

MFACE: A Multicast Backbone-Assisted Face Traversal Algorithm for Arbitrary Planar Ad Hoc and Sensor Network Topologies

Hannes Frey and François Ingelrest

Abstract—Face is a well-known localized routing protocol for ad hoc and sensor networks which guarantees delivery of the message as long as a path exists between the source and the destination. This is achieved by employing a left/right hand rule to route the message along the faces of a planar topology. Although face was developed for the unicast case, it has recently been used in combination with multicasting protocols, where there are multiple destinations. Some of the proposed solutions handle each destination separately and lead thus to increased energy consumption. Extensions of face recovery to the multicast case described so far are either limited to certain planar graphs or do not provide delivery guarantees. A recently described scheme employs multicast face recovery based on a so called multicast backbone. A multicast backbone is a Euclidean spanning tree which contains at least the source and the destination nodes. The idea of backbone assisted routing is to follow the edges of the backbone in order to deliver a multicast message to all spanned destination nodes. The existing backbone face routing scheme is however limited to a certain planar graph type and a certain backbone construction. One of the key aspects of the multicast face algorithm MFACE we propose in this work is that it may be applied on top of any planar topology. Moreover, our solution may be used as a generic framework since it is able to work with any arbitrary multicast backbone. In MFACE, any edge of the backbone originated at the source node will generate a new copy of the message which will be routed toward the set of destination nodes spanned by the corresponding edge. Whenever the message arrives at a face edge intersected by a backbone edge different from the initial edge, the message is split into two copies, both handling a disjoint subset of the multicast destinations which are defined by splitting the multicast backbone at that intersection point.

I. INTRODUCTION

Ad hoc networks consist of independent wireless devices which are communicating without a predefined network infrastructure. Such technology may be useful, for instance, in order to rapidly setup a communication infrastructure in disaster recovery scenarios. Moreover, ad hoc networking may provide electronic data exchange at conferences, universities, companies, or any public area like train stations or airports. It may also be used in order to extend the coverage of wireless access points providing a link to the internet. Sensor networks are a special class of ad hoc networks, which received significant attention in recent years. The idea is to combine small sensing, computation, and wireless communication capabilities in small

low-powered devices. Once deployed, a sensor network should last many years providing measurements on some physical phenomenon like vibration, temperature, or humidity.

In this paper, we are interested in multicast routing, where a source host decides to send a message to a set of several destination nodes. In wireless ad hoc and sensor networks, communication ranges are limited due to the path loss of radio communications, and thus many other intermediate nodes must participate in the multicasting, enabling the message to reach all destination nodes. The simplest algorithm to achieve this may be to use the well-known *blind flooding* method, where each node that receives the packet for the first time forwards it to its neighborhood. This method obviously ensures a total coverage of the network, provided that the latter is connected, and thus a coverage of all destination nodes. Unfortunately, it requires *every* node to participate, leading to a lot of duplicated packets and wasted energy: as nodes rely on limited onboard power supply, other protocols, more energy aware, must be designed.

Routing algorithms may be classified into localized and centralized ones. The latter require information about the global network structure to be acquired and maintained. Such schemes are obviously not scalable, since the control overhead increases with the number of network nodes. Moreover, the energy consumption needed for that gathering is generally higher than the savings obtained thanks to the centralized knowledge. Localized algorithms are a promising approach to provide networks of any size. In these schemes, a global objective is achieved by using only local neighborhood information. A special class of localized routing schemes, referred to as geographic routing, requires nodes to be able to determine their own location. Incorporating geographic information in routing decisions enables novel communication paradigms: in geocasting communication, for instance, rather than delivering a message to a specific network address, all nodes in a specific geographic area are addressed.

All localized multicast routing algorithms described so far employ a greedy heuristic in order to forward a message using information about the destinations and the 1-hop neighbors of the current forwarding node. The next hop nodes are selected according to a local objective function which has to be maximized or minimized. In order to provide loop-free operation, greedy routing requires a premature stopping criterion which prevents a message from visiting a node twice.

H. Frey is with IMADA, Syddansk Universitet, DK-5230 Odense M, Denmark. E-Mail: frey@imada.sdu.dk

F. Ingelrest is with LCAV, I&C School, EPFL, CH-1015 Lausanne, Switzerland. E-Mail: Francois.Ingelrest@epfl.ch

As a consequence, a message might get dropped although there exists a path from the source node to the destinations. Face traversal is a recovery mechanism which maintains the localized nature of greedy routing schemes. However, such a scheme can not be applied on the underlying network graph directly. The latter must be planarized to be able to use Face routing on it.

Face routing which is well-studied for the unicast case has recently been used in combination with multicast greedy schemes as well. Some multicast algorithms described employ traditional unicast face recovery in order to recover from a greedy routing failure for each destination node individually. Thus multicast forwarding “degenerates” to individual unicast tasks as soon a greedy routing failure occurs. Multicast extensions to face routing are addressing this problem by handling several destinations in a single planar graph traversal. All existing multicast extensions to face recovery are either limited by the fact that they sacrifice the delivery guarantees of the unicast case, or they are limited to a specific planar graph construction method. In this paper, we describe for the first time MFACE, a multicast extension which both guarantees delivery and works on any planar network topology. Our generic framework does not suffer from the limitation encountered in existing solutions, and is able to use any given underlying multicast backbone to decide how face traversal should be done, and when a message should be split into multiple packets in an efficient way.

The remainder of this paper is organized as follows. In the next section, we give the needed preliminaries, while Sec. III proposes a literature review of the related work. In Sec. IV, we describe MFACE, our generic face multicast routing framework. This section is followed by Sec. V which provides a proof of correctness of MFACE in terms of loop free operation and delivery guarantees. We finally conclude and discuss open research issues in Sec. VI.

II. PRELIMINARIES

The common representation of a wireless network is a graph $G = (V, E)$, where V is the set of vertices (the hosts, or nodes) and $E \subseteq V^2$ the set of edges giving the available communications: if a node v is a physical neighbor of a node u (v lies within the communication range of u and thus receives its messages), then there exists $(u, v) \in E$. If we assume circular communication ranges and that all nodes have the same communication radius R , then the set E is defined by:

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

$|uv|$ being the Euclidean distance between nodes u and v . This model is known as the *unit disk graph* model. A generalization of this model, the *quasi unit disk graph* model, allows a slight variation in each node’s communication range in the magnitude of $\sqrt{2}$ between a minimum and maximum communication radius r_{\min} and r_{\max} . More precisely, for $r_{\max}/r_{\min} = \sqrt{2}$ any edge set E complies with the quasi unit disk assumption if the following holds:

$$\begin{aligned} (u, v) \in E &\Rightarrow |uv| \leq r_{\max} \\ |uv| \leq r_{\min} &\Rightarrow (u, v) \in E \end{aligned}$$

We assume that nodes are able to retrieve their absolute or relative location, by any hardware or software means, and maintain 1-hop neighborhood information thanks to regular beacon messages. For instance, absolute locations may be available thanks to a GPS, while relative ones may be obtained by setting up a relative coordinate system based on signal strength measurements [1]. As for any other geographic routing scheme, the location of destination nodes is needed. Whenever such a scheme is used as a replacement for non geographic routing, the position for a given node address is required. This information may be acquired thanks to a location database running as a separate network service [2].

A planar graph is a graph in which no edges intersect. The known localized planar graph construction methods are Gabriel graph (GG), relative neighborhood graph (RNG) and localized Delaunay triangulation (LDT). Without any additional provision, planar graph construction according to these schemes requires the unit disk graph assumption. For Gabriel graph construction, however, a localized extension has been described in [3], [4] which works also under quasi unit disk graphs.

The Gabriel graph, described in [5], is constructed as follows. 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 $[uv]$. The relative neighborhood graph (see Fig. 1(b)), described in [6], uses a similar removal strategy. An edge (u, v) is removed if there exists a node w such that $(u, w), (w, v) \in E$ and $|uw|, |wv| < |uv|$. A simpler definition would be “for any triangle in the graph, remove the longest edge”. In the localized Delaunay triangulation, each node u applies the Delaunay triangulation on its 1-hop neighborhood set. In the Delaunay triangulation, a triangle exists if there is no other vertex inside the circle passing through all the end points of this triangle. This locally constructed Delaunay triangulation has to be exchanged with all 1-hop neighbor nodes in order to remove Delaunay triangulation edges which have not been preserved by other neighbor nodes. Different variants of localized Delaunay triangulation have been described in [7], [8], [9]. All discussed edge removal strategies are illustrated by Fig. 1.

III. RELATED WORK

A class of multicasting protocols relies on a backbone to achieve delivery of the message in an effective way. The idea is to construct a geometric overlay multicast tree, and then to use the edges of this tree to route the message in a unicast fashion between each of its vertices. Such a backbone may be for instance an Euclidean minimum spanning tree or a Steiner tree constructed over the source node and the set of destinations. It may actually be any tree spanning all destination nodes.

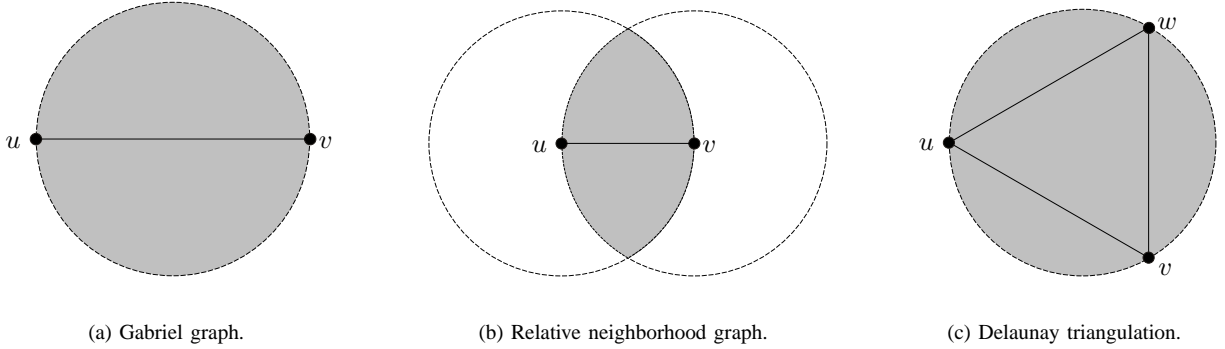


Fig. 1. Edges removal strategies for constructing planar graphs: if a node exists in the grey area, the corresponding edges are removed.

In [10], the construction of such backbones is proposed in order to minimize miscellaneous metrics (e.g., bandwidth, energy consumption). Another example of a backbone-assisted multicast protocol is MSTEAM [11], which uses a minimum spanning tree to decide when the message should be split into multiple packets and which next hop to choose toward a set of destination nodes. In this paper, we apply this idea to extend face routing to the multicast case.

A. Face traversal in unicast communication

A unicast face routing algorithm for ad hoc and sensor networks was originally described in [12]. The algorithm makes use of an overlay planar graph to achieve the message delivery by routing packets in a localized way along the face edges of this graph. The faces used to route the message are the ones which are intersected by the straight line joining the source node s to the destination t . Face traversal is repeatedly done by using the left/right hand rule: a receiver node sends the message along the edge which is lying next in counterclockwise/clockwise direction of the edge it was received from. For instance, when starting at node s in Fig. 2, the face F_1 will be traversed along the path sv_1v_2 when using right hand rule and along the path sv_4v_3 when using the left hand rule. When the message is going to cross the edge v_2v_3 , a face switch occurs, so that the packet will be next routed inside face F_2 . It can be observed that face traversal might produce long detours from the shortest possible path, in particular when the network is sparse. Greedy routing in contrast, if successful, often achieves a path length close to the shortest one. Thus, face routing is generally used as a recovery mechanism in case of a greedy routing failure. When greedy routing can be applied again, face traversal is stopped and greedy routing is resumed again. It has been observed when face routing is used as a recovery mechanism only, under the GG and RNG topology, a specific rule for switching between faces is not required [13]. Under these graph models returning into greedy mode is always possible when traversing the very first face only. However, this holds not in general. For instance, when applying combined greedy and face routing on a LDT, switching between adjacent face might be necessary before greedy routing can be started again [13].

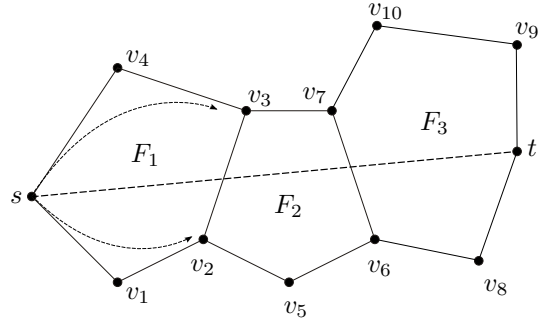


Fig. 2. Face unicast routing: the message travels along the faces intersected by the source-destination line st .

B. Face traversal in multicast communication

In the multicast schemes GMR [14] and GMREE [15], face traversal is employed for each of the destination nodes as a separate instance. More precisely, whenever a node s decides that a set of destination $\{t_1, \dots, t_k\}$ can no longer be handled in greedy forwarding mode, a unicast face traversal according to the source-destination line st_i is started for each of these destination nodes t_i . Each of these individual face traversals are accomplished until encountering a node which enables greedy forwarding again. In order to reduce bandwidth requirements the routing scheme merges packets visiting the same next hop node in a single transmission. However, despite this optimization, conceptually the algorithm performs face traversal for each destination node individually. This leads to disjoint routes and thus to a higher energy consumption. Since face recovery is handled by individual unicast face traversals, correctness of the scheme in terms of loop-free operation and guaranteed delivery is ensured by the well-known results for the unicast greedy-face-greedy schemes [13].

A face recovery approach which can handle several destination nodes in a single face traversal was described in [16] for the first time. In the described multicast scheme PBM, face recovery is used as well in order to recover from greedy routing failures. When a node s has no neighbor node which provides positive progress for a given set of destination nodes $\{t_1, \dots, t_k\}$, multicast face recovery is started. The destinations are handled in a single traversal of the face which

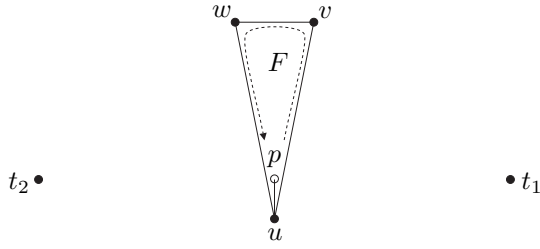


Fig. 3. A message drop in PBM: the message will traverse face F and will never find a node closer to a destination node t_i than u .

contains s on its boundary and which is intersected by sp , p being the center of gravity $(t_1 + \dots + t_k)/k$ of all destination nodes. In each face recovery step, however, it is tested for each destination if the current node is closer to t_i than the node where the message entered face recovery mode. For all destinations which satisfy this condition, greedy forwarding is started again. The remaining destinations are kept in face recovery mode.

A proof on correctness of this scheme was still missing by now, and in fact, it turns out that the described variant is merely a good heuristic in order to recover from greedy routing failures. A simple example which shows that delivery guarantees are not provided by this scheme is depicted in Fig. 3. Node u is located closer to t_1 and t_2 as it is v and w . Since none of the neighbors are providing some progress towards t_1 and t_2 , multicast face recovery is started for the destination set $\{t_1, t_2\}$. The center of gravity p of t_1 and t_2 is located within the face F . However, during face traversal of F no node is encountered which is located closer to t_1 or t_2 than $|ut_1| = |ut_2|$. Thus, the message will traverse the entire face and will be dropped after passing the first traversal edge uw for the second time. This happens independently of the existence or nonexistence of a path connecting u with t_1 or t_2 .

The first scheme implementing a multicast extension of face routing which provides delivery guarantees is presented in [11]. The described multicast algorithm MSTEAM applies face traversal in order to handle a message to a set of destinations by using one of these destinations, t , as the gateway to all others. The recovery scheme is designed to run over a planar topology resulting from Gabriel Graph construction. In this case, planar graph recovery with respect to a source destination line st can be simplified to traversal of the very first face which contains s on its boundary and which is intersected by st . Correctness of the scheme is proved in [11] for Gabriel Graphs. More precisely, the Gabriel Graph property is required to prove case 1 of Lemma 1. The proof of this case requires that for an edge uv intersecting st at least one of the nodes u and v is lying closer to t than the Euclidean distance between s and t . Thanks to Corollary 1 in [13], this property also holds for Relative Neighborhood Graphs. It follows that the face recovery mechanism of MSTEAM provides delivery guarantees and loop-free operation as well in Relative Neighborhood Graphs.

Table I provides a summary of all known face routing

variants applied in multicast schemes. As we shall see in the next section, the proposed MFACE scheme is the first generic variant that provides delivery guarantees while assuming any planar topology and any underlying multicast backbone. All schemes given in this table are loop-free.

IV. A GENERIC MULTICAST FACE ALGORITHM

In the following section, we describe an extension of the face routing mechanism for the multicast case, referred as MFACE, which is intended to be used under any planar topology. We assume a message addressed from a source node s to a set of destination nodes $T = \{t_1, \dots, t_k\}$. Furthermore, we assume that any construction mechanism was applied on $S = \{s\} \cup T$ in order to calculate a multicast backbone being a spanning tree Δ which includes at least all nodes in S .

We first consider the source node s which initiates the multicasting task. Let Δ be the multicast backbone which is to be used to reach all destination nodes in T . The message thus has to be routed along the edges of Δ , and must be split at node s if multiple paths start from this node. Actually, each of these 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. The algorithm thus instantiates a separate multicast task for each edge st from the multicast backbone which originates at s . More precisely, for each such edge st , the algorithm initiates a multicast task which only considers the subtree of Δ which is reachable by the edge st . A special case occurs, when the subtree contains no destination nodes. In this case the subtree can simply be ignored.

For any other subtree originating from st which contains at least one destination node, face traversal for this multicast task is started by selecting the face adjacent to s which is intersected by the open line segment $(st]$. When s has only one neighbor node v , face traversal start is done via the edge sv . When s has at least two neighbors, the selection of the first traversed edge is done according to the *best angle heuristic* [17]: from the possible face edges originating at s , the edge sv is selected which minimizes the angle between st and sv . This procedure is summarized by algorithms 1 and 2.

Algorithm 1 Starting multicast face traversal

```

 $s \leftarrow$  current node
 $t_1, \dots, t_k \leftarrow$  multicast destinations
 $\Delta \leftarrow$  multicast backbone over  $\{s, t_1, \dots, t_k\}$ 
for all edges  $st$  in  $\Delta$  do
   $\Delta' \leftarrow$  subtree of  $\Delta$  reachable via  $st$ 
  if  $\Delta'$  contains at least one destination node then
     $su \leftarrow$  edge minimizing  $\angle ust$ 
     $rule \leftarrow$  determine traversal rule for  $s, u$ , and  $t$ 
    begin traversal along  $su$  according to  $rule$  and subtree  $\Delta'$ 
  end if
end for

```

Algorithm	Delivery Guarantees	Planar Topology	Multicast Rule
GMR, GMREE	yes	any	separate unicasts
PBM	no	any	average over destination points
MSTEAM	yes	GG, RNG	multicast overlay
MFACE	yes	any	multicast overlay

TABLE I

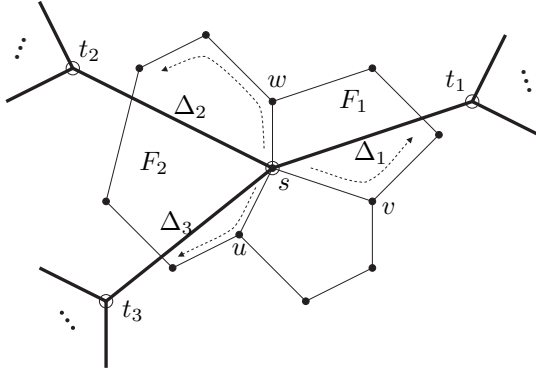
ALL KNOWN FACE RECOVERY VARIANTS APPLIED IN MULTICAST COMMUNICATION.

Algorithm 2 Determining the traversal rule

```

s ← current node
u ← next node
t ← destination node
if u is located right of st then
    rule ← clockwise
else
    rule ← counterclockwise
end if

```

Fig. 4. Starting multicast face traversal: 3 edges originate at s , and a copy of the message will be sent along each of them.

Refer to Fig. 4 for an example. The multicast backbone edges originating at s are st_1 , st_2 , and st_3 . Thus, s initiates 3 multicast tasks, one for each edge st_i . The multicast task created for st_1 will traverse the face F_1 , while the two multicast tasks created for the edges st_2 and st_3 will both traverse the face F_2 . According to the best angle heuristic, the start edge for traversal of F_1 will be sv since $\angle vst_1 < \angle wst_1$ is satisfied. The remaining multicast backbone considered in this case is Δ_1 . Traversal of face F_2 will be done in two directions, along sw for t_2 since $\angle wst_2$ is minimum, and along start edge su for t_3 since $\angle ust_3$ is minimum. In this example, Δ_2 will be the remaining multicast backbone considered in the first case, while Δ_3 will be the remaining multicast backbone in the latter case.

We now describe how faces are switched whenever the current multicast message arrives at a face edge which is intersected by any of the edges of the remaining multicast backbone handled by that message. Let st be the backbone tree edge where traversal of the face was started at (for instance, edge st_1 in the traversal of face F_1 in Fig. 4), let u be the current forwarding node, and let uv be the next face edge to be traversed. We consider two different cases of uv intersecting with any of the edges from the multicast backbone.

Intersection with the starting edge of the backbone

Two cases arise, when uv intersects with st in a point p . When face traversal has already visited an intersection with st which is located closer to t than it is p , then this intersection is simply ignored. If the intersection p is the one closest to t compared to all other intersections visited so far, then the multicast face routing mechanism restarts traversal of the face which has the current node on its boundary and which intersects with the open line segment $(pt]$. Traversal start of this face is done according to the best angle method, i.e., the current forwarding node u selects the outgoing edge which minimizes the angle with respect to the line segment ut .

This rule is illustrated by Fig. 5, where we assume that the current multicast subtree handling was started at node s . The message is traveling along the edges of face F_1 since this face is intersected by the edge st_1 . When it arrives at node u , a face switch occurs. Edge uv is intersecting with the backbone edge st_1 , the edge which was used in order to determine face traversal start. Thus, one single instance of the multicast message is kept and the face intersected by the remaining line segment of st_1 is traversed. According to the best angle variant, traversal is started along edge uv since $\angle vut_1 < \angle wut_1$ is satisfied.

Intersection with any other edge of the backbone

When uv intersects with any other multicast backbone edge t_it_j different from st , then the edge st is removed from the multicast backbone and the remaining tree is split into two subtrees Δ_1 and Δ_2 . Tree Δ_1 denotes the subtree which is reachable over the directed edge t_it_j , while tree Δ_2 denotes the subtree which is reachable over the directed edge t_jt_i . The multicast message is split into two copies in order to address each of both subtrees in a different multicast task. Selecting the face which is to be traversed next is done in the same way as it is done for the case uv intersecting with st . Let p be the point of intersection between uv and t_it_j . For handling the remaining subtree Δ_1 , that face is selected which is intersected by the open line segment $(pt_j]$. For subtree Δ_2 , the face is selected which is intersected by the opposite open line segment $(pt_i]$. Face traversal start is done according to the best angle variant, i.e., the edge is selected which minimizes the angle with respect to the edges ut_j and ut_i , respectively. A special case occurs, when the remaining multicast backbone Δ_i consists only of nodes which are no message destinations. In this case the multicast task for Δ_i can simply be ignored.

Finally, we have to consider the case that the current visited node v is equal to one of the multicast destinations. In this

Algorithm 3 Continuing multicast face traversal

```

rule ← traversal rule
 $\Delta$  ← remaining multicast backbone
e ← traversal start edge
xt ← edge multicast backbone is originating from
p ← last intersection point with xt
u ← previous node
v ← current node
w ← next node according to rule
if v is a multicast destination then
  remove v from multicast destinations
  start face traversal at v
else if vw does not intersect with an edge of  $\Delta$  then
  if vw ≠ e then
    send message along vw
  else
    drop message
  end if
else
  if vw intersects with pt then
    vw ← edge minimizing  $\mathcal{L}wvt$ 
    rule ← determine traversal rule for v, w, and t
    begin traversal along vw according to rule and subtree  $\Delta$ 
  else
    remove xt from  $\Delta$ 
     $t_1t_2$  ← edge in  $\Delta$  intersecting with vw
    p ← intersection point of vw and  $t_1t_2$ 
     $\Delta_1$  ← subtree of  $\Delta$  reachable via  $t_2t_1$ 
     $\Delta_2$  ← subtree of  $\Delta$  reachable via  $t_1t_2$ 
    for k = 1, 2 do
      remove  $t_{3-k}$  from multicast destinations for message instance k
      if  $\Delta_k$  contains at least one destination node of message instance k then
        vw ← edge minimizing  $\mathcal{L}wvt_k$ 
        rule ← determine traversal rule for v, w, and  $t_k$ 
        begin traversal along vw according to rule and subtree  $\Delta_k$ 
      end if
    end for
  end if
end if

```

case, the multicast message is passed to the upper protocols layer, then node v is removed from the set of destinations, and multicast face traversal is restarted at node v using Alg. 1. Refer to Alg. 3 for a summary of the whole described procedure.

For an illustration of the described strategy refer to Fig. 5. When traveling along the face F_2 , the message arrives at the intersection point p of backbone edge t_1t_3 and F_2 . Thus, the message is split into two multicast subtasks. One instance is handled in face F_2 in order to address the subtree originating from the directed edge t_3t_1 . The other instance is handled in face F_3 addressing the subtree originating from the edge t_1t_3 .

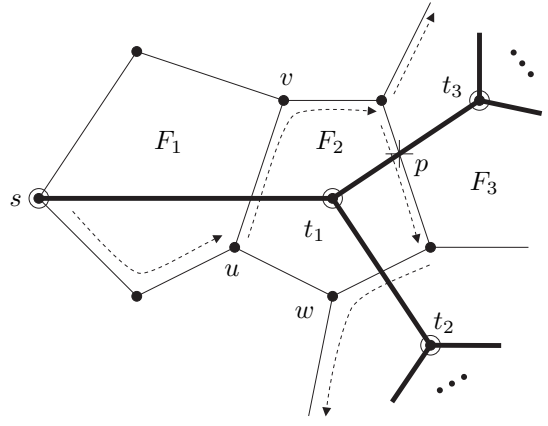


Fig. 5. Switching faces on intersected boundaries.

V. CORRECTNESS OF MFACE

In the following section we prove correctness of MFACE in terms of loop free operation and delivery guarantees. The proof is organized as follows. We first show in Lemma 1, when at least one destination node is reachable, face exploration from an edge intersected by the multicast backbone will always arrive at an additional intersection point. Then we show in Lemma 2 that after a message split each message instance is addressing a disjoint multicast destination set and that no destination is erased in this step. These two algorithm invariants are finally used in order to prove the proposition of Theorem 1 and Theorem 2.

Lemma 1: Let Δ be a connected backbone which originates from xt . Let vw be an edge which intersects with xt at a point p . Let F be the face which has vw as a boundary edge and which intersects with pt . If v can reach at least one destination node in Δ then F also intersects with Δ in a point q different from p .

Proof: If t is a boundary node of F or if t is located outside of F then pt is also intersecting the boundary of F in a point q with $|qt| < |pt|$ (see Fig. 6(a)). Since pt is a segment of the backbone edge xt it follows that Δ intersects with F in a point different from p .

Assume that t is located within face F (see Fig. 6(b)). By assumption v can reach at least one destination node t_i . It follows from Lemma 3 in [13] that destination node t_i is either a node of the boundary of F or is located outside of F . By assumption Δ is connected and xt is the edge Δ originates from. Thus, there exists at least one edge in Δ which is different from xt and which intersects with the face boundary of F , i.e., we have again an intersection point $q \neq p$. ■

Lemma 2: Let T be the set of destinations handled by a multicast message instance. Whenever the message is split into several instances, each instance is addressing a disjoint nonempty set of destination subsets. The union over those destination subsets is T .

Proof: When multicast face exploration is started in a node s the destination set is split into the subsets which are reachable over exactly one of the backbone edges originating

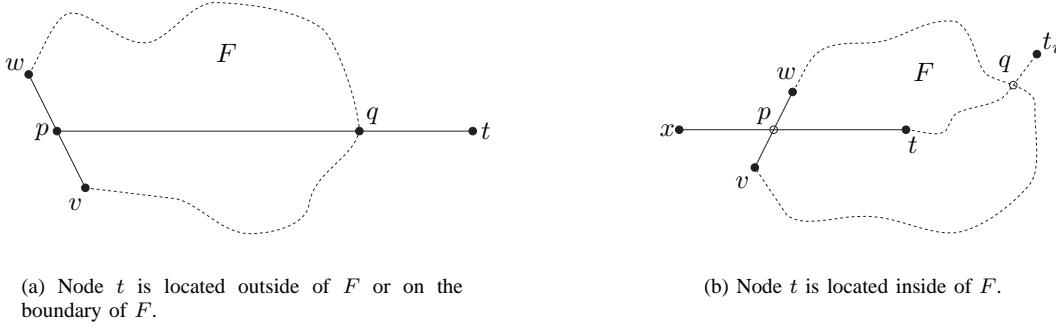


Fig. 6. The cases in the proof of Lemma 1

from s (see st_1 , st_2 , and st_3 in Fig. 4 for an example). For each subtree, a multicast message is created only if it contains at least one destination node.

When a message is split into two instances due to an intersection with a backbone edge t_1t_2 , the edge st the backbone Δ originates from is removed from Δ . The resulting tree Δ' is still spanning all destinations addressed by this multicast instance since s is not a destination node in this case. In fact, either traversal of Δ was started in s or traversal of Δ was started due to a visited edge intersecting with the multicast backbone. In the first case, s was already visited and is thus no longer a destination node. In the latter case s was removed from the destination nodes when traversal of Δ started.

The remaining multicast backbone Δ' is split into the subtree Δ_1 reachable via t_2t_1 and the subtree Δ_2 reachable via t_1t_2 . Let T_i be the destination nodes which are connected by Δ_i . The destination nodes addressed by the first message instance are $T_1 \setminus \{t_2\}$. The destination set addressed by the second message is $T_2 \setminus \{t_1\}$. In both cases an instance of a multicast message is created only, if it addresses at least one multicast destination. Finally, it holds $T_1 \setminus \{t_2\}$ and $T_2 \setminus \{t_1\}$ are disjoint and satisfy $T_1 \setminus \{t_2\} \cup T_2 \setminus \{t_1\} = T$. ■

Theorem 1: The described MFACE routing algorithm is loop free, independently of the underlying multicast backbone.

Proof: Whenever a message is split, it addresses a subset S of the destinations T which were handled before that message split, while $S \neq T$ is satisfied by Lemma 2. Moreover, two multicast messages are never merged. It follows, that for each possible subset of T at most one instance of a message may exist during a multicast task. It is thus sufficient to show that each potential message instance is forwarded only a finite number of steps.

Let m be a message instance for a given destination subset and let xt be the originating edge of the multicast backbone Δ handled by m . As soon as the message arrives at an edge intersecting an edge of Δ which is different from xt , the message instance disappears. It is split into several multicast sub tasks.

Two cases arise when the message arrives at an edge wv intersecting with xt . When the intersection point q satisfies

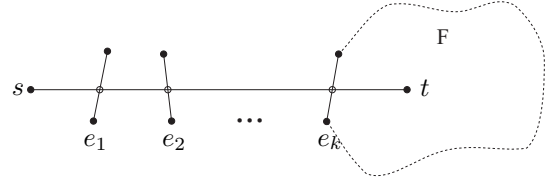


Fig. 7. A sequence of intersections leading towards t .

$|qt| < |pt|$ compared to the previous encountered intersection p with xt ($p = x$ if no previous intersection was encountered so far), it will begin traversal of the face having wv as a boundary node and which intersects the open line segment $(qt]$. When the intersection point q satisfies $|qt| \geq |pt|$, the intersection is simply ignored.

Since the number of edges are finite, it follows that the message can only visit a finite sequence of edges e_1, \dots, e_k intersecting with xt while after e_k no closer intersection to t exists (see Fig. 7). If the message arrives at e_k without being split so far, the face F containing t is traversed. The message will either be split during this face traversal or visit the first face traversal edge twice in the same direction. In the latter case the message will be dropped. ■

Theorem 2: Let s be the multicast start node and Δ be any connected backbone which connects s with a given set of destinations $T = \{t_1, \dots, t_k\}$. The described MFACE routing algorithm provides guaranteed delivery for each t_i which is reachable from s .

Proof: Let t_i be a destination node which is reachable from s . Due to Lemma 2 node t_i appears in a sequence of message instances, all addressing a destination subset of the previous one, until either the message is delivered to t_i or dropped without delivery. Assume for the sake of contradiction that the message is dropped.

A message is dropped only if it traverses a face F without finding an edge intersecting with the handled remaining multicast backbone Δ . Traversal of face F is either started due to Alg. 1 or Alg. 3. Let xt be the edge which was used in order to define the multicast backbone on face traversal start. By Alg. 1 the start node s selects the face which has s on its boundary and which intersects with xt . By Alg. 3 face traversal is started due to an intersection of xt with a face edge e . Let p be the

intersection point. The algorithm starts traversal of the face which has e as a boundary edge and which intersect with pt . In both cases the precondition of Lemma 1 is satisfied.

The face traversal start node v has received a message from the multicast start node s . Since t_i is reachable from s , it follows, that v is able to reach node t_i as well. Due to Lemma 1 F intersects also with Δ in a point which is different from p and s , respectively. Thus, face traversal will encounter an intersection with Δ , a contradiction. ■

VI. CONCLUSION AND OPEN RESEARCH TOPICS

Under a localized protocol, changes in the network graph due to appearing and disappearing links require message exchange only among those nodes which are immediately affected by that change. Consequently, localized routing does not suffer from control message overhead of their centralized counterparts. They scale well with any network size, making such schemes attractive in particular for large scale sensor network scenarios consisting of thousands of nodes.

In this paper, we presented MFACE, which is the very first multicast extension of localized face routing which is able to work with any multicast backbone on any given planar topology. For instance, the variant used in MSTEAM is restricted to two planar graphs (Gabriel graph and relative neighborhood graph) and to a specific class of multicast backbones (no additional points, like Steiner points, are allowed). MFACE does not have these limitations and may be used in combination with any multicast backbone assisted greedy multicast scheme. Moreover, we have shown that MFACE is loop-free and guarantees delivery as long as a path exists between the source node and the destinations.

The multicast face algorithm described in this work performs an individual unicast message forwarding task for each multicast backbone edge. As it can be observed in Fig. 4 the case of several backbone edges intersecting the same face might result in traversal of the face in two different directions. As future work, we plan to investigate how in this situation a message can be aggregated into one copy traversing the face in one direction only while maintaining loop free operation and delivery guarantees. We expect that traversing face sequences intersected by the multicast backbone instead of traversing individual multicast backbone edges will result in energy and communication bandwidth savings.

Moreover, for the unicast case it has been observed that each localized routing algorithm might produce a routing path of length $O(l^2)$, while l is the length of the shortest path [18]. A sophisticated improvement of unicast face traversal has been proposed in [19], [20] which provably produces path lengths which are limited by the bound $O(l^2)$. Such a scheme can thus be denoted as an asymptotically optimal localized unicast routing algorithm. Multicast routing opens a new field to the question of asymptotical optimality of localized algorithms. In future research we plan to investigate this bound in the light of the multicast problem and how the face routing improvement from [19], [20] can be applied in the case of multicast routing in order to achieve asymptotical optimal routing path lengths.

There are some fundamental questions which arise with respect to both localized unicast and localized multicast routing. By now planar graph traversal is the only known solution which achieves delivery guarantees in a pure localized manner. An unresolved problem, however, remains how far one can go beyond the required unit disk or quasi unit disk graph network models. It has been shown that under a centralized topology control mechanism referred as CLDP [21] planar graph routing works well even if there might exist intersections among some of the remaining network edges. CLDP works under any undirected network topology. However, the scheme sacrifices the localized nature of planar graph routing. The protocols utilize probing messages which might travel any distance which can not be bound by a constant k . A fundamental question remains if in arbitrary networks it is possible at all to locally construct a topology, possibly with intersecting edges, which supports delivery guarantees of a face traversal based routing algorithm. Moreover, in three dimensional networks even for the unit disk graph model no localized routing algorithm has been described so far which provides delivery guarantees. A question remains if localized routing and guaranteed delivery is possible at all in the three dimensional case. Finally, all localized routing algorithms require that nodes are able to determine their physical location (either relative to others or absolute). A fundamental question remains if geographic information is a necessary condition to enable the definition of localized unicast or multicast schemes which provide delivery guarantees. In other words, it remains open if it is possible at all to explore a network graph in a depth first search manner without memorizing state in the messages and the network nodes.

ACKNOWLEDGMENTS

This work was partially financed by the European Commission under the Framework 6 IST Project “Wirelessly Accessible Sensor Populations (WASP)” and by the Danish Agency for Science, Technology and Innovation, grant number 272-06-0581.

REFERENCES

- [1] S. Capkun, M. Hamdi, and J. Hubaux, “GPS-free positioning in mobile ad-hoc networks,” in *Proceedings of Hawaii International Conference on System Sciences*, Jan. 2001.
- [2] I. Stojmenovic, “Location updates for efficient routing in ad hoc networks,” in *Handbook of Wireless Networks and Mobile Computing*, I. Stojmenovic, Ed. Wiley, 2002, ch. 21, pp. 451–471.
- [3] L. Barriere, P. Fraigniaud, L. Narayanan, and J. Opatrný, “Robust position-based routing in wireless ad hoc networks with unstable transmission ranges,” in *Proceedings of the 5th ACM International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL M 01)*, 2001, pp. 19–27.
- [4] F. Kuhn, R. Wattenhofer, and A. Zollinger, “Ad-hoc networks beyond unit disk graphs,” in *ACM DIALM-POMC Joint Workshop on Foundations of Mobile Computing*, San Diego, Sept. 2003, pp. 69–78.
- [5] K. Gabriel and R. Sokal, “A new statistical approach to geographic variation analysis,” *Systemic Zoology*, vol. 18, no. 3, pp. 259–278, Sept. 1969.
- [6] G. Toussaint, “The relative neighborhood graph of finite planar set,” *Pattern Recognition*, vol. 12, no. 4, pp. 261–268, 1980.

- [7] J. Gao, L. J. Guibas, J. Hershberger, L. Zhang, and A. Zhu, "Geometric spanner for routing in mobile networks," in *Proceedings of the second ACM International Symposium on Mobile Ad Hoc Networking and Computing MobiHoc '01*, Long Beach, CA, USA, Oct. 2001, pp. 45–55.
- [8] X.-Y. Li, G. Calinescu, and P.-J. Wan, "Distributed construction of a planar spanner and routing for ad hoc wireless networks," in *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Society (INFOCOM '02)*, vol. 3. Piscataway, NJ, USA: IEEE Computer Society, June 23–27 2002, pp. 1268–1277.
- [9] X. Li, I. Stojmenović, and Y. Wang, "Partial delaunay triangulation and degree limited localized bluetooth scatternet formation," *IEEE Transactions on Parallel and Distributed Systems*, vol. 15, no. 4, pp. 350–361, Apr. 2004.
- [10] K. Chen and K. Nahrstedt, "Effective location-guided overlay multicast in mobile ad hoc networks," *International Journal of Wireless and Mobile Computing*, vol. 3, 2005, special issue on Group Communications in Ad Hoc Networks.
- [11] 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," 2007, submitted for publication.
- [12] P. Bose, P. Morin, I. Stojmenović, and J. Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks," *ACM Wireless Networks*, vol. 7, no. 6, pp. 609–616, Nov. 2001.
- [13] 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, Sept. 2006.
- [14] J. Sanchez, P. Ruiz, J. Liu, and I. Stojmenović, "Bandwidth-efficient geographic multicast routing protocol for wireless sensor networks," vol. 7, no. 5, May 2007, pp. 627–636.
- [15] J. Sanchez, P. 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.
- [16] M. Mauve, H. Füllner, J. Widmer, and T. Lang, "Position-based multicast routing for mobile ad-hoc networks," Department of Computer Science, University of Mannheim, Germany, Tech. Rep. TR-03-004, 2003.
- [17] H. Frey and I. Stojmenović, "On delivery guarantees of face and combined greedy face routing in ad hoc and sensor networks," 2007, journal publication in preparation.
- [18] F. Kuhn, R. Wattenhofer, and A. Zollinger, "Asymptotically optimal geometric mobile ad-hoc routing," in *Proceedings of the 6th International Workshop on Discrete Algorithms and Methods for Mobile Computing & Communications (DIAL M 02)*. New York: ACM Press, Sept. 28 2002, pp. 24–33.
- [19] —, "Worst-case optimal and average-case efficient geometric ad-hoc routing," in *Proceedings of the 4th ACM International Symposium on Mobile Computing and Networking (MobiHoc 2003)*, Annapolis, Maryland, USA, 2003.
- [20] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger, "Geometric ad-hoc routing: Of theory and practice," in *Proceedings of the 22nd ACM International Symposium on the Principles of Distributed Computing (PODC)*, Boston, Massachusetts, USA, July 13–16 2003, pp. 63–72.
- [21] Y.-J. Kim, R. Govindan, B. Karp, and S. Shenker, "Geographic routing made practical," in *Proceedings of USENIX Symposium on Network Systems Design and Implementation*, Boston, Massachusetts, USA, May 2005.