Geographic Routing with Partial Position Information

Tony Ducrocq, Michael Hauspie, Nathalie Mitton

Geographic Routing with Partial Position Information*

Tony Ducrocq[1], Michaël Hauspie[2] and Nathalie Mitton[1]
[1]*Inria Lille - Nord Europe.*
{*tony.ducrocq, nathalie.mitton*}*@inria.fr*

[2]*Université Lille1*
michael.hauspie@univ-lille1.fr

Keywords: Wireless Sensor Networks, Geographic Routing Algorithm, Position Based Routing.

Abstract: Geographic routing protocols show good properties for Wireless Sensor Networks (WSN). They are stateless, local and scalable. However they require that each node of the network is aware of its own position. While it may be possible to equip each node with GPS receiver, even if it is costly, there are some issues and receiving a usable GPS signal may be difficult in some situations. For these reasons, we propose a geographic routing algorithm, called **HGA**, able to take advantages of position informations of nodes when available but also able to continue the routing in a more traditional way if position information is not available. We show with simulations that our algorithm offers an alternative solution to classical routing algorithm (non-geographic) and offers better performances for network with a density above 25 and more than 5% of nodes are aware of their position.

1 Introduction

Wireless sensor networks consist of sets of mobile wireless nodes without the support of any fixed infrastructure. Such wireless sensor networks offer great application perspectives. Sensors are tiny devices with hardware constraints (low memory storage, low computational resources) that rely on battery. Sensor networks thus require energy-efficient algorithms to make them work properly in a way that suits their hardware features and application requirements.

Low power sensor nodes have limited transmission power, thus they can communicate only to a limited number of nodes. This set of nodes is called the neighborhood of the node. In order to send messages at longer range, nodes are using multi-hop communication. Multi-hop communication means that data will need to be routed from source to destination by other nodes. An efficient way to route messages is to use nodes position information. A node uses a metric based on its own position, its neighbors positions and the destination position in order to choose the next hop for the route. To use such a technique it is possible to equip each node of the network with a GPS receiver or to configure them at setup with their position if they are static.

On the other hand there exist non geographic routing algorithms. They do not require node position to route data but only neighborhood knowledge.

In the Smart Cities context, it is likely that the network is heterogeneous (Al-Hader et al., 2009). Some nodes may be aware of their position while some other are not. The reason to not equip nodes with GPS receiver may be to save money or energy. It is also not possible to setup position of mobile nodes as it will change over time. Furthermore, even if all nodes are equipped with a GPS receiver, it is possible that some of them do not receive the signal because of environmental factors. For instance, two districts of a city may be connected through a tunnel, in which GPS signal is not received.

In such contexts, the availability of a routing algorithm taking advantages of nodes positions when possible but also functioning when the position is not available is interesting. Thus, we introduce the **HGA** algorithm (**H**ybrid-**G**reedy-**A**ODV). The algorithm works like the Greedy geographic routing while position information allows it, and if position information is not available, a route request message is sent in the node neighborhood in order to find a path toward the destination.

*This work is partially supported by CPER Nord-Pas-de-Calais/FEDER Campus Intelligence Ambiante and ANR ECOTECH BinThatThinks.

The rest of the paper is organized as follows: Section 2 presents related work concerning geographic and pseudo geographic routing algorithms. We introduce some background works on which our algorithm relies in Section 3. Section 4 describes the **HGA** algorithm operation in details and Section 5 present simulation and results of its performances. Finally we conclude in Section 6.

2 Related Works

To the best of our knowledge, there is no existing routing solution that considers that only a subset of nodes is aware of its position. Classical routing algorithms and those using virtual coordinates are the closest approaches to the one we propose. In this section we describe related works as well as works on which our proposal relies.

The main idea of geographic routing protocols is to route data closer to the destination at each step of the routing. The Greedy algorithm, on which part of our work relies, chooses the closest node to the destination in a node's neighborhood as the next forwarder of the data packet. The Greedy algorithm and some variants from literature are shown on Figure 1.

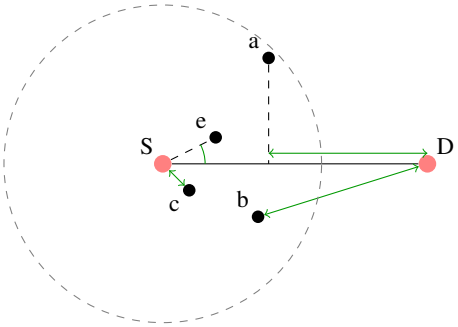


Figure 1: Comparison of geographic algorithms. Node *a* is chosen by *MFR* algorithm, node *b* by *Greedy*, node *c* by *NFP* and node *e* by *Compass*.

Some algorithms allow the deduction of position for nodes by using their neighbors (Čapkun et al., 2002), (Ermel et al., 2005). Nevertheless, these solutions exhibit poor performances due to the difficulty to determine distance between two nodes without specialized hardware (Benkic et al., 2008). Moreover, Seada et al. (Seada et al., 2004) showed that a small error on position can generate several losses of packets in geographic routing algorithms.

Routing algorithms over virtual coordinates allow the benefits of geographic routing algorithms without the need of real geographic position of nodes. The main drawback of these solutions resides in the setup of the virtual coordinates system. Indeed, it is needed to flood all the network several times at the beginning of the network lifetime.

The VCap solution (Caruso et al., 2005) relies on virtual coordinates and a greedy routing. Some particular nodes are chosen for their interesting properties to act as starting points for the different coordinates of the system. These nodes are named anchors. The coordinate system relies on the distance in number of hops between a node and the different anchors. A predefined sink node is used to initiate anchors election (or any other node). This node, by broadcasting a message in the whole network allows to elect a first anchor. At its turn, this anchor broadcast a message in the whole network to elect a second one and so on. Anchors are chosen to be close to the network boundaries a far away to each other to limit duplicates in coordinates. In VCap the network is flooded at least 5 times. Otherwise, because uniqueness of coordinates is not guaranteed, the delivery can not be guaranteed.

The *cost over progress* concept idea (Stojmenovic, 2006) is to optimize the ratio between the cost of a transmission by the progress made by the transmission. Vcost (Elhafsi et al., 2007) relies on this idea and on VCap coordinate system to reduce energy needs for the routing.

The LTP algorithm (Chávez et al., 2007) uses a root and a tree construction as the only coordinate system instead of anchors. Thanks to the tree, this routing algorithm guarantees delivery but the chosen routes may be long. Above all, the tree construction is costly and hard to maintain.

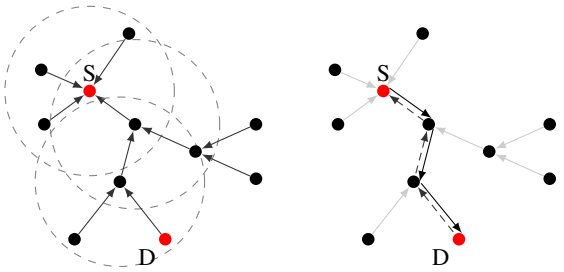
HECTOR (Mitton et al., 2012) combines both coordinates systems of LTP and VCap. Its chooses the best node in the VCap coordinate system at the same level or that allow a progress in LTP coordinate system. HECTOR is then able to guarantee delivery of packets and to reduce the energy used for the routing.

Other virtual coordinates solutions have been proposed, some are without guaranteed delivery (Benbadis et al., 2006), (Fang et al., 2005), (Niculescu and Nath, 2001), some other with guaranteed delivery (Chávez et al., 2007), (Liu and Abu-Ghazaleh, 2008). However, algorithms using virtual coordinates have the drawback to not support node mobility. Furthermore, it is needed to flood the whole network at the beginning, introducing high latency at startup when the network is big.

3 Background

Our proposal relies on *route requests* and *route replies* similar to those used in AODV (Perkins and

Royer, 1999). We introduce their behavior in the following. Figure 2 shows a route search in AODV.



(a) *Reverse path* setup. (b) *Forward path* setup.

Figure 2: Route search example from S to D in AODV.

AODV is a non-geographic routing, totally reactive. Indeed, routes are established only on-demand, when they are needed. To route a data packet toward the destination, a node broadcasts a RREQ (route request) in its neighborhood. Each node receiving the RREQ keeps track of it until expiration delay expires (defined to 3000 ms in AODV). Keeping track of RREQ allows the creation of the *reverse path* as shown in Figure 2(a) that will be used when destination is found. If a node S does not know a route towards the destination D, S is not the destination and it has not received this RREQ earlier, it broadcasts a RREQ at its turn in its neighborhood. When a node knows a route toward the destination, or is itself the destination, it sends a RREP packet (route reply) to the node that has sent the RREQ. The RREP packet then follows the *reverse path* and each node on the route saves the source of the RREP in its neighborhood table in order to create the *reverse path*. Figure 2(b) shows the creation of the *forward path* and expiration of *reverse paths*.

4 Proposal

In this paper, we propose **HGA**, a **H**ybrid **G**reedy-**A**ODV algorithm. The main idea of **HGA** is to apply a greedy geographic routing whenever it is possible. When a greedy routing is not possible, a RREQ is sent in the k-neighborhood of the node. When a node needs to route a data packet and none of its neighbors allows a progress toward the destination, a route is searched in AODV fashion (Perkins and Royer, 1999).

We first describe the notations and the model used and then the algorithm details.

4.1 Notations and Model

The k parameter is the maximum depth for searching a route and defines the variant of HGA : HGA-k. The set of nodes aware of their position is P, then if u knows its position $u \in P$. R defines the routing table of a node, then $R(u)$ is the routing table of node u. $RREQ\_TIMEOUT$ is the delay until a RREQ is ignored if no RREP has been received. The Euclidean distance between nodes u and v is noted $|uv|$. $N(u)$ is the set of neighbors in communication range of u. RREQs in the "waiting" state belong to the set W.

4.2 Algorithm Description

The header of a data packet contains the destination position and the last known position where the packet passed by.

To route a data packet, a node u starts by verifying if one of its neighbors allows a progress toward the destination. If u knows its own position, the progress depends of the distance between u and the destination. If u is not aware of its position, the progress depends on the last known position contained in the data packet if available, otherwise, any node aware of its position will be considered as making a progress. If no neighbor of u allows a progress, $R(u)$ is sought to find a route allowing a progress in an AODV fashion. If there is no such a route, a RREQ is sent $N(u)$ and a new research in routing table is made after the delay $RREQ\_TIMEOUT$. Algorithm 1 sums up route search on a node u.

When u receives a RREQ, it can either transmit it, answer with a RREP or discard it. Node u starts by analyzing $R(u)$ to verify, by comparing the sequence number and the source node of RREQ to those in the routing table entries, whether u has already received this RREQ. If so, the RREQ is ignored and nothing happens, otherwise u search in $N(u)$ if a node is closer to the destination than the source of the RREQ. If a route is found among neighbors of u, a RREP is sent to the nodes that transmitted the RREQ to u. The RREP contains the source id and sequence number from the RREQ, it also contains position of the closest neighbor of u to destination. If no route is found, the RREQ is forwarded if the maximum depth k for RREQ (MAX_HOP_RREQ) has not been reached. When a RREQ is transmitted, a new entry in the routing table is created with a 'waiting" flag allowing to know that a RREQ has been sent but no route is known yet. Algorithm 2 describes operations made at RREQ reception.

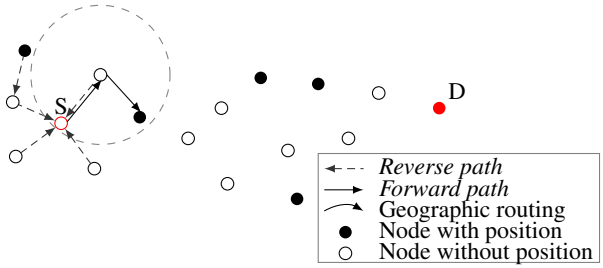
When a RREQ is received by a node u, the corresponding entry with "waiting" flag in the routing table is updated. The flag is removed and the neighbor allowing the destination to be reached is written is the table (this information is known thanks to the RREQ). Likewise, position of the destination is updated, indeed, the RREP may point to a position allowing to approach

Algorithm 1: Data packet reception at node u towards D with l being last node with known position.

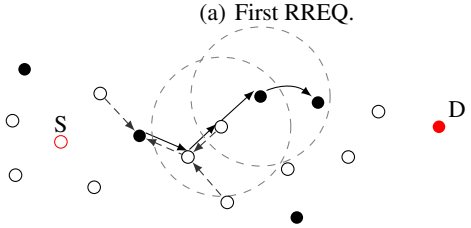
```
1  if u = D then
2      exit /* success */ ;
3  next ← −1 ;
4  if u ∈ P then
5      dist ← |uD| ;
6      l ← u ;
7  else if l ≠ −1 then
8      dist ← |lD| ;
9  else
10     dist ← +∞ ;
11 for v ∈ N(u) do
12     if |vD| < dist then
13         next ← v ;
14         dist ← |vD| ;
15 if next < 0 then
16     /* no route found in direct neighborhood */
17     broadcast(RREQ) ;
18     R ← R ∪ RREQ ;
19     W ← W ∪ RREQ ;
20     wait(RREQ_TIMEOUT) ;
21     if ¬(∃r, r ∈ R ∧ |rD| < dist) ∧ r ∉ W then
22         R ← R \ RREQ ;
23         /* no route found */
24         drop(data) ;
25         exit /* fail */ ;
26     else
27         for r ∈ R do
28             /* For each known route */
29             if |rD| < dist then
30                 dist ← |rD| ;
31                 next ← r ;
32 send(data, next) /* data transmitted */ ;
```

Algorithm 2: Reception of a RREQ initiated by S with sequence number *seqnum* on node u for destination D.

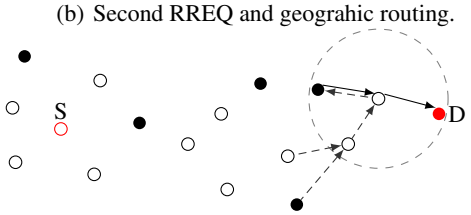
```
1  if (S, seqnum) ∈ R then
2      exit /* RREQ already sent */;
3  next ← ∅ ;
4  dist ← |SD| ;
5  for v ∈ N(u) do
6      if |vD| < dist then
7          next ← v ;
8          dist ← |vD| ;
9  if next = ∅ then
10     hop ← hop − 1 ;
11     R ← R ∪ RREQ ;
12     if hop > 0 then
13         broadcast(RREQ) ;
14 else
15     send(RREP, u, next) ;
```



(a) First RREQ.



(b) Second RREQ and geograhic routing.



(c) Third RREQ.

Figure 3: Routing example in HGA.

the destination but not necessarily the destination itself. By saving position of the node contained in the RREQ, routing table informations are more accurate.

HGA does not guarantee delivery, indeed if no node in the k-neighborhood of a node is aware of its position, the node will not be able to route packets. However, the route search mechanism allows to circumvent dead ends by searching new routes in the k-hops limit. Figure 3 shows an example of routing from node S, which is not aware of its position, to D, aware of its position. Figure 3(a) highlights a *route request* with a maximum depth of 3. Nodes receiving the RREQ and knowing their position answer with a RREP and S chooses the one which allow the maximum progression to destination. Once the packet has been routed to the intermediate destination, a new RREQ is initiated, the packet continues its progression then it is routed using classical geographic routing technique as intermediate destination knows a neighbor allowing a progression through destination 3(b). Finally, on Figure 3(c) a last RREQ is performed to allow the destination to be reached.

5 Simulation and Results

We choose to compare HGA to VCap, a routing algorithm using a similar greedy routing technique but based on virtual coordinates. We also simulate a greedy routing algorithm, slightly adapted to operate with some nodes without position awareness. In this variant, nodes without position knowledge do not send HELLO messages, then they do not participate in routing but can emit data packets. When a node needs to send a data packet (because it is the source or it needs to transfer it), it chooses in its neighbors with position the closest to the destination allowing a progress. If the node sending the data packet is not aware of its position, the notion a progress does not exist, then the closest position aware neighbor to destination is chosen.

We simulated HGA with route search depth of 1, 2, 3 and 5 hops. VCap has been simulated with 3 and 5 anchors.

Simulations have been performed using WSNET (Fraboulet et al., 2007) simulator. Each simulation lasts for 10 minutes. We generated random connected topologies of 500, 750, 1000, 1500, 2000, 2500 and 3000 nodes in a field of 500 m $\times$ 500 m to achieve average density of 10, 15, 20, 30, 40, 50 et 60 nodes per communication range. Nodes have a communication range of 40 m. Each combination (algorithm, average density) is launch 50 times on different topologies. The same topology is used for a combination of average density and iteration number for each different algorithm. For instance the first simulation of HGA-1 uses the same topology as the simulation of VCap-3, Greedy, etc.

To simulate a data traffic, every 10 seconds, each node chooses randomly a destination node aware of its position. The source node sends the data packet in respect to the algorithm it executes. Nodes are desynchronized in order to distribute emissions in time.

We measure delivery ratio, control message number and memory used. The delivery ratio is the number of messages received divided by the number of messages sent. Control messages are HELLO, RREQ, RREP and messages to initiate the coordinate system. Finally, the memory used is the average number of neighbors saved by a node plus the number of known routes for each node. Simulation parameters are summed up in Table 1.

For the sake of clarity, in the following we refer to HGA with a depth search of 1 hop by HGA-1 ($k = 1$). Moreover, HGA-2 (resp. HGA-3 and HGA-5) refer to HGA algorithm with depth search of 2 (resp 3 and 5) hops. Finally VCap-3 and VCap-5 refer to the VCap algorithm with 3 and 5 anchors. VCap results curves have been duplicated on figures with 1%, 5% and 10% located nodes in order to ease readings.

Parameter	Value
Duration (m)	10
MAC layer	idealmac
Interferences	none
Data size (bytes)	10
Header size (bytes)	88
Field size	500 m $\times$ 500 m
Communication range	40 m
Number of nodes	500, 750, 1000, 1500, 2000, 2500, 3000
Density	10, 15, 20, 30, 40, 50, 60
Iterations	50

Table 1: Simulation parameters

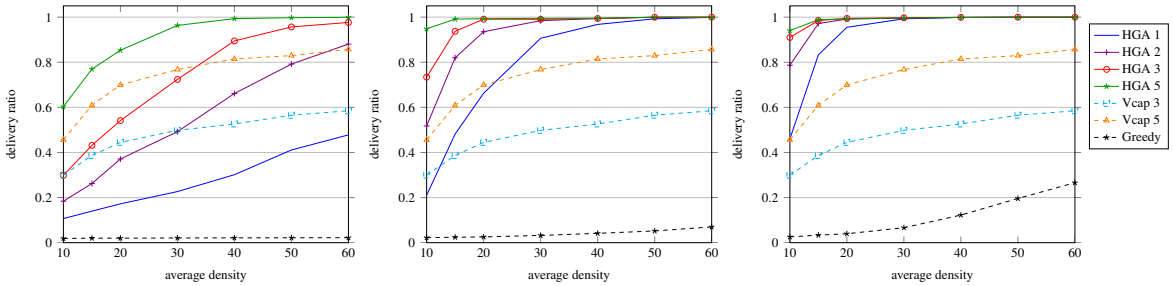
5.1 Delivery ratio

Figure 4 shows delivery ratio for the four variants of HGA, the two variants of VCap and the Greedy algorithm. We can see on Figure 4(a) that whatever the average density of the network, with 1% of nodes aware of their position, HGA-5 has a delivery ratio higher that the two variants of VCap and Greedy. We also observe than from an average density of 40, HGA-5 reaches almost 100% delivery ratio. As we expected, HGA-1 offers lower performances than HGA-2, which offers lower performances than HGA-3 while HGA-5 gets the best performances. As well, VCap is better with 5 anchors than with 3 as expected since it reduces the This hierarchy is also observed with 5% and 10% of nodes aware of their position as seen on Figures 4(b) and 4(c).

We can observe that HGA-1 performs much better than Greedy. HGA-1 has a delivery ratio 5 times higher for a density of 10 and almost 15 times higher for a density of 40 although only 1% of nodes are aware of their position. It shows that with only a search depth of 1 hop ($k = 1$), we can achieve really interesting performances. We will see how it impact the cost in messages in the next section.

One notices that HGA algorithms take a better advantage for higher densities of nodes. Indeed, for 1% of nodes aware of their position (Fig. 4(a)), the delivery ratio increases for densities from 10 to 20. It is similar for all variants of HGA and VCap. Besides, from a density of 20, the increase is stronger for HGA than for VCap. It allows HGA-2 to outperform VCap-3 from a density of 25 and HGA-3 is over VCap-5 from a density of 35.

With 5% of nodes aware of their position (Fig. 4(b)), performances of HGA are improved. This behavior is logical as a RREQ is more likely to find a node aware of its position and then a node in the direc-



(a) 1% position. (b) 5% position. (c) 10% position.

Figure 4: Delivery ratio

tion of the destination. It is interesting to highlight that the delivery ratio increase is higher that with 5% of nodes aware of their position. It allows all variants of HGA to outperform both VCap variant from a density of 25. We can observe that this behavior is even more true with a position knowledge of 10% (Fig. 4(c)). These results show that HGA benefits more of a higher density of nodes than VCap. Indeed with VCap, the number of nodes with the same coordinates do not necessary decrease with the increase of node density while RREQs of HGA have a higher probability to reach a node in the direction of the destination as explained.

5.2 Control Messages Cost

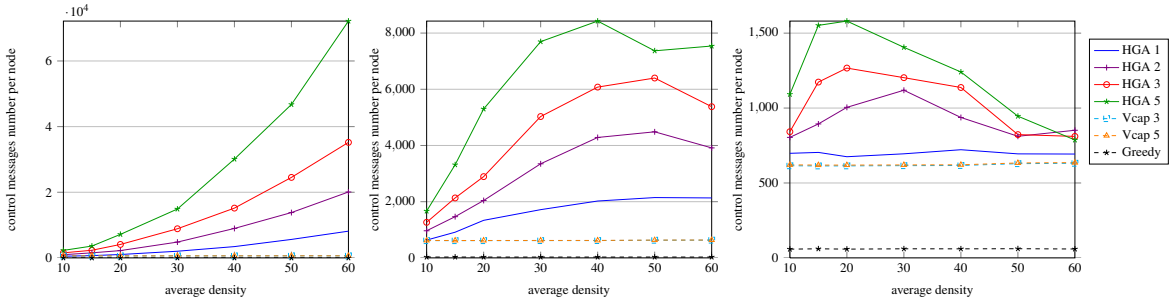
Figure 5 shows the average number of control messages sent by nodes. For 1% of nodes aware of their position (Fig. 5(a)), we observe for HGA that, the deeper is the search, the more there is control messages. Likewise, the higher is density, the more messages are sent. It can be explained by the fact that more nodes receive the RREQs when the density is higher, each node receiving a RREQ potentially re-emits it, explaining the exponential increase of the number of messages emitted with the average density. It also explains why the increase is stronger when the route search if deeper. Figures show that the number of control messages sent by Greedy is very low because only nodes aware of their position emit control messages. For Greedy and VCap, the number of control messages sent is constant as it does not depend on density. For a low density, the control messages number is close for all algorithms. The gap increases with the density. For low densities, it is interesting to prefer HGA-5 which offers high delivery ratio, compared to other solutions, while adding a reasonable amount of control messages. When the average density increases, VCap is more suitable as it offers a delivery ratio close to, if not better than some variants of HGA while generating less control messages.

When the number of nodes aware of their position increases to 5%, the number of control messages strongly decreases for HGA, especially with deep search. This behavior is explained because the probability of finding a route during a RREQ increases when the number of nodes aware of their position increases. This probability also increases when the density of nodes increases (there is more nodes in the neighborhood to satisfy the request). This explains the inversion of curves for HGA for 5% and 10% of nodes aware of their position (Fig. 5(b) and 5(c)). However, there is no need to use deep variants for 10% of nodes aware of their position as the delivery ratio rapidly reaches 100% in that case.

5.3 Memory Cost

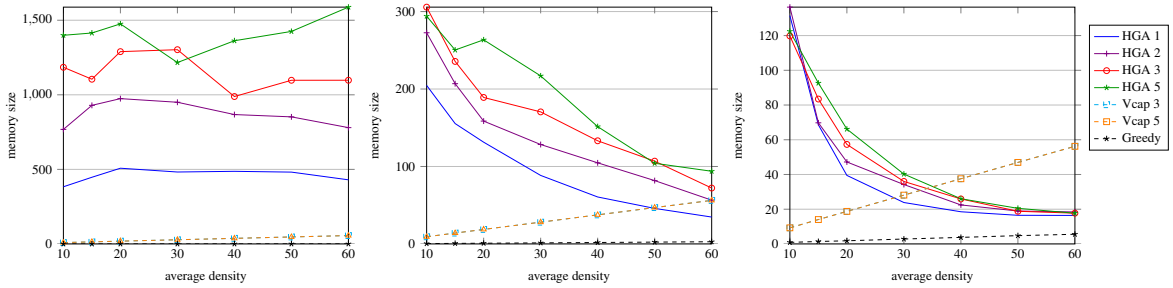
Figure 6 shows memory usage for the different algorithms. We can see that memory needs are much higher for HGA than VCap when 1% of nodes are aware of their position (Fig. 6(a)). With 5% (Fig 6(b)), the gaps for memory usage are lower and even really close to VCap when average density increase. On Figure 6(c), with 10% of nodes with position knowledge, memory needs are much lower for HGA and even lower than VCap when density is over 40.

The memory size for Greedy is only the number of neighbors aware of their position for each node. The high amount of memory needed by HGA is explained because in simulations, every route is kept until the end of the simulation. It would be possible to lower the memory needs by “forgetting” old routes with the counterpart of having to send more control messages. We can highlight that experimental conditions are demanding concerning memory needs. Indeed, sources and destinations of data packet are chosen randomly at a high frequency. In real applications, the number of destinations of data packets are often limited if not unique in case of data collection. In case of actuators, one base station sends messages to different sources. In that case the source is unique, thus the number of routes to memorize for each node is reduced.



(a) 1% position aware nodes. (b) 5% position aware nodes. (c) 10% position aware nodes.

Figure 5: Message control number



(a) 1% position. (b) 5% position. (c) 10% position.

Figure 6: Average size in memory.

6 Conclusion

In this article we propose a novel solution for geographic routing, that allow to use nodes position when available but also able to deal with unavailable positions. To the best of our knowledge, there is no solution in the literature which relies on the same assumptions. Indeed, we can find geographic routing solutions for networks with all nodes aware of their positions as well as solutions using virtual coordinates where no node is aware of its position. We propose an hybrid solution, taking advantages of real node position when available but able to operate if not always available. The geographic and the reactive methods illustrated in this paper could be replaced by some other methods from literature or new ones. Both part should be chosen regarding application needs. The underlying topology of the application will favor some geographic routing techniques while the traffic pattern will favor the classical routing part. For low data rates, the classical routing part could be proactive or even hybrid as proposed in ZRP (Haas et al., 2002).

Our solution, **HGA** offers much better performances than Greedy geographic routing if part of the nodes are not aware of their position. We also compare **HGA** with a routing solution based on virtual coordinates, VCap, and show that for different scenarios, our solution offer better performances with a limited or no overhead.

For future works, we consider to validate our works with realistic physical and MAC layers. The use of real platform such as FIT or SmartSantander (Sanchez et al., 2011) would allow us to validate these aspects while testing realistic environments.

Studying node mobility is another interesting aspect, we could characterize the needed timeout before routes expires with regards to nodes speed. Comparison with routing algorithm using virtual coordinates would also bring some arguments for **HGA** since these algorithms need to frequently rebuild coordinates system, and then, increase number of control messages exchanged in the network.

Finally it could be interesting to compare our solution with a variant in which each node keeps a k-hop neighborhood table instead of using k-hop RREQ. The study of control messages number would allow us to find threshold to know which variant is better in some cases.

REFERENCES

Al-Hader, M., Rodzi, A., Sharif, A., and Ahmad, N. (2009). Smart City Components Architecture. In *Proceedings of the International Conference*

on Computational Intelligence, Modelling and Simulation, CSSim, pages 93–97, Brno, Czech Republic. IEEE.

Benbadis, F., Puig, J.-J., de Amorim, M. D., Chaudet, C., Friedman, T., and Simplot-Ryl, D. (2006). Jumps: Enhancing hop-count positioning in sensor networks using multiple coordinates. *International Journal on Ad Hoc and Sensor Wireless Networks*, abs/cs/0604105.

Benkic, K., Malajner, M., Planinsic, P., and Cucej, Z. (2008). Using RSSI value for distance estimation in wireless sensor networks based on ZigBee. In *Proceedings of the 15th International Conference on Systems, Signals and Image Processing*, IWSSIP, pages 303–306, Bratislava, Slovakia.

Caruso, A., Chessa, S., De, S., and Urpi, A. (2005). GPS free coordinate assignment and routing in wireless sensor networks. In *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, INFOCOM, pages 150–160, Miami, FL, USA. IEEE.

Chávez, E., Mitton, N., and Tejeda, H. (2007). Routing in Wireless Networks with Position Trees. In Kranakis, E. and Opatrny, J., editors, *Ad-Hoc, Mobile, and Wireless Networks*, ADHOC-NOW, pages 32–45, Cancun, Mexico. Springer Berlin Heidelberg.

Elhafsi, E. H., Mitton, N., and Simplot-Ryl, D. (2007). Cost over Progress Based Energy Efficient Routing over Virtual Coordinates in Wireless Sensor Networks. In *Proceedings of International Symposium on a World of Wireless, Mobile and Multimedia Networks*, WoWMoM, pages 1–6, Helsinki, Finland. IEEE.

Ermel, E., Fladenmuller, A., Pujolle, G., and Cotton, A. (2005). On Selecting Nodes to Improve Estimated Positions. In *Proceedings of the 7th international Mobile and Wireless Communication Networks*, MWCN, pages 449–460, Marrakech, Morocco. Springer US.

Fang, Q., Gao, J., Guibas, L., de Silva, V., and Zhang, L. (2005). GLIDER: gradient landmark-based distributed routing for sensor networks. In *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, INFOCOM, pages 339–350, Miami, FL, USA. IEEE.

Fraboulet, A., Chelius, G., and Fleury, E. (2007). Worldsens: Development and Prototyping Tools for Application Specific Wireless Sensors Networks. In *Proceedings of the 6th International Symposium on Information Processing in Sensor Networks*, IPSN, pages 176–185, Cambridge, MA, USA. ACM.

Haas, Z. J., Pearlman, M. R., and Samar, P. (2002). The Zone Routing Protocol (ZRP) for Ad Hoc Networks. IETF Internet Draft.

Liu, K. and Abu-Ghazaleh, N. (2008). Stateless and guaranteed geometric routing on virtual coordinate systems. In *Proceedings of the 5th International Conference on Mobile Ad Hoc and Sensor Systems*, MASS, pages 340–346, Atlanta, GA, USA. IEEE.

Mitton, N., Razafindralambo, T., Simplot-Ryl, D., and Stojmenovic, I. (2012). Towards a hybrid energy efficient multi-tree-based optimized routing protocol for wireless networks. *Sensors*, 12(12):17295–17319.

Niculescu, D. and Nath, B. (2001). Ad hoc positioning system (APS). In *Proceedings of the Global Telecommunications Conference*, GLOBECOM, pages 2926–2931, San Antonio, TX, USA. IEEE.

Perkins, C. and Royer, E. (1999). Ad-hoc on-demand distance vector routing. In *Proceedings of the 2nd Workshop on Mobile Computing Systems and Applications*, WMCSA, pages 90–100, New Orleans, LA, USA. IEEE.

Sanchez, L., Galache, J., Gutierrez, V., Hernandez, J., Bernat, J., Gluhak, A., and Garcia, T. (2011). Smartsantander: The meeting point between future internet research and experimentation and the smart cities. In *Future Network Mobile Summit*, FutureNetw, pages 1–8, Warsaw, Poland.

Seada, K., Helmy, A., and Govindan, R. (2004). On the effect of localization errors on geographic face routing in sensor networks. In *Proceedings of the 3rd international symposium on Information processing in sensor networks*, IPSN, pages 71–80, New York, NY, USA. ACM.

Stojmenovic, I. (2006). Localized network layer protocols in wireless sensor networks based on optimizing cost over progress ratio. *IEEE Network*, 20(1):21–27.

Čapkun, S., Hamdi, M., and Hubaux, J.-P. (2002). GPS-free Positioning in Mobile Ad Hoc Networks. *Cluster Computing*, 5(2):157–167.