LEMA: Localized Energy-Efficient Multicast Algorithm based on Geographic Routing

Juan A. Sanchez, Pedro M. Ruiz DIIC, University of Murcia, {jlaguna,pedrom}@dif.um.es

Abstract

Most usage scenarios for ad hoc and sensor networks require some deegree of one-to-many or many-to-many interactions. In particular, for the case of sensor networks there is a number of scenarios in which a node has to send the same data to multiple destinations. Given that sensor networks have very limited resources, multicasting is a very interesting approach to deliver the same data packet to multiple destinations while reducing the amount of bandwidth and power consumption. In this paper, we present LEMA (Localized Energy-Efficient Multicast Algorithm), a new energy-efficient multicast routing protocol based on geographic routing. The protocol is localized, as long as a node currently routing a multicast data packet is only required to know the coordinates of the destinations (locally obtained from the data packet to route) and the positions of its next hops. The protocol uses a localized source routing scheme to reach next hops which, unlike traditional geographic routing schemes, allows for the use of neighbors not providing advance to reduce the overall energy consumption. Our simulation results show that for networks with enough density the protocol is able to outperform even well-known centralized heuristics such as Minimum Incremental Power (MIP) as well as Shortest Path Tree (SPT) based on energy. For low-density networks, the protocol is still competitive for most scenarios.

1 Introduction and Related Work

There is a growing number of applications being developed to take advantage of the properties of Wireless Sensor Networks (WSN). They are considered as one of the most promissing network technologies, both in academia and industry. On of the typical usage scenarios is monitoring of environmental parameters in physically complex or dangerous areas. The new applications designed to work over WSN usually require of group-based communications. Changing the behavior of some sensor of the network or sending commands to a group of nodes are examples of one-to-many communications. Multicast is a network primitive specifically designed to provide that kind of communications to multiple destinations.

One the other hand, WSN are made upon hundreds (probably thousands) of sensor nodes. Sensors are tiny devices that sense their environment and communicate among them using an on chip integrated radio interface. They are usually operated by batteries. Thus, their energy and computational power is limited. For that reason, the software designed to be part of a WSN must take care of saving as much resources as possible.

The problem of building a minimal multicast tree or Steiner Minimum Tree (SMT) has been prooben to be NP-complete even when every link has the same cost. Karp demonstrated it in [10] by a transformation from the exact cover by 3-sets.

Therefore, solutions to multicast routing in the literature have used more computationally simple trees such as Shortest Path Trees (SPT) or prunned Minimum Spanning Trees. Other authors have used heuristics algorithms. For instance, the MST heuristic [11][12] provides a 2-approximation of the SMT.

There has been a big amount of work in the field of energy-efficient multicast routing algorithms for wireless networks. A famous example of centralized algorithms is the Minimum Incremental Power (MIP) [5]. Some others algorithms are designed to be decentralized. For example, G-REMiT [13] makes use of the Core Based Tree (CBT) algorithm [6]. Unfortunetely, even the adaptations of CBT to wireless networks described in [7] are unable to be deployed in a WSN. Those algorithms cannot be applied to WSN due to their centralized architecture or because they are solutions oriented to build multicast shared trees that relay on the concept of Rendezvous Points (RP). Each sender registers itself in the RP and consecuently it is not adequate to be used in a network where nodes cannot be considered stable enough to be a RP. The reason is that, usually, nodes in a WSN operate in duty-cycle, changing from awake to sleep periods to reduce energy consumption. For that reason we center our study in Geographic routing protocols, which are most suited for WSN.

Geographic routing was first introduced by Finn in [1] is a decentralized routing technique in which nodes select their next hops solely based on the positions of destinations and their own. The main goal is to make progress, i.e., forwarding the message to a neighbour being closer to the destination that the current one. That greedy routing guarantees the protocol to be loop-free. The problem arises when there is no neighbour providing advance towards the destination. These situations, called local optimum, appear when there are areas without nodes in the middle of more dense areas. These holes can be avoided using what the authors of [2] and [3] call perimeter routing. Routing in that mode allows surrounding the void area to eventually reach a node in where greedy routing can be resumed. The problem of perimeter routing is that the selected path can be arbitrary long and evidently energy inefficient.

However, geographic routing is a routing technique which is well-suited for WSN. It is state-free because nodes do not need to maintain complex routing tables and is flexible enough to adapt of the non-static topology of WSN. Position Based Multicast [8] and Scalable Position-Based Multicast [9] algorithms are two examples of multicast geographic routing protocols. The first one is an adaptation to multicast of the Greedy Perimeter Stateless Routing (GPSR) protocol which tries to find a trade-off between path length and total number of messages by configuring a parameter. The problem of this algorithm is that in each step (each node routing a message), all possible subsets of neighbors are computed to select the best one. These neighbors are called forwarding nodes, and, for each destination, one of them takes responsibility for forwarding. Given n neighbors, the number of possible subsets is $\sum_{k=0}^{n}\binom{n}{k} = 2^n$. This leads to an exponential time complexity, too high, to be feasible in a sensor node. On the other hand, SPBM is more a group management protocol than a multicast routing protocol and it makes use of multiunicast to deliver messages. Unfortunately, neither of them are designed to build energy efficient multicast trees.

In this paper we have developed and tested a new geographic multicast routing for building energy efficient multicast tree in WSN. The basic idea is to extend GPSR as PBM does but our contribution is twofold. First of all, each node routing a message to a set of destinations, locally applies Kruskal's MST algoritm with edge cost based on distance between that subset of nodes. This allows the node to get an idea of the shape of the part of the energy minimal tree starting on them. Thus, nodes are able to decide wether it is time to split the destinations in subsets reachable through different paths or not. For that reason the multicast trees built by our protocol are an aproximation to the Steiner Minimun Tree for energy. Secondly, each node computes the best energy efficient path towards selected forwarding nodes. Although, next forwarding nodes are directly reachable, it can exist an energy efficiently way to reach them. Once those paths are computed localy, messages are forced to follow them by using the well-known Source Routing technique. Several simulations have been made showing that our protocol is as good as centralized aproaches such as MIP and even a 28% better as density increases.

The rest of the paper is organized as follows: Our proposition is described in section 3. In section 4 we show an analysis of the performance of our solution. Finally, section 5 provides some conclusions and discusses open issues.

2 Physical Model

2.1 Network Model

We represent a WSN 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 $\iff$ $dist(v_1, v_2) \leq r$ (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 5, working with more realistic models considering lossy links between nodes is something we plan to work as an extension of this work.

2.2 Energy Model

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

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

Being α the media attenuation factor satisfying $2 \leq \alpha \leq 6$ and C a constant representing the power used to process the radio signal. One key aspect of wireless comunication is the well-known *broadcast advantage*. That is the posibility of reaching several nodes with the same transmission. It allows a node to send a message to any number of its neighbours with a single transmission. In particular, if a node transmits a message with enough power to reach a node placed at distance d, all the nodes placed in the circle of radius d centered at that node, will receive the same message without any extra energy consumption.

3 Energy Efficient Multicast Routing

The main goal of a multicast routing protocol is the delivery of messages to a subset of nodes in the network minimizing the resources needed to do it. In particular it is important to find out paths that can be shared by different destinations. Paths can be shared because the message is the same for all the destinations. This is the same as to find out a tree rooted in the source node and having the destinations as leaves or intermediate nodes. This tree is formed by the set of destination nodes, the source node and some other intermediate nodes. Being a tree means that each node has a single parent but might have zero (leaves or destinations), one (intermediate nodes) or more than one (bifurcation nodes) children. The last one are of great interest because, in these nodes, path towards different subsets of destinations diverge. The more difficult task is to find nodes in where to make those divergences, to reduce te overal cost of the tree. Diverging too early (close to the root) means each individual path from the root to the destination is going to be shorter but the total cost of the tree will be higher. Diverging too late (close to destinations) means longer individual paths but, up to certain limit, lower total cost of the tree.

In multicast geographic routing nodes do not know the complete topology of the network. The only information they have is usually its position and the position of its neighbours. A node forwarding a message has to decide the best way to do it. Usually, the message includes the list of receivers or a way to get it. Each node also knows the list of nodes in its neighbourhood. With this information, the node has to choose between forwarding the message to another node or splitting the list of destinations in subsets and then find the best suitable next hop for every subset. Finally, the message should be delivered to the next hop. Furthermore, routing protocols are normally designed to optimise some criteria such as delay, bandwidth usage, or as in our case, energy consumption. This means that the paths found should be as energy efficient as possible.

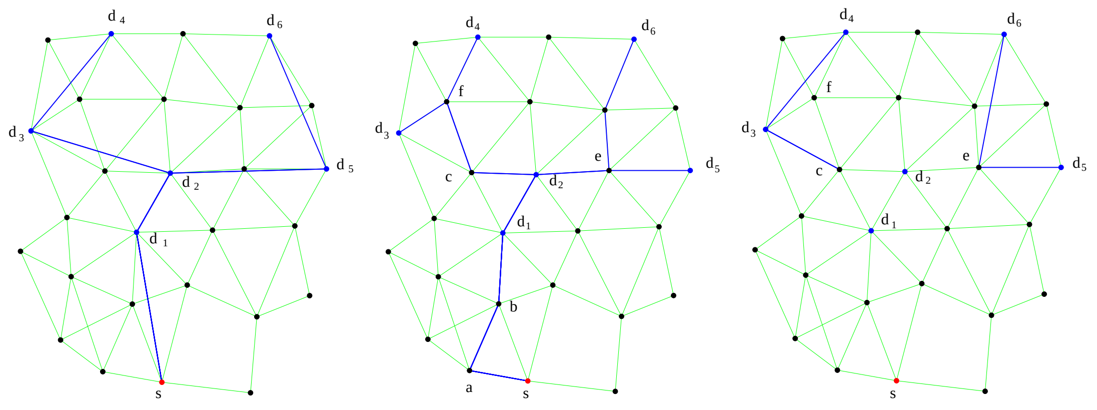
In addition, there are situations in which a node does not have any neighbours providing advance towards the destination. These situations, also called local optimum, have already been studied and solutions such as face2 [2] or GPSR [3] can be applied. Those solutions are based in graph planarization algorithms that can be run locally to find out an alternative way to suround the local maximum. Messages being routed in this way are said to be in perimeter mode. Eventually, a node in which to revert to normal greedy mode is found but the path followed can be arbitrary long. Next section details how our algorithm has adapted the unicast perimeter routing to be used in multicast scenarios.

3.1 Building the Multicast Tree

In geographic routing nodes make decision based solely on the information about its position, the position of its neighbours and the position of the destinations. As we have already commented, the most important aspect of the algoritm is deciding where to split the paths towards the destinations. Kruskal algorithm [14] to find the Minimum Spanning Tree of a graph has a computation time of $O(n^2)$ being n the number of nodes of the graph. Moreover, it has been shown that the MST-heuristic [11][12] is a 2-approximation of the optimal Steiner Tree and using as edges weights the euclidean distance between nodes, the resulting tree can be seen as a energy efficient multicast tree because energy consumption is related to distance as section 2 explains. Nevertheless, nodes in a WSN do not have the information about the complete topology nor the computational power to run this algorithm for potencially thosands of nodes. But, one to many comunications in WSN are usually employed to send messages to a reduced group of nodes. Moreover, the list of destinations might be included in the message itself. Thus, a node can compute the MST of the metric closure and use it to decide the correct direction. In the best case, with enough density in the network, the tree found will resemble the one found using the centralized MST-heuristic, therefore, we are building locally a multicast tree near a 2-approximation of the Steiner Tree.

The proposed algorithm works as follows. A node s included as a relay for some destinations in a received a message, examines the destination list (D) which it is responsible for. All the destinations for which there exist at least one neighbour providing advance are used to build a graph in which s is also included. This graph called $G_D = \{D \cup s\}$, has edges between each two nodes whose weight are the Euclidean distance between them. Then, node s applies kruskal's algorithm to compute the Minimun Spaning Tree (MST) of G_D. MST is then splited in subtrees whose roots are the different direct childs of s in the tree. Each direct child c, represent all the destinations (including itself) that share at least the path between s and c. The last step is starting geographic routing towards each direct child, i.e., towards some concrete destination nodes. That means selecting the best neighbours to route in a greedy way towards each root of the found subtrees. Finally, each greedy step is optimized to save as much energy as possible. This optimization is explained in the next section and it is based in the well-known technique of Source Routing.

Fig. 1(a) shows an example of a WSN and the MST of its metric closure. s is the node that needs to send a message to the set of nodes $D = \{d_1, d_2, ..., d_6\}$. Node s computes the MST of $G_D = \{D \cup S\}$ obtaining the tree shown in the figure. The result is that s takes the decision of performing greedy routing towards d_1 because it is the only direct



(a) MST of the metric enclosure (b) Path found by our protocol (c) Sub-MST trees

Figure 1. Building the MST-like tree structure

child of s in the MST. Thus, s select as best forwarding neighbour node b (Fig.1(b)). As we will see in the next section, the best energy efficiently way of reaching b is through node a, therefore, s sends a mesage to node a. When node a receives the message it repeates the process building the MST of the its own E. In this case the resulting MST is the same and a selects b as next forwarding. Node b does the same to reach node d_1. At node d_1, the list of destinations is reduced by one but the MST computed at this node remains the same, next forwarding neighbour is d_2. At this node, the new MST computed makes a branch, i.e., d_2 has two direct childs in the MST, d_3 and d_5. This implies that the list of destinations (at this point consisting of d_3, d_4, d_5 and d_6) has to be splitted in two subsets. One subset is represented by d_3 and includes d_4 and itself. The other subset, represented by d_5, includes d_6 and itself. Both subtrees are depicted in Fig. 1(c). Node d_2 selects neighbour c as best forwarding node to go towards d_3 and neighbour e to deal with d_5. Only a message is sent by d_2 including in its header the list of destinations that each relay is responsible for. Thus, two different paths start at this point. As we can see in the figure, the MST computed at node e is not the same as the one computed at s, a or even d_1. The same occurs at node f, it was the best forwarding node of c to reach d_3 but now the new MST computed at f is different and a bifurcation is made.

Each forwarding node takes care of a subset of the destinations. Applying Kruskal's algoritm over their G_D as explained before, is the key point to decide wether to split or not those destinations in different paths. The closer to the final destinations, the better the approximation obtained. That allows our protocol to achieve a good performance because the final shape of the tree build is very similar to the one obtained applying the centralized MST-heuristic.

Finally, we have explained the behavior of the routing protocol when destinations can be reached in a greedy way. For those destinations for which there is no neighbour providing advance towards them, perimeter routing is applied to get out of the local optimum reached. The algorithm that we use is an extension of GPSR. Described by Karp and Kung in [3], it is based in the well-known rule of the right hand for traversing graphs. This way only works in a graph where there are no links crossing each other. Thus an algorithm for planarization of graphs is needed. Usually, Relative Neighbourhood Graph (RNG) is used. This algorithm can be run locally using only the information about the position of the nodes's neighbours. Once the RNG is computed, perimeter routing starts following the right hand rule. Perimeter routing finishes when a node closer to the destination than the one in which perimeter routing started, is found. The adaptation of this unicast algorithm to the multicast scenario is very simple. When a node routing a message in greedy mode detects that there are not any neighbour providing advance for some of the destinations it is in charge of, perimeter routing algorithm is run for each of the destinations individually. Let be D_p the subset of destinations for which no greedy route can be found at current node. Perimeter algorithm is applied to each $d \in D_p$. The result of each run is the neighbour that must be the perimeter forwarding relay in which to start the perimeter routing for each destination. It can occur than more than one destination share the same perimeter forwarding relay node. Thus the path for those destinations can be shared minimizing energy consumption.

In all cases, only a message is sent in every step of the routing algorithm. This message includes information about which neighbours are relays and what subset of destinations they must take care of. The message includes also an entry for each perimeter forwarding relay node and the subsets of destinations sharing that same perimeter relay node as next hop. Therefore when a node decides that, for example, some destinations must be routed in greedy mode through neighbour a and some other destinations must be routed in perimeter mode through neighbour b, a message is sent including an entry for each one. Each entry includes the subset of destinations covered by each relay and entries for perimeter forwarding relays nodes also include perimeter information defined by GPSR algorithm such as the node in which perimeter routing started.

3.2 Optimizing each step with Source Routing

The well-known Dijkstra algorithm, to find the shortest path between two nodes in a graph, can be used to find energy shortest paths as long as edge weights reflect the energy needed to send messages between nodes connected by those edges. Shortest Path Tree (SPT) algorithm works applying this algorithm over the complete topology. This is a centralised algorithm that finds out the best path between the source and each destination in terms of energy consumption. The different paths found are then combined to create a multicast tree in which overlaped paths may be shared by multiple destinations to save bandwidth and energy.

The idea is to use the same approach to the local neighbourhood of the node currently routing the message. Taking into account that each node knows its neighbourhood, we can use this information, to locally compute the shortest path from one node to every neighbour. These optimal paths can be used to reach every desired neighbour saving as much energy as possible. In our algorithm, each node finds the energy shortest path to the selected next relay neighbours and force the message to follow these locally computed optimal paths. Including the computed paths in a special header called Source Routing Header (SRH) forces relays to route the message through the correct precomputed path. A node receiving a message with a SRH in it does not compute the next hop, it just remove itself from the SRH and forward the message to the next node in the SRH. This behaviour guarantees that each hop is saving as much energy as possible.

Unlike traditional geographic routing algorithms, our algorithm is able to make use of every neighbour of a node. Traditional geographic routing protocols only use neighbours providing advance towards the destinations to avoid the creation of routing loops. Our approach avoids the loops but at the same time takes advantage of using every possible neighbour because the energy shortest path can include nodes placed farther from the destinations than the current

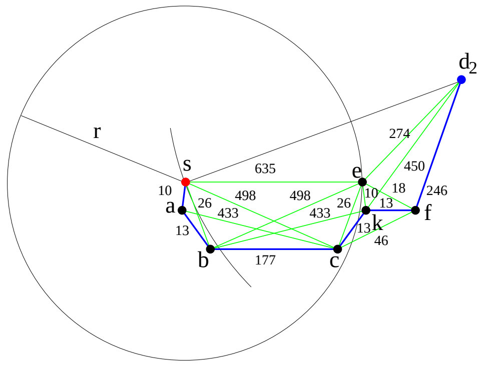


Figure 2. Unicast Source Routing Optimization

node. Therefore we can assure that, in each step of the algorithm the delivery of the message is made using the lowest energy possible. Nevertheless, a globally bad path might be built even if each step is optimized. This is the consecuence of chossing a bad next hop due to the lack of global knowledge.

Our algorithms tries to fix this problem not forcing messages to follow completely the shortest path computed. The path is examined to include only the hops necesary to reach a node on that path, providing advance towards the destination. This node will run the routing algorithm because it is in the end of the SR path. A new decision will be taken in a node closer to the destinations and with some other neighbours hence a better decision can now be taken based on the new information known by the new current node. In fact, the best next hop might differ. In the worst case, no new information might be available and the same next hop will be chosen. Thus, the proposed use of source routing, can only reduce energy.

To explain how SR optimization works we start with an unicast example. Fig. 2 shows an example in which node s currently holding the packet has selected e as the next forwarding node to route the message to the destination d_2. Each link is labeled with the cost in energy of sending a message through it. From the point of view of s, e is the neighbour that provides most advance towards the destination whereas nodes a and b do not provide any advance. But if we compute the energy shortest path from s to e, the resulting path is $[s, a, b, c, e]$. Notice that, globally, the energy shortest path includes also node k but, k is not a neighbour of s, therefore, it does not know about its existence and can not include it in the computation of the path. Following this path completely is the locally optimal decision to reach node e. That is, the best local decision with the knowledge that node s has. However, going to e using that path may not be the best global decision. Our algorithm selects c as

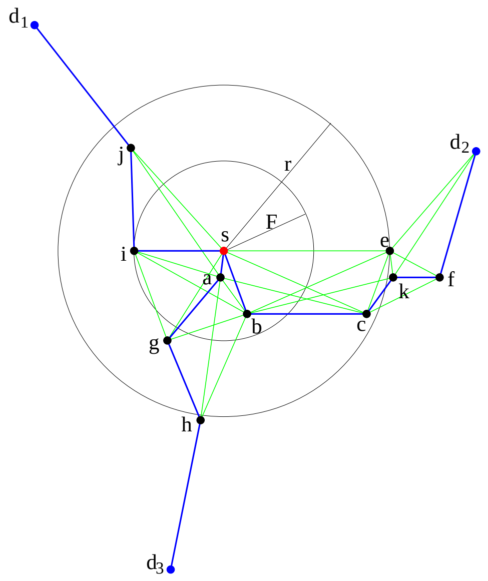


Figure 3. Multicast Source Routing Optimization

next hop node because it is the first node in the local shortest path providing advance towards d_2. Therefore s creates a message including in the header the path that the message should follow to reach c using the previously computed energy shortest path. The list included in the messages is $[a, b]$ because it is not necessary to include the origin nor the destination of the SR as they are already included in the header of the message. When node a receives the message, it checks if it has a SR header or not. In our example, the header exists so, a removes itself from the list and forwards the message to the next node in the SRH. Node b receives the message and behaves in the same way but this time the list remain empty so b forwards the message to the original relay, the node c. At node c source routing ends and then it can recompute the next best hop. Notice that unlike, the rest of greedy routing algorithms, we can select nodes not providing advance as forwarders and that can save energy. In this example, node a is the relay chosen by s and its distance to d_2 is greater than the distance from s.

Node c was in the best path that s could compute with its locall knoledge but c has different neighbours than s so runing again the routing protocol could change the original plans of node s finding a bether path by considering new nodes in c's neighbourhood. In the worst case the route remains the same as s found. In this example, node c does have more information than node s thus a new election is made and node f is selected as the next relay. Notice that e, the original best next hop selected by s, was not the best ellection and not following completely the original best path computed gives the protocol the oportunity to correct the error. Our approach takes advantage of the progressive increment in knowledge that nodes located closer to the destination have. The energy shortest path to reach f is through node k and as it does provide advance towards d_2, thus, it is selected as next relay. In node k a new run of the algorithm is made but, again, the result is that, f is still the best next relay. Although node k has the destination node as a neighbour, going directly is not the best energy efficient path. Node f finish the routing delivering the message to d_2 directly. In this example, the energy shortest path computed with global knowledge is the same path found by our algorithm that does not have global knowledge.

Facing a multicast scenario is a litle bit complicated because it must be taken into account the broadcast advantage that the shared medium used in wireless communications have. As commented in section 2, a message sent using certain power, reaches all the nodes placed closer to a certain distance. In the multicast scenario, some nodes need to split the message into different paths through different subsets of destinations. Not all the paths start with the same relay node because each one might have a different closest neighbour. Let $D = \{d_1, d_2, ..., d_k\}$ be the set of the k next destinations that are childs of current node, s, in the MST graph previously computed. $R = \{r_1, r_2, ..., r_k\}$ is the set of neighbours closest to each destination. Each r_i is the neighbour closest to d_i with $1 \leq i \leq k$. Being a path a list of nodes, Dijkstra algorithm is applied on the local graph made upon node s and its neighbours and the energy shortest path from s to each r_i is found. Calling $P = \{P_1, P_2, ...P_k\}$ to the set of energy shortest path computed for each r_i. We define $f_i = first(P_i)$ as the first node in the list of the path P_i and $H = \{f_1, f_2, ..., f_k\}$ the set of first nodes in each path. The protocol computes the value of $F = max(dist(s, f_i))\ \forall f_i \in H$ as the farthest first node of every energy shortest path computed. Thus, F is the minimun distance that a message sent by s must reach to cover all the first nodes of the paths from s to R. This allows s to send a single message for all branches saving energy and bandwidth.

We show its operation with an example that is an extension of the previous one to deal with multiple destinations. In fact, the links and weights for nodes a to f are the same. Fig 3, shows node s currently holding the message. d_1, d_2, d_3 are the set of destinations included in the message and e, g and j are the neighbours (they are inside the circle of radius r which is the coverage radio of a node) selected to be the relays for d_1, d_2 and d_3 respectively. The energy shortest path from s to e is $[s, a, b, c, e]$ and c is the first node providing advance towards d_1 so the path is reduced to $[a, b]$. The energy shortest path from s to g is $[a, g]$ and again it is cut to $[a]$ because it is the first node in the path providing advance towards d_2. Reaching node j can be done consuming the lowest energy going through node i. Node s computes the value of $F = max(dist(s, a), dist(s, a), dist(s, i))$ to evaluate the energy needed to reach such a distance. When the message is sent, every node inside the circle of radius F will be reached. Given that a and b in our example, the forwarding between a and b of the path for d_1 is not necesary and can be eliminated saving its energy consumption. Thus, we remove node a from the SRH for the relay c leaving only node b on it. Finally, the message is sent and received by a, b, g, and i. Node a is the relay for destination d_3 and must run the routing protocol to find out that its next hop is g (the same decision that took s). The same ocurs in the case of node i that also decides its best next hop to destination d_1 is node j. Node b is in the SRH of relay c, therefore, removes himself from the SRH and forward the message to c without running the routing protocl. From that point, routing continues as in the unicast case because each node currently routing messages (c, g and j) only has a destination to deal with.

4 Experimental Results

We have developed an event-driven simulator to evaluate the performance of our new routing protocol. The most important parameter evaluated is energy consumption. Considering density of a WSN as the mean number of neighbours we wanted to evaluate the impact of density in the behavior of our protocol. For that reason we have generated 100 different graphs for 9 different mean densities varying between 8 and 41 neighbours per node. The basic scenario is a square of 250x250 metres with the source and destination nodes randomly placed and with guarantee that all destinations are reachable from the source. The number of nodes is determined by the mean density we want to have. In the scenarios generated that number goes from 79 to 397 nodes. We consider the same radio range for all the nodes and we set it to 50 meters. As energy consumption grows with distance, we keep the coverage radio and the area of deployment of the sensors fixed. That guarantees fair comparisons between different densities. To evaluate the efficiency of multicast we have chosen four different configurations for the number of destinations. Each configuration is determined by the number of nodes of the network that is actually a destination. The four settings selected are: 1, 5, 10 and 25.

The algorithms simulated are two centralized (SPT and MIP) and a distributed algorithm based on multiunicast routing that we call Energy Efficient Multi Unicast (EEMU). It is based on IPowerProgress [4] and it is probably the easiest and most commonly used alternative to simulate multicast without using an specifically designed multicast routing protocol. By EEMU we understand the use of the IPowerProgress algorithm for each destination independently. SPT is the centralized application of Dijkstra's shortest path tree, in which edge costs are the energy of traversing that edge. Lastly, MIP is the well-known centralized Multicast Incremental Protocol [5].

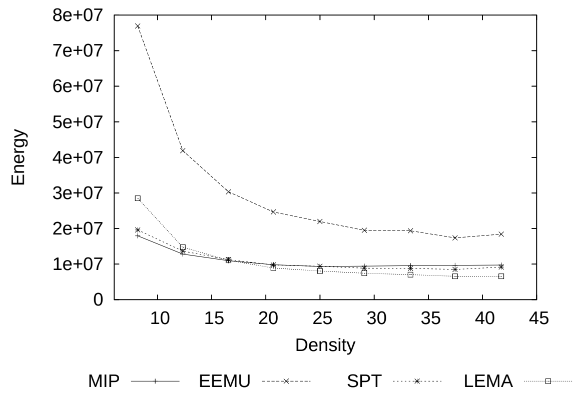


Figure 4. Total energy for 10 destinations

In order to achieve an acceptable level of confidence in the results obtained, we have run a total of 3600 simulations for each algorithm. This is the result of testing each algorithm in 100 different scenarios for each combination of density and number of destinations having a confidence interval for the 95% reduced enough.

4.1 Effect of the density

Figure 4 shows the total energy consumption of each protocol at increasing density for the multicast scenario with 10 destinations. As expected, EEMU obtanis worst results than the multicast oriented protocols. LEMA is a 65% better than EEMU even at low densities. Moreover, the effect of increasing density in centralized algorithms (MIP, SPT) is much lower compared with the localized ones (EEMU and LEMA). This is due to the high probability of finding void zones (local optimum) during the routing in greedy mode when density is very low. In fact, as table 1 shows, for densities below 12, the percentage of messages sent in perimeter mode is significative. Thus, the performance of the greedy part of our algorithm improves as density increases. Only when density is higher than 12 the energy consumption is mostly due to the work of the greedy part of the algorithm. For that reason, MIP and SPT obtain better results than our protocol for densities lower than 12. On the other hand, when density is higher than 12, the results are almost the same and even better than the centralized algorithms. Fig. 5 shows the percentage of energy reduction that our protocol has over MIP and SPT. It goes from a 2% when density is 16 up to a 28% of energy reduction against MIP with a mean density of 41.

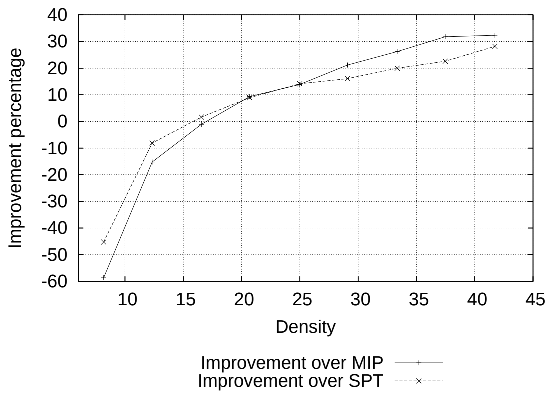


Figure 5. Energy reduction for 10 destinations

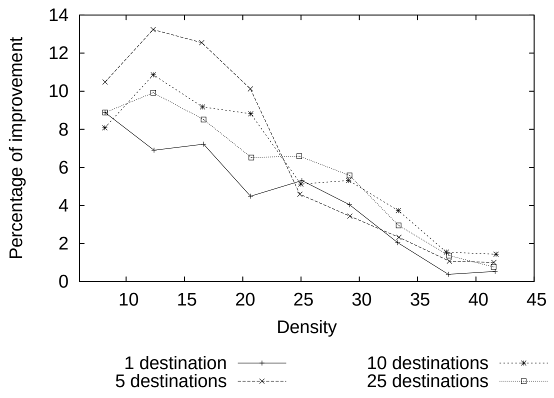


Figure 7. Source Routing Improvement

	Density								
Algoritm	8	12	16	20	24	29	33	37	41
LEMA	25%	4%	2%	0.5%	0%	0%	0%	0%	0%
EEMU	17%	1%	0.5%	0%	0%	0%	0%	0%	0%

Table 1. Percentage of messages sent under perimeter routing

4.2 Effect of the number of destinations

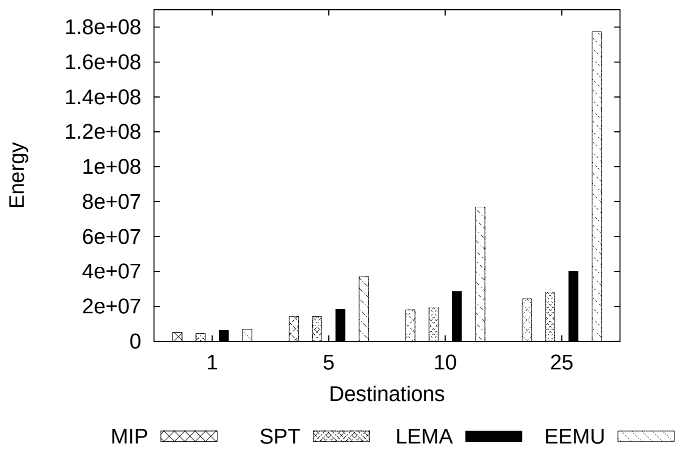
As expected, the higher the number of destinations the higher the energy consumption. Nevertheless, the increase is not linear except for EEMU. EEMU does not take advantage of the multicast philosophy, thus its energy consumption grows lineally with the number of destinations. Figures 6(a) to 6(d) show the total energy consumption of the protocols at increasing the number of destinations. Notice that the difference in energy consumption between 25 and 10 is almost the same as between 10 and 5, but, the increasing in destinations is three times higher. On the other hand, the figures show how our protocol equals and outperform MIP and SPT when density is higher than 12.

4.3 Source Routing Improvement

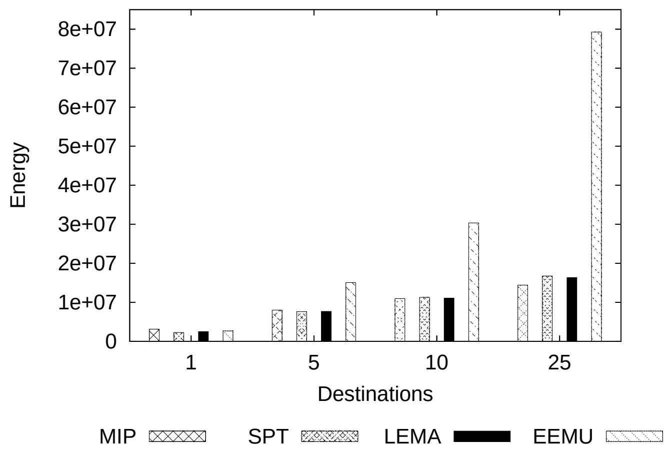
As our protocol makes use of the Source Routing technique, in this section we analyze the improvement achieved by that. Fig. 7 shows the reduction of energy due to the use of SR for different densities. To obtain this values we have run LEMA with and without using SR and we have compared the results. The use of SR can only reduce the energy needed, thus, in the worst case the total energy migh be the same. Nevertheless, our results show that SR does play an important role in the reduction of energy. In the figure, each curve represents a different multicast scenario (different number of destinations). The figure shows that the number of destinations does not affect significantly to the amount of SR applied SR because the four curves are very similar. Nevertheless, the density does have an effect in the probability of finding useful nodes to be used in SR. For that reason, at low densities the improvement of SR is low and at high densities, optimal paths between forwarders do not usually have nodes not providing advance. Thus, SR ends at the very beginning and can not be of great help. The maximum contibution of SR appears when mean density is around 12 (being of a 13%) and as density increases the percentage of improvement due to SR decreases. The minimum improvement is set on a relatively high value of 1%.

5 Conclusions and Future Work

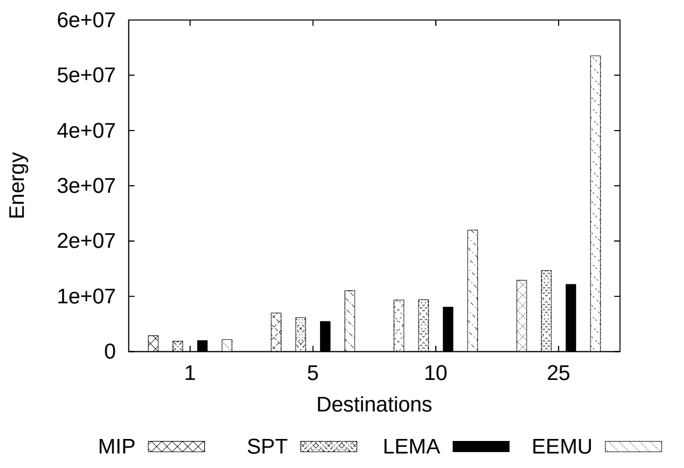
Many new applications developed to take advantage of the characteristics of WSN needs group communications. Motivated by the necessity of a multicast network primitive to support those new applications, we have developed a new multicast routing algorithm. It is based on geographic routing. Thus, it is not necessary to know in advance the topology of the network and it is able to adapt to changes in the network. At the same time, not using routing tables makes the protocol scalable with the size of the network. On the other hand, two innovative ideas have been introduced to deal with the scarce resources of WSN, specially the lack of a guaranteed source of energy. We apply two common centralized algorithms (MST and Dijkstra) in localized conditions with only partial information. Nodes use MST algorithm to find out the best next subset of destinations towards which forwarding the messages. Nevertheless, geographic routing is used to reach them. Finally, Dijkstra algorithm is also applied locally to reduce the energy consumption of



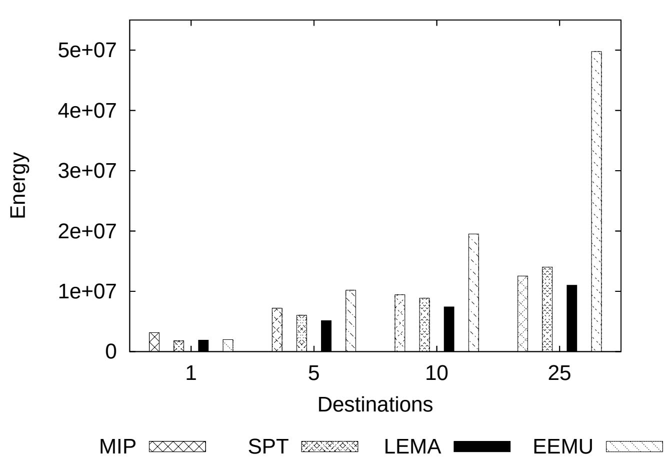
(a) density = 8



(b) density = 16



(c) density = 25



(d) density = 29

Figure 6. Total Energy for varying destination number

forwarding a message. Nodes compute the energy optimal path between them and its next forwarding neighbour and force the message to follow that path using the well-known Source Routing technique. Several simulation have been made showing that our algorithm achieves very good results outperforming classic centralized algorithms such as MIP and SPT. For future work we are working with more realistic physical model that take into account losses and make use of ARQ methods to guarantee delivery while maintaining reduced the energy needed to route messages.

References

[1] Finn, G. G., Routing and addressing problems in large metropolitan-scale internetworks, Tech. Rep. ISI/RR-87-180, Information Sciences Institute, Mar. 1987.

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

[3] Brad karp and H.T.Kung, GPSR: Greedy Perimeter-Stateless Routing for Wireless networks, in Proceedings of the sixth annual ACM/IEEE International Conference on Mobile computing and networking (MobiCom ’00), Boston, Massachusetts, August 2000, pp. 243-254

[4] Ivan Stojmenovic and Xu Lin, Power aware localized routing in wireless networks, IEEE Transactions on Parallel and Distributed Systems, Vol. 12, No. 11, November 2001, 1122-1133

[5] Jeffrey E. Wieselthier, Gam D. Nguyen and Anthony Ephremides, Energy-Efficient Broadcast and Multicast Trees in Wireless Networks, Mobile Networks and Applications, Vol. 7, 481-492, 2002

[6] T. Ballardie, P. Francis, and J. Crowcroft, Core Based Trees (CBT) – An architecture for scalable inter-domain multicast routing, in Proceedings of ACM SIGCOMM’93, San Francisco, CA, October 1993, pp.85–95.

[7] C.-C. Chiang, M. Gerla, and L. Zhang, Adaptive Shared Tree Multicast in Mobile Wireless Networks, in Proceedings of GLOBECOM ’98, pp. 1817–1822, November 1998.

[8] M. Mauve, H.F$\ddot{u}\beta$ler, 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.

[9] M. Transier, H. Füßler, J. Widmer, Martin Mauve and Wolfgang Effelsberg, Scalable Position-Based Multicast for Mobile Ad-hoc Networks, First International Workshop on Broadband Wireless Multimedia: Algorithms, Architectures and Applications (BroadWim 2004),San Jose, CA, USA, 2004

[10] R.-M. Karp, Reducibility among combinatorial problems, in Complexity of computer computations, Plenum Press, New York, 1975, pp.85103.

[11] L. Kou, G. Markowsky, and L. Berman A fast algorithm for Steiner trees. Acta Informatica, no. 15, vol. 2, 1981, pp. 141145.

[12] J. Plesnik The complexity of designing a network with minimum diameter. in Networks, no. 11, 1981, pp. 7785.

[13] Bin Wang and Sandeep K. S. Gupta G-REMiT: An Algorithm for Building Energy Efficient of Multicast Trees in Wireless Ad Hoc Networks. in Proceedings of NCA’03, 2003

[14] J. B. Kruskal, On the shortest spanning subtree of a graph and the traveling salesman problem, in Proceedings of the American Mathematics Society, vol. 7(1), pp. 4850, 1956.