Localized Minimum Spanning Tree Based Multicast Routing with Energy-Efficient Guaranteed Delivery in Ad Hoc and Sensor Networks

Hannes Frey
University of Paderborn
D-33098 Paderborn
hannes.frey@uni-paderborn.de

François Ingelrest
LCAV, EPFL
CH-1015 Lausanne, Switzerland
Francois.Ingelrest@epfl.ch

David Simplot-Ryl
IRCICA/LIFL, Univ. Lille 1
CNRS UMR 8022, INRIA Futurs, France
David.Simplot@lifl.fr

Abstract

We present a localized geographic multicast scheme, MSTEAM, based on the construction of local minimum spanning trees (MSTs), that requires information only on 1-hop neighbors. A message replication occurs when the MST spanning the current node and the set of destinations has multiple edges originated at the current node. Destinations spanned by these edges are grouped together, and for each of these subsets the best neighbor is selected as the next hop. This selection is based on a cost over progress metric, where the progress is approximated by subtracting the weight of the MST over a given neighbor and the subset of destinations to the weight of the MST over the current node and the subset of destinations. Since such greedy scheme may lead the message to a void area (i.e., no neighbor providing positive progress), we propose a new multicast generalization of the well-known face recovery mechanism. We provide a theoretical analysis proving that MSTEAM is loop-free, and achieves delivery of the multicast message as long as a path to the destinations exists. Our results demonstrate that MSTEAM outperforms the best existing localized multicast scheme, and is almost as efficient as a centralized scheme in high densities.

1 Introduction

We consider multi-hop ad hoc and sensor networks, and localized routing schemes, where nodes maintain knowledge only about spatially nearby nodes. Thus, unlike in centralized schemes, changes in the network require only local message exchanges. Because of small batteries, routing must be energy-efficient to maximize network lifetime.

We especially consider *geographic* routing, where each node knows its own location, and where the next hop is chosen based on the position of neighbors with respect to the destination. Almost all such schemes are based on greedy heuristics: the best neighbor according to an evaluation function is selected. For instance, a solution is to choose the closest neighbor to the destination. Obviously, the message may be blocked (*i.e.*, no neighbor closer to the destination than the current node). To guarantee delivery, an additional scheme, named face routing [1], is generally used to escape from these void areas.

In multicast routing, a message is to be delivered from a source node to a set of destinations. When using geographic routing to solve the multicast problem, the most challenging question is to decide when the message should be replicated into different packets. The best solution is to route the message using a common path among all destinations, and then to replicate it at the end of this path. Of course, the difficulty lies in determining the best common path by using only local information at each hop. Moreover, even with a global knowledge, this problem is NP-complete.

In this paper, we present a *Minimum Spanning Tree based Energy Aware Multicast* scheme (MSTEAM), a generalization for the multicast case of the cost over progress framework described in [6]. In such scheme, greedy routing is done by selecting at each hop the neighbor v providing the smallest ratio between the cost needed to transmit the message to v and the progress toward the destination provided by v. We use a minimum spanning tree (MST) as a backbone to decide when a message has to be replicated. In addition, we use an MST-based localized next hop selection scheme which considers energy consumption of sending a message to a neighbor v over the progress achieved thanks to this node. To escape from void areas, MSTEAM uses a multicast generalization of the face protocol [1].

A key aspect of our solution is that it highly fits wireless dynamic networks since it is localized. Indeed, forwarding nodes need to construct local MSTs using only information on their 1-hop neighborhood, which may be obtained thanks to beacon messages. MSTEAM is also well-suited for constrained mobile devices, since an MST may be efficiently computed in time $O(n \log n)$. Moreover, MSTEAM is loop-free and always achieves delivery.

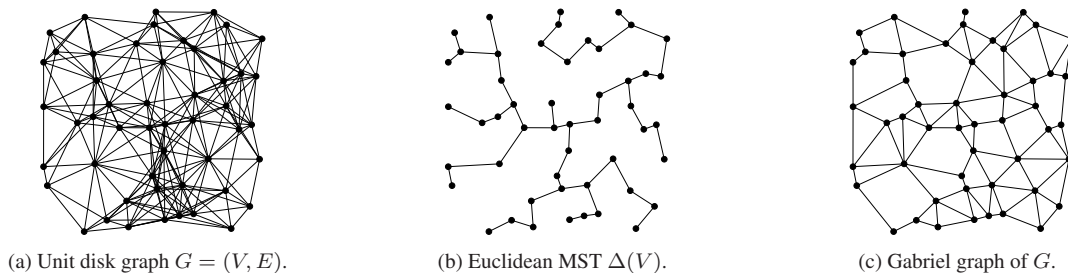| (a) Unit disk graph $G = (V, E)$. | (b) Euclidean MST $\Delta(V)$. | (c) Gabriel graph of G. |

Figure 1: A unit disk graph $G = (V, E)$ and two subgraphs.

2 Preliminaries

2.1 Network model

A wireless network is a graph $G = (V, E)$, V being the set of vertices (network nodes) and $E \subseteq V^2$ the set of communication links: there exists a pair $(u, v) \in E$ if u is physically able to communicate with v. The neighborhood set $N(u)$ of a node u is defined as:

$$N(u) = \{v \in V \mid v \neq u \wedge (u, v) \in E\}.$$

The density d is the average number of neighbors per node. The set of edges E depends on the underlying physical model. The most well-known one is the *unit disk graph* (UDG) model. Given a set of nodes V and a maximum communication range R, it defines E as:

$$E = \{(u, v) \in V^2 \mid u \neq v \wedge |uv| \leq R\},$$

$|uv|$ being the Euclidean distance between u and v. Fig. 1a provides an example of a UDG. For a given multicast task, the set of destinations is denoted as $T = \{t_1, \ldots, t_k\}$. We assume that nodes can adjust their transmitting power, *i.e.*, sending a message from u to v requires the smallest possible power. We assume that nodes collect 1-hop neighborhood information by sending beacons.

2.2 MAC layer model

When a message is replicated, the forwarding node sends the packet to more than one neighbor. We consider two MAC layer models. In the *unicast MAC* layer, sending a message to j neighbors is performed by j independent unicast transmissions. In the *multicast MAC* layer, sending a message is done by only one transmission, exploiting the broadcast capabilities of the wireless media. A detailed investigation on how a reliable communication is achieved—either in the unicast or the multicast MAC layer—is beyond the scope of this work.

2.3 Geometric concepts

The minimum spanning tree (MST) is a well-known graph construction: a tree $\Delta(u_1, \ldots, u_n)$ is an MST if its weight $|\Delta(u_1, \ldots, u_n)|$ is minimal. The weight of the tree denotes the sum of the weight over all tree edges. In a Euclidean MST, illustrated in Fig. 1b, the weight of an edge is equal to its Euclidean length. Such trees may be efficiently computed in time $O(n \log n)$. Note that we use throughout the paper the notation $\Delta(S)$, which is equivalent to $\Delta(u_1, \ldots, u_n)$ for any set $S = \{u_1, \ldots, u_n\}$. The Steiner tree problem is similar to the MST one: the goal is to construct a tree $\Gamma(u_1, \ldots, u_n)$ with minimal weight, while allowing the insertion of additional intermediate *Steiner points*, to reduce the weight of the resulting spanning tree. This problem is NP-complete.

A planar graph is a graph in which no edges intersect. *Gabriel graph* construction (see Fig. 1c) is a localized construction method based on a geometric concept, introduced in [4]. Starting from a UDG $G = (V, E)$, each edge $(u, v) \in E$ is removed if there exists a vertex w located inside the circle $U(u, v)$ of diameter $|uv|$ centered at the midpoint of the segment $[uv]$. This graph is interesting for decentralized networks since the removal algorithm may be applied independently by each node, and does not require any message exchange.

3 Related Work

One of the first multicast algorithm for ad hoc networks is MIP [11], a variant of the BIP algorithm. In BIP, a tree is built from a source node by adding at each step the less expensive node. Both schemes are centralized – they cannot be applied in a distributed network without a large overhead –, and MIP tree is composed of subtrees coming from BIP, which is explicitly built for broadcast. As a result, routing is not optimal because there is not enough path reuse.

The pioneering work in *localized* multicast is [7], which describes the *Position-Based Multicast* (PBM) protocol. Forwarding is performed by determining the neighbor sub-

set maximizing a weighted sum over two conflicting objectives: maximizing the number of next hops and minimizing the remaining distance to the destinations. The weight of the two objectives is controlled by a parameter $\lambda \in [0,1]$. An early packet replication is achieved by a λ close to 0. Less frequent replications are obtained with λ close to 1. PBM requires testing each possible subset and selecting the one maximizing the objective function. The complexity of PBM is thus $O(2^m)$ for m neighbors. To recover from void areas, face recovery computes an average point p over the destinations and starts traversal of the face intersected by the straight line sp. When a destination can be handled in greedy mode again, the message is replicated and the packet is handled in greedy mode again, while the other part continues with face recovery. Even under the Gabriel graph construction, routing loops may occur with PBM.

The protocol described in [2] performs message forwarding along a backbone defined by constructing a spanning tree over the source node and the destinations. Three spanning tree heuristics are considered. For each spanning tree edge st originated at the start node s, a single multicast instance is sent to the end node t. The destination subset addressed by this message consists of those destinations that are reachable over this tree edge. This is comparable to the replication strategy we describe. However, the mechanism described in [2] differs from MSTEAM in the following ways: first, messages must follow a backbone edge st until they eventually arrive at the destination. Only when arrived at this node messages might be replicated again. This is in contrast to MSTEAM where replications are possible at any node. Second, in PBM, all destinations reachable over t in the backbone will be disconnected whenever node t is not reachable from the source node. Third, multicast routing described in [2] is only concerned with localized construction of multicast overlays. Routing a message along an overlay edge uv is done in a centralized way by calculating the shortest path from u to v. In this work, we describe a localized metric in order to select an energy-efficient next hop by using information about the destinations and the neighbors only. Fourth, with global information, recovery from greedy routing failures is not an issue in PBM.

A recent localized scheme, *Geographic Multicast Routing* (GMR), can be found in [8]. The algorithm is based on the cost over progress framework [6] and, opposed to PBM, does not require setting a proper network-dependent parameter λ. In the unicast case, cost over progress denotes the relation between cost produced in the next hop and the progress achieved by this next hop. The multicast extension of this framework minimizes the number of selected next hops over the progress achieved by the selected set. Progress is the difference between the sum over all individual distances between the current node and destinations, and the sum over distances of each next hop and the destinations

covered by this node. A node v *covers* a destination if this destination is closest to v compared to all other next hop. In contrast to [7], this scheme describes also an efficient neighbor set selection strategy which reduces the cost from $O(2^m)$ to $O(mk \min(m,k)^3)$ in the worst case, where k is the number of destinations and m the number of neighbors. Since this scheme is greedy, a message can be blocked. The scheme describes a face recovery strategy which applies traditional unicast face traversal for each destination. To save bandwidth, however, face messages traveling the same face are aggregated into a single message.

All these schemes consider hop count as the optimization criteria. When the hardware provides signal strength adaptation, a single transmission over a large distance can be more expensive than many small-distance transmissions. The first localized scheme considering this possibility is described in [9]. This scheme, GMREE, is an extension of GMR considering the cost of the total energy consumption of the next transmission.

4 MST based Multicast

In this section, we provide a motivation of the goodness of using Euclidean MSTs to decide a message replication, and to determine the best next hop node for a given set of destinations. Unless specified, MST always denotes Euclidean MST.

4.1 Why MST-based Backbones?

When global information is available, finding the optimal multicast tree is possible, though, being an NP-complete problem. Our goal is to define a localized heuristic to find a tree with a "low" cost. Quantifying "low" in terms of a formal analysis is beyond our scope, but we will show empirically that the cost obtained thanks to the described heuristic does not significantly depart from an efficient centralized solution.

The quality of an algorithm depends on two factors:

- The message replication strategy, which should aim at message forwarding along a cost effective backbone.

- The next hop selection, which should aim at cost-effective message forwarding along this backbone.

Message forwarding along an edge of the multicast backbone selects the "best" neighbor with respect to the considered metric (*e.g.*, hop count, Euclidean distance) and the destinations reachable along this backbone edge.

Let s be the source and $T = \{t_1, \ldots, t_k\}$ the destinations. Let $C(u,v)$ denote the weight of the shortest weighted path from u to v. Under the unicast MAC assumption, a weighted Steiner tree $\Gamma(s, t_1, \ldots, t_k)$, using

(a) All destinations in the same set.
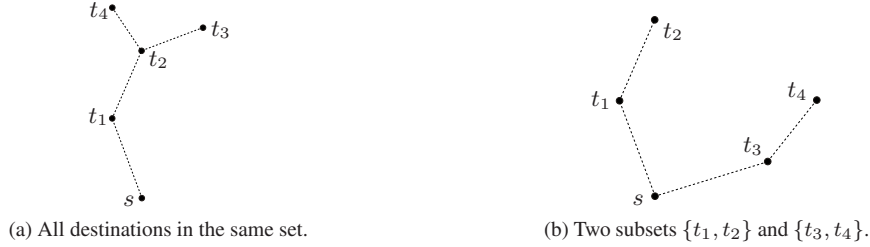
(b) Two subsets $\{t_1, t_2\}$ and $\{t_3, t_4\}$.

Figure 2: The replication strategy used by MSTEAM: $\Delta(s, t_1, t_2, t_3, t_4)$ is used to replicate the message at node s.

$C(u, v)$ as the cost function, defines the cost-optimal multicast backbone. We do not assume that a node is aware of all network nodes to compute such a Steiner tree. Moreover, we do not assume that the cost function $C(u, v)$ is even known to the nodes. Thus computing a Steiner tree as an optimal backbone is impossible in this general setting. As an approximation we use the concept of weighted MST, which may be computed even by constrained devices. Since the energy model is unknown, we approximate the routing cost by the simplified assumption that $|uv| < |uw|$ implies $C(u, v) < C(u, w)$. Then, the weighted MST is equivalent to the Euclidean MST.

Under the multicast MAC assumption, energy savings are possible at nodes where the message is replicated. At this point, any set of next hops might produce the same routing cost. In a small-scale multicast, it might thus be more efficient to perform a single direct "large" broadcast transmission instead of many "short" transmissions. In a large-scale multicast, however, we expect that the cost savings which are possible at replication nodes will be outweighed by the cost required to route the message between those replication points. Thus, we use the same MST approximation under both multicast MAC assumptions.

4.2 Replication strategy

Let s be the current forwarding node and $T_i \subseteq T$ the set of destinations handled by s. Node s computes the MST $\Delta(\{s\} \cup T_i)$ over itself and T_i. This tree provides the backbone to use to reach all destinations in T_i from s. The message thus has to be routed along the edges of this tree, and must be replicated at node s if multiple paths start at this node. Actually, each of these paths is represented by an edge originating at s, and spans a subset of destinations. These are forming exactly a destination subset to which s has to send an individual message copy.

This strategy is illustrated in Fig. 2, where s handles the destinations t_1, t_2, t_3 and t_4. In Fig. 2a, the MST $\Delta(s, t_1, t_2, t_3, t_4)$ has only one edge originated at node s, so all destinations are grouped together. In this case, the

message is not replicated and will be routed along (s, t_1). In Fig. 2b, there are two edges originated at node s: the first one spans t_1 and t_2, the second one spans t_3 and t_4. The message is thus replicated into two packets. The first one is routed along (s, t_1) toward $\{t_1, t_2\}$, the second one is routed along (s, t_3) toward $\{t_3, t_4\}$.

4.3 Energy-efficient metric

Greedy routing is done with a generalization of the framework described in [6] for the multicast case. In this framework, two things need to be estimated: the *cost* (in terms of energy consumption) of choosing a given neighbor v as the next hop, and the *progress* toward the destination subset T_i provided by the replication strategy. The neighbor with the smallest cost over progress ratio is chosen as the next hop for T_i.

Since the current forwarding node u considers the MST $\Delta(\{u\} \cup T_i)$ as the backbone to route the message toward T, $|\Delta(\{u\} \cup T_i)|$ may be used as a local estimation of the remaining distance the message has to travel. Considering a neighbor v, $|\Delta(\{u\} \cup T_i)| - |\Delta(\{v\} \cup T_i)|$ is an estimation of the progress provided by v toward T_i. The greedy scheme assumes that the current node u always selects a candidate node v providing positive progress.

Sending a message from a node u to its neighbor v requires an amount of energy denoted $f(u, v)$. Thus, the cost over progress ratio $Q(u, v, T)$ at node u of a neighbor node v is:

$$Q(u, v, T) = \frac{f(u, v)}{|\Delta(\{u\} \cup T)| - |\Delta(\{v\} \cup T)|}.$$

The expression of $Q(u, v, T)$ is a generalization of the unicast routing metric $f(u, v)/(|ut| - |vt|)$ described in [6]. In this connection, u is the current node, v the next hop candidate, and t the destination node. When $T = \{t\}$, the expression $Q(u, v, T)$ reduces to this formula since an MST over a pair of nodes is the straight line connecting them.
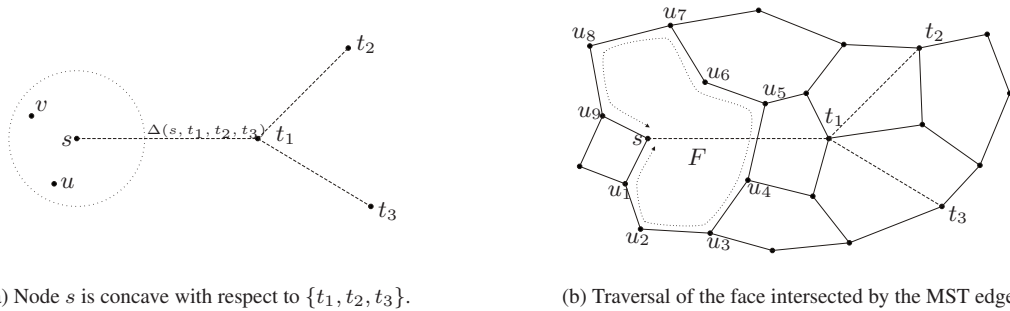
(a) Node s is concave with respect to $\{t_1, t_2, t_3\}$.

(b) Traversal of the face intersected by the MST edge.

Figure 3: Minimum spanning tree based face multicast routing is used in order to recover from concave nodes.

4.4 Recovery strategy

Suppose that in Fig. 3a s has to send a message toward $\{t_1, t_2, t_3\}$. Destinations are connected over (s, t_1) of $\Delta(s, t_1, t_2, t_3)$. However, s may not select any of its neighbors u and v, since they satisfy $|\Delta(u, t_1, t_2, t_3)|, |\Delta(v, t_1, t_2, t_3)| > |\Delta(s, t_1, t_2, t_3)|$: the message is blocked. This happens whether any of these destinations are reachable from s or not. We denote such node as *concave* with respect to the destination subset.

Face is a unicast routing scheme that can be used to handle greedy routing failures for each destination individually. We describe for the first time a multicast extension of face that can handle all destinations at once. Similar to unicast face routing, the multicast scheme requires a topology control mechanism which transforms the underlying network into a planar graph. Here, we employ the Gabriel graph which requires the network to comply to the UDG model.

As depicted in Fig. 3b, a planar graph partitions the plane into faces that can be traversed by employing the left/right hand rule; a node sends the message along the edge which is lying next in clockwise/counterclockwise direction of the edge it was received from. For instance, when starting at node s in Fig. 3b, the face F will be traversed along the path $su_1u_2 \ldots u_9$ when using the right hand rule. Unicast face recovery has different variants; We employ the one that transmits the message along the sequence of faces that are intersected by the line st connecting the source s to the destination t. When the message arrives at a node closer to t than s, greedy routing is used again. With a Gabriel graph, this mechanism simplifies to traversing the first face.

The idea of multicast face is as follows. Suppose that a node s has computed a destination subset T_i for which no better greedy neighbor exists. Let st be the edge connecting s to $\Delta(\{s\} \cup T_i)$. By using any of the two rules – right or left hand – node s starts traversal of the face intersected by the outgoing MST edge st. Face traversal continues until the message arrives at a node u satisfying $|\Delta(\{u\} \cup T_i)| < |\Delta(\{s\} \cup T_i)|$. At this node, the destination subset T_i is

handled in greedy mode again. A special case occurs when no such node u is found during face traversal. In this case, to avoid loops, the message is dropped if it is about to be sent again over the first face traversal edge in the same direction.

Referring to Fig. 3b, edge st_1 connects s to $\Delta(s, t_1, t_2, t_3)$. Since s is concave with respect to $\{t_1, t_2, t_3\}$, it starts traversal of face F, *i.e.*, the face intersected by st_1. Assuming the right hand rule, face traversal visits nodes u_1, u_2, and u_3. Since u_3 is the first one satisfying $|\Delta(u_3, t_1, t_2, t_3)| < |\Delta(s, t_1, t_2, t_3)|$, it handles the destination subset $\{t_1, t_2, t_3\}$ in greedy mode again.

5 The MSTEAM protocol

5.1 Description

Given the destinations $T = \{t_1, \ldots, t_k\}$, the source s first decides whether a message replication occurs. It thus computes the MST $\Delta(\{s\} \cup T)$, and groups all destinations spanned by edges originated at s (see Sec. 4.2).

For each subset $T_i \subseteq T$ obtained, s computes a subset $N_i(s) \subseteq N(s)$ containing all neighbors $v \in N(s)$ such that $|\Delta(\{v\} \cup T_i)| < |\Delta(\{s\} \cup T_i)|$ (neighbors providing positive progress toward T_i). If $N_i(s) \neq \emptyset$, s computes the cost over progress ratio $Q(s, v, T_i)$ for each neighbor $v \in N_i(s)$ (refer to Sec. 4.3). The neighbor providing the best ratio is the next hop toward T_i. If $N_i(s) = \emptyset$, then the message is blocked, and face recovery must be used. Node s applies the strategy presented in Sec. 4.4 to select the face node v as the router toward T_i. The process is repeated until all subsets T_i have been considered.

In the unicast MAC case, a packet is sent for each subset T_i, with the set of destinations, the selected router and the mode (greedy or face). With face routing, the packet also contains the first edge traversed by the packet in this mode, and the weight of the MST at the starting node ($|\Delta(\{s\} \cup T_i)|$ in this example). If multicast MAC is considered, this information is aggregated into the same packet. This means that this packet contains a list of all next hops
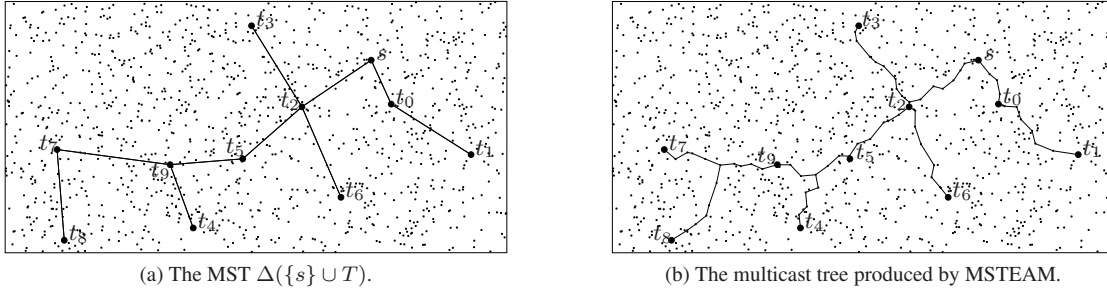
(a) The MST $\Delta(\{s\} \cup T)$.

(b) The multicast tree produced by MSTEAM.

Figure 4: A sample run of MSTEAM for a set T of 10 destinations and a density $d = 35$.

and for each of them, the set of destinations they serve, the mode to use and the additional face information. In both cases, the packet is sent using the minimum energy needed for successful transmission to the next hop(s).

When a node u receives a packet, it checks whether it is a forwarder. If not, the packet is dropped. If so, it checks the routing mode currently used for the given set of destinations $T_i \subseteq T$. In greedy mode, u repeats the process followed by s. In face mode, it checks whether it is closer to the destinations (e.g., $|\Delta(\{u\} \cup T_i)|$ is less than the weight written in the packet). If so, it handles T_i in greedy mode. If not, face recovery is applied once again (refer to Sec. 4.4). Of course, if u is a destination, it removes itself from T_i.

Fig. 4 shows a sample run of MSTEAM over a random network. Figure 4a gives the MST $\Delta(\{s\} \cup T)$, while Fig. 4b provides the multicast tree produced by MSTEAM. The MST spanning all destinations was used at the source s. Since two edges originate at s, the message has been replicated into two packets at s. The first one was sent toward t_0 and t_1 along (s, t_0), while the second one was sent toward the other nodes along (s, t_2). Figure 4b shows that MSTEAM was able to follow these edges in an effective way. One can also observe that the replication strategy correctly works by looking at the path followed to reach t_4 and t_9 from the node close to t_5. Instead of following (t_5, t_9), MSTEAM routed the message along a common path among t_4 and t_9, and then replicated it at the end of this path.

Regarding the complexity of MSTEAM, a node in greedy mode computes an MST for the replication strategy, which has a time complexity in $O(k \log k)$, k being the number of destinations. In the worst case, all destinations are handled separately. For each of them and for each neighbor, a new MST is computed. In this case, the complexity in time of MSTEAM is $O(mk^2 \log k)$ for the greedy mode, m being the number of neighbors. This complexity may be better estimated since a MST has a maximum degree of 6, regardless of k. Since face mode has a complexity in $O(k \log k)$, the complexity of MSTEAM in the worst case is $O(mk \log k)$, which is lower than the complexity of GM-REE ($O(mk \min(m, k)^3)$, still considering the worst case).

5.2 Correctness of MSTEAM

We now prove that MSTEAM is loop-free and guarantees delivery. We assume the UDG model with radius R, and the use of Gabriel graph to construct the planar graph used during face recovery. The proof of the two lemmas may be found in [3].

Lemma 1 *Let s be a node where face recovery for the destinations $\{t_1, \ldots, t_k\}$ was started, and st the edge connecting s to $\Delta(s, t_1, \ldots, t_k)$. If s can reach at least one destination, then traversal of face F, intersected by st, always arrives at a node u satisfying $|\Delta(\{u, t_1, \ldots, t_k\})| < |\Delta(\{s, t_1, \ldots, t_k\})|$.*

Lemma 2 *A message addressed to $S = \{t_1, \ldots, t_k\}$ will either be dropped, replicated, or delivered after a finite number of steps.*

Theorem 1 *The described multicast routing scheme MSTEAM is loop-free and provides delivery guarantees.*

Proof 1 *Let s be the source and T the set of destinations. In MSTEAM, a message is either kept or replicated in a forwarding step, and two messages are never merged. Thus, for each possible subset $S \subseteq T$, at most one instance of a message addressing this set may exist. It follows that the number of possible message instances is finite. Finally, due to Lemma 2 each instance is handled a finite number of forwarding steps. It follows that the total number of forwarding steps is finite, i.e., no routing loop occurs.*

Let t be an element of T. Suppose there exists a path from s to t, and that t is dropped during routing. A message might only be dropped when it is handled in face mode. Let u be the node where face traversal was started, and S the subset of destinations handled in face mode. Since the message is dropped, all node v visited during face traversal satisfies $|\Delta(\{v\} \cup S)| \geq |\Delta(\{u\} \cup S)|$. Since u was reached by s, and since s can reach t, u can reach at least one node in S. By Lemma 1, it follows that face traversal will visit a node w satisfying $|\Delta(\{w\} \cup S)| < |\Delta(\{u\} \cup S)|$, a contradiction.

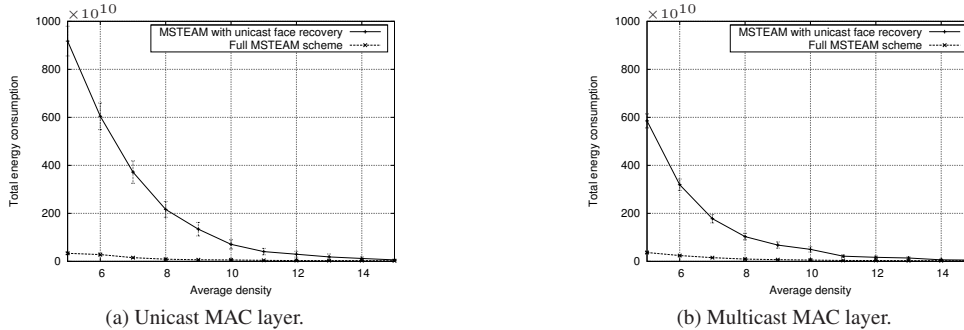(a) Unicast MAC layer.

(b) Multicast MAC layer.

Figure 5: Performance of the MST based face recovery mechanism used in MSTEAM (10 destination nodes).

6 Performance evaluation

6.1 Simulation settings

We used a home-made simulator with the UDG model. Nodes have a maximum range $R = 250$, and a number of them are randomly put on an area of size 2500×2500 to obtain a given density. The 95% confidence interval is given on each figure. The energy consumption f(u, v) of a node u transmitting a message to a node v is:

$$\text{f}(u, v) = |uv|^\alpha + c_e,$$

α being the path loss constant. We used $\alpha = 4$ and $c_e = 10^8$, used in [9].

We selected GMREE [9] as it is the only other geographic energy-efficient *localized* multicast scheme. GMREE uses an underlying subgraph, and authors showed that the Local Shortest Path Tree (LSPT) [10] using f(u, v) as the cost function is the most efficient one. We thus restricted our simulations of GMREE to the LSPT. GMREE was compared in [9] to ESP (Energy-Efficient Shortest Path), the centralized application of Dijkstra's shortest path tree, using f(u, v) as the cost function. We found that ESP has variable performances based on the maximum communication range: thanks to its centralized knowledge, ESP generates efficient routes, where the length of each hop is close to the optimal, while replicating messages too often. When the communication range is too low, routes are not enough efficient to counterbalance the wastes coming from replications. We believe that this problem has been overlooked in [9]. We thus used another centralized scheme, using an approximated weighted Steiner tree [5] as the multicast tree.

6.2 Experimental results

Fig. 5 shows the efficiency of the new MST-based face recovery mechanism. Small densities were used, to maximize the number of void areas. The new scheme is by far

superior to the basic one: considering the density $d = 5$, energy consumption is divided by approximately 27 in Fig. 5a and 16 in Fig. 5b. Even with $d = 10$, the ratio is still around 10 in both figures. As the density increases, the number of void areas decreases and face recovery is less used, so the two schemes become equal. Even in the multicast MAC layer case, where packets are aggregated before transmission, the basic scheme is inferior because each destination is handled separately, so that the protocol quickly uses different paths and lead to increased energy consumption.

Fig. 6 shows results at increasing density. As aforementioned, ESP replicates the message too early and is less efficient as the density increases. The Steiner scheme uses a centralized backbone and has very good results for all densities. As the density increases, the Steiner tree becomes closer to the optimal tree, and energy consumption decreases. MSTEAM is able to provide close results: when $d = 40$, it consumes only 6% more energy than the Steiner scheme in Fig. 6a and 8% more in Fig. 6b. MSTEAM is always better than GMREE: since face routing is not used in such densities, the two reasons are the replication strategy and the MST-based progress estimation, which is better than the one used in GMREE. This is true even in Fig. 6b where the multicast wireless advantage is considered. Overall, this case does not change the relative position of all schemes, and just leads to lower energy consumption.

We finally give results at increasing number of destinations in Fig. 7 ($d = 35$). As expected, the energy consumption of all schemes increases with the number of destinations, since more packets and more paths are generated. The increase is not fully linear because of the common paths used among destinations. However, Steiner scheme and MSTEAM are more scalable than the other ones since their energy consumption increases more slowly. For instance, when changing the number of destinations from 10 to 30, the consumption of MSTEAM in Fig. 7a is multiplied by 1.7 while the one of GMREE is multiplied by 1.9. This difference is obviously visible in both unicast and multicast MAC layers, and is caused by a better path reuse strategy.
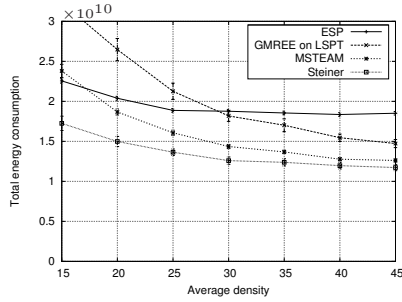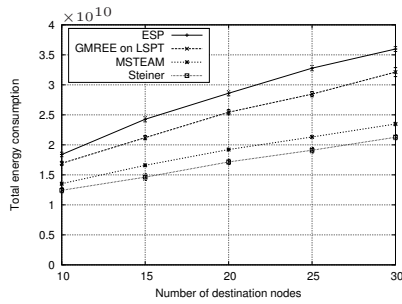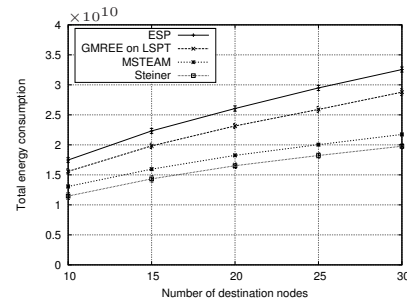
(a) Unicast MAC layer.
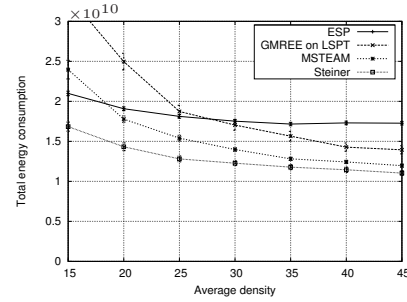
(b) Multicast MAC layer.

Figure 6: Performance of all selected schemes at increasing density (10 destination nodes).

(a) Unicast MAC layer.

(b) Multicast MAC layer.

Figure 7: Performance of all selected schemes at increasing number of destinations ($d = 35$).

7   Conclusion and future work

We have presented MSTEAM, a geographic localized multicast scheme, that uses an MST as an approximation of the optimal backbone. Besides guaranteeing delivery, our protocol is well-suited to ad hoc and sensor networks because it is localized and has low time complexity. We experimentally demonstrated that MSTEAM is very energy-efficient, even compared to a centralized scheme, and outperforms the best existing localized multicast protocol.

We plan to improve MSTEAM using a Steiner tree approximation. We focused on MST because it is a reasonable approximation of an optimal tree. Moreover, its computation has a very low time complexity, while a Steiner tree introduces more complexity. However, under the exponential path loss model used for our experiments, calculating a Steiner tree is possible and one can expect better energy savings. Of course, a trade-off is then needed.

References

[1] P. Bose, P. Morin, I. Stojmenović, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. *ACM Wireless Networks*, 7(6), Nov. 2001.

[2] K. Chen and K. Nahrstedt. Effective location-guided overlay multicast in mobile ad hoc networks. *International Journal of Wireless and Mobile Computing*, 3, 2005. Special issue on Group Communications in Ad Hoc Networks.

[3] H. Frey, F. Ingelrest, and D. Simplot-Ryl. Localized minimum spanning tree based multicast routing with energy-efficient guaranteed delivery in ad hoc and sensor networks. Technical Report RT-0337, INRIA, 2007.

[4] K. Gabriel and R. Sokal. A new statistical approach to geographic variation analysis. *Systemic Zoology*, 18(3), Sept. 1969.

[5] L. Kou, G. Markowsky, and L. Berman. A fast algorithm for Steiner trees. *Acta Informatica*, 15(2), June 1981.

[6] J. Kuruvila, A. Nayak, and I. Stojmenović. Progress based localized power and cost aware routing algorithms for ad hoc and sensor wireless networks. *International Journal of Distributed Sensor Networks*, 2(2), June 2006.

[7] M. Mauve, H. Füßler, J. Widmer, and T. Lang. Position-based multicast routing for mobile ad-hoc networks. Technical Report TR-03-004, Department of Computer Science, University of Mannheim, Germany, 2003.

[8] J. Sanchez, P. Ruiz, X. Liu, and I. Stojmenović. GMR: Geographic multicast routing for wireless sensor networks. In *Proc. of the IEEE Communications Society Conference on Sensor, Mesh, and Ad Hoc Communications and Networks (SECON)*, Sept. 2006.

[9] J. Sanchez, P. Ruiz, and I. Stojmenović. Energy efficient geographic multicast routing for sensor and actuator networks. *Computer Communications*, 30, 2007. Special issue on Sensor-Actuator Networks.

[10] S. Wang, D. Wei, and S. Kuo. SPT-based topology algorithm for constructing power efficient wireless ad hoc networks. In *Proc. of the ACM Int. World Wide Web Conference*, May 2004.

[11] J. Wieselthier, G. Nguyen, and A. Ephremides. Energy-efficient broadcast and multicast trees in wireless networks. *Mobile Networks and Applications*, 7(6), Dec. 2002.