Available online at www.sciencedirect.com
ScienceDirect
Computer Communications 30 (2007) 2519–2531
computer communications
www.elsevier.com/locate/comcom

Energy-efficient geographic multicast routing for Sensor and Actuator Networks

Juan A. Sanchez [a,*], Pedro M. Ruiz [a], Ivan Stojmenovic [b]

[a] Department of Communications and Information Engineering, University of Murcia, Espinardo, 30071 Murcia, Spain
[b] SITE, University of Ottawa, Ottawa, ON KIN 6NR, Canada

Available online 21 June 2007

Abstract

The case in which the same information or events need to be sent from a single sensor node to multiple actuator nodes, is very common in many applications of Sensor and Actuator Networks (SANET). Sensors have very limited resources in terms of energy, bandwidth, and computational power. Thus, routing messages preserving energy and network bandwidth is a challenging requirement of paramount importance. In this paper we present a novel energy-efficient multicast routing protocol called GMREE which is specifically designed to achieve that goal. Our protocol builds multicast trees based on a greedy algorithm using local information. The heuristic we use is based in the concept of cost over progress metric and it is specially designed to minimize the total energy used by the multicast tree. GMREE incorporates a relay selection function which selects nodes from a node's neighborhood taking into account not only the minimization of the energy but also the number of relays selected. Nodes only select relays based on a locally built and energy-efficient underlying graph reduction such as Gabriel graph, enclosure graph or a local shortest path tree. Thus, the topology of the resulting multicast trees really takes advantage of the benefit of sending a single message to multiple destinations through the relays which provide best energy paths. Our simulation results show that our proposed protocol outperforms the traditional energy-efficient multiunicast routing over a variety of network densities and number of receivers. In addition, for dense networks, the performance approximates the one achieved using the centralized shortest weighted path tree (computed by Dijkstra's algorithm).

Keywords: Multicast; Energy efficient; SANET; WSN

1. Introduction

SANETs consist of a set of networked sensor and actuator nodes that communicate among each other using wireless links. They work in a distributed way, and collaborate to perform automated tasks requiring sensing and actuation capabilities. Sensor nodes are usually small, inexpensive and with limited communication, computation, and energy resources. The number of such sensors in a SANET is expected to be large, in the order of hundreds or thousands. Unlike sensors, actuators usually have more resources and their number is usually much more limited.

Both sensors and actuators are usually static, although some actuators may also be mobile. Even if all of them are static, the topology of the network changes over time. The reason is that nodes usually operate in duty-cycle, with awake and sleeping periods to save energy. Thus, the topology formed by active sensors and actuators changes as they change their state over time. In fact, a dynamic routing protocol is employed to communicate nodes which are not within radio range. Communication between such nodes in these networks is made upon the concept of relay node. That is, when source and destination are not within radio range, a multihop path is created using some intermediate nodes between them as relays. The selection of the next hop relay is performed by each intermediate node based on some routing algorithm. For instance, in geographic routing protocols this decision is taken based on the position of the neighbors with respect to the

* Corresponding author.
E-mail addresses: jlaguna@dif.um.es (J.A. Sanchez), pedrom@dif.um.es (P.M. Ruiz), ivan@site.uottawa.ca (I. Stojmenovic).

0140-3664/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2007.05.032

destination. Both neighbors and their positions are discovered by beacon messages which are broadcasted periodically by each node to its 1-hop neighborhood.

SANETs have a lot of potential for a wide range of monitoring and control applications such as healthcare, improvement of productive processes, automatic control, etc. In many of those scenarios, sensors may need to send the same information to multiple actuators. For instance, we are developing a system for water irrigation control in agricultural exploitations. In such system, actuators are in charge of opening or closing valves depending on the need to water some areas based on information measured from sensors. In our design, sensors need a very lightweight and energy-efficient algorithm to send data and commands to multiple actuators. Given that the position of actuators is well-known and static, a geographic routing scheme is most suited to our scenario.

In general, although sensor nodes consume energy sensing and processing, the most energy consuming operation is communication. In most hardware platforms existing today, the ratio between the energy cost of sending one bit and the cost of processing one instruction is in the order of 190–2900. For instance, Telos motes based on the IEEE 802.15.4 physical layer, consume 54.5 μA for the CPU and 17.4–8.5 mA for radio transmission (depending on output power) [1]. Thus, it is of vital importance for many applications to count on energy-efficient routing protocols to deliver data from a sensor to a set of actuators. The multicast communication paradigm is an ideal solution for effectively supporting this kind of communication because it is specifically thought of to reduce the consumption of network resources to route data packets to multiple destinations by sending a single copy of the message, duplicating it at relay nodes only when it is strictly necessary. Although multicast can be seen as a special case of broadcast, designing efficient multicast routing protocols is a very challenging task. In fact, the problem of energy-efficient multicast routing is well-known to be NP-complete. In addition, given that geographic routing is also a very interesting alternative for these scenarios in which position of the nodes is known, we use a localized greedy multicast routing heuristic to build energy-efficient multicast paths.

In this paper we present GMREE, which is, to the best of our knowledge, the first energy-efficient geographic multicast routing protocol designed specifically to fulfill the requirements of wireless sensor networks. GMREE is an adaptation of the GMR protocol [2] which only uses the position of neighbors and destinations to take routing decisions. Our algorithm has two key components: the neighbor selection function and the relay set optimization phase. Neighbor selection function takes decisions about which neighbor is the best candidate relay for each destination based on a cost over progress ratio. That is, a tradeoff between the cost of reaching that neighbor, and the progress it provides towards that particular destination. The second phase improves the initial relay set selected in the first stage by reducing the number of forwarders. This allows for a higher path reuse among destinations, which also means an overall reduction in energy. Our proposed scheme can use different energy-efficient underlying graphs to enhance path selection, and reduce the amount of energy consumed. We have tested three of those underlying graphs: Gabriel Graph, Enclosure Graph, and Dijkstra's Local Shortest Path Tree, being the latter the one getting best results.

The remainder of the paper is organized as follows: Section 2 presents an overview of the state of the art in the topic. We describe our network and energy models in Section 3. Section 4 describes the GMREE protocol. Our proposed greedy neighbor selection algorithm is explained in Section 5. We evaluate in Section 6 the performance of our solution using simulation. Finally, Section 7 provides some conclusions and discusses open issues.

2. Related work

Previous distributed routing protocols for mobile ad hoc networks are not able to operate over SANETs due to their limited scalability and intensive use of flooding to find routes. Location-based routing has emerged as one of the most effective schemes to deal with the scalability of routing in large-scale wireless sensor networks, with very light requirements in terms of computing power and memory of the devices.

The idea of greedy routing based on position information was first introduced by Finn and many other greedy-based routing protocols based on the same concept but with different metrics have been proposed later on in the literature [9]. Recently, Stojmenovic [20], proposed a general framework for those protocols called cost over progress. The idea is to define a metric for the cost of selecting next hop (e.g., energy required to reach it) as well as a function to quantify the progress (e.g., reduction of overall distance remaining to reach the destination). Then, the node with the best ratio is the selected neighbor. We now show examples how the cost over progress metric can be used to select next hop (in a routing task) towards destination when nodes know position information. The cost metric depends on the assumptions and metrics used, while progress accounts for the advance towards the destination. Several known protocols are special cases of this design.

In a unicast routing scenario, let C be the node currently holding the message, D the destination node, A the forwarding neighbor being evaluated, $|\overline{CD}| = c$, $|\overline{AD}| = a$, and $|\overline{CA}| = r$ (see Fig. 1). A simple special case of this general design is greedy routing in sensor networks by Finn [9] for the generally accepted unit disk graph (UDG) model. The cost metric normally used in UDG is the hop count. Therefore, the cost of going to any neighbor is one hop. The progress made by forwarding from C to A is $c - a$, which is difference between distances to destination from current node and neighboring node. Therefore, the protocol will minimize $\frac{1}{(c-a)}$, or maximize $c - a$; that is, neighbor closest to destination would be selected. An alternative

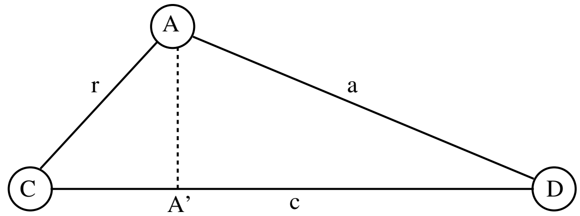


Fig. 1. Selecting the best neighbor C in localized routing schemes.

measure of progress is the distance $|\overline{CA'}|$, where A' is the projection of A on line $\overline{CD}$ (see Fig. 1).

This method is known as MFR (Most Forward within Radius), proposed by Takagi and Kleinrock [10]. Note that, it is necessary that only neighbors closer to the destination than current node are considered. This guarantees the protocol to be loop-free. To guarantee delivery when such neighbor does not exist, an additional procedure usually called face routing is described by Bose et al. [5].

Face routing routes along the faces of the graph. Thus, the face routing algorithm is applied over a planar subgraph of the original network in which crossing edges are removed. The combination of greedy routing and face routing is called Greedy–Face–Greedy (GFG). It uses greedy routing and if a local maximum is eventually found, face is used until greedy routing can be applied again or the destination is reached. These algorithms do not offer geographic multicast communications but their contribution is very valuable starting point to design new multicast variants.

Given the difficulty of adapting existing ad hoc multicast protocols to a fully localized operation, and the relative novelty of the commented geographic approaches in unicast scenarios, only a few new protocols have been recently proposed provide geographic multicast routing. One of such protocols is Position Based Multicast Routing (PBM) protocol proposed by Mauve et al. in [11]. This protocol, although not initially thought for sensor networks, fulfills most of desired design criteria of locality and limited network overhead. It is a generalization of GFG [5] routing to operate over multiple destinations. It builds a multicast tree, whose shape can vary from the shortest path tree, to an approximation of a minimum cost multicast tree depending on a parameter denoted as λ. However, the election of the proper λ for each scenario remains unsolved. Sanchez et al. describe in [2] a free-of-parameters algorithm called GMR, which is based on the cost over progress framework. Results show that GMR outperforms PBM regardless of the λ value selected over a variety of network scenarios. The reason is that the neighbor selection function manages to achieve a very good approximation of the minimum bandwidth consumption multicast tree. In addition, GMR is more efficient in computation time, thanks to a greedy set merging scheme. Scalable Position Based Multicast for Mobile Ad-Hoc Networks (SPBM [16]) is another multicast protocol designed to improve scalability. It uses the geographic position of nodes to provide a scalable group membership scheme and to forward data packets. SPBM is mainly focused on the task of managing multicast groups in an scalable way. However, they fail to provide efficient multicast forwarding, because it uses one separate unicast geographic routing for each destination, which turns out to be inefficient for most network scenarios. Unfortunately, these protocols are not designed to take into account the energy spent by the network doing the task of routing.

The problem of routing in a energy efficient way is even harder than the routing itself. There are many energy-efficient geographic unicast routing protocols. For instance, iterative power described in [15] and GPSR-S [18] are the most representative. In both cases, an enhanced cost function considering energy is used to select next hops based on the optimization of energy consumption. Regarding energy-efficient broadcast, there are some centralized heuristics such as BIP [7] which tend to outperform distributed ones (this is valid only if the energy cost needed to maintain global information in dynamic network is ignored). But there are also some very efficient localized solutions. For instance, Ingelrest et al. describe TR-LBOP in [14] which is one of the few localized algorithms having a similar performance to BIP. They develop the idea of the target radius, that is the optimal radius to achieve energy-efficient broadcast, combined with a localized neighbor elimination scheme (NES).

Regarding energy-efficient multicast, MIP [7] is the variant of BIP for multicast but its main problem is that it is computed by selecting a subtree consisting of the paths which connects the source to the destinations in the original BIP tree. Thus, given that BIP is explicitly built for broadcast, the selected paths are not optimal because in some cases paths are not reused for multiple destinations. Recently, Guo and Yang [8] proposed an energy-efficient multicast routing algorithm for ad hoc networks which makes use of an initial flooding to build a first multicast tree, which is then refined by local operations coordinated by the multicast source, and performed one at a time. Those local operations are based on information carried out by periodic in-tree floodings initiated by the multicast source. They assume that distance to neighbors is known by each node and their protocol is thought of to deal with mobility. During the optimization phase, the source sends along the tree the whole route from the source to destinations. Hence, this protocol is not expected to be scalable for large-scale wireless sensor networks. Thus, we propose a energy-efficient geographic multicast routing protocol working with local information and being able to scale as their unicast counterparts.

One common option to enhance routing solutions is to execute them on a subgraph of the original network, which is constructed to improve some particular aspect of the protocol. For instance, to reduce energy consumption only those links providing energy-efficient paths to the rest of neighbors are really interesting for a node. The algorithms to build those local subgraphs are commonly called topology control algorithms [22]. For energy efficiency, we have considered three well-known subgraphs: Gabriel Graph (GG), Enclosure Graph (EG), and Local Shortest Path Tree (LSPT). GG was proposed in [19] is a well-known planar graph used in several disciplines. It is defined as follows. GG has an edge uv if and only if the disk with diameter uv contains no other node inside it. Each common neighbor w of nodes u and v should be located at distance at least $\frac{|uv|}{2}$ from the midpoint of uv for uv to be included in GG. Rodoplu and Meng [RM] introduced the enclosure graph for localized power aware topology control in ad hoc networks. The power needed for transmitting between two nodes at distance r is proportional to $u(r) = r^\alpha + c$, where α is power attenuation factor (a number between 2 and 6), while c is a constant that accounts for the cost of running hardware at nodes and minimum reception power. Although most researchers assume $c = 0$ which enables to prove some nice properties, in reality is $c \geqslant 0$ which means that selecting very close forwarding neighbor may not be the best choice when energy is the criterion. An edge AB is in the enclosure graph if and only if direct transmission between A and B is power optimal solution for given set of nodes. That is, $u(|AB|) \leqslant u(|AC|) + u(|CB|)$ for any common neighbor C of A and B. The enclosure graph is unidirectional. In case $\alpha = 2$ and $c = 0$, the enclosure graph becomes equivalent to GG. Wang et al. [17], used Local Shortest Path Tree (LSPT) in a topology control protocol. In it, each node u applies Dijkstra's algorithm to find shortest weighted (weight of an edge of length r is $r^\alpha + c$) path to each of its neighboring nodes, using only its local 1-hop information (the concept can be extended to the k-hop knowledge). Node u then keeps only outgoing edges in this structure (towards neighbors with direct link requiring less power than the sum of power on any path between them), and removes the others. The result is sent to its neighbors. Each node then removes unidirectional links and adjusts its transmission radius according to the remaining logical links. The experimental data show that this tree has average degree about 2.4. LSPT is a connected localized structure. We shall see that using those underlying energy-efficient subgraphs, our proposed scheme obtains a very good performance.

3. Physical model

3.1. Network model

We represent a SANET as an undirected graph $G = (V, E)$ where V is the set of vertices and E is the set of edges. We assume that every node, represented by a vertex $v \in V$, is embedded in the plane, i.e., there are no great differences in height between nodes. Each node $v \in V$ has a maximum transmission range r that can be considered, without losing generality, the same for all nodes. Let $dist(v_1, v_2)$ be the Euclidean distance between two vertices $v_1, v_2 \in V$. An edge between two nodes $v_1, v_2 \in V$ exists $\iff dist(v_1, v_2) \leqslant r$ (i.e., v_1 and v_2 are able to communicate directly).

This model, known as unit disk graph (UDG), is a generally accepted approximation to make the problem tractable. As we will mention latter in Section 6, working with more realistic models considering lossy links between nodes is something we plan to work on for future works.

Definition 1. Complete neighborhood. Given the previous model we define $N(n)$, $n \in V$, the complete neighborhood of a node n, as the set of nodes directly reachable by n.

$N(n)$ is, by its definition, the set of all the nodes that n can reach setting its transmission power to the maximum. We assume that nodes have the capacity of adjusting its transmission power. So, we can define the reduced neighborhood of a node n for the distance d as follows.

Definition 2. Reduced neighborhood. Given a node n and a distance $d \geqslant 0$, we define the reduced neighborhood of node n at distance $d < r$ (denoted by $N(n, d)$) as the subset of $\{n_i \in N(n) | dist(n, n_i) \leqslant d\}$.

It is clear that $|N(n, d)| \leqslant |N(n)|$ and that reaching the reduced neighborhood will always imply using a lower energy than reaching the complete neighborhood.

3.2. Energy model

There are different energy models used to estimate the energy a node n needs to send a message far enough to reach a specific neighbor placed at distance d. In the most commonly used model, the energy consumption for transmitting a fixed size message at distance d is:

$$E(n, d) = d^\alpha + C_e$$

where α is the media attenuation factor satisfying $2 \leqslant \alpha \leqslant 4$ and C_e is a constant representing the power used to process the radio signal. To consider the energy consumed by every node receiving the message, we can extend the model adding a new constant called C_r. Every node $u \in N(n, d)$ will receive the message sent by n at distance d. Although only some of them will be responsible of acting as relay nodes, all of them must read the header of the message to check that fact. C_r represents the energy consumed by this checking phase. We consider realistic enough the values of $\alpha = 4$ and $C_e = 10^8$ derived in the work performed by Rodoplu and Meng in [4]. The value of C_r is generally accepted to be a third of the energy consumed by a emission at the maximum power. But, the model is general enough to accept any other value depending on the concrete technology being used.

$$C_r = \frac{r^\alpha + C_e}{3}$$

So, taking into account C_r, the extended energy model can be defined as:

$$E(n, d) = d^\alpha + C_e + |N(n, d)| \times C_r$$

Some authors do not consider necessary the use of C_r whereas some other do, for instance in IEEE 802.15.4, the cost of listening the channel is nearly the same as receiving messages. In our work we have considered both approaches. All our simulations are made for both variants $C_r = 0$ and $C_r \neq 0$, and the proposed protocols are able to operate effectively regardless of the energy model in use.

3.3. Wireless broadcast advantage

Unlike in wired networks, communications in wireless networks such as SANETs where nodes usually have an omni-directional antenna are broadcast in nature. Delivering a message to a set of relay nodes can be done with a single message transmission. This message has to be sent with enough power to reach the most distant of the selected relay nodes. Using that power, not only the desired node but all the nodes included in the reduced neighborhood for such distance are reached.

First advantage is on the data forwarding itself. To reach several relay nodes we can issue a single transmission, saving thus bandwidth and energy. Another benefit of the broadcast nature of the wireless medium happens when a node routing a message has a neighbor which is also one of the destinations. In this case, if any of the relays nodes is further away from the current node than such neighbor, the same transmission can deliver the message to it, and selected relay nodes to reach remaining destinations.

The second advantage is that nodes have different alternatives to route a message. Fig. 2 shows the case in which node n wanted to reach nodes u_1 and u_2. It can set up its transmission power to reach u_2 directly. The other alternative is using lower energy to reach only u_1, leaving it the task of reaching u_2 with a second transmission at distance d_2. Due to the relation between distance and energy consumption there are situations in which issuing more transmissions saves energy:

$$E(n, d_3) \geqslant E(n, d_1) + E(u_1, d_2)$$

By designing a protocol which takes into account this effect, the total amount of energy needed to route messages from sources to destinations can be reduced.

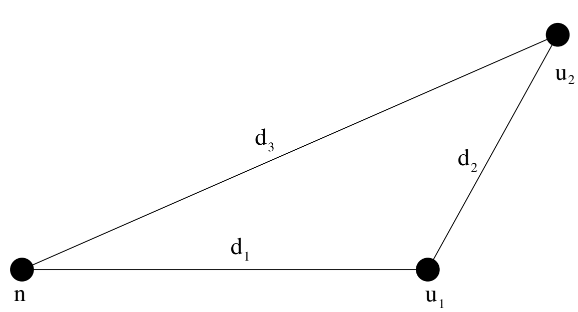


Fig. 2. Broadcast advantage.

4. GMREE: energy-efficient geographic multicast routing

4.1. Introduction

The goal of the protocol is to build multicast trees minimizing the criteria of the total energy used to deliver data packets. Our protocol is based solely on local information. Being more specific, each node uses the position of its neighbors and the position of destinations to decide next hops toward destinations. We assume that nodes know this information. The way in which this can be achieved is out of the scope of this work. Previous works such as [3] worked out elegant solution for these issues using static high-power directional antennas, and other works have worked out even solutions to work with virtual geographic coordinates [13].

Our protocol is based on the concept of cost over progress [20] as a metric of the goodness of the different neighbors as candidate relays. As cost we consider the energy needed to reach the furthest neighbor in the selected set of relays plus the energy that such amount of nodes will need to process the message. As progress we consider the summary difference between the distance to destinations from the node taken the routing decision and the distance from the set of relays being evaluated. The use of such a ratio lets us achieve a good trade-off between maximizing the advance towards the destinations but maintaining the energy consumption at low.

Like almost every geographic routing algorithm, GMREE selects as forwarding neighbors of a node those which are closer to destinations than itself. When this is not possible (i.e., a node has no such neighbors) GMREE uses a recovery mode called face routing that we will explain later in detail.

In addition, we have also considered the use of localized and energy-efficient underlying subgraphs on which GMREE is applied. It is desirable that the selected nodes to take part in this new Reduced Neighborhood are those taking part of paths with minimum energy consumption. Therefore, using these nodes as relay we are routing using the most promising paths. We have considered three different underlying subgraphs: Gabriel Graph, Enclosure Graph, and Local Shortest Path Tree. As we will explain later on, each approach has its theoretical advantage. Section 5 will show how the best one is the one which uses Local Shortest Path Tree.

Regardless of the set of neighbors a node has, the main difference between routing in unicast and multicast is the neighbor selection function. In the first case this function must return a single node, the one selected as a relay. There are at most n candidates being n the number of neighbors. In multicast scenarios it has to be decided which neighbor will be relay for which destinations. A neighbor can be responsible for one, more than one or none of the destinations. So the problem is which neighbors to select and which destinations assign to each neighbor selected. The number of the neighbors selected will also be of vital importance because each relay selected will start a new and separated path toward destinations that the node is responsible for.

Therefore, there are as many options as subsets of neighbors, i.e., $2^{|N|}$. Choosing too many neighbors means that in the next step, there will be too many routes toward only a few destination as in unicast routing, and consequently the consumption of energy will be high because the protocol will not be exploiting multicast operation. Choosing less relays implies that some of them will be in charge of routing toward more than one destination. This is usually a better decision from the point of view of saving energy because several messages are sharing the same route, but can lead to a common longer path so, again, the waste of energy could be significant.

The problem now is how to select the subset of relays. Let *selected ratio* be the proportion of selected nodes, the fewer the selected ratio, the greater the distance to destinations because we might have left out the ones whose progress toward destinations is higher. But at the same time we are consuming a lower energy in the next step (higher path reuse). This is where the goodness of the relation between cost and progress function will be a key aspect.

In the next sections we will explain in detail the different components of our algorithm and its variants. First of all, we will see how the cost over progress metric can be adapted to be used in a multicast scenario and with an energy metric. Next we explain the tree different approaches we have followed to obtain a reduced subset of neighbors. Each reduced neighborhood has advantages and disadvantages over the rest. Finally, we will explain the detailed neighbor selection function and explain the different operation modes of the protocol.

4.2. Cost over progress metric

We adapt cost over progress ratio paradigm to our energy requirements. The cost has to reflect the energy needed to reach relay nodes. Following the energy model proposed in Section 3, and being $d = \overline{CA}$ the distance between C and A, the cost of C selecting node A as relay is

$$E(C,d) = d^{\alpha} + C_e + |N(C,d)| \times C_r$$

Considering the multicasting problem, where a source node wishes to send a packet to a number of destinations (sinks) with known positions. Assume that a node C, after receiving a multicast message is responsible for destinations D_1, $D_2,\ldots,D_5$, and that it evaluates neighbors A_1, A_2 as possible candidates for forwarding. The whole task could be sent to one neighbor only (e.g., if there exist one that is closer to all destinations than C), or could be split to several neighbors, each with a subset of destinations to handle.

Consider the case in Fig. 3 as illustration of the general principle. The current total distance for multicasting is $T_1 = |\overline{CD_1}| + |\overline{CD_2}| + |\overline{CD_3}| + |\overline{CD_4}| + |\overline{CD_5}|$. Let $F_i = \{A_1, A_2\}$ be the forwarding set of neighbors currently being considered to cover D_1, D_2, D_3, and D_4, D_5, respectively, the new total distance is $T_2 = |\overline{A_1D_1}| + |\overline{A_1D_2}| + |\overline{A_1D_3}| + |\overline{A_2D_4}| + |\overline{A_2D_5}|$, and the progress made is $T_1 - T_2$.

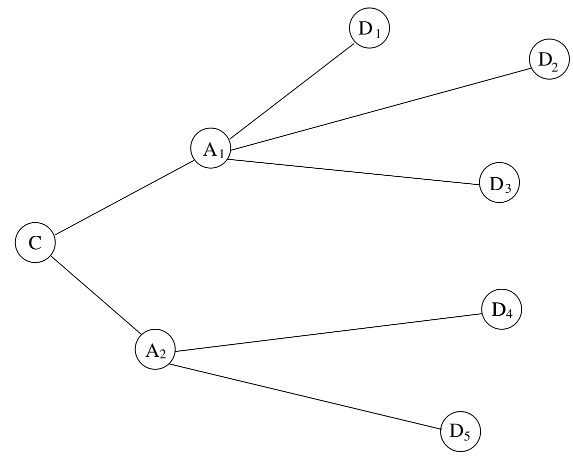


Fig. 3. Evaluating the candidate forwarding from C to A_1 and A_2.

The cost of forwarding packet through F_i must include not only the energy to reach the most distant selected neighbor A but also the cost associated to the fact that the selected neighbors start $|F_i|$ new separated paths towards the destinations covered by F_i. The cost metric should also include a new term $|F_i|C_e$, because each of the forwarders, with its retransmission, will require at least energy C_e.

Thus, each forwarding set F_i, $i = 1,\ldots,2^N$ is evaluated as $\frac{E(C,d)+|F_i|C_e}{(T_1-T_2)}$, being $d = \mathrm{Max}(\overline{CA_i})|A_i \in F_i$. Among all candidate forwarding sets, the one with minimal value of this expression is selected. If there is no neighbor closer than C towards one or more of the destinations, then we have to enter into face mode. Section 4.4.2 describes how to manage this situation.

4.3. Underlying energy-efficient subgraphs

One of the most important aspects of our algorithm is the neighbor selection function. To improve such selection, we use underlying locally computed energy-efficient graph reductions. The general idea is to preserve only a subset of the edges, which provide low energy paths to the rest of neighbors. Taking into account the wireless advantage we can leave out of the selection process those nodes that can be reached with a lower energy consumption through some other nodes. Fig. 4 describe the different graph reductions we use.

From the point of view of energy conservation, Gabriel Graph (GG) links may belong to suboptimal paths between a node and its neighbors. This is especially the case in dense networks, when $C_e > 0$. In general, longer links, not belonging to GG, may be needed to produce energy optimal paths. However, GG links are generally short and energy efficiency may be expected.

Another interesting underlying structure is the enclosure graph [4]. It only contains a link between two neighbors, if

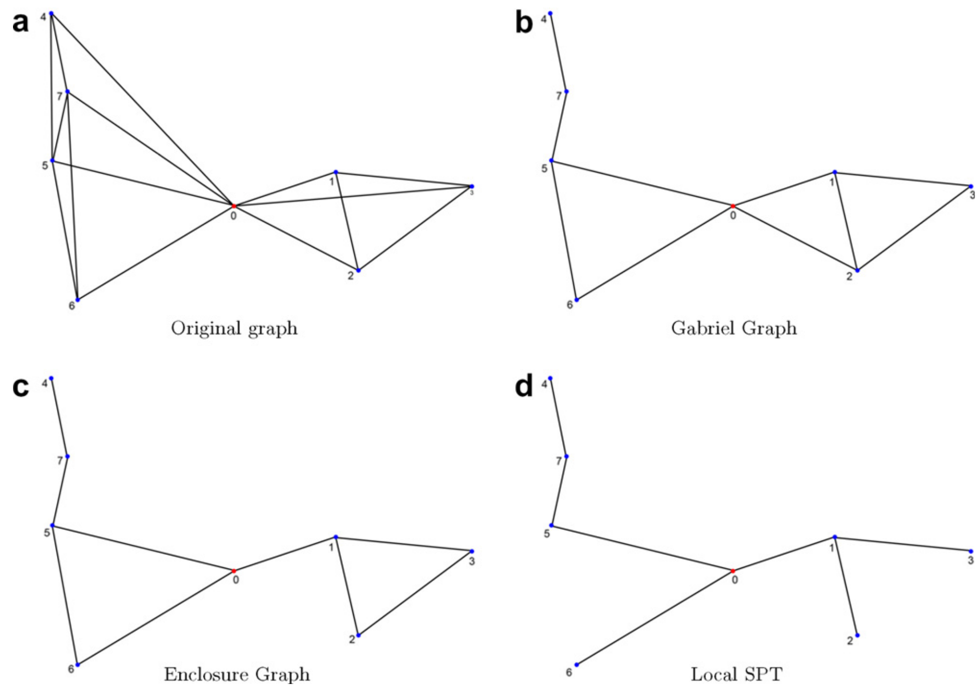


Fig. 4. Different graph reduction for energy saving.

there is no other common neighbor that reduces the overall energy by acting as a relay to connect them. It is possible to build this graph locally. Here we can say this is a optimal graph because as the authors show in his work, this graph include all the energy optimal path between every two nodes in the network. But, of course, it is the duty of multicasting protocols to find them. By using only those remaining paths, greedy multicasting protocols can get better energy consumptions.

Finally, the other interesting alternative we tested is using the well-known Dijkstra algorithm locally to find the shortest paths in terms of energy between a node and all its neighbors. A node selects only those links going out of the current node, and sends that selection to neighbors. In that way, they eliminate unidirectional links, and remaining links form the Local Shortest Path Tree (LSPT). Again, multicasting considering only these links can help at reducing energy consumption.

4.4. GMREE operation

Let's now see how the algorithm works. GMREE has two different modes of operation: greedy and face. Greedy multicasting can be applied to those destinations for which there are neighbors of the current node providing advance toward them. Nodes start face mode whenever they have no neighbors providing advance towards one or more destinations.

Messages must include some information indicating not only the destinations but also if the message is being routed in greedy or face mode. As explained in [5], the minimum information required includes the node in which face mode started called L_p, the first edge of the current face, and L_f the cross point with the line from the node where face started to the destination. On the other hand, messages include a list of relay nodes, the destinations for which each relay is in charge and the mode the multicasting is being done (greedy or face). In this way we are taking advantage of the broadcast nature of the wireless links used by the nodes. Each node sends only one message after its routing algorithm has been executed. This message reaches only the nodes placed at the distance selected by the sender, it is, the distance to the most distant node selected as a relay. When neighbors receive the message, they check whether they are included in the relay list or not. If they are not, they just drops the packet. Otherwise, they repeat the process by running the routing algorithm to select their relay nodes.

Finally, when a node is forwarding the message to its selected relays, it adjust the transmission power to reach the most distant one, so that all of them can receive the message.

4.4.1. Greedy multicasting

A node executes the greedy multicasting algorithm for the Greedy Destination List (GDL). This list contains destinations for which there is at least one neighbor in the node's neighbor list providing advance towards them. The Greedy Neighbor Selection function is then executed and the result is a list of relays nodes and its associated destination lists. This function uses as neighbors those belonging to the concrete Reduced Neighborhood that it is being used.

The node examines the list of relays and includes an entry for each one in the new message. Notice that each relay can be in charge for several destinations. When a destination is included in the neighborhood (reduced or not) the message includes an entry for it without destination list.

When a node receives a message and it is one of the relays included in the header it realizes that it has to execute the routing algorithm to determine next hops for the message to be routed towards the list of associated destinations. If the destination list is empty, this means that current node is the destination of the message so it only has to pass the message to the internal module in charge of processing it. After examining the destination list and creating the GDL, it might happen that some destinations are left out of it, these ones are included in the face destination list. Next subsection explains how to route towards those destinations for which face mode is required.

4.4.2. Multicasting in face mode

Multicast in greedy mode can end up reaching a void zone with no neighbor providing progress towards one or more destinations. The list containing these destinations for which no progress can be achieved is called Face Destination List (FDL). This issue is solved for the unicast case as explained in [5], but there is not any multicast specific approach to face routing. The alternative we use is to use the well-known face routing individually for each destination requiring face routing. We are working in a new multicast specific face routing algorithm for future work.

Routing algorithm in face mode acts as follows. For each destination in the FDL it executes face routing algorithm determining the next hop. The message to broadcast will include an entry for each destination in FDL. It is possible that different destinations share the same next hop, that is, the same relay node. However, we cannot merge those destinations in the same entry, as we do with greedy destinations, because each one will probably follow a different face path and must be processed independently. Each of these entries in the message have the routing mode set to "Face Mode".

When a node receives a message and it is the relay node for some of the entries in "Face Mode" it first checks whether itself or some of its neighbors is closer to the destinations than the node where face started. In this case, those destinations are included in the list of greedy destinations to be processed in that way. Destinations for with it is necessary to continue in face mode are processed using face routing algorithm and included in the resulting message but with next hops updated accordingly to the result of face routing algorithm.

Notice that face routing algorithm always uses Gabriel Graph as underlying graph. So, its behavior is independent of the energy-efficient underlying neighbor reduction being applied.

5. Greedy neighbor selection

In the previous section we explained how GMREE works, and gave detailed explanations of the different aspects of its operation. However, we did not explain into detail how is the neighbor selection algorithm implemented. That is, what is the concrete algorithm used by the current node to decide which subset of its neighbors forwards multicast data messages for which subset of destinations. In this section we describe that part of the protocol, showing the benefit in terms of computational cost compared to previous works.

Given k destinations, a possible algorithm can consider all S_k partitions of the set of destinations. For each subset in a given set partition, the node checks whether it is possible to find a neighbor that is closer to all destinations in that subset than the current node C. If this is not possible for a subset, then this partition is ignored. If this is possible for each subset in the given set partition, then we measure the cost/progress ratio. Finally, after all the evaluations, we choose the best one among all measured ones. A fast algorithm to compute those set partitions is given in [12]. This solution is only applicable for small number of destinations, e.g., up to 5. For larger number it becomes exponential in k, and therefore a faster greedy solution is needed.

We start with the set of destinations $\{D_1, D_2, \ldots, D_k\}$ for which there is a neighbor of the current node providing advance. We first group together into the same subset, those destinations for which the neighbor providing the largest advance is the same. For instance, in Fig. 3, the initial set partition to consider would be $\{\{D_1, D_2, D_3\}, \{D_4, D_5\}\}$, where A_1 serves D_1, D_2, D_3 and A_2 serves D_4 and D_5. In general, the set partition could be $\{M_1, M_2, \ldots, M_m\}$ with each destination being in exactly one of these subsets and node C can have n neighbors. Each M_i has its own cost-progress ratio, and the whole set partition also has its own cost-progress ratio as we explained before. The cost-progress ratio for the subset $\{D_1, D_2, D_3\}$ in Fig. 3 can be computed as follows: The current total distance for multicasting is $T_1 = |\overline{CD_1}| + |\overline{CD_2}| + |\overline{CD_3}|$. The new total distance is $T_2 = |\overline{A_1D_1}| + |\overline{A_1D_2}| + |\overline{A_1D_3}|$, and the progress made is $T_1 - T_2$. The cost is the energy needed to reach A_1. Thus, the forwarding cost-progress is evaluated as $\frac{E(C,|\overline{CA_1}|)+C_e}{(T_1-T_2)}$. This reflects the fact that after receiving the message, A_1 will process it and then forward it.

Let P_i be the progress in M_i (each P_i is $T_1 - T_2$ from above explanation). The cost in each M_i is $E(C, |\overline{CA_i})+C_e$ since all destinations in a given subset are served by the same (one) neighbor and taking into account the Wireless Broadcast Advantage, one message sent to distance d will reach every node whose distance to the sender is lower than d. Therefore, the overall cost-progress ratio for the selected set partition is then $\frac{E(C,d)+mC_e}{\sum_{i=1}^{m} P_i}$ being $d = \max dist(C, A_i), i \in 1, \ldots, n$. We are looking for the set partition with optimal ratio.

The greedy set partition selection multicast algorithm presented in Algorithm 1 works as follows. First, the initial partition set $M = \{M_1, M_2, \ldots, M_m\}$ is initialized as we explained before. The current cost-progress ratio is computed accordingly as $\frac{E(C,d)+mC_e}{\sum_{i=1}^{m} P_i}$. Then the algorithm computes for each of the partitions in M, which are the ones that can be combined and provide the higher improvement in the overall cost-progress ratio. Two partitions can be combined only if there are neighbors of the current node, providing advance towards all the destinations in both partitions M_i and M_j. The process is repeated until no more benefit is obtained by further merging. We explain below how are two partitions merged.

Algorithm 1. Greedy set partition selection algorithm

$M = \{M_1, M_2, \ldots, M_m\}$ so that $M_i = \{D_i |$ same neighbor provides most advance$\}$
BestDistance = max $dist(C, A_i)$, $i \in 1, \ldots, m$
$CRatio = \frac{E(C,BestDistance)+mC_e}{\sum_{i=1}^{m} P_i}$
while CRatio improved **do**
  repeat
    Find cost reduction by merging of $\{M_i, M_j\} \in M$
    if reduction < BestReduction **then**
      BestReduction = reduction
      BestMerge = $\{M_i, M_j\}$
      BestDistance = max $dist(C, A_i)$, $i \in 1, \ldots, m$
    end if
  until All pairs tested
  $M = \{M_1, M_2, \ldots, \{M_i, M_j\}, \ldots, M_{m-1}\}$
  $CRatio = \frac{E(C,BestDistance)+|M|C_r}{\sum_{i=1}^{|M|} P_i}$
end while

In order to merge two subsets M_i and M_j as one single set $M_{i,j}$, the algorithm considers the set of neighbors that are closer to all destinations in $M_i \cup M_j$ than current node C. If that set is empty then merging is not possible. Otherwise, among all such neighbors A, we select the one which provides the best cost-progress ratio for this new subset $M_{i,j}$.

This algorithm, instead of testing all possible subsets, only needs to test $O(k^3)$ of them (in the worse case) being k the total number of destinations. In fact when the number of neighbors of the current node is lower than the number of destinations, there is no need to test more than n^3 subsets, being n the number of neighbors.

The reason why we merge subsets rather than using those nodes providing the biggest advance to a set of destinations, is because there can be nodes not providing the maximum advance, which may certainly reduce the cost of the multicast tree, which still providing an important progress.

To see that, let's assume that $M = M_1, M_2, \ldots, M_m$ is the initial set partition in which all destinations in every M_i are served by the same closest node (B_i) among all neighbors of the current node A. Let $N(A)$ be the set of A's neighbors. Given two subsets M_i, $M_j \in M$ we analyze the conditions under which some neighbor with a lower progress can still provide a better tradeoff.

Let $E(A, d_1) + mC_r$ be the cost of the initial partition, d_1 the distance from node A to the B_i which is furthest away, and $\sum_{k=1}^{m} P_k$ be the progress made with such election. If we merge M_i and M_j into a single set served by a single node $B_{i,j} \in N(A)$, $B_{i,j} \neq B_i$, $B_{i,j} \neq B_j$, the cost of the new subset $M_i \cup M_j$ is $E(A, \overline{AB_{ij}}) + C_e$. The overall cost after merging is then $E(A, d_2) + (m-1)C_e$ being d_2 the distance from node A to the furthest of $B_k \cup B_{i,j}$, $B_k \in N(A)$, $k \neq i$, $k \neq j$. In addition, the new progress made after merging would be $\sum_{k \neq i, k \neq j} P_k + P_{i,j}$, being $P_{i,j}$ the progress made by $B_{i,j}$ towards destinations in $M_{i,j}$.

For the new cost-progress to be better than before merging, the following inequality must be satisfied:

$$\frac{E(A, d_1) + mC_e}{\sum_{k=1}^{m} P_k} > \frac{E(A, d_2) + (m-1)C_e}{\sum_{k \neq i, k \neq j} P_k + P_{i,j}}$$

In Fig. 5, we can see that current node S and three neighbors n_1 and n_2 within its radio range. We also see destinations D_1, D_2. n_1 and n_2 are the neighbors providing more progress to D_1 and D_2, respectively. The shadowed zone, is the one in which there might exist other neighbors that can improve the cost-progress ratio, even though they do not provide the best advance for any of the individual destinations. The legend of the graph represents the improvement in cost/progress ratio over the configuration before merging. Of course, for different position of nodes, and destinations, a different improvement area is obtained.

Thus, we can see how merging subsets is really effective at reducing the overall energy consumption of the protocol. Next section analyses the performance of the different alternatives being evaluated.

6. Experimental results

We have performed a simulation-based analysis to assess the performance of the proposed schemes. Simulations have been performed with a custom-made java simulator for geographic routing. Each simulation has been performed considering the same area in which nodes are randomly deployed. To generate scenarios with different mean densities, we vary the number of nodes. Nine different densities, from 8 to 41 mean neighbors per node have been simulated. In addition we have also considered four different sets of destination nodes $(1, 5, 10, 30)$ for each density. We have considered two different energy models ($C_r = 0$ and $C_r \neq 0$) as explained before. For each density, number of destinations and energy model 50 different scenarios have been evaluated, giving a total of 2800 simulations per variant of each protocol.

Given that there is no other geographic energy-efficient multicast routing reported in the literature, we have considered a multi-unicast greedy based routing scheme for comparison. In addition, although not directly comparable, we also compare against two centralized schemes (energy-efficient shortest path tree and MIP) to assess how close is the performance of our protocol from a centralized scheme. Being more specific, the different protocols and variants evaluated are multi-unicast (MU), Energy-Efficient

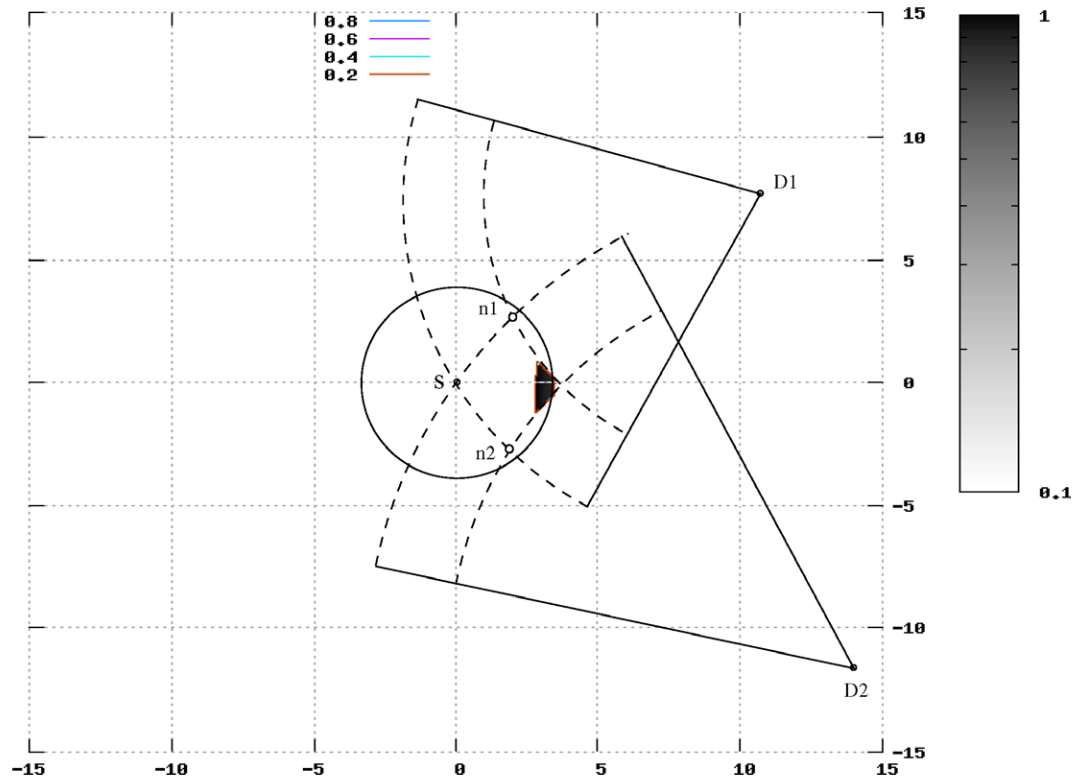


Fig. 5. Identification of nodes according to their goodness.

Shortest Path Tree (ESP), Minimum Incremental Power (MIP), Energy-efficient GMR over the Enclosure Graph (GMREE_EG), Energy-efficient GMR over the Gabriel Graph (GMREE_GG), and Energy-efficient GMR over the Local Shortest Path Tree (GMREE_SPT). By MU we understand the use of the IPowerProgress [15] algorithm for each destination independently. This is the easiest way to send data messages to multiple destinations without using flooding. The ESP is the centralized application of Dijkstra's shortest path tree, in which edge costs are the energy of traversing that edge.

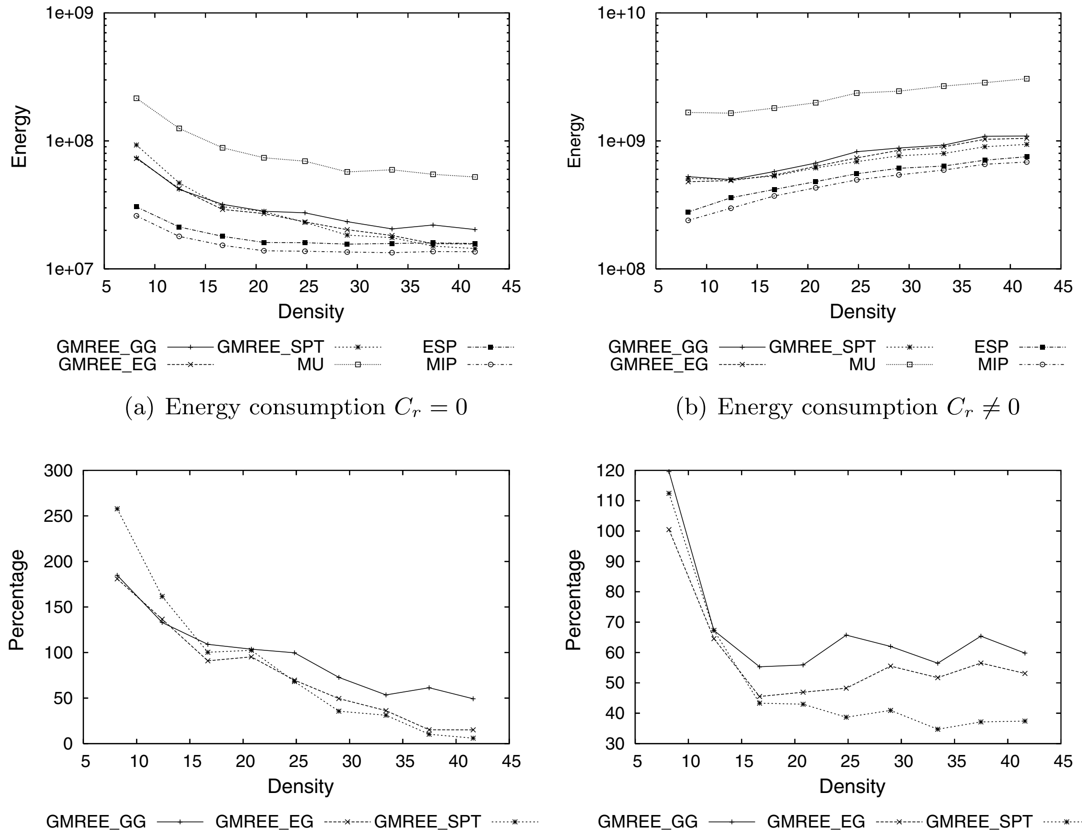
The main performance metric is the mean energy consumed and the energy consumption ratio compared to the (MIP) centralized tree based on energy. This ratio is computed based on the energy consumption of the different protocols averaged over the 50 different graphs for each value of density, number of destinations and energy model. The confidence interval for the 95% is small enough, indicating that the mean obtained with this amount of repetitions is highly representative.

In the next subsections we analyze the performance of each of these alternatives.

6.1. Energy consumption at increasing densities

In Fig. 6, we show the energy consumption of the different protocols as the mean density increases for the two energy models considered. As we can see in Figs. 6(a) and (b), there is a great difference between the two energy models tested. With $C_r = 0$ the density helps protocols to find out more energy-efficient paths. There are two main reasons. The first one is the fact that the higher the density, the lower the number of times that the protocol has to enter into face mode. Secondly, the higher the number of neighbors the easier is the task of finding out good relays for all the protocols. On the other hand, when $C_r \neq 0$, the higher the density the greater the energy consumption. The energy consumed by all the protocols increases linearly with the mean density. This increasing is slow but the cause is that with this energy model, the density does not help at reducing the energy consumption because the energy due to reception represents a significant amount of the total.

Regarding the performance across variants, in both cases $C_r = 0$ and $C_r \neq 0$ MU consumes, as expected, much more energy than our multicast approaches. The reason is that MU does not take any advantage of links which are common to multiple destinations. In Figs. 6(c) and (d), we can see the percentage of additional energy consumed for each of our variants compared to the centralized protocol MIP. When $C_r = 0$, we can see in Fig. 6(c), the higher the density, the closer become our alternatives to the centralized MIP. This is again due to the reduction in the number of times face is required, which means that multicast trees become less suboptimal. As we can see, as the density increases, the differences across variants become more evident. In fact, for densities higher than 25 we see how GMR_GG starts getting a worse performance than the other two alternatives. The reason is that the very short



(a) Energy consumption $C_r = 0$

(b) Energy consumption $C_r \neq 0$

(c) Energy consumption ratio $C_r = 0$

(d) Energy consumption ratio $C_r \neq 0$

Fig. 6. Energy consumption for increasing density and 30 receivers.

edges preserved by the Gabriel Graph, force the protocol to select paths consisting on many very short hops, which require at least C_e energy. The variant based on the enclosure graph (GMR_EE) considers energy consumption to eliminate those edges whose endpoint can be reached with a lower energy consumption by going though an intermediate node. Thus, it outperforms the GG variant. However, given that EE does not consider possible paths going through multiple intermediate nodes, it does not outperform the algorithm based on the local shortest path tree (GMR_SPT) scheme. In fact, we can see that this latter scheme outperforms ESP and it is very close to MIP for high densities.

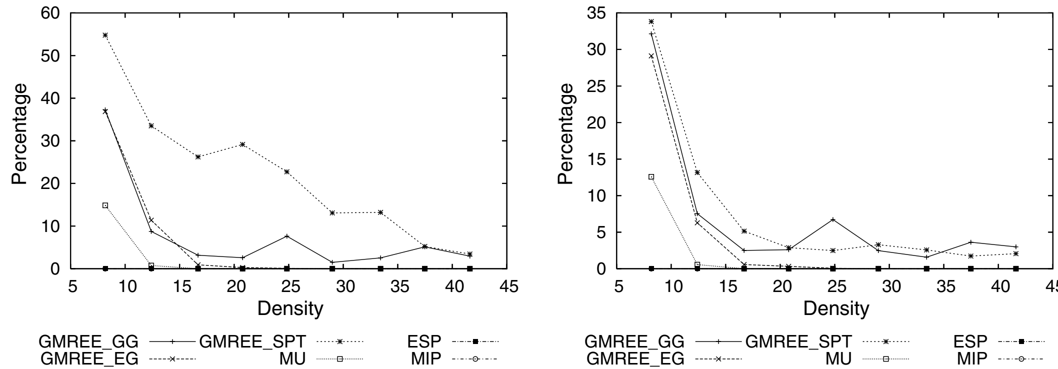
For the case in which $C_r \neq 0$ we can see in Fig. 6(d) very similar results. The higher the density the lower the difference against MIP and for mean densities over 12, GMR_SPT outperforms the other variants. As it can be seen, in this case, the energy ratio is better than in the previous one. The reason is explained in Fig. 7. For lower densities, the proportion of energy due to perimeter routing in the case when $C_r \neq 0$ is less than in the $C_r = 0$ case. Therefore, the difference against the centralized approach is lower, and the ratio better.

Fig. 7 shows the percentage of energy due to routing in perimeter mode of each algorithm tested. In both cases, $C_r = 0$ and $C_r \neq 0$ the percentage decreases as density increases. The high percentage of energy consumed due to perimeter routing when density is lower than 15 is the main reason of the bad behaviour of the protocols at these lower densities.

In the next subsection we analyze how does the protocol perform when a varying of receivers are considered.

6.2. Energy consumption at increasing number of receivers

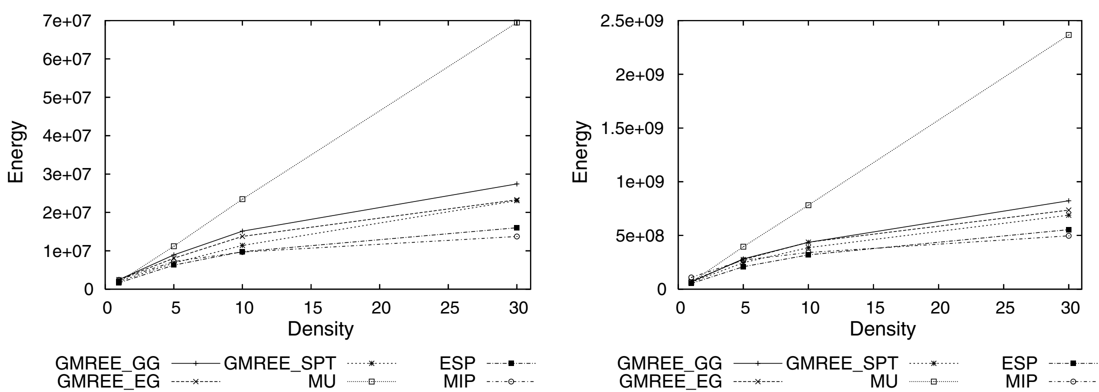
Fig. 8 shows the performance of the simulated schemes as the number of destinations increase. As expected, the higher the number of destinations the higher the overall energy consumption for all protocols. The obvious reason is that reaching more destinations often requires creating more paths, which is more expensive in terms of energy. Again, MU has the higher energy consumption with a linear increase as the number of receivers increases. This shows its limited scalability regarding the number of destinations. If we compare Figs. 8(a) and (b), we can see that the performance of our proposed schemes are much more scalable in terms of energy consumption to the increase of the number of receivers. In addition, we see that the energy model does not have a big influence on the energy consumption for an increasing number of receivers. Of course, the total energy consumption with the model in which $C_r \neq 0$ is higher because it also counts energy of receptions but the trend is similar for both models.



(a) Perimeter energy $C_r = 0$

(b) Perimeter energy $C_r \neq 0$

Fig. 7. Percentage of energy due to perimeter routing at density increases with 30 receivers.



(a) Energy consumption $C_r = 0$

(b) Energy consumption $C_r \neq 0$

Fig. 8. Energy consumption for increasing number of receivers for GMREE_SPT algorithm.

Regarding the comparison across protocol variants we can see that for the density shown in these graphs MIP offers the best performance. In addition, we can also see that regardless of the number of receivers, the performance advantage of GMREE_SPT over GMREE_GG and GMREE_EE still remains. The explanation for such differences is again the ability of the LSPT underlying graph to use the whole neighborhood to assess which local links are really in the shortest path tree based on energy. Thus, very short or very long links are avoided.

7. Conclusions and future work

We introduced GMREE, an energy-efficient and fully localized geographic multicast routing algorithm for Sensor and Actuator Networks. The protocol assumes that the position of destinations is known either in advance, or it is provided by some location service. GMREE uses a cost over progress based function to select the subset of neighbors needed to forward the message to multiple destinations saving as much energy and network bandwidth as possible. It uses only local information to select routes, making it able to scale to large network deployments. We evaluated several variants of GMREE based on different energy-efficient localized neighborhood reductions. We have shown that these different underlying subgraphs have an significant impact on the overall energy consumption of multicast trees computed. From the different subgraphs evaluated, (i.e., Gabriel Graph, Enclosure Graph, and Local Shortest Path Tree) simulations have shown that GMREE_SPT is the one getting the best results by leaving only those edges in the subgraph which are really energy efficient. This prevents the greedy neighbor selecting from selecting very promising neighbors which do not lead to overall energy efficient paths.

The performance of each variant of GMREE has been compared against two other algorithms: Multi Unicast Greedy Face Greedy (MU) based on the iPowerProgress metric to save energy, and the centralized Energy Shortest Path Tree based on Dijkstra's algorithm. The first one is the easiest solution to send messages to multiple destinations using geographic routing. However, its energy consumption is really high because it does not take advantage of the benefit of sharing links across multiple destinations. ESP is a centralized algorithm that builds multicast trees merging the energy shortest paths calculated using Dijkstra algorithm. Simulations show, that GMREE outperforms MU for all underlying subgraphs, and achieves a performance similar to ESP for network densities in which the impact of face routing is negligible.

For future work we are studying mechanisms to reduce energy consumption for face routing, as well as to be able achieve energy and bandwidth efficiency by sharing paths created in face mode between multiple destinations for which face routing is needed.

Acknowledgements

This work was supported in part by the Spanish MEC by means of the "Ramon y Cajal" Research Program, in part by the SMART TIN2005-07705-C02-02 Project, and in part by the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

[1] Joseph Polastre, Robert Szewczyk, David Culler, Telos: enabling ultra-low power wireless research, in: Proceedings of the Fourth International Conference on Information Processing in Sensor Networks, April 2005.

[2] Juan A. Sanchez, Pedro M. Ruiz, Ivan Stojmenovic, GMR: geographic multicast routing for wireless sensor networks, in: Proceedings of the third Sensor and Ad Hoc Communications and Networks (SECON '06). Reston, VA, USA, Sept. 2006, pp. 20–29.

[3] Loukas Lazos, Radha Poovendran, SeRLoc: secure range-independent localization for wireless sensor networks, in: ACM Workshop on Wireless Security (ACM WiSe 2004), Philadelphia, PA, 2004.

[4] V. Rodoplu, T.H. Meng, Minimum energy mobile wireless networks, IEEE J. Select. Areas Commun. 17 (8) (1999) 1333–1344.

[5] P. Bose, P. Morin, I. Stojmenovic, J. Urrutia, Routing with guaranteed delivery in ad hoc wireless networks, ACM Wireless Netw. 7 (6) (2001) 609–616.

[7] Jeffrey E. Wieselthier, Gam D. Nguyen, Anthony Ephremides, Energy-efficient broadcast and multicast trees in wireless networks, Mobile Netw. Appl. 7 (2002) 481–492.

[8] S. Guo, O. Yang, Localized operations for distributed minimum energy multicast algorithm in mobile ad hoc networks, IEEE Trans. Parallel Distr. Syst. 18 (2) (2007) 186–198.

[9] S. Giordano, I. Stojmenovic, Position based routing in ad hoc networks, a taxonomy, in: X. Cheng, X. Huang, D.Z. Du (Eds.), Ad Hoc Wireless Networking, Kluwer, Dordrecht, 2003.

[10] H. Takagi, L. Kleinrock, Optimal transmission ranges for randomly distributed packet radio terminals, IEEE Trans. Commun. 32 (3) (1984) 246–257.

[11] M. Mauve, H. Füßler, J. Widmer, T. Lang, Position-based Multicast Routing for Mobile Ad-Hoc Networks, TR-03-004, Department of Computer Science, University of Mannheim, March 2003.

[12] B. Djokic, M. Miyakawa, S. Sekiguchi, I. Semba, I. Stojmenovic, A fast iterative algorithm for generating set partitions, Comput. J. 32 (3) (1989) 281–282.

[13] A. Caruso, A. Urpi, S. Chessa, S. De, GPS-free coordinate assignment and routing in wireless sensor networks, in: Proceedings of IEEE Infocom 2005, vol. 1, March 2005, Miami, USA, 2005, pp.150–160.

[14] F. Ingelrest, D. Simplot-Ryl, I. Stojmenovic, Optimal transmission radius for energy efficient broadcasting protocol in ad hoc and sensor networks, IEEE Trans. Parallel Distr. Syst. 17 (2) (2006) 536–547.

[15] Ivan Stojmenovic, Xu Lin, Power aware localized routing in wireless networks, IEEE Trans. Parallel Distr. Syst. 12 (11) (2001) 1122–1133.

[16] Matthias Transier, Holger Füßler, Jörg Widmer, Martin Mauve, Wolfgang Effelsberg, Scalable position-based multicast for mobile ad-hoc networks, in: First International Workshop on Broadband Wireless Multimedia: Algorithms, Architectures and Applications (BroadWim 2004), San Jose, CA, USA, 2004.

[17] S.-C. Wang, D.S.L. Wei, S.-Y. Kuo, SPT-based topology algorithm for constructing power efficient wireless ad hoc networks, ACM International World Wide Web Conference, 2004.

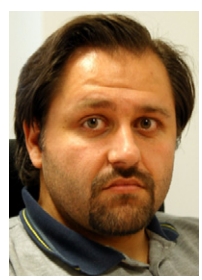
[18] Hyuntae Cho, Yunju Baek, Location-based routing protocol for energy efficiency in wireless sensor networks, in: Lecture Notes in Computer Science, vol. 3823, 2005, pp. 622–631.

[19] K.R. Gabriel, R.R. Sokal, A new statistical approach to geographic variation analysis, Syst. Zool. 18 (1969) 259–278.

[20] I. Stojmenovic, Localized network layer protocols in sensor networks based on optimizing cost over progress ratio, IEEE Netw. 20 (1) (2006) 21–27.

[22] J. Hou, N. Li, I. Stojmenovic, Topology construction and maintenance in wireless sensor networks, in: I. Stojmenovic (Ed.), Handbook of Sensor Networks: Algorithms and Architectures, Wiley, New York, 2005, pp. 311–341.

Juan A. Sanchez received his B.Sc. (1999), M.Sc. (2004), and Ph.D. (2006) degree in Computer Science from the University of Murcia, Spain. For over four years he was involved in several national and international R&D projects in the field of wireless and mobile networks, multimedia applications and next generation networks as member of the R&D Department of Agora Systems S.A. Currently, he is Adjunct Professor at the Department of Communications and Information Engineering (DIIC), at the University of Murcia (UMU).



Pedro M. Ruiz received his B.Sc. (1999), M.Sc. (2001), and Ph.D. (2002) degrees in Computer Science from the University of Murcia, Spain. He is a *Ramon y Cajal* Researcher in the Department of Information and Communication Engineering (DIIC) at the University of Murcia (UMU), and he has also held Post-doctoral research positions at ICSI in Berkeley, King's College London and University of California at Santa Cruz. During these years he has acted as Principal Investigator in several projects mainly funded by the European Union, Spanish government and private companies, and has published over 60 refereed papers in international journals and conferences. **Dr. Ruiz** is in the editorial board for the International Journal on Parallel, Emergent, and Distributed Systems, he participates in the Technical Committee of several conferences, and he has served as a reviewer for major IEEE journals and conferences. His main research interests include sensor networks, mobile and ad hoc wireless networks and distributed systems. He is a member of the IEEE Communications Society.

Ivan Stojmenovic received Ph.D. in mathematics from University of Zagreb, Croatia, 1985. He held positions in Serbia, Japan, USA, Canada, France, and Mexico. He published over 200 different papers, and edited four books on wireless, ad hoc and sensor networks and applied algorithms with Wiley/IEEE. He is currently editor of several journals including IEEE TPDS, and founder and editor-in-chief of three journals. Stojmenovic is in the top 0.56% most cited authors in Computer Science (Citeseer 2006). One of his articles was recognized as the **Fast Breaking Paper**, for October 2003 (as the only one for all of computer science), by *Thomson ISI Essential Science Indicators*.