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

Hannes Frey, François Ingelrest, David Simplot-Ryl

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

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

Hannes Frey — François Ingelrest — David Simplot-Ryl

N° 0337

Juin 2007

Thème COM

Rapport technique

ISSN 0249-0803 ISRN INRIA/RT-0337--FR+ENG

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

Hannes Frey* , François Ingelrest† , David Simplot-Ryl‡

Thème COM — Systèmes communicants
Projets POPS

Rapport technique n° 0337 — Juin 2007 — 18 pages

Abstract: We present a minimum spanning tree based energy aware multicast protocol (MSTEAM), which is a localized geographic multicast routing scheme designed for ad hoc and sensor networks. It uses locally-built minimum spanning trees (MST) as an efficient approximation of the optimal multicasting backbone. Using a MST is highly relevant in the context of dynamic wireless networks since its computation has a low time complexity ($O(n \log n)$). Moreover, our protocol is fully localized and requires nodes to gather information only on 1-hop neighbors, which is common assumption in existing work. In MSTEAM, a message split occurs when the MST over the current node and the set of destinations has multiple edges originated at the current node. Destinations spanned by each of 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 localized scheme may lead the message to a void area (i.e., there is no neighbor providing positive progress toward the destinations), we also propose a completely new multicast generalization of the well-know face recovery mechanism. We provide a theoretical analysis proving that MSTEAM is loop-free and always achieves delivery of the multicast message, as long as a path exists between the source node and the destinations. Our experimental results demonstrate that MSTEAM is highly energy-efficient, outperforms the best existing localized multicast scheme and is almost as efficient as a centralized scheme in high densities.

Key-words: Ad Hoc Networks, Sensor Networks, Energy Efficiency, Multicast Routing, Localized Protocol.

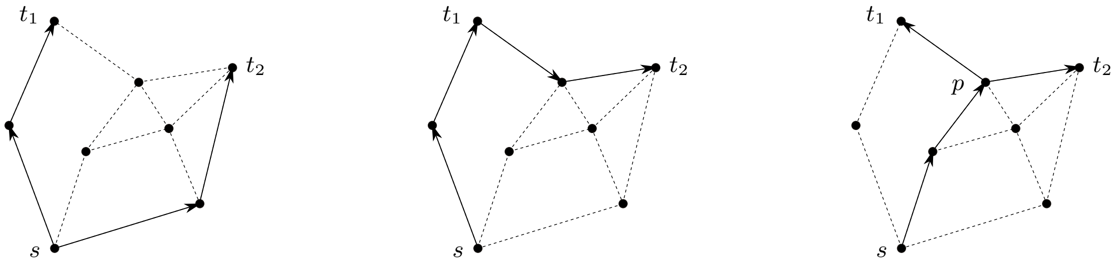
* IMADA, Syddansk Universitet, DK-5230 Odense M, Denmark. Email: frey@imada.sdu.dk
† LCAV, I&C School, EPFL, CH-1015 Lausanne, Switzerland. Email: Francois.Ingelrest@epfl.ch
‡ IRCICA/LIFL, Univ. Lille 1, CNRS UMR 8022, INRIA Futurs, France. Email: David.Simplot@lifl.fr

Unité de recherche INRIA Futurs
Parc Club Orsay Université, ZAC des Vignes,
4, rue Jacques Monod, 91893 ORSAY Cedex (France)
Téléphone : +33 1 72 92 59 00 — Télécopie : +33 1 60 19 66 08

Routage multicast localisé à base d'arbres couvrants minimums avec acheminement économique garanti pour réseaux ad hoc et de capteurs

Résumé : Nous présentons un protocole de routage multicast économique d'un point de vue énergétique et basé sur le calcul d'arbres couvrants minimums (MSTEAM). Il s'agit plus précisément d'un protocole localisé de routage géographique conçu spécifiquement pour les réseaux ad hoc et de capteurs. Il utilise des arbres couvrants minimums (MST) calculés de manière locale afin d'obtenir une approximation de l'arbre optimal de routage multicast. L'utilisation de MSTs est particulièrement adaptée dans le contexte des réseaux dynamiques sans fil car sa complexité de calcul est très basse ($O(n \log n)$). De plus, notre protocole est entièrement localisé et ne nécessite qu'une connaissance à 1 saut à chaque nœud, cette hypothèse étant extrêmement répandues dans la littérature. Dans MSTEAM, une scission du message est effectuée quand le MST reliant le nœud courant à l'ensemble des destinations possède plusieurs arêtes enracinées au nœud courant. Chaque sous-ensemble de destinations forme alors un groupe pour lequel le meilleur voisin est choisi comme prochain saut. Cette sélection s'effectue à partir d'une métrique de type coût sur progrès, où le progrès est approximé en soustrayant le poids du MST reliant le voisin considéré et le sous-ensemble de destinations au poids du MST reliant le nœud courant à ce sous-ensemble. Puisqu'une telle heuristique gloutonne est susceptible d'amener le message dans un cul-de-sac (un nœud qui ne possède aucun voisin offrant un progrès positif vers les destinations), nous proposons également une généralisation originale du protocole *face*, un mécanisme permettant de se sortir de telles situations. Nous fournissons une analyse théorique prouvant que MSTEAM ne génère pas de circuit et parvient toujours à acheminer le message de multicast, tant qu'il existe effectivement une route entre la source et l'ensemble de destinations. Nos résultats expérimentaux démontrent que MSTEAM est très efficace d'un point de vue énergétique, surclasse le meilleur protocole localisé existant, et est presque aussi efficace qu'un algorithme centralisé dans les hautes densités.

Mots-clés : Réseaux ad hoc, Réseaux de capteurs, Économie d'énergie, Routage multicast, Protocole localisé.



(a) The message is split at the source node. (b) The message is never split. (c) The message is split at the end of a common path.

Figure 1: Some message splitting strategies for geographic multicast routing: s is the source node and $T = \{t_1, t_2\}$ the set of destinations.

1 Introduction and motivation

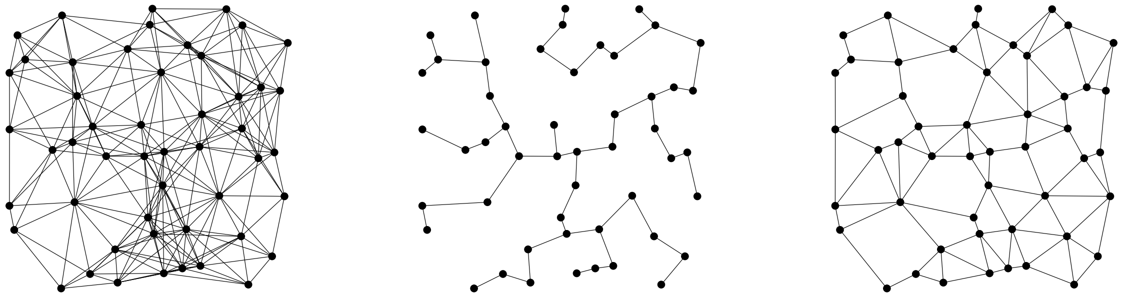
1.1 Context

Telecommunication technologies are becoming more and more important in everyone's life as well as in the industrial world. It is envisioned that the world of tomorrow will be based on the "Internet of Things", because of the advanced integration of electronic chipsets into the thousands of objects composing our world [1]. This will be possible thanks to a new generation of wireless networks: dynamic, mobile and self-organized. Wireless ad hoc and sensor networks are an important step toward this future, as both of them show these characteristics.

In such networks, because of the path loss of radio communications, only close hosts may directly communicate to each other. Long-distance communications require the messages to be forwarded by multiple intermediate nodes, from the source to the destination. Localized message forwarding is a resource-efficient communication paradigm which is well tailored to such decentralized networks. In localized routing schemes, each intermediate node is only expected to maintain knowledge about spatially nearby network nodes. Thus, unlike in centralized schemes, a change in the network requires only a local message exchange with those nodes which are immediately affected by that change. In particular, in sensor network scenarios with thousands or even ten thousands of sensor nodes, localized communication appears to be a promising approach. Because of limited battery capacity, routing must be done in an energy-efficient manner (e.g., by minimizing the global energy consumption) in order to maximize the network lifetime.

In this paper, we are especially interested in *geographic* routing, which requires each node to be able to determine its own location. In this category, the local choice of the next forwarder is based on the position of the neighbors with respect to the destination. Almost all geographic schemes are based on greedy heuristics: an intermediate node will choose as the next hop its best neighbor according to a given evaluation function. For instance, a simple behavior may be to select the closest neighbor to the destination. Due to the nature of such heuristics, the message may arrive at a void area (i.e., there is no neighbor closer to the destination than the current node) and routing may fail. To guarantee delivery, an additional scheme, named face routing [2,9,13,12], is generally used to escape from those void areas. In the simplest setting, greedy routing is used again as soon as the message arrives at a node closer to the destination than the one where face routing was started. This is repeatedly done until the destination is reached. Such a combination of greedy and face routing is simply called Greedy-Face-Greedy.

Multicast routing is a generalization of unicast routing (message forwarding toward a single destination), where a message is to be delivered from a given 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 split into different packets. A first solution, depicted in Fig. 1(a), may be to split the message at the source node into k packets, k being the number of destinations, and then to separately route each packet toward a destination t_i. A second solution, depicted in Fig. 1(b), is to never split the message and to route it along a circuit among all destinations (in Fig. 1(b), the message is sent from s to t_1 and then from t_1 to t_2). Finally, a last and generally better solution, depicted in Fig. 1(c), is to route the message using a common path among the destination nodes, and then to split the message at the end of this path. Of course, the whole difficulty lies in determining the best common path by using only local information at each hop. Moreover, even under the assumption of global knowledge, the problem of computing such an optimal multicast tree is actually a NP-complete problem [17].



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

Figure 2: A unit disk graph $G = (V, E)$ and two subgraphs.

1.2 Contribution

In this paper, we present a *Minimum Spanning Tree based Energy Aware Multicast* scheme (MSTEAM in short, pronounced "em-steam"), which is a generalization for the multicast case of the cost over progress framework described in [14]. In a cost over progress scheme, greedy routing is done by selecting at each hop the neighbor providing the lowest ratio between the cost needed to transmit the message to this neighbor and the progress toward the destination provided by this neighbor.

In this work, we use a minimum spanning tree (MST) as a multicast backbone in order decide when a message has to be split into multiple packets addressing each a destination subset. In addition, we use a 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. Since this scheme is just a greedy heuristic, the message might get stuck at nodes having no better neighbor. Thus, MSTEAM also makes use of a new multicast generalization of the face recovery mechanism which is described for the first time in this work to the best of our knowledge.

One of the key aspects of our solution is that it highly fits wireless ad hoc and sensor networks since it is fully localized. Indeed, forwarding nodes need to construct local MST's using only information on their 1-hop neighborhood, which may be obtained thanks to simple beacon messages. MSTEAM is also well-suited for constrained mobile devices, since a MST may be efficiently computed in time $O(n \log n)$. Moreover, our scheme is loop-free and always achieves delivery, as long as a path exists between the source node and the destinations. We provide a theoretical analysis proving this assertion, as well as some experimental results which demonstrate that MSTEAM is very energy-efficient and outperforms existing geographic multicast schemes.

The remainder of this paper is organized as follows. In the next section, we give the needed preliminaries, while Sec. 3 proposes a literature review of the related work. In Sec. 4, we provide a discussion about the goodness of using minimum spanning trees for multicast routing, and then we present the key components of our scheme. In Sec. 5, we give a detailed description of MSTEAM and a formal proof of its correctness. We then provide significant experimental results about the efficiency of our scheme in Sec. 6, and we finally conclude in Sec. 7.

2 Preliminaries

2.1 Network model

We model a wireless network by a graph $G = (V, E)$, where V is the set of vertices (the network nodes) and $E \subseteq V^2$ the set of edges that gives the available communications: there exists a pair $(u, v) \in E$ if the node u is physically able to communicate with v. The neighborhood set $\mathrm{N}(u)$ of a node u is defined as:

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

The density d of the network is the average value of neighbors per node. The construction of the set of edges E depends on the considered underlying physical model. The most well-known one is the *unit disk graph* model. Given a set of nodes V and a maximum communication range R, E is defined as:

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

$|uv|$ being the Euclidean distance between nodes u and v. Fig. 2(a) provides an example of a unit disk graph. For a given multicast task, the set of destinations is denoted as $T = \{t_1, \ldots, t_k\}$. We assume that nodes are able to adjust their transmitting power, i.e., sending a message from a node u to a neighbor v is done by using the smallest possible power for that. We also assume that nodes regularly collect 1-hop neighborhood information by using beacon messages. This is a fairly common assumption in literature.

2.2 MAC layer model

Whenever a message is split during a multicast, the current forwarding node has to send the packet to more than one immediate neighbor. In this work, we consider two simplified MAC layer models. In the *unicast MAC* layer, sending a message to j next hop neighbors is performed by j independent reliable unicast transmissions. In the *multicast MAC* layer, sending a message is done by only one single transmission. In other words, the latter refers to a MAC layer implementation which exploits the broadcast capabilities of the wireless communication media. A detailed investigation on how such 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 a MST if its weight $|\Delta(u_1, \ldots, u_n)|$ is minimal. At this, the weight of the tree denotes the sum of the weight over all tree edges. In a Euclidean MST, illustrated by Fig. 2(b), 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 this 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 somewhat similar to the MST problem: the goal is to construct a tree $\Gamma(u_1, \ldots, u_n)$ with minimal weight, while allowing the insertion of additional intermediate vertices (called *Steiner points*) in order to reduce the weight of the resulting spanning tree. This problem is known to be NP-complete [10].

A planar graph is a graph in which no edges intersect[1]. *Gabriel graph* construction is a prominent localized construction method which is based on a geometric concept which was introduced by Gabriel and Sokal in [8]. Starting from a unit disk graph $G = (V, E)$, each edge $(u, v) \in E$ is considered and 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 very interesting for decentralized networks since this removal strategy may be applied independently by each node, and does not require any message exchange. Refer to Fig. 2(c) for an illustration of a Gabriel graph constructed over an entire network.

3 Related Work

Limited on-board power supply – in particular in sensor networks scenarios – drives current research on localized communication protocols. Such protocols are generally scalable and resource-efficient due to the minimum amount of control overhead they need. In recent years, many localized unicast algorithms, being a combination of greedy and face routing, have been considered [5, 6].

The pioneering work especially in localized multicast routing can be found in [15] which describes the *Position-Based Multicast* (PBM) protocol, the first fully localized operating multicast scheme. Multicast forwarding is performed by determining the neighbor subset which maximizes a weighted sum over two conflicting objectives: maximizing the number of next hop nodes and minimizing the remaining overall distance from the next hop nodes toward the destination nodes. The impact of the two objectives is controlled by a parameter $\lambda \in [0, 1]$. An early packet split and thus individual message transmissions toward each destination node is achieved by a λ close to 0. Less frequent packet splits and thus longer paths keeping several multicast destination in one message are obtained with a λ close to 1. Without any additional provision the algorithm requires testing each possible subset and selecting the one which maximizes the objective function. Thus, the complexity of this scheme is $O(2^m)$ for m neighbor nodes.

This scheme, being greedy in nature, can only consider nodes which provide positive progress toward the destination nodes. Thus, recovery from greedy routing failures becomes an issue in PBM. Two variants of face recovery are considered. In the plain setting, face recovery computes an average point p over the destination

[1]Strictly speaking a graph is planar if it has any planar embedding, where no edges are intersecting. Within localized routing protocols, however, planarity refers to the property that no edges are intersecting in the natural embedding where each graph node is placed at the device position

nodes and starts traversal of the face intersected by the straight line *sp*. As soon as at least one destination node can be handled in greedy mode again the message is split into a part handled in greedy mode again and the remaining part continuing with face traversal. A proof of correctness of this scheme was not provided so far and in fact even under the good natured Gabriel graph construction a routing loop can be constructed. In order to avoid greedy and face messages to travel independently into the same direction, PBM includes an optional combination of greedy and face routing. In this setting, face and greedy messages are handled in one message as long as the next face traversal node provides progress for the greedy routing node.

The multicast routing protocol described in [3] performs message forwarding along a multicast backbone which is defined by constructing a spanning tree over the multicast start node and the message destinations. Three spanning tree heuristics are considered and compared to each other.

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 multicast message consists exactly of these destination nodes which are reachable over this tree edge. This is comparable to the message splitting strategy described in this work. However, the multicast mechanism described in [3] differs from MSTEAM in the following ways: First, any multicast message is forced to follow a backbone edge st until it eventually arrives at the destination node. Only when arrived at this node the message might be split again. This is in contrast to MSTEAM where premature message splits are possible at any node. Second, in contrast to MSTEAM, all destination nodes reachable over t in the multicast backbone will be disconnected whenever node t is not reachable from the multicast start node. Third, multicast routing described in [3] is only concerned with localized construction of multicast overlays. Routing a message along a multicast overlay edge uv is done in a centralized way by just calculating the shortest path from u to v in terms of hop count. In this work, we are describing a localized metric in order to select an energy-efficient next hop node by using information about the destinations and the hop neighbors only. Fourth, with global information the shortest path can always be found. Thus, recovery from greedy routing failures is actually not an issue in [3]. Moreover, PBM is calculating the shortest path by using global knowledge which makes the definition of a localized greedy next hop selection function obsolete. Finally, energy consumption due to exponential path loss is not considered in that work. The empirical studies consider the shortest path in terms of hop count.

A recently proposed localized multicast routing scheme, *Geographic Multicast Routing* (GMR), can be found in [18]. The multicast algorithm is based on the cost over progress framework [14] and, opposed to PBM, does not require setting a proper network-depending parameter λ. In the unicast case, cost over progress denotes the relation between cost produced in the next hop and the progress achieved by the next hop node. The multicast extension of this framework minimizes the number of selected next hop nodes over the progress achieved by this selected set. Progress is measured as the difference between the sum over all individual distances between current forwarding node and destinations, and the sum over distances of each next hop node and the destinations covered by this node. A node v is said to *cover* a destination if this destination is closest to v compared to all other next hop nodes. In contrast to [15], 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 is the number of neighbor nodes.

Since this scheme is greedy as well, only node sets providing positive progress are considered. Consequently, a message can get caught at a network void and greedy recovery becomes an issue. The scheme describes a plain face recovery strategy which applies traditional unicast face traversal for each destination node individually. In order to save communication bandwidth, however, face messages traveling the same face are aggregated into one single message.

All the revised schemes consider minimizing hop count as the optimization criteria. When the communication hardware provides signal strength adaptation, however, a single transmission over a large distance can significantly exceed the power consumption of many small distance transmissions over several intermediate hops. The first localized multicast scheme which considers energy-efficient multicast tree construction is described in [19]. This scheme, named GMREE, is basically an extension of GMR considering the cost of the total energy consumption of the next transmission over the progress provided by the next hop nodes. Progress is computed in the same way as it is done in GMR.

Overall, the current literature covers well energy-efficient unicast routing, while the issue of multicasting starts only to be addressed, like in [19], leaving room for further developments and improvements.

4 MST based Multicasting

In this section, we first provide a motivation on the goodness of using Euclidean MST's in order to locally decide a message split and to determine the best next hop node with respect to a given set of destinations. There follows a description of how localized path segmentation and message forwarding may be performed according to a Euclidean MST backbone. A new face recovery mechanism developed for the multicast case is then described. Unless otherwise specified, MST always denotes Euclidean MST in the following.

4.1 A Motivation for using MST based Multicast Backbones

Whenever global information is available, finding the multicast tree with minimum cost is possible, though, being a NP-complete problem [17]. Our goal in this work is to define a localized heuristic which aims at finding a multicast tree with a "low" cost. Quantifying the latter in terms of a formal analysis is beyond the scope of this work, but we will show empirically that the cost obtained thanks to the described heuristic does not significantly depart from an efficient centralized solution.

From our point of view, the quality of a localized multicast algorithm depends on two factors:

- The message splitting strategy, which should aim at message forwarding along a cost effective multicast backbone.
- The next hop selection criterion, which should aim at cost-effective message forwarding along this multicast backbone.

For instance, the multicast backbone in Fig. 1(a) is the tree consisting of edges st_1 and st_2. The backbone in Fig. 1(b) is the tree consisting of edges st_1 and t_1t_2. Finally, the backbone in Fig. 1(c) utilizes some point p in between s, t_1, and t_2 and consists of the edges sp, pt_1, and pt_2.

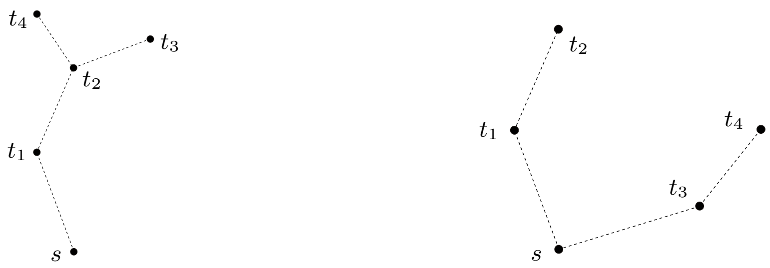
Cost-effective message forwarding along an edge of the multicast backbone will select the "best" neighbor with respect to the metric being applied (e.g., based on hop count, Euclidean distance, energy consumption) and the destination nodes which are reachable along this backbone edge. For instance, in Fig. 1(a) next hop selection along the multicast backbone edge st_1 will consider the best node with respect to the metric and the destination node t_1. Next hop selection along the multicast backbone edge sp in Fig. 1(c) will consider the best node with respect to the metric and the destination nodes t_1 and t_2.

Suppose s being the source node and $T = \{t_1, \ldots, t_k\}$ being the message destinations. Furthermore, let $\mathrm{C}(u, v)$ 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 $\mathrm{C}(u, v)$ as the cost function, defines the cost optimal multicast backbone. In this work we do not assume that a node is able to request all network nodes for computing 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 a cost optimal multicast backbone is not possible in this general multicast setting. As an approximation we utilize the concept of weighted MST instead which may be efficiently computed even by constrained devices. Since the exact energy model is not known, we approximate the routing cost by the simplified assumption that $|uv| < |uw|$ always implies $\mathrm{C}(u, v) < \mathrm{C}(u, w)$. In this case, the weighted MST is equivalent to the Euclidean MST, for which only locations of nodes are needed.

Under the multicast MAC assumption, energy savings are possible at nodes where the message is split. At this point, any set of next hop nodes might produce the same routing cost as addressing a single next hop node. In a small scale multicast, it might thus be more cost-efficient to perform a single direct "large" broadcast transmission instead of performing many "small" transmissions in order to reach the destinations. In a large scale multicast, however, we expect that the cost savings which are possible at the nodes where the message is split will be outweighed by the routing costs which are required in order to route the message between those split points. Thus, we use the same MST approximation even under the multicast MAC assumption.

4.2 Message splitting strategy

The following describes the rule which is performed at each forwarding node in order to decide if a message has to be kept as one single packet or if it has to be split and sent toward different destination subsets. Let s be the current forwarding node and let $T_i \subseteq T$ be the set of multicast destinations which have to be handled by s. Node s has to calculate the MST $\Delta(\{s\} \cup T_i)$ over itself and T_i. This tree provides the multicast backbone which is to be used to reach all destination nodes in T_i from s. The message thus has to be routed along the edges of this tree, and must be split at node s if multiple paths start from this node. Actually, each of these



(a) All destinations are kept in the same set.

(b) Two subsets $\{t_1, t_2\}$ and $\{t_3, t_4\}$ are created.

Figure 3: The message splitting strategy used by MSTEAM: $\Delta(s, t_1, t_2, t_3, t_4)$ is used to split the message at node s.

paths is represented by an edge which originates at node s and spans a subset of destination nodes. These are forming exactly a destination subset to which s has to send an individual message copy.

This strategy is illustrated by Fig. 3, where node s has to handle the destination nodes t_1, t_2, t_3 and t_4. In Fig. 3(a), the resulting 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 will not be split and will have to be routed along the edge (s, t_1). In Fig. 3(b), there are two edges originated at node s: the first one spans t_1 and t_2, while the second one spans t_3 and t_4. The message will thus be split into two packets. The first one will be routed along the edge (s, t_1) toward the set of destinations $\{t_1, t_2\}$, and the second one will be routed along (s, t_3) toward the set of destinations $\{t_3, t_4\}$.

4.3 Energy-efficient metric

In MSTEAM, greedy routing is done thanks to a generalization of the cost over progress framework described in [14] for the multicast case. In this framework, as it is implied by its name, 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 which was provided by the message splitting strategy. The neighbor for which the cost over progress ratio is minimum is simply chosen as the next hop for T_i.

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

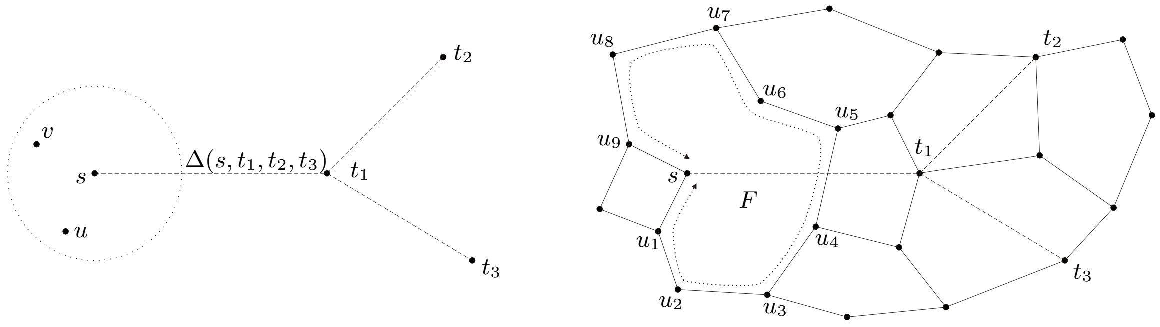
The cost of sending a message from the current node u to its neighbor v requires a specific amount of energy denoted as $\mathrm{f}(u, v)$. Thus, the cost over progress ratio $\mathrm{Q}(u, v, T)$ at node u of a neighbor node v is equal to:

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

Note, the expression of $\mathrm{Q}(u, v, T)$ is in fact a generalization of the unicast routing metric $\mathrm{f}(u, v)/(|ut| - |vt|)$ described in [14]. In this connection, u is the current node, v the next hop candidate, and t the unicast destination node. When having only one destination node $T = \{t\}$, the expression $\mathrm{Q}(u, v, T)$ actually reduces to this formula since a MST over a pair of nodes is simply the straight line connecting them.

4.4 Recovery strategy

As previously stated, the message may arrive at a void area, i.e., the set of possible next hop nodes might be empty for a given destination subset. For instance, suppose that in Fig. 4(a) node s has to send a multicast message toward the destination set $\{t_1, t_2, t_3\}$. All destinations are connected over the link (s, t_1) of $\Delta(s, t_1, t_2, t_3)$. However, node s may not select any of its two neighbors u and v as the next hop, since they both satisfy $|\Delta(u, t_1, t_2, t_3)|, |\Delta(v, t_1, t_2, t_3)| > |\Delta(s, t_1, t_2, t_3)|$. Without any further provision, multicast routing



(a) Node s is concave with respect to $\{t_1, t_2, t_3\}$.

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

Figure 4: Minimum spanning tree based face multicast routing is used in order to recover from concave nodes.

toward these destination nodes will be stopped at s. This happens independently whether any of these destination nodes are reachable or not reachable from s. In accordance to the notion for unicast greedy algorithms, we denote such a node as *concave* with respect to the destination subset.

Face routing is a well-known unicast routing scheme which can be used in order to handle greedy routing failures for each destination individually. In the following we describe for the first time a multicast extension of face routing which can handle all multicast destinations at once. Similar to unicast face routing, the multicast scheme requires a localized topology control mechanism which transforms the underlying wireless network into a planar graph. In this work, we employ the previously described common Gabriel graph construction which requires the wireless network to comply with the unit disk graph model.

As depicted in Fig. 4(b), a planar graph partitions the plane into faces which can be traversed in a localized way by employing the left/right hand rule; a receiver 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. 4(b), the face F will be traversed along the path $su_1u_2 \ldots u_9$ when using the right hand rule and along the path $su_9u_8 \ldots u_1$ when using the left hand rule.

Unicast face recovery has been described in different variants. In this work we employ the variant which transmits the message along the sequence of faces which are intersected by the straight line st connecting source node s and destination node t. Whenever the message arrives at a node which is closer to t than the start node s, face recovery is switched back into greedy mode again. Note that under the Gabriel graph assumption, this unicast face recovery mechanism simplifies to traversing the very first face only [7].

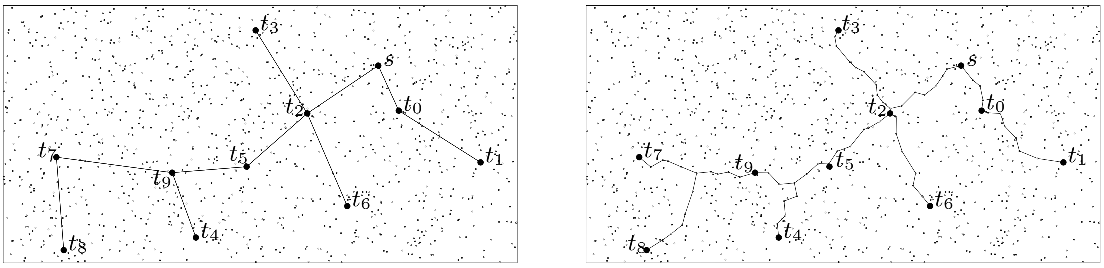
The idea of multicast face recovery is as follows. Suppose that the current forwarding node s has computed a destination subset T_i for which no better greedy neighbor node exists. Let st be the edge connecting node s with $\Delta(\{s\} \cup T_i)$. By using any of the two rules – right hand rule or left hand rule – node s starts traversal of the face which is intersected by the outgoing minimum spanning tree edge st. Face traversal continues until the message arrives at a node u which satisfies $|\Delta(\{u\} \cup T_i)| < |\Delta(\{s\} \cup T_i)|$. 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, in order to avoid a message loop, the message is dropped if it is about to be sent again over the first face traversal edge in the same direction.

Refer to Fig. 4(b) for an example. The edge st_1 connects node s with $\Delta(s, t_1, t_2, t_3)$. Since node s is concave with respect to $\{t_1, t_2, t_3\}$, it will start traversal of face F, i.e., the face which is intersected by st_1. Assuming the right hand rule, face traversal will visit the nodes u_1, u_2, and u_3. Since node 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 will handle the destination subset $\{t_1, t_2, t_3\}$ in greedy mode again.

5 The MSTEAM protocol

5.1 Description

The protocol may be described as follows. The source node s, which initiates the multicasting task toward the destination set $T = \{t_1, \ldots, t_k\}$, first has to decide if a message split should be performed. It thus computes



(a) The MST $\Delta(\{s\} \cup T)$). (b) The multicast tree produced by MSTEAM.

Figure 5: A sample run of MSTEAM for a set T of 10 destinations and a density $d = 35$.

the MST $\Delta(\{s\} \cup T)$, and groups together all destinations spanned by edges originated at s (refer to Sec. 4.2 and Algorithm 1).

For each subset $T_i \subseteq T$ obtained in this way, s computes a subset $\mathrm{N}_i(s) \subseteq \mathrm{N}(s)$, which contains all neighbors $v \in \mathrm{N}(s)$ such that $|\Delta(\{v\} \cup T_i)| < |\Delta(\{s\} \cup T_i)|$ (these are the neighbors providing positive progress toward T_i). If $\mathrm{N}_i(s)$ is not empty, s computes the cost over progress ratio $\mathrm{Q}(s, v, T_i)$ for each neighbor $v \in \mathrm{N}_i(s)$ (refer to Sec. 4.3). The neighbor providing the best ratio is chosen as the next hop toward T_i. If $\mathrm{N}_i(s)$ is empty, then s is concave with respect to T_i, and face recovery must be used to escape from this void area. Node s thus applies the strategy presented in Sec. 4.4 to select the face node v as the router toward T_i (refer to Algorithm 2). The whole process is repeatedly done until all subsets T_i have been considered.

If unicast MAC is considered, a packet is sent for each subset T_i. Each of them contains the set of destination nodes, the selected router and the mode (greedy or face) that must be used. In the case of face routing, the packet also contains the very first edge traversed by the packet in face mode, and the weight of the MST at the starting node ($|\Delta(\{s\} \cup T_i)|$ in this example). If multicast MAC is considered, all this information is aggregated into the same packet. This means that the latter will contain a list of the selected next hops and for each of them, the set of destinations they have to serve, the mode to use and the additional face information if needed. 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 multicast packet, it has to check if it has been designated as the next hop by the previous transmitter. If not, the packet is simply 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 same process followed by s. In face mode, it checks whether it is closer to the set of 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 and Algorithm 3). Of course, if u was one of the destinations, it removed itself from T_i and stopped the process if the latter became empty.

Fig. 5 illustrates a sample run of MSTEAM over a randomly deployed network. In Fig. 5(a) is given the MST $\Delta(\{s\} \cup T)$, while Fig. 5(b) provides the multicast tree produced by MSTEAM. The MST spanning all destinations was used at the source node s. Since two edges originate at s, the message was split into two packets at this node. The first one was sent toward t_0 and t_1 along the edge (s, t_0), while the second one was sent toward the rest of the destination nodes along the edge (s, t_2). One can observe on Fig. 5(b) that MSTEAM was able to follow these edges in an effective way. One can also observe that the message splitting strategy correctly works by looking at the path followed to reach t_4 and t_9 from the node lying close to t_5. Instead of following the edge (t_5, t_9), MSTEAM routed the message along a common path among t_4 and t_9, and then split it at the end of this path. The same observation applies to nodes t_7, t_8 and t_9.

Regarding the complexity of MSTEAM, a forwarding node in greedy mode will have to compute a MST for the message splitting strategy, which thus has a time complexity in $O(k \log k)$, k being the number of destination nodes. In the worst case, all destination nodes are handled separately. For each of them and for each neighbor, a new MST must be computed. In this case, the complexity in time of MSTEAM is thus $O(mk^2 \log k)$ for the greedy mode, m being the number of neighbor nodes. This complexity may actually be better estimated since a MST has a maximum degree of 6, regardless of the value of k. Since the face mode has a complexity in $O(k \log k)$, the final complexity of MSTEAM in the worst case is $O(mk \log k)$, which is lower than the complexity of GMREE ($O(mk \min(m, k)^3)$, still considering the worst case).

Algorithm 1 Handle message in greedy mode

```
s ← current node
if s is a multicast destination then
   pass message to upper protocol layer
   remove this node from multicast destinations
end if
t_1, ..., t_k ← multicast destinations
T ← Δ(s, t_1, ..., t_k)
for all edges st in T do
   T_i ← destinations reachable over st in T
   S ← Δ({s} ∪ T_i)
   if ∃v ∈ N(s) with |Δ({v} ∪ T_i)| < |S| then
      v ← neighbor which minimizes Q(s, v, T_i)
      greedy forward message to v
   else
      start face recovery
   end if
end for
```

Algorithm 2 Start face recovery

```
s ← current node
N_GG(s) ← Gabriel graph neighbors of s
v ← node in N_GG(s) lying next in cw direction from st
face forward message to v
```

Algorithm 3 Handle message in face mode

```
s ← face traversal start node
e ← face traversal start edge
v ← previous node
u ← current node
t_1, ..., t_k ← multicast destinations
if |Δ(u, t_1, ..., t_k)| < |Δ(s, t_1, ..., t_k)| then
   handle message in greedy mode
else
   N_GG(u) ← Gabriel graph neighbors of u
   w ← node in N_GG(u) lying next in cw direction from uv
   if vw = e then
      drop message
   else
      face forward message to w
   end if
end if
```

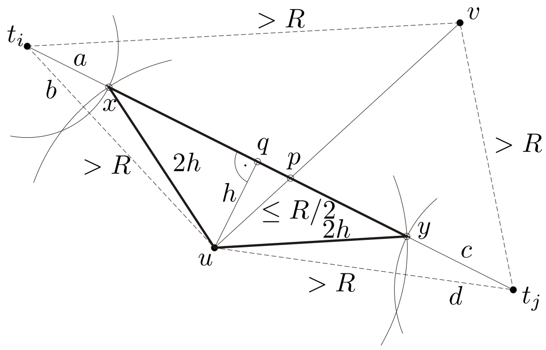


Figure 6: Geometric construction of case *2.b*.

5.2 Correctness of MSTEAM

The following section provides a formal proof of the correctness of MSTEAM in terms of loop-free operation and delivery guarantees. We assume the node set V to be finite and the network to be modeled as a unit disk graph with radius R. Furthermore, we assume that the planar graph used during face recovery results from local Gabriel graph construction. We will first show that under this setting, traversal of the very first face is always sufficient in order to recover from a concave node. Subsequently, we show that a single instance of a multicast message always travels a finite number of hops before it is either dropped, delivered, or split. These two results are finally used in order to show correctness of the described algorithm.

Lemma 1 *Let s be a node where face recovery for the destination nodes $\{t_1, \dots, t_k\}$ was started and let st be the edge connecting s with $\Delta(s, t_1, \dots, t_k)$. If s can reach at least one of the destination nodes, then traversal of the face F which is intersected by st will always arrive at a node u which satisfies $|\Delta(\{u, t_1, \dots, t_k\})| < |\Delta(\{s, t_1, \dots, t_k\})|$.*

Proof. We first assume that at least one of the destination nodes t_i is lying on the face boundary, i.e., there is one face boundary edge uv such that t_i is located on the straight line connecting u and v. Due to the Gabriel graph construction we have that the circle $U(u, v)$ around u and v is empty. It follows that t_i is either node u or node v. WLOG we assume $u = t_i$. We have $|\Delta(u, t_1, \dots, t_k)| = |\Delta(t_1, \dots, t_k)| < |\Delta(t_1, \dots, t_k)| + |st| = |\Delta(t_1, \dots, t_k) \cup st| = |\Delta(s, t_1, \dots, t_k)|$. It follows that at the latest when face traversal visits node u, greedy mode is employed once again. Thus, in the remainder of the proof we can assume that the destination nodes are not located on the face boundary.

When all destination nodes are lying within the traversed face F, then it can be shown that node s cannot reach any of the destination nodes [7]. Thus, at least one of the destination nodes is lying outside of F and it follows that at least one edge of $\Delta(\{s, t_1, \dots, t_k\})$ is intersecting the boundary of F. Let uv be such an intersected boundary edge of F. We consider two different cases.

Case 1, uv is intersected by st: Due to Lemma 1 of [7] we have that at least one of the edge end points of uv is lying closer to t than s. WLOG we assume $|ut| < |st|$. We have that $(\Delta(s, t_1, \dots, t_k) \setminus \{st\}) \cup \{ut\}$ is a spanning tree over $\{u, t_1, \dots, t_k\}$. Thus, it follows: $|\Delta(\{u, t_1, \dots, t_k\})| \leq |(\Delta(s, t_1, \dots, t_k) \setminus \{st\}) \cup \{ut\}| = |\Delta(s, t_1, \dots, t_k)| - |st| + |ut| < |\Delta(\{s, t_1, \dots, t_k\})|$.

Case 2, uv is intersected by any edge t_it_j which is different from st: We consider two subcases.

Case 2.a, at least for one of the nodes u or v and one of the destination nodes t_i or t_j the Euclidean distance is less than $|st|$: WLOG we assume that $|ut_i| < |st|$ is satisfied. Again, $(\Delta(s, t_1, \dots, t_k) \setminus \{st\}) \cup \{ut_i\}$ is a spanning tree over $\{u, t_1, \dots, u_k\}$ and we obtain $|\Delta(\{u, t_1, \dots, t_k\})| \leq |(\Delta(s, t_1, \dots, t_k) \setminus \{st\}) \cup \{ut_i\}| = |\Delta(s, t_1, \dots, t_k)| - |st| + |ut_i| < |\Delta(\{s, t_1, \dots, t_k\})|$.

Case 2.b, the mutual distances between u, v and t_i, t_j satisfy $|ut_i|, |ut_j|, |vt_i|, |vt_j| \geq |st|$: Refer to Fig. 6 for the following geometric construction.

We have that $|\Delta(s, t_1, \dots, t_k) \setminus \{st\}|$ is a spanning tree over $\{t_1, \dots, t_k\}$. It follows, $|\Delta(t_1, \dots, t_k)| \leq |\Delta(s, t_1, \dots, t_k) \setminus \{st\}| = |\Delta(s, t_1, \dots, t_k)| - |\{st\}| < |\Delta(s, t_1, \dots, t_k)|$. Thus, t is not a neighbor of s. Otherwise, face recovery would not have been started. Since s is not connected with t it follows $|st| > R$ and thus also $|ut_i|, |ut_j|, |vt_i|, |vt_j| > R$.

Let p be the intersection point between uv and t_it_j. Nodes u and v are connected by a Gabriel graph edge and are thus also connected in the network graph, i.e., we have $|uv| \leq R$. Thus, the distance of at least one of these nodes toward p is not greater than $R/2$. WLOG we assume that $|up| \le R/2$ is satisfied. Let q be the perpendicular of u with respect to $t_i t_j$ and let $h = |uq|$.

For $h = 0$ we have that u is located on the straight line connecting t_i and t_j. Thus, we have $|uq| + |t_i t_j| = |t_i t_j| = |t_i u| + |u t_j|$.

For $h > 0$ we consider first the triangle connecting $\{u, x, y\}$, while the points x and y are lying on $t_i t_j$ and satisfying $|ux|, |uy| = 2h$. Since (u, q, x) describes a perpendicular triangle, we have $(2h)^2 = h^2 + |qx|^2$ which implies $|qx| = \sqrt{3}h$. Due to symmetry it also holds $|qy| = \sqrt{3}h$. It follows $|uq| + |xy| = (2\sqrt{3} + 1)h > 4h = |ux| + |uy|$. Since $h \le R/2$ we have $|ux| = 2h \le R < |ut_i|$. Thus, there exist $a \ge b \ge 0$ with $|qt_i| = |qx| + a$ and $|ut_i| = |ux| + b$. Due to symmetry there also exists $c \ge d \ge 0$ with $|qt_j| = |qy| + c$ and $|ut_j| = |uy| + d$. This finally yields $|uq| + |t_i t_j| = |uq| + |xy| + a + c > |ux| + |uy| + a + c \ge |ux| + |uy| + b + d = |ut_i| + |ut_j|$.

Thus, for either $h = 0$ or $h > 0$ the inequality $|ut_i| + |ut_j| \le |uq| + |t_i t_j|$ holds. We apply this inequality in order to estimate $|\Delta(u, t_1, \ldots, t_k)|$. Since $(\Delta(s, t_1, \ldots, t_k) \setminus \{st, t_i t_j\}) \cup \{ut_i, ut_j\}$ is a spanning tree over $\{u, t_1, \ldots, t_k\}$ we have $|\Delta(u, t_1, \ldots, t_k)| \le |(\Delta(s, t_1, \ldots, t_k) \setminus \{st, t_i t_j\}) \cup \{ut_i, ut_j\}| = |\Delta(s, t_1, \ldots, t_k)| - (|st| + |t_i t_j|) + (|ut_i| + |ut_j|) < |\Delta(s, t_1, \ldots, t_k)| - (|st| + |t_i t_j|) + (|uq| + |t_i t_j|) = |\Delta(s, t_1, \ldots, t_k)| + |uq| - |st|$. Finally, we have $|uq| \le R/2 < |st|$ which implies $|\Delta(u, t_1, \ldots, t_k)| < |\Delta(s, t_1, \ldots, t_k)|$. ∎

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

Proof. Starting in greedy mode the described forwarding mechanism handles the message instance according to an alternating sequence of greedy and face recovery forwarding steps. It is thus sufficient to show the following two invariants: First, in both greedy and face recovery mode, the message is handled only in a finite number of forwarding steps. Second, there are only a finite number of switches between greedy and face recovery mode until the message is either delivered, dropped, or split.

To ease the representation, let $\Delta(v_i)$ denote $\Delta(v_i, t_1, \ldots, t_k)$ in the following. Whenever the message instance is handled in greedy mode, it will only traverse edges uv which satisfy $|\Delta(u)| > |\Delta(v)|$. Thus, during greedy mode the message visits each network node at most once. Since the number of network nodes is finite, it follows that a message in greedy mode will after a finite number of forwarding steps either be delivered, split, or changed into face recovery mode.

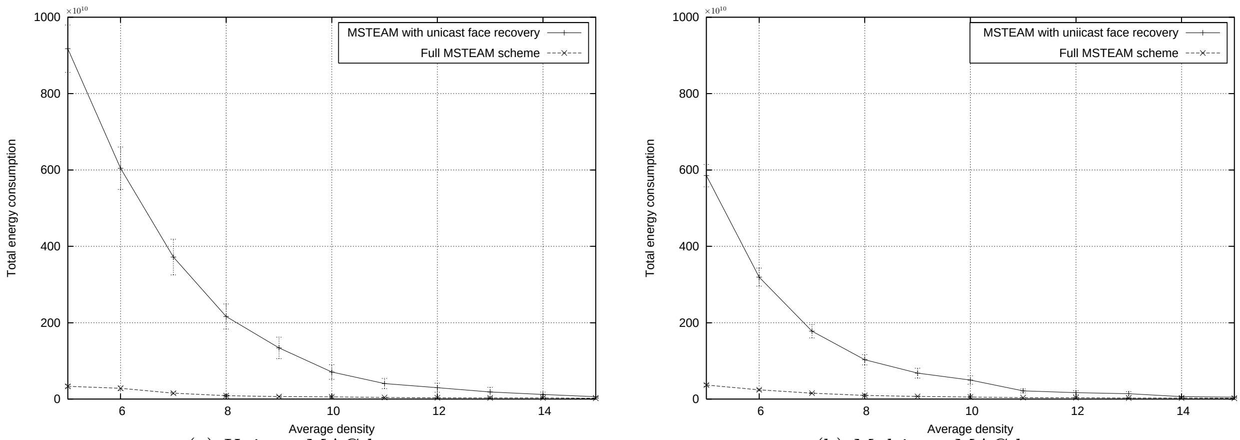
Whenever the message instance is handled in face recovery mode, it traverses a single face only. Since there are a finite number of network nodes, the face also consist of a finite number of boundary nodes. The message will be dropped when it visits the first face traversal edge in the same direction twice. It follows that after a finite number of forwarding steps the message will either be dropped or handled in greedy mode again.

Finally, assume for the sake of contradiction that the message is infinitely often switched between greedy and face recovery mode. Let $u_1, u_2, \ldots$ be the nodes where the message is switched into greedy mode, and let $v_1, v_2, \ldots$ be the nodes where the message is switched into face recovery mode. I.e., in node u_1 the message is starting in greedy mode, in node v_1 it is switched into face recovery mode, in node u_2 it returns into greedy mode, and so on. It follows, $|\Delta(u_1)| \ge |\Delta(v_1)| > |\Delta(u_2)| \ge |\Delta(v_2)| > \ldots$. Consequently, in any network node switching from face into greedy mode is performed at most once. Since the number of network nodes is finite, the number of switches between greedy and face recovery mode is finite as well, a contradiction. ∎

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

Proof. Let s be the multicast originator and let T be the set of all multicast destinations. In the described multicast algorithm a multicast message is either kept or split in a forwarding step and two multicast messages are never merged together. Thus, for each possible subset $S \subseteq T$ at most one instance of a multicast message addressing this set may exist during multicast execution. It follows that the number of possible multicast message instances is finite. Finally, due to Lemma 2 each instance is handled only a finite number of forwarding steps. It follows that the total number of forwarding steps is finite as well, i.e., no routing loop occurred.

Let t be an element of T and suppose that there exists a path from s to t. Suppose that the destination t is dropped during multicast 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 be the subset of destinations handled in face mode. Since the message is dropped, each node v visited during face traversal satisfies $|\Delta(\{v\} \cup S)| \ge |\Delta(\{u\} \cup S)|$. Since node u was reached by s and since s is able to reach node t, we have that u can reach at least one node in S. By Lemma 1 it follows that face traversal will visit a node w which satisfies $|\Delta(\{w\} \cup S)| < |\Delta(\{u\} \cup S)|$, a contradiction. ∎



(a) Unicast MAC layer.

(b) Multicast MAC layer.

Figure 7: Performance of the MST based face recovery mechanism used in MSTEAM (10 destination nodes).

6 Performance evaluation

6.1 Simulation settings

Our experimental results were obtained thanks to a home-made simulation tool, using the unit disk graph model. A detailed investigation of MAC and physical layer impact on our multicast scheme is lying beyond the scope of this work. We schedule this as future research and set the focus on the network layer in this work. In our simulations, nodes have a maximum range $R = 250$ as defined in IEEE 802.11 and are uniformly and randomly deployed over a square area of size 2500×2500. The exact number of nodes n depends on the required density. The 95% confidence interval is given on each figure.

The energy consumption $\mathrm{f}(u, v)$ of a node u transmitting a message to a node v is equal to:

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

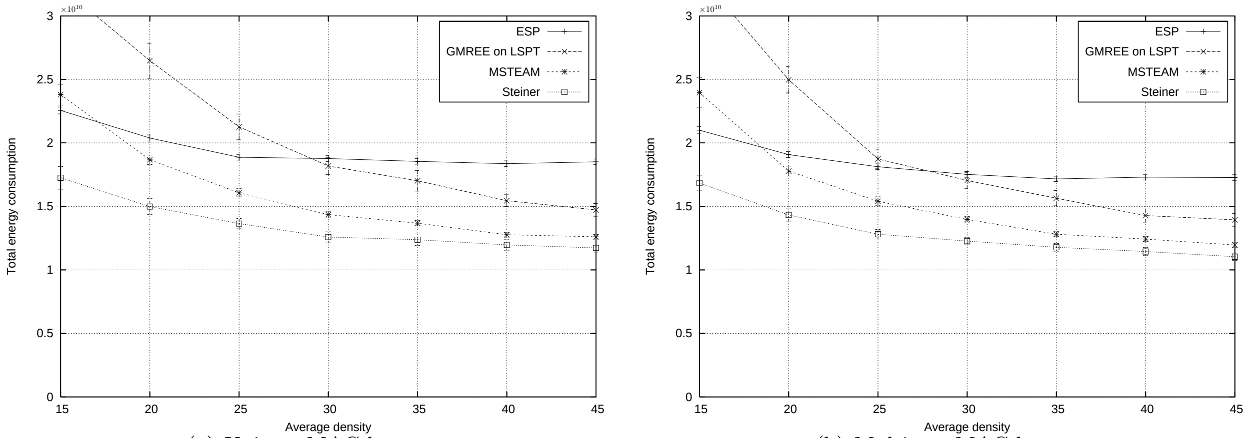
α being the path loss constant: the higher the value of α, the higher the energy needed to cover a given range. The constant c_e accounts for miscellaneous constant costs (e.g., preparation of the packet, MAC layer processing) as stated in [4]. We used the values $\alpha = 4$ and $c_e = 10^8$, derived from [16].

To evaluate MSTEAM, we selected the GMREE scheme [19] since it is the only other geographic energy-efficient localized multicast protocol. Authors of this scheme used an underlying subgraph in their performance evaluation, and experimentally showed that the Local Shortest Path Tree (LSPT) [21] using $\mathrm{f}(u, v)$ as the cost function was the most efficient one. We thus restricted our simulations of GMREE to this subgraph.

GMREE was compared in [19] to a centralized scheme named ESP (Energy-Efficient Shortest Path), which is simply the centralized application of Dijkstra's shortest path tree, using $\mathrm{f}(u, v)$ as the cost function. However, the performance of this scheme highly depends on the considered maximum communication range. There indeed exists an optimal routing radius, and with its centralized knowledge, ESP is able to generate efficient routes where the length of each hop is close to this optimal range. While ESP splits the message into too many packets and wastes a lot of energy, the savings obtained thanks to the efficient routes are higher and provide good results. However, when the maximum communication range is not high enough, routes are no longer sufficiently efficient to counterbalance the wastes coming from message splits. We believe that this problem was overlooked in [19]. We thus introduced another centralized scheme, which computes an approximated weighted Steiner tree using the heuristic given in [11] and uses this tree as the multicast tree. This scheme is simply referred to as Steiner, and $\mathrm{f}(u, v)$ is the cost function.

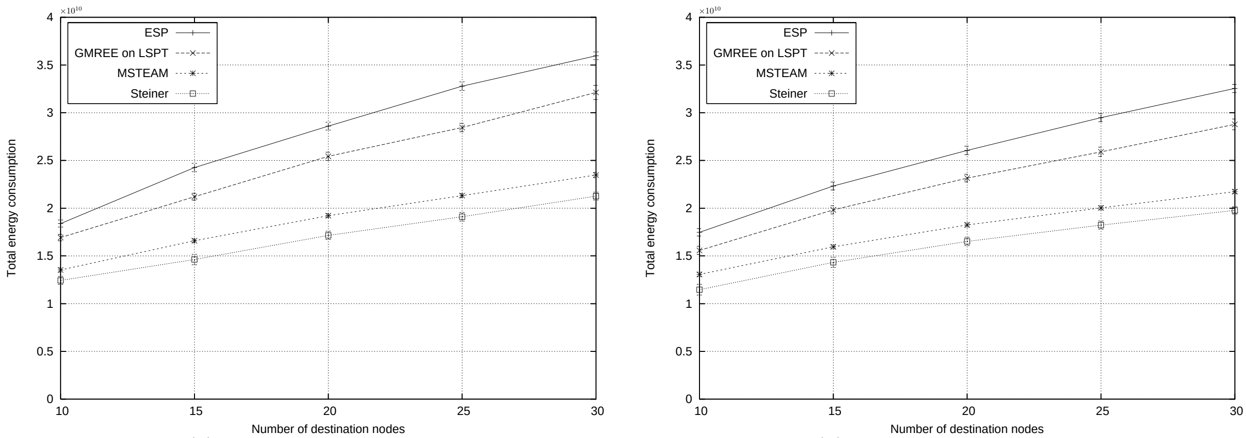
6.2 Experimental results

Fig. 7 illustrates the efficiency of the new MST based face recovery mechanism, compared to the basic unicast one. In the latter, when the message arrives at a void area, it is split into k packets, k being the number of destinations, and each packet is handled separately. Small densities were used, to maximize the number of void areas. One can observe that in particular in small densities, the new scheme is by far superior to the basic



(a) Unicast MAC layer. (b) Multicast MAC layer.

Figure 8: Performance of all selected schemes at increasing density (10 destination nodes).



(a) Unicast MAC layer. (b) Multicast MAC layer.

Figure 9: Performance of all selected schemes at increasing number of destinations ($d = 35$).

	$d = 20$ $\lvert T\rvert = 10$	$d = 45$ $\lvert T\rvert = 10$	$d = 35$ $\lvert T\rvert = 20$	$d = 35$ $\lvert T\rvert = 30$
GMREE	0%	0%	0%	0%
MSTEAM	29%	14%	24%	27%
Steiner	43%	20%	33%	34%

Table 1: Summary of performance evaluation.

one. Considering the density $d = 5$, the energy consumption is divided by a factor of approximately 27 in Fig. 7(a) and 16 in Fig. 7(b) thanks to the MST based face mechanism. Of course at such small density, the quantity of void areas is high, but 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. Note that in the multicast MAC layer case, packets are aggregated before transmission. The basic face scheme is nevertheless inferior because each destination is handled separately, so each destination be again handled in greedy mode at different nodes, i.e., message forwarding will quickly use different paths and lead to increased energy consumption.

Fig. 8 provides results for all selected schemes at increasing density. As we explained before, ESP provides good results only for some correctly chosen sets of parameters. As the density increases, the shortest paths share too few common parts and message splits are too numerous for the routes efficiency to compensate the wasted energy. As a consequence, it becomes less efficient than localized schemes which use a better message splitting strategy. The Steiner tree based scheme uses a centralized efficient multicast backbone and is thus able to provide very good results for all densities. As the density increases, the approximated Steiner tree becomes closer to the optimal one, and energy consumption is thus lower. MSTEAM, with its localized computation, is however able to provide very close results in high densities. At the density $d = 40$, MSTEAM consumes only 6% more energy than Steiner in Fig. 8(a) and 8% more in Fig. 8(b). For all densities, MSTEAM provides better energy savings than GMREE. Since face routing is not used in such densities, the two main reasons are the message splitting strategy and the MST-based progress estimation, which is better than the one used in GMREE where the sum over all individual distances to the destinations is used. This is true even in Fig. 8(b) where the multicast wireless advantage is considered. Overall, the latter does not change the relative position of all schemes, it only leads to a lower energy consumption.

We finally provide results at increasing number of destination nodes in Fig. 9. The density considered was $d = 35$. 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 destination nodes. However, one can notice that the Steiner scheme and MSTEAM are more scalable than the other schemes since their energy consumption increases in a slower way. For instance, when changing the number of destinations from 10 to 30, the consumption of MSTEAM in Fig. 9(a) 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 layer cases, and is caused by a better path reuse strategy. Once again, MSTEAM is close to the Steiner scheme and thus provides very good results.

Table 1 provides a summary of the observed results. For each scheme is given the improvement over GMREE.

7 Conclusion and future work

In this paper, we presented and analyzed MSTEAM, a new geographic localized multicast scheme that uses a minimum spanning tree as an approximation of the optimal multicast backbone. Besides the fact that our protocol guarantees delivery of the multicast message in an efficient way, it is especially suited for ad hoc and sensor networks since it is fully localized and has a low time complexity. We experimentally demonstrated that MSTEAM is really energy-efficient, even when compared to a centralized scheme, and outperforms the best existing localized multicast routing protocol.

For future work, we plan to work on improving MSTEAM by considering the use of a Steiner tree approximation instead of a MST. Actually, we focused on MST because it provides a reasonable approximation of an optimal multicast tree when the edge weight function is not known. Moreover, its computation has a very low time complexity, while even an approximated Steiner tree introduces more complexity in the multicast algorithm. However, under the specific exponential path loss model used for experimental results in this paper, calculating a Steiner tree based on this power consumption is possible and one can expect better energy savings by considering a good weighted Steiner tree approximation. Of course, a trade-off between energy savings and complexity is needed in this case. Another improvement might be introduced during face routing. When two hop neighbor information is available face routing can employ shortcuts along the planar graph in order to save energy as described in [20]. A trade-off, however, exists in this case since maintaining the local view on the planar graph within the transmission range requires more control overhead than just exchanging position information with one hop neighbor. Finally, the only planar graph we considered in this paper for face routing is the Gabriel graph. It should be possible to construct a more efficient planar graph by constraining the resulting edges to be of optimal length for the considered energy model. However, a trade-off is once again involved since the construction of this graph would be more complex while face routing is used only in special cases, and especially when the network density is low.

References

[1] G. Borriello, editor. *RFID: Tagging the world*, volume 48. Communications of the ACM, September 2005.

[2] P. Bose, P. Morin, I. Stojmenović, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. *ACM Wireless Networks*, 7(6):609–616, November 2001.

[3] 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.

[4] L.M. Feeney. An energy-consumption model for performance analysis of routing protocols for mobile ad hoc networks. *ACM Journal of Mobile Networks and Applications*, 3(6):239–249, June 2001.

[5] H. Frey and I. Stojmenović. Geographic and energy aware routing in sensor networks. In *Handbook on Sensor Networks*. Wiley, 2006.

[6] H. Frey and I. Stojmenović. Geographic routing algorithms for ad-hoc networks. In *Wireless Ad Hoc and Sensor Networks*. Springer, 2006.

[7] H. Frey and I. Stojmenović. On delivery guarantees of face and combined greedy face routing in ad hoc and sensor networks. In *Proceedings of the ACM Annual International Conference on Mobile Computing and Networking (MobiCom)*, Los Angeles, USA, September 2006.

[8] K.R. Gabriel and R.R. Sokal. A new statistical approach to geographic variation analysis. *Systemic Zoology*, 18(3):259–278, September 1969.

[9] B. Karp and H.T. Kung. GPSR: Greedy perimeter stateless routing for wireless networks. In *Proceedings of the ACM Annual International Conference on Mobile Computing and Networking (MobiCom)*, Boston, USA, August 2000.

[10] R.M. Karp. *Complexity of Computer Computations*, chapter Reducibility among combinatorial problems. Plenum press, 1972.

[11] L. Kou, G. Markowsky, and L. Berman. A fast algorithm for Steiner trees. *Acta Informatica*, 15(2):141–145, June 1981.

[12] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger. Geometric ad-hoc routing: Of theory and practice. In *Proceedings of the ACM International Symposium on the Principles of Distributed Computing (PODC)*, Boston, USA, July 2003.

[13] F. Kuhn, R. Wattenhofer, and A. Zollinger. Worst-case optimal and average-case efficient geometric ad-hoc routing. In *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, Annapolis, USA, June 2003.

[14] 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):147–159, June 2006.

[15] 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.

[16] V. Rodoplu and T.H. Meng. Minimum energy mobile wireless networks. *IEEE Journal on Selected Areas in Communications*, 17(8):1333–1344, August 1999.

[17] P.M. Ruiz and A.F. Gomez-Skarmeta. Approximating optimal multicast trees in wireless multihop networks. In *Proceedings of the IEEE International Symposium on Computers and Communications (ISCC)*, La Manga del Mar Menor, Spain, June 2005.

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

[19] J.A. Sanchez, P.M. Ruiz, and I. Stojmenović. Energy efficient geographic multicast routing for sensor and actuator networks. *Computer Communications*, 2007. Special issue on Sensor-Actuator Networks (SANETs), to appear.

[20] Ivan Stojmenović and Susanta Datta. Power and cost aware localized routing with guaranteed delivery in unit graph based ad hoc networks. *Wireless Communications and Mobile Computing*, 4:175–188, 2004.

[21] S. Wang, D. Wei, and S. Kuo. SPT-based topology algorithm for constructing power efficient wireless ad hoc networks. In *Proceedings of the ACM International World Wide Web Conference*, New York city, USA, May 2004.

Unité de recherche INRIA Futurs
Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)

Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-0803