, Geocasting in ad hoc and sensor networks, SITE, University of Ottawa, TR-04-02, March 2004.

[SW] Stojmenović I. and J. Wu, Broadcasting and activity scheduling in ad hoc networks, in: Ad Hoc Networking (S. Basagni, M. Conti, S. Giordano and I. Stojmenović, eds.), IEEE Press, to appear.

[WXZLP] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, C.D. Gill, Integrated coverage and connectivity configuration in wireless sensor networks, Proc. ACM SenSys, Los Angeles, November 2003.

[ZH] H. Zhang and J. C. Hou, Maintaining Sensing Coverage and Connectivity in Large Sensor Networks, UIUCDCS-R-2003-2351, June 2003.