

Simple and Efficient Geographic Routing around Obstacles for Wireless Sensor Networks

Olivier Powell* and Sotiris Nikolettseas**

Computer Engineering and Informatics Department of Patras University and
Research Academic Computer Technology Institute (CTI), Greece.

Abstract. Geographic routing is becoming the protocol of choice for many sensor network applications. The current state of the art is unsatisfactory: some algorithms are very efficient, however they require a preliminary planarization of the communication graph. Planarization induces overhead and is thus not realistic for some scenarios such as the case of highly dynamic network topologies. On the other hand, georouting algorithms which do not rely on planarization have fairly low success rates *and* fail to route messages around all but the simplest obstacles. To overcome these limitations, we propose the GRIC geographic routing algorithm. It has absolutely no topology maintenance overhead, almost 100% delivery rates (when no obstacles are added), bypasses large convex obstacles, finds short paths to the destination, resists link failure and is fairly simple to implement. The case of hard concave obstacles is also studied; such obstacles are hard instances for which performance diminishes.

1 Introduction

Recent advances in micro-electromechanical systems (MEMS) have enabled the development of very small sensing devices called sensor nodes [1–3]. These are smart devices with sensing, data-processing and wireless transmission capabilities meant to collaboratively form *wireless sensor networks* (sensor nets) instrumenting the physical world by collecting, aggregating and propagating environmental information to regions of interest such as mobile users or fixed base stations possibly linked to a satellite or the Internet. Some applications imply deployment in remote or hostile environments (battle-field, tsunami, earth-quake, isolated wild-life island, space exploration program) to assist in tasks such as target tracking, enemy intrusion detection, forest fire detection or environmental or biological monitoring. Other applications imply deployment indoors or in urban or controlled environments, for example with the purpose of industrial supervising, indoor micro-climate monitoring (e.g. to reduce heating cost by detecting poor thermal insulation of buildings), smart-home applications, patient-doctor health monitoring or blind and impaired assisting. Because of a few characteristics that differentiate them from otherwise similar ad hoc wireless nets such as

* Work supported by the *Swiss National Science Foundation*, ref. PBGE2 - 112864.

** Work partially supported by the E.U. IST program num. IST-2005-15964 (AEOLUS).

MANETS, sensor nets raise a multitude of algorithmic challenges [4, 5]. Among characteristic that make sensor nets very different [6] are strong resource limitations (energy, memory, processing power), high-density and size (which can be orders of magnitude greater than for other technologies) and the necessity to operate unattended under the constraint of environmental hazard.

Problem statement: We consider the problem of routing messages in a localized sensor net, a problem commonly called geographic routing (or georouting). We address the problem of finding a simple and efficient georouting algorithm which delivers messages with high success rate even in regions of low density (routing holes) and large communication blocking obstacles. The routing algorithms we allow ourselves to consider should be *lightweight*, *on demand* (thus making our algorithm all-to-all), *efficient* and *realistic*.

On the importance of geographic routing: According to [7], “*the most appropriate protocols [for sensor nets] are those that discover routes on demand using local, lightweight, scalable techniques, while avoiding the overhead of storing routing tables or other information that is expensive to update such as link costs or topology changes*”. This is due to the severe resource limitations of sensor devices and the high dynamics of the ad hoc networks they spontaneously establish. In view of this, geographic routing is very attractive [6, 7]. The early and simple greedy georouting protocols [8] where messages are sent to the neighbour maximising progress towards the destination meet those idealistic requirements: the only information required to route is, assuming the nodes are localized, the destination of the message. One may wonder how realistic the assumption of localized nodes is, and to what extent it confines georouting to a specialised niche. Our point of view is that “*...geographic routing is becoming the protocol of choice for many emerging applications in sensor networks...*” [9] because location aware nodes are likely to be available since “*...in many circumstances, it is useful and even necessary for a node in a wireless sensor network to be aware of its location in the physical world. For example, tracking or event-detection functions are not particularly useful if the [sensor net] cannot provide any information where an event has happened*” [6]. Node localization is achievable through one of the many localization systems that use a combination of GPS like technology to localize a few beacon nodes followed by a distributed localization protocol [6, 7, 10–12]. Interestingly, it turns out that georouting can even be used when nodes are not location aware by using *virtual coordinates* as was proposed in [13].

State of the Art: The major problem of the early greedy georouting algorithms [8] is the so called *routing hole problem* [6, 7, 14] where messages get trapped in “local minimum” nodes which have no neighbours closer to the destination of the message than themselves. The incidence of routing holes increases as network density diminishes and the success rate of the greedy algorithm drops very quickly with network density. In order to bypass routing holes (and obstacles), very ingenious georouting algorithms have been developed. The most successful ones are probably the celebrated GFG and GPSR algorithms [15, 16] (as well as incremental improvements such as GOAFR [17], c.f. [18] for details) which have the very strong property of *guaranteeing successful routing* if the network is con-

nected. GFG and GPSR are very similar and were, to our knowledge, developed independently. We use the encompassing term of *face routing* algorithms to refer to GFG, GPSR and their incremental successors. They all share, as a central idea, the use of a greedy propagation phase until the message reaches a local minimum. At this point a temporary *rescue mode* is used to escape the local minimum. The rescue mode uses (a variant of) the FACE algorithm originally described in [15] where messages are routed along the faces of the polygons of a planar subgraph of the communication graph. The use of a planar subgraph, which is necessary for face routing, is a crucial and restrictive characteristic: it implies that a graph planarization component has to be included in the routing algorithm. Until very recently, a major pitfall [19] of face routing algorithms was that no practical planarization algorithm was known: “... *all currently proposed geographic routing algorithms rely on idealized assumptions about radios and their resulting connectivity graphs [...] which are grossly violated by real radios [...] causing persistent failures in geographic routing, even on static topologies*” [20]. In a recent breakthrough paper [21] the first practical planarization algorithm with reasonable message overhead was proposed, *lazy cross-link removal* (LCR). Although reasonable, at least in static nets, LCR still induces a high topology maintenance overhead to discover impairing “cross-links”, c.f. [21]. Another interesting approach is the BOUNDHOLE algorithm from [22] which uses the TENT rule to discover local minimum nodes and then “bounds” the contour of routing holes. Although it has a high overhead when compared to the algorithm we propose, the information gained during the contour discovery phase may be used for other application than routing such as path migration, information storage mechanisms and identification of regions of interest, c.f. [22] for details. Some other solutions are the probabilistic PFR, VTRP, LTP and CKN protocols [23–26]. These approaches are very different from face routing algorithms in the sense that, at the cost of accepting lower success rates (particularly in low density networks), they induce very little topology maintenance overhead. Another drawback is that they fail to bypass large obstacles [27].

Our approach: To overcome the limitations of previous approaches we propose a new algorithm: **GeoRoutIng** around obstaCles (**GRIC**), pronounced “Greek” in reference to its design location: the University of Patras in Greece. The main idea of **GRIC** is to appropriately combine movement directly towards the destination (to optimize performance) with an inertia effect. Inertia forces messages to keep moving along the “current” direction and to closely follow the perimeter of obstacles in order to efficiently bypass them. Inertia permits to get out of many routing holes and to bypass some quite strongly blocking convex obstacles. To further improve our algorithm, a “right-hand rule” inspired component is used in combination with a virtual compass. The right-hand rule is a well known “wall follower” technique to get out of a maze [28] which is also used for face routing. However, unlike face routing algorithms, **GRIC** has the advantage of using the right-hand rule on the complete communication graph, thus eliminating the planarization phase overhead. The right-hand rule permits to route messages around large obstacles, not only convex but also concave. It is useful even in the

absence of obstacles, making the success rate of **GRIC** close to 100% even for very low density networks. We implement our algorithm and comparatively evaluate its performance against those of other representative algorithms (greedy, LTP and FACE). We focus on two performance measures: success rate and hop count. We study the impact on performance of several types of obstacles (both convex and concave) and representative regimes of network density.

Strengths of our approach: **GRIC** is very simple (for example when compared to face routing relying on LCR planarization) and thus easy to implement. It has a very high success rate, even in the case of low density networks. It is capable of bypassing large emission blocking obstacles (although for the hardest obstacles performance decreases with network density) using a short path, close to optimal in the absence of global knowledge of the network. It is particularly suitable for highly dynamic networks where links go up and down, e.g. because of environmental fluctuation or network congestion. This follows from the fact that, for a start, **GRIC** has absolutely no topology maintenance overhead (the only information required, at the node level, is a list of outbound neighbours) and from the fact that, as shown in our experiments, **GRIC** is not only robust when confronted with link failure: it also has the surprising property of actually *performing better when confronted to limited link instability*. To our knowledge, the near 100% success rate of **GRIC** (without obstacles) and its effective obstacle avoidance property is unique among lightweight routing protocols. It also offers a competitive alternative to face routing, probably with a different dedicated application niche: **GRIC** would be preferred for highly dynamic networks whereas face routing may be preferred in the case of more stable networks where the planarization overhead is paid off over time if the topology is static, implying that planarization does not need to be recomputed frequently. As a consequence, we feel that **GRIC** considerably improves the state of the art of geographic routing.

2 The **GRIC** Algorithm

Sensor Net Model: When a node needs to route a message according to the **GRIC** algorithm, it needs some network topology information. More precisely, nodes should be aware of their 1-hop away outbound neighbours, as well as their coordinates. In mathematical language, this is equivalent to assuming a *directed dynamic communication graph* (i.e. connectivity can change over time) embedded in the Euclidean plane. Although this may seem quite abstract at first sight, it is in fact very realistic: **GRIC** is a network layer protocol, it therefore relies on the data-link, MAC and physical layers. Many different MAC and data-link protocols exist, and although the study of the impact of different possible combinations is beyond the scope of this paper, most of them would provide, at the network layer, the level of abstraction we assume in this paper, c.f. [18] for details. A final minor assumption is that we allow messages to piggy-back $\mathcal{O}(1)$ bits of information encoding the position of the last node visited, the position of the targeted message destination and a mark-up flag.

Overview: Like face routing algorithms, **GRIC** uses two different routing modes:

a normal mode called *inertia mode* and a *rescue mode*. Intuitively, the inertia mode is used when the message makes progress towards the destination, and the rescue mode when it is going away from the destination. The inertia mode is inspired by physics and we use it to control the trajectory of messages inside the network. Messages are “attracted” to their destination but also have an incentive to follow the “straight line”, like a celestial body is attracted in a planet system. The rescue mode adds a right-hand rule component to the inertia mode. The right hand-rule is a “wall follower” technique to get out of a maze [28], also used by face routing algorithms. GRIC combines it with inertia to bypass complex obstacles by following their contour.

Routing with inertia: We consider a node n at position p receiving a message m . n needs to take a routing decision for m . First, n reads the information piggy-backed on m to learn p' and p'' , the position of the node which sent m to n and the destination position of m respectively, as illustrated in figure 1(a). Next, n computes $v_{\text{prev}} = p - p'$ and $v_{\text{dest}} = p'' - p$. If attached at position

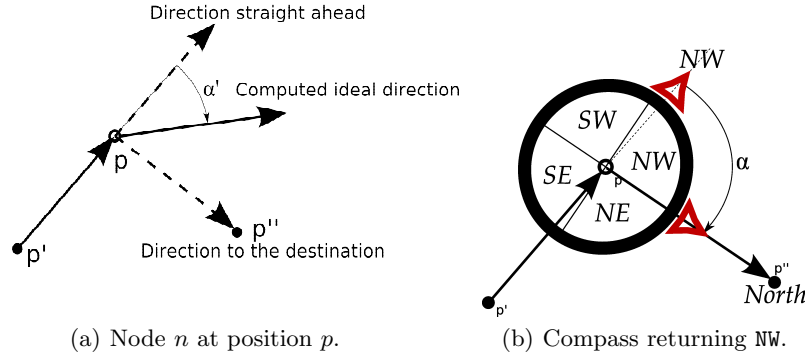
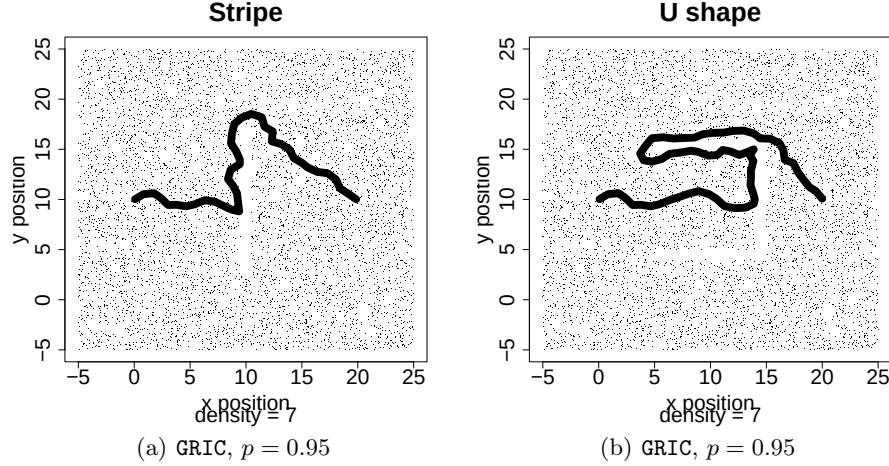


Fig. 1. The compass device.

p , v_{prev} is a vector pointing in the previous direction travelled by m whereas v_{dest} is a vector pointing in the direction of m 's destination. Using elementary trigonometry, n computes the angle between v_{prev} and v_{dest} , which is uniquely defined if we allow only values in $[-\pi, \pi[$. n will try to send m in a computed ideal direction $v_{\text{ideal}} = \mathcal{R}_{\alpha'} \cdot v_{\text{prev}}$, where $\mathcal{R}_{\alpha'}$ is a rotation matrix of angle α' defined by $\alpha' = -\beta\pi$ if $\alpha < -\beta\pi$, $\alpha' = \beta\pi$ if $\alpha > \beta\pi$, $\alpha' = \alpha$ otherwise. The parameter β ranges in $[0, 1]$. If $\beta = 1$, $v_{\text{ideal}} = v_{\text{dest}}$ and inertia routing is equivalent to the greedy routing algorithm. At the other extreme when $\beta = 0$, $v_{\text{ideal}} = v_{\text{prev}}$ and the ideal direction is equal to the previous direction: inertia is maximal. The inertia factor can thus be controlled by adjusting β . In our simulations, setting $\beta = \frac{1}{6}$ proved to be a good choice for practical purposes. The final routing decision of n is to send m to its neighbour n_2 maximizing progress towards the ideal

direction v_{ideal} , i.e. n_2 maximizes the scalar product $\langle v_{\text{ideal}} | \text{pos}(n_2) - p \rangle$.

Routing around obstacles: Our experiments show that inertia routing by-



passes routing holes with high probability and routes messages around some large convex obstacles such as the one in figure 2(a). It is therefore used as the normal routing mode of **GRIC**. However, more complex concave obstacles as in figures 2(b), 2(c) and 2(d) cannot be bypassed by inertia routing and we therefore add a *rescue mode* to **GRIC**. The first difficulty is to know *when* to switch to rescue mode. A *virtual compass device* and the use of a flag fulfill this purpose. We keep notations of the previous section and consider that node n receives a message m for which it needs to take a routing decision and describe below the steps required to implement the obstacle avoidance feature of **GRIC**.

- *The compass device*: n considers the north to be p'' , i.e. the destination of m , and it wants to know what cardinal direction describes the last hop of m : north-west, north-east, south-west or south-east, c.f. figure 1(b). The answer depends on α : the virtual compass indicates SW if $\alpha \in [-\pi, -\pi/2[$, NW if $\alpha \in [-\pi/2, 0[$, NE if $\alpha \in [0, \pi/2[$ and SE otherwise.

- *The flag*: Intuitively, when m is being routed around an obstacle the flag should be up, otherwise it should be down. We metaphorically consider a walker following the path along which m is routed. When the walker follows the path and assuming m stays close to the obstacle's contour, two cases can occur: the obstacle's perimeter is either on the right or on the left of the walker. When this is so, we say the message is routed around the obstacle according to the right or left-hand rule respectively and **GRIC** acknowledges it by raising the flag and tagging it with SW or SE respectively. Formally, when n receives m , it starts by adjusting the flag's value in the following way. If the flag is down, the algorithm looks at the compass. If the compass points north, the flag stays down. However, if the compass points south, *the flag is raised*. and tagged with SW or SE respec-

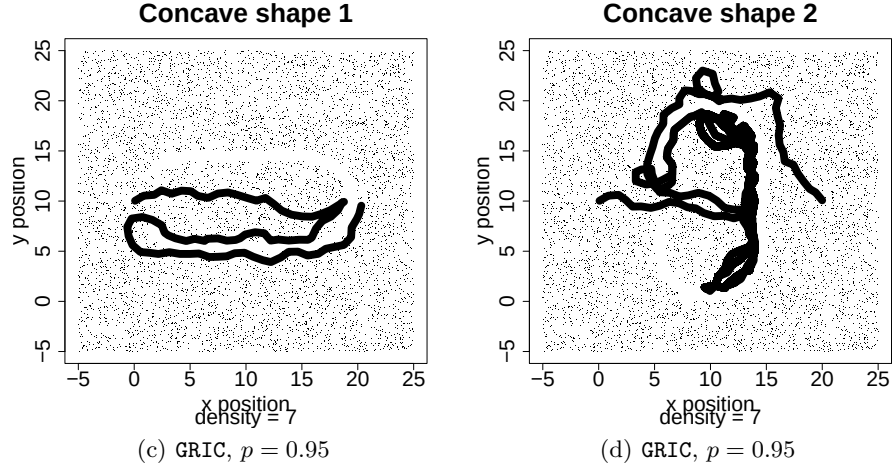


Fig. 2. Typical behavior for different obstacle shapes.

tively. If the flag is up, n has two options: leave the flag up (without changing the tag), or put the flag down. The flag goes down only if it was SW-tagged and the compass points NW, or if the flag was SE-tagged while the compass points NE.

- *Mode selection:* n receives the message m and it has to take a routing decision. The decision is taken in steps. First, n adjusts the flag according to the procedure described above. Next, it chooses whether to operate in the normal mode or whether it should switch to rescue mode. Only then will the routing decision be made and we shall soon describe how, but we first describe how to choose the routing mode. When the flag is down the normal mode is always used. When the flag is up, it depends. Suppose the flag is up and for the sake of simplicity let us assume it is SW-tagged, the other case being symmetric. Note that by case assumption this implies that the compass does not point NW, since otherwise the flag would be down. Also, in this case, intuitively GRIC tries to route around the obstacle using the right-hand rule. We next have to consider two cases. If the compass points SW, recalling the definition of v_{prev} , v_{ideal} and α , it is easy to see that by case assumption v_{ideal} is obtained by applying to v_{prev} a rotation of angle α' with α' in $[-\pi/6, 0]$, c.f. figure 1. In other words, inertia routing gives the message an incentive to turn to the right. This is consistent with the right-hand rule and GRIC thus chooses the normal routing mode. If the message points SE or NE, a similar reasoning shows that α' is in $[0, \pi/6]$ and that inertia routing will give an incentive for the message to turn left. However, this will get the message away from the obstacle (in the expected case where the obstacle's contour is indeed closely followed and kept on the right of the message). This is contrary to the right-hand rule idea and therefore GRIC will switch to rescue-mode.

- *The rescue mode:* By case assumption, the flag is up and we assume without loss of generality the tag to be SW, i.e. the right-hand rule applies. The selec-

tion procedure previously described chooses rescue mode when inertia routing would give the message an unwanted incentive to turn left by computing an α' value in $[0, \pi/6]$. Intuitively, the rescue mode simply inverts the rotation angle α' in the following way: let $\alpha_2 = -\text{sign}(\alpha)(2\pi - |\alpha|)$. v_{ideal} is then defined by $v_{\text{ideal}} = R_{\alpha'_2} \cdot v_{\text{prev}}$, where $\alpha'_2 = \beta\alpha_2$ and where β is the same inertia conservation parameter as the one used for inertia routing. Putting all things so far together, GRIC is formally described by the (non-randomized version of) algorithm 1.

Algorithm 1 GRIC, running on the node n which is at position $\text{pos}(n)$.

```

1: if the flag is down and compass indicates SW (or SE) then
2:   raise the flag and tag it with SW (or SE respectively)
3: else if the flag is up and the tag is SW (or SE) then
4:   lower flag if the compass points NW (or NE respectively)
5: Decide if mode should be normal or rescue {c.f. “Mode selection” subsection}
6: if mode = normal then
7:    $\gamma := \alpha'$  {c.f. “Routing with inertia” section}
8: else if mode = rescue then
9:    $\gamma := \alpha'_2$  {c.f. “Rescue mode” subsection}
10:  $v_{\text{ideal}} := R_{\gamma} \cdot v_{\text{prev}}$ 
11: Let  $\mathcal{V}$  be the set of neighbours of  $n$  and let  $\mathcal{V}'$  be an empty set.
12: if running the random or non-random version of GRIC then
13:   set  $p = 0.95$  or  $p = 1$  respectively
14: for all  $v \in \mathcal{V}$  do
15:   add  $v$  to  $\mathcal{V}'$  with probability  $p$ 
16: Send the message to the node  $n_2 \in \mathcal{V}'$  maximizing  $\langle v_{\text{ideal}} | \text{pos}(n_2) - \text{pos}(n) \rangle$ 

```

Randomization and robustness: While designing GRIC, we decided to test its robustness when confronted to link instability. We found that *it performs better in the case of limited link instability*. Although surprising and perhaps counterintuitive, this feature of GRIC is easy to understand. Indeed, routing failure is likely to occur when a message deterministically starts looping around a local minimum. A small random perturbation breaks GRIC’s determinism and messages eventually escape the loop. This is a very nice property of GRIC which also suggests an easy way to improve its behavior in the context of stable networks: each time n needs to take a routing decision for a message m it starts by temporarily and randomly discarding each of its outbound links with probability p , c.f. line 12 of algorithm 1. For practical purposes experiments show the choice of $p = 0.95$ to be good. Further decreasing the value of p , to the best of our understanding, is tantamount to decreasing node density and thus performance slowly decreases.

3 Experiments

We validate the performance of GRIC through extensive experiments. A single experiment consists of randomly deploying a sensor net. A message is then gen-

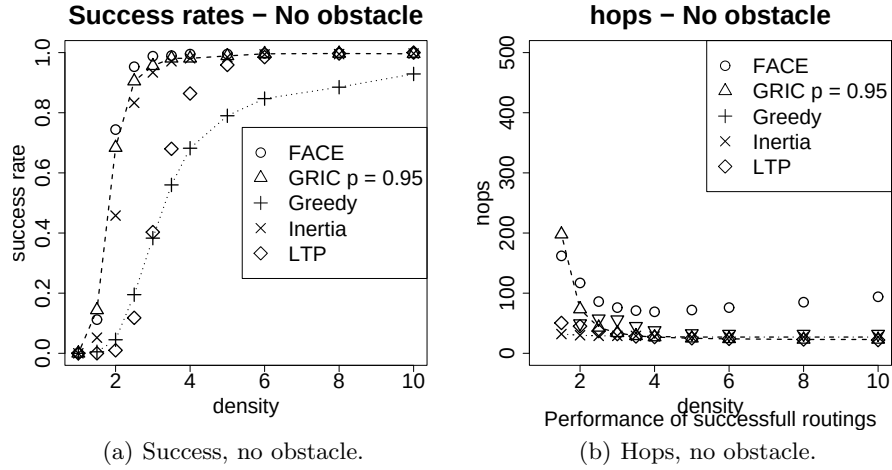
erated by a single node. The message has a destination, and the network tries to route the message to it. The experiment is successful if the message reaches its destination before a given timeout value, it is deemed to have failed otherwise. To verify the quality of successful outcomes, we measure the length in hops of the path that leads the message from source to destination. As a first experimental validation of GRIC and for practical purposes we resolve ourselves to simulation rather than experimenting with real sensor nets: even small size sensor nets are still quite prohibitively expensive and choosing and implementing a full protocol stack (MAC and data-link) on top of which our network layer algorithm operates implies a substantial amount of work which we delay for possible future work.

Simulation platform: We developed a high-level simulation platform using the Ruby programming language. As was previously explained, the GRIC algorithm is a network layer algorithm assuming a list of reliable collision free data-links to be made available by lower protocol stack layers. This assumption is weak since most physical/MAC/data-link suites would indeed provide this level of abstraction. We choose as a communication model the unit disc graph. Arguably, this model is the most commonly used for sensor net simulations. However, we acknowledge that this choice is not completely satisfactory and that more realistic communication graph models would be more appropriate. Defining a reasonable model suitable for simulation purposes is a challenging task. To our knowledge, only recently did the research community start to investigate this problem [29, 30] and defining such a model is beyond the scope of this paper. Nevertheless, because our algorithm is robust in the presence of link failure, because it requires only one-hop-away neighbourhood discovery, because it does not even require links to be symmetric and because it implies *absolutely no topology maintenance*, we are confident that the unit disc communication graph is good enough to give a reasonable approximation of the behavior GRIC will have in real sensor nets.

Simulation details: We deploy randomly and uniformly N sensor nodes in the region of the Euclidean plane defined by $\{(x, y) \in \mathbb{R}^2 \mid -5 \leq x, y \leq 25\}$. The density of the network d is defined as $N/900$. Using the unit disc graph model, the expected number of neighbours per node is thus close to $d \cdot \pi$. A message m is generated at the point $(0, 10)$ and attached to the closest node. The destination of m is $(20, 0)$. m is propagated in the network according to the routing protocol considered. The experiment is considered successful if m gets within distance 1 of its destination (there may not be a node at the exact destination of the message). The outcome is a failure if m does not reach its destination in less than N steps. For precaution, we also consider the outcome to be failure if m approaches within distance 1 from the border of the network.

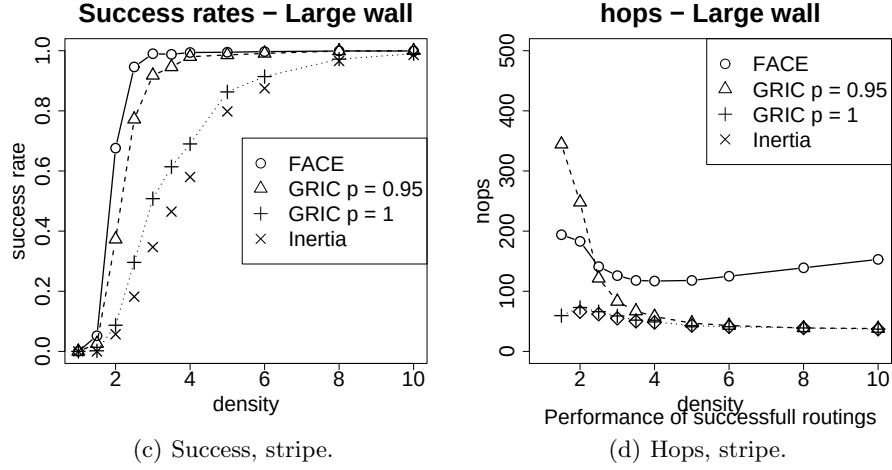
Preliminary results: We first consider the case where no obstacle is added to the network. We considered the FACE algorithm of [15]. FACE is not the most competitive algorithm in the face routing family in terms of path length, because it always runs in rescue mode. However, like all face routing algorithms it has the very strong “guaranteed delivery” property to always route a message to its destination if a path exists from the source. In view of this, figure 3(a)

reveals the following: in the absence of obstacles, the random version of GRIC (with $p = 0.95$) has almost 100% success rate even for low densities where the network is disconnected with high probability since the success rate is very close to that of FACE. We also observe in figure 3(b) that the path length of GRIC is close to optimal since it competes with greedy routing which is known to find very short paths, c.f. [18]. The performance of inertia routing is less than for GRIC (and slightly less than GRIC with $p = 1$, but for clarity we do not show this on the graph, c.f. [18] for details), but is still quite good and it outperforms the LTP protocol of [25]. (LTP uses a limited backtracking, we allowed LTP a maximum of 5 consecutive backtracking steps).



Obstacles: We consider four different types of large communication blocking obstacles.

- Large wall:* The first obstacle is convex, c.f. figure 2(a). The random version of GRIC bypasses this large convex obstacle with high probability, c.f. figure 3(c). In terms of network design, it makes no sense to deploy a sensor net that has high probability of being disconnected and thus of being non-operational. We see from the performance of FACE that this critical density is around $d = 3$, and even for such a low density the performance of GRIC is very good, both in terms of success rate and path length. It is interesting to see that in the presence of this large obstacle, the randomization ($p = 0.95$) implies a high improvement over the deterministic version ($p = 1$), which not only implies that GRIC is resistant to link instability, it actually performs better in the presence of limited link instability. Interestingly, although not competitive we see that inertia is capable of bypassing the obstacle when the density is not too low. In [27], LTP was shown to have a good obstacle avoidance property. However, LTP never routes around a large wall such as the one we consider in this work (thus LTP is not included on plots). Therefore, to the best of our knowledge



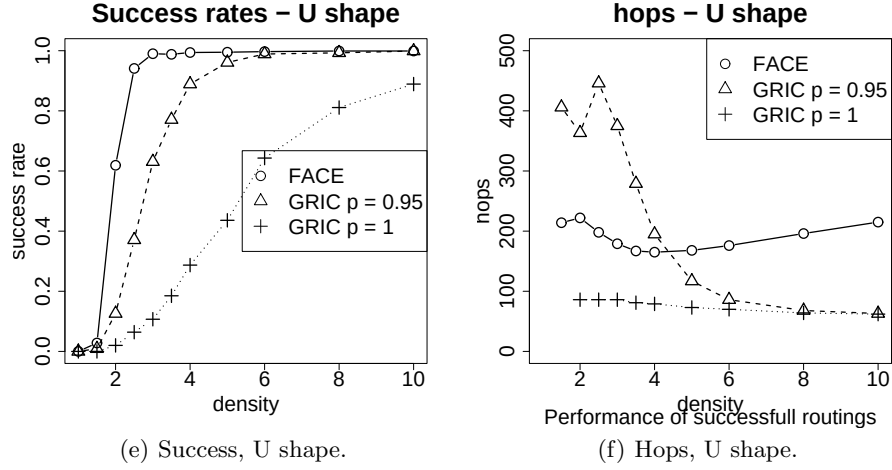
GRIC outperforms other lightweight georouting protocols, both in the presence and absence of obstacles. In figure 3(d) we verify that path length is kept very low, considering that the message has no a priori knowledge of the network and discovers the obstacle only when reaching it.

- *U shape obstacle*: First of all, using a rule of thumb when looking in figures 2(b) and in figure 3(f) shows that the deterministic version of GRIC (i.e. when $p = 1$) rarely routes messages successfully to their destination but when it does so it uses a near optimal path. In light of this observation, we conclude that GRIC ($p = 0.95$) *bypasses this hard concave obstacle and uses a short path*. However, the performance is only good when the network density is around 5 and higher. This is a medium density for sensor nets since, again, a density below 3 is probably not acceptable in terms of network design since it yields a disconnected network with non-negligible probability.

- *Concave shape 2*: We skip results for the first concave shape in figure 2(c) because they are similar to those of the U shaped obstacle and turn to the final obstacle. As seen in figure 2(d), this obstacle is problematic. The message is routed out of the obstacle only to fall back in with high probability. As a consequence, the random version of GRIC only reaches acceptable performances for very high network densities: success rate is bad for densities below 5 and path length is prohibitive even for densities below 8, c.f. figures 3(g) and 3(h).

4 Conclusion

We have studied geographic routing in the presence of hard communication blocking obstacles and proposed a new way of routing messages which substantially improves the state of the art by somehow combining the best of two worlds: the lightweight (no topology maintenance overhead), robustness (to link failure) and simplicity of the greedy routing algorithm with the high success rates and



obstacle avoidance features of face routing. The simplicity of **GRIC** suggests that it would be a protocol of choice for routing in mobile networks. We shall investigate this in future work. We have shown that **GRIC** resists (and actually performs better) in the presence of limited link failure. Future work will investigate this matter more in depth, as well as the question of localization errors. At first sight, there seems to be no reason to believe **GRIC** to be sensitive to them. **GRIC** proposes an alternative to the face family of protocols. We believe it has a slightly different application niche and is preferable in the case of highly dynamic networks (because frequent topology changes increase the topology maintenance overhead of the planarization phase required for face routing), whereas face routing may be better in sparse but stable topologies where some overhead is acceptable. Deeper understanding of the differences between face and **GRIC** routing will require further investigation, more realistic communication graph models and possibly turning to real world experiments.

References

1. J. M. Rabaey, M. Josie Ammer, J. L. da Silva, D. Patel, and S. Roundy. Picoradio supports ad hoc ultra-low power wireless networking. *Computer*, 2000.
2. B. Warneke, M. Last, B. Liebowitz, and K.S.J. Pister. Smart dust: communicating with a cubic-millimeter computer. *Computer*, 2001.
3. I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 2002.
4. M. Haenggi. Challenges in wireless sensor networks. In *Handbook of Sensor Networks: Compact Wireless and Wired Systems*. CRC Press, 2005.
5. D. Estrin, R. Govindan, J. Heidemann, and S. Kumar. Next century challenges: Scalable coordination in sensor networks. In *Mobile Computing and Networking*. ACM, 1999.

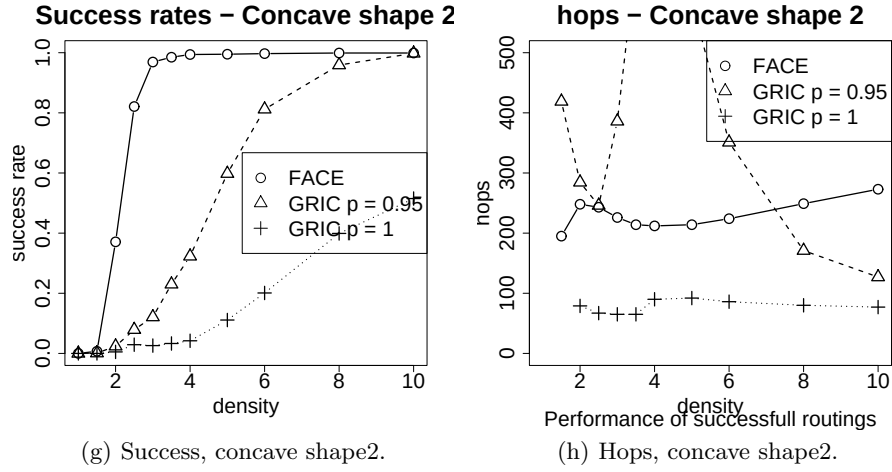


Fig. 3. Summary of simulation results.

6. H. Karl and A. Willig. *Protocols and Architectures for Wireless Sensor Networks*. Wiley, 2005.
7. F. Zhao and L. Guibas. *Wireless Sensor Networks, an Information Processing Approach*. Elsevier, 2004.
8. G. G. Finn. Routing and addressing problems in large metropolitan-scale inter-networks. Technical report, Information Sciences Institute, 1987.
9. K. Seada, A. Helmy, and R. Govindan. On the effect of localization errors on geographic face routing in sensor networks. In *Information Processing in Sensor Networks*, 2004.
10. J. Beutel. Location management in wireless sensor networks. In *Handbook of Sensor Networks: Compact Wireless and Wired Systems*. CRC Press, 2005.
11. J. Hightower and G. Borriello. Location systems for ubiquitous computing. *Computer*, 2001.
12. Pierre Leone, Luminita Moraru, Olivier Powell, and Jose Rolim. A localization algorithm for wireless ad-hoc sensor networks with traffic overhead minimization by emission inhibition. In *Algorithmic Aspects of Wireless Sensor Networks*, 2006.
13. A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, and I. Stoica. Geographic routing without location information. In *Mobile Computing and Networking*, 2003.
14. N. Ahmed, S. S. Kanhere, and S. Jha. The holes problem in wireless sensor networks: a survey. *SIGMOBILE Mob. Comput. Commun. Rev.*, 2005.
15. P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. In *Discrete Algorithms and Methods for Mobile Computing and Communications*, 1999.
16. B. Karp and H. T. Kung. GPSR: greedy perimeter stateless routing for wireless networks. In *Mobile Computing and Networking*, 2000.
17. F. Kuhn, R. Wattenhofer, and A. Zollinger. Worst-case optimal and average-case efficient geometric ad-hoc routing. In *Mobile ad hoc Networking & Computing*, 2003.

18. Olivier Powell and Sotiris Nikolettseas. Geographic routing around obstacles in wireless sensor networks. Technical report, Computing Research Repository (CoRR), 2007.
19. Y.-J. Kim, R. Govindan, B. Karp, and S. Shenker. On the pitfalls of geographic face routing. In *Foundations of mobile computing*, 2005.
20. Y.-J. Kim, R. Govindan, B. Karp, and S. Shenker. Geographic routing made practical. In *Networked Systems Design & Implementation*, 2005.
21. Y.-J. Kim, R. Govindan, B. Karp, and S. Shenker. Lazy cross-link removal for geographic routing. In *Embedded Networked Sensor Systems*, 2006.
22. Q. Fang, J. Gao, and L. Guibas. Locating and bypassing holes in sensor networks. *Mobile Networks and Applications*, 2006.
23. I. Chatzigiannakis, T. Dimitriou, S. Nikolettseas, and P. Spirakis. A probabilistic forwarding protocol for efficient data propagation in sensor networks. *Journal of Ad hoc Networks*, 2006.
24. T. Antoniou, I. Chatzigiannakis, G. Mylonas, S. Nikolettseas, and A. Boukerche. A new energy efficient and fault-tolerant protocol for data propagation in smart dust networks using varying transmission range. In *Annual Simulation Symposium (ANSS)*. ACM/IEEE, 2004.
25. I. Chatzigiannakis, S. Nikolettseas, and P. Spirakis. Smart dust protocols for local detection and propagation. *Journal of Mobile Networks (MONET)*, 2005.
26. I. Chatzigiannakis, A. Kinalis, and S. Nikolettseas. Efficient and robust data dissemination using limited extra network knowledge. In *IEEE Conference on Distributed Computing in Sensor Systems (DCOSS)*, 2006.
27. I. Chatzigiannakis, G. Mylonas, and S. Nikolettseas. Modeling and evaluation of the effect of obstacles on the performance of wireless sensor networks. In *Annual Simulation Symposium (ANSS)*. ACM/IEEE, 2006.
28. A. Hemmerling. *Labyrinth Problems: Labyrinth-Searching Abilities of Automata*. B.G. Teubner, Leipzig, 1989.
29. Y. Yu, B. Hong, and V. Prasanna. On communication models for algorithm design for networked sensor systems: A case study. *Pervasive and Mobile Computing*, 2005.
30. A. Cerpa, J. Wong, L. Kuang, M. Potkonjak, and D. Estrin. Statistical model of lossy links in wireless sensor networks. In *Information processing in sensor networks*, 2005.