

Geographic Routing with Logical Levels forwarding for Wireless Sensor Network

Yassine SABRI

STIC Laboratory
Chouaib Doukkali University, B.P: 20
El Jadida MOROCCO

Najib EL KAMOUN

STIC Laboratory
Chouaib Doukkali University, B.P: 20
El Jadida MOROCCO

ABSTRACT

Location information is essential in many applications of WSNs, it is natural to use this information for routing as well. Location-based protocols or geographical routing protocols to exploit the location information of each node to provide efficient and scalable routing. Various routing algorithms that feat geographic information (e.g., GPSR) have been aimed to attain this goal. These algorithms refer to all nodes by their location, not address, and use those coordinates to route greedily, when possible, towards the destination. However, there are dozen's situations where location information is not available at the node. This paper presents a new geographical routing protocol for Wireless Sensor Networks (WSN) energy-efficient data forwarding, called GRPW (geographic routing protocol washbasin). Protocol GRPW ensures a load balancing, minimizing energy consumption and the rate of message delivery using a routing policy with logical levels, inspired from the water flow in a washbasin, without making the assumption that all sensors are localized. GRPW protocol performance compared to the protocol GPSR show that maximizes the lifetime of the network, provides quality service parameterizable, and is appropriate for dense sensor networks confronting our method to an optimal algorithm.

General Terms:

Routing, wireless sensor network (WSN)

Keywords:

Wireless Sensor Network (WSN), Geographical Routing, Localization.

1. INTRODUCTION

Wireless Sensor Networks (WSN) are constituted of a large number of tiny sensor nodes randomly distributed over a geographical region whose power consumption is low. However, as shown in current research [11], the classical routing protocols are not applicable to sensor networks in a real environment, mainly because of specific radio conditions. Noise, interference, collisions and the volatility of the node neighborhood leading to a significant drop in performance. Many applications for sensor networks such as monitoring of forest fires, the remote meter reading, ... For these cases, The Geographic routing of data in this type of network is an important challenge. Geographic routing uses nodes' locations as their addresses, and forwards packets (when possible) in a greedy manner towards the destination. The most widely known proposal is [5][8], but several other geographic routing schemes have been proposed [14]. One of the key challenges in geographic routing is how to deal with dead-ends, where greedy

routing fails because a node has no neighbor closer to the destination; a variety of methods (such as perimeter routing in GPSR/GFG) have been proposed for this. More recently, GOAFR [10] proposes a method for routing approximately the voids that is some asymptotically worst case optimal as well as average case efficient. Geographic routing is scalable, as nodes exclusively maintain state for their neighbors, and supports a full general any-to-any communication pattern without explicit route establishment. However, geographic routing requires that nodes know their location. While this is a natural assumption in some settings (e.g., sensornet nodes with GPS devices), there are many circumstances where such position information isn't available. are most often require information about the position of their voisins to function effectively. Or, this assumption is far from the reality. The other, the localization of protocols, used as a preliminary step by geographical routing protocol are not necessarily precise. For example, in [15], the authors proposed localization methods with which sensors determine their positions with a rate of less than about 90% positioning in large scale. or, if a node that does not know its location, the node risk of never communicate with other node of networks, and no information will be transmitted to the user and the base station never knows that node.

In this paper, we propose a geographic routing protocol which is based on the construction of a logical topology levels. This strategy allows the construction the incline level directed to the sink node. To make the sensor network similar to a washbasin and routing the packet to the flow of water in this washbasin. We assume that among the node deployed that there are well-positioned nodes and other non-positioned. Our work is decomposed into two phases: Phase 1: one starts by constructing a logical topology by forming the logic levels, each level is characterized by a width that depends on the network density, Then for each node will be well localized we assigned to it a level depending on the distance which separates him to the node sinks. subsequently the non-localized nodes will join the level closest according to neighborhood nodes by sending specific messages, at the end of this phase each node must have a level of belonging.

Phase 2: This phase consists of a transmit data to sink node, data transmission is performed in three modes of progression: the first mode to route the packet to a neighbor who belongs to a higher level, however, if they find no neighbor in the level above it switches to Second mode to transmit the packet to a neighbor who belongs to the same level, otherwise the packet will delivering to third mode in case there are some holes must reroute the packet to the lower level.

The rest of the paper is organized as follows: In Section 2, we summarize related work on localization algorithms. In Section 3, introduces basic notions for this problem. In Section 4, we

present our new localization algorithm GRPW. In Section 5, we evaluate the proposed scheme through comprehensive simulation studies and compare it with GPSR technique. We conclude the paper in Section 6.

2. RELATED WORK AND BACKGROUND

With the development of wireless ad hoc networks like roofnets [17][18] and sensornets [7][12], there has been a proliferation of geographic routing algorithms in recent years.

The geographical routing emphasizes the fact that each node knows its coordinates geographically and uses those to the final destination for routing decisions. Early work in geographic routing considered only greedy forwarding [16] by using the locations of nodes to move the packet closer to the destination at each hop. Greedy forwarding fails when reaching a local maximum, a node that has no neighbor closer to the destination. CompassII [9] presents a face routing algorithm that guarantees message delivery on a geometric graph by traversing the edges of planar faces intersecting the line between the source and the destination. [5] discuss algorithms for extracting planar graphs from unit graphs and for face routing in the planar graphs to guarantee delivery. Due to the inefficient paths resulting from face routing, they propose combining face routing with greedy forwarding to improve the path length.

GPSR protocol [8] is the earliest geographical routing protocols for ad hoc networks which can as well be applied for WSN surroundings. The GPSR adapts a greedy forwarding strategy and perimeter forwarding strategy to route messages. It makes use of a neighborhood beacon that sends a node's identity and its position. Each packet contains the position of the destination and nodes necessitate only local selective information concerning their emplacement and their immediate neighbors' positions to forward the packet. Each node forwards the packet to the neighboring nodes nearest to the destination using greedy forwarding. When greedy forwarding fails, face routing is employed to route roughly dead-ends until closer nodes to the destination are found.

Recently, there have been proposals [13][14] for geographic routing without location information. The approach is to assign logical coordinates to each node and then apply greedy forwarding over the logical coordinates. To route from u to v with GEM primitive routing algorithm, a message is delivered following a node's parents until a parent's virtual angle contains v 's virtual angle. Then the message is delivered through every child that comprises v 's virtual angle until v . In order to build the logical space, a relatively large establish overhead is necessitated in the form of various global floods or iterations of coordinates' relaxation. Extra overhead is incurred to detect and propagate changes in the logical coordinates due to topology changes. Although it has some benefits, GEM is not as efficient and scalable as routing with location information. Perimeter routing as well cannot be used in these strategies, and in [14] there is no solution provided for dead-ends that guarantees delivery. In addition, for data-centric memory, these schemes do not allow for robust reproduction and consistency.

Another problem of geographic routing is that packets may be routed to a dead end. The distance from a sensor to the base station is marked alongside the sensor. Suppose there is no sensor to the right of the dashed line, which outlines an open void. The void may be a disaster area where all sensors are destroyed, or it may be a bay where the sensors cannot survive. There is a single dead end, which does not have any neighbor that is closer to the base station. Once a packet is routed to the dead end, it cannot proceed any further. The packets from the gray nodes will be forwarded to the dead end based on geographic routing. To solve the above problem, GPSR use the "right-hand rule" to route packets along the boundary of the void

until they reach the other side of the void. This approach finds a specific detour path out of the dead end.

In this paper we address how to keep back the gains of geographic routing in the absence of localization information. This problem has been partially addressed in [6], but there they consider the case where some nodes don't have geographical data; here we focus mainly on the case where no (or only the perimeter) nodes have location information. Our approach involves assigning virtual logical levels to each node and then applying standard geographic routing over these levels of belonging. These levels virtual need not be precise representations of the inherent geography but, in order to serve as a basis of routing, they must reflect the underlying connectivity. Thus, we construct these virtual coordinates using mainly local connectivity information and the destination position sink. Since local connectivity information is ever available (nodes perpetually know their neighbors), this technique can be applied in most settings. Moreover, as our simulation consequences demonstrate, there are scenarios, such as in the presence of obstacles, where greedy routing performs better using virtual logical levels than using true geographic coordinates.

3. MODEL

First we describe the system model on which our protocol design is based. In this paper, we focus on static networks. Moreover, it assumes that all sensors have identical reachability radius r . However, it is easy to adapt our method to sensors having different reachability radius. A wireless sensor networks is represented as a bidirectional graph $G(V, E)$ where V is the set of n nodes representing sensors and E is the set of m edges representing communication links. If two nodes $u, v \in V$ are neighbors, then they are linked that means distance between u and v is smaller than r . The set of neighbors for a node $u \in V$ is noted $N(u)$. The nodes have knowledge of their location through some other means, such as GPS or simply explicit programming. The set of nodes is noted Λ . The set of neighbor anchors for a node u is noted $N_{\Lambda}(u)$ ($N_{\Lambda}(u) = N(u) \cap \Lambda$) and the set of non-neighbor nodes that know their position is noted $\bar{N}_{\Lambda}(u)$ ($\bar{N}_{\Lambda}(u) = \Lambda / N_{\Lambda}(u)$). Note that all identical nodes (anchors or others nodes) have the same capabilities (energy, processing, communication, ...). The coordinate of a position of node u is noted (x_u, y_u) . \mathcal{P} is the set of all possible positions in a network.

4. GEOGRAPHIC ROUTING PROTOCOL WASHBASIN

We assume in this work that each node can get its own location information either by GPS or other location services [19][15]. Each node can get its one-hop neighbor list and their locations by beacon messages. We consider the topologies where the wireless sensor nodes are roughly in a plane. Our approach involves three steps:

- (1) **The distribution of the sink address to the set of sensors networks:** In the first step, The communications in this step are made in three steps:
 - When a node wants to transmit the sink position to its neighbors, it first emits ADV message containing the location of sink.
 - A node receiving a message ADV. If interested in this information, it sends a message REQ to its neighbor.
 - In Receiving a message REQ, the transmitter transmitted to the node concerned the sink position as a DATA message.

- (2) **Construction of logical levels:** In this step the node networks determine its level of belonging through the sink node position, each node u well localized, calculate its level based on the received position of sink in the *Phase 1*, with which u calculates the distance d_{uSink} which separates him with the sink node. the levels is calculated so that the width level η be constant is less than and inversely proportional to the density of networks δ .

The level l of the node u defined by:

$$Level_u = \{l \in \mathbb{N} / \frac{d_{uSink}}{\eta} \leq l \leq \frac{d_{uSink}}{\eta} + 1\}$$

Set of the neighbor nodes that are well localized and which belongs to the same level as u :

$$L_{N_{\Lambda}(u)} = \{v \in N_{\Lambda}(u) / Level_u = Level_v\}$$

Set of the neighbor nodes that are well localized and which belongs to the higher level than u :

$$L_{N_{\Lambda}(u)}^+ = \{v \in N_{\Lambda}(u) / Level_u = Level_v - 1\}$$

Set of the neighbor nodes that are well localized and which belongs to the lower level than u :

$$L_{N_{\Lambda}(u)}^- = \{v \in N_{\Lambda}(u) / Level_u - 1 = Level_v\}$$

For non-localized node, they determined their level of belonging by sending a message belonging to neighboring nodes, the neighboring node that received the message, reply with a message containing its level of belonging, the level of u is equal to the level of maximum neighbors that they like the nivau.

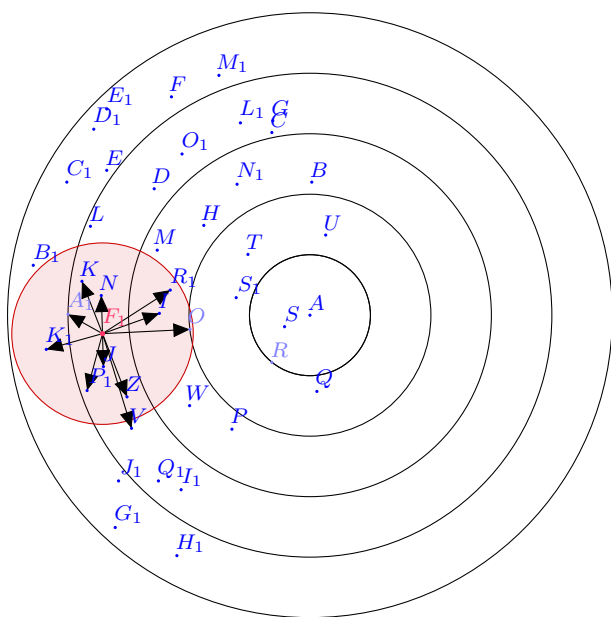


Fig. 1. The calculated level for a not localized node

Figure 1 shows an example of a sensor F_1 not localized who is seeking to determine its level of belonging, the node begin

scanning its neighbor by sending a *HELLO* message, to calculate the number of node of each level $\{2, 3, 4\}$, F_1 ranking neighbor as following $L_{N_{\Lambda}(F_1)}^- = \{A_1, K_1\}$, $L_{N_{\Lambda}(F_1)}^+ = \{R_1, O, I\}$ and $L_{N_{\Lambda}(F_1)} = \{P_1, K, N, Z, V\}$. like that he decided to join the level 3 of $L_{N_{\Lambda}(F_1)}$ that has the maximum number of nodes.

- (3) **Data forwarding :** the routing decision is done in our approach in three modes (Figure 5 shows the pseudo code of GRPW), depending on dispoibilites neighboring nodes and of their level of belonging: the *Even Forwarding*, *Anterior Forwarding* and the *Rear Forwarding* (respectively called EF, AF and RF).

—The AF constructs a road traversing the nodes of the source to the destination which each node receiving a packet *DataPacket* with the mode of transport *ANTERIOR_FORWORD*, will move toward the intermediate node in its coverage area and belongs to the level 1, the intermediate node select among the neighboring node which belongs to the levels using a lookup function.

Lookup function is used by a node in order that he can determine the next hop to reach the next level, to determine the next hop function, lookup based on the principle of Round Robin (RR), Round-robin (RR) is one of the simplest scheduling algorithms for processes in an operating system. As the term is generally used, time slices are assigned to each process in equal portions and in circular order, handling all processes without priority (also known as cyclic executive). Using RR will enable us to balance the traffic on all node resaux and exploit several paths for routing the packet. The routing metric which we define is composed of the residual energy and node density.

The figure 2 is an example of the *Anterior Forward* routing technique represent by the segment in green color, the sensor Z belongs to level $Level_Z = 3$ seeks the next hop neighbor among the set $L_{N_{\Lambda}(Z)}^+ = \{I, W\}$ of neighbors belonging to level 4, Z selects I as next hop based on the lookup function.

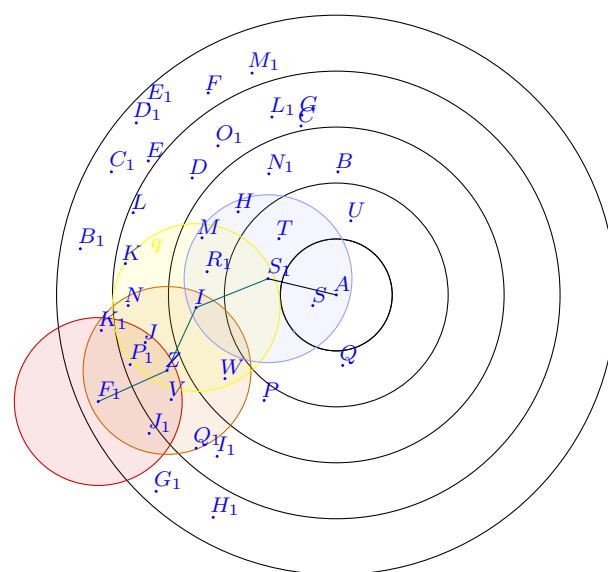


Fig. 2. Anterior Forward for the node Z

Because of the frequent failures of nodes, the mobility of nodes or policy scheduling of activities used, disconnections can occur in the network generates, so, what are called holes In this situation, all the nodes one is empty, its neighbors that they can transmit packets to a nivau L the node u will

change the routing mode to *EVEN_FORWARD* mode to reroute the packet in EF Mode. u will also delegate the responsibility to transmit the packet he received to the nearest neighbor node sink that he has the same level as u . This process is repeated recursively until reaching a node that can move the packet to the nodes of level L_{NA}^+ .

—Because of the frequent failures of nodes, node mobility or policy scheduling of activities used, disconnections can occur in the network generates, well, what are called "Voids". In this situation, we switching to the EF mode. The EF consists of finding a path for a packet that did not find a next hop neighbor in level L_{NA}^+ using the AF mode, EF is defined as follows: When a packet *DataPacket*, arrives at a node x from node y , the way forward is the next one is at the same level of belonging $L_{NA}()$ while avoiding the "crossing links" (route already traveled), this is why each packet keeps his footsteps (noded list, crossing EF in Mode or RF) in the vector *DataPacket*: *IntermediateNodes* the next hop is selected by the lookup function that returns a noded neighbor who belong to the same level as x and that do not belongs to all the nodes already traversed *DataPacket*: *IntermediateNodes* to of avoids the routing loop

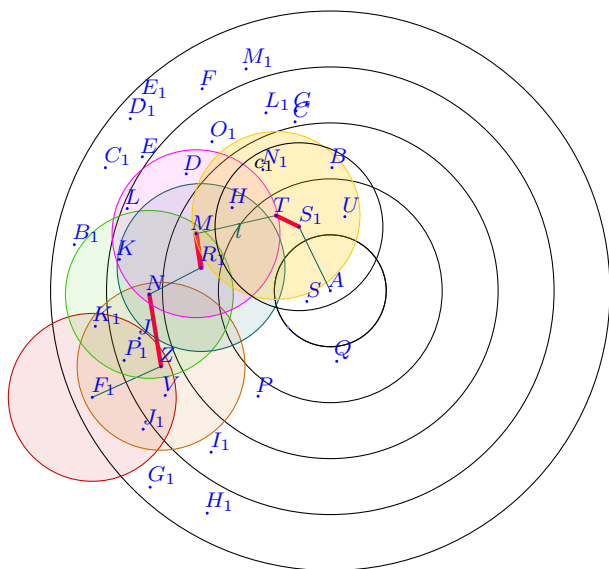


Fig. 3. Even Forwarding for the node Z

The figure 3 is an example of the *Even Forwarding* routing technique represented by the red segment, the sensor Z belongs to the level $Level_Z = 3$, Z seeks the next hop neighbor among the set of neighbors belonging to the level 4 but when there is no node that belongs to level $L_{NA}^+(Z)$, Z will route the packet to the neighbor nodes of the same level as Z by selecting from the set $L_{NA}(Z) = \{V, N, P\}$ based on the lookup function, Z will rerouting the packet to the node N.

—RF consists of reroute a packet *DataPacket*, who was failed in AF and EF, RF fact sends a packet to the low level $L_{NA}^-()$ by seeking the next hop among neighboring based on the lookup function. RF is leaning on same technique used in Ef, for avoids the routing loop we safeguard the sets of node traversed by the packet *DataPacket* in a vector-type structure *DataPacket*: *IntermediateNodes*. The figure 4 is an example of the *Rear Forwarding* routing technique represented by the green segment, the sensor C belongs to the

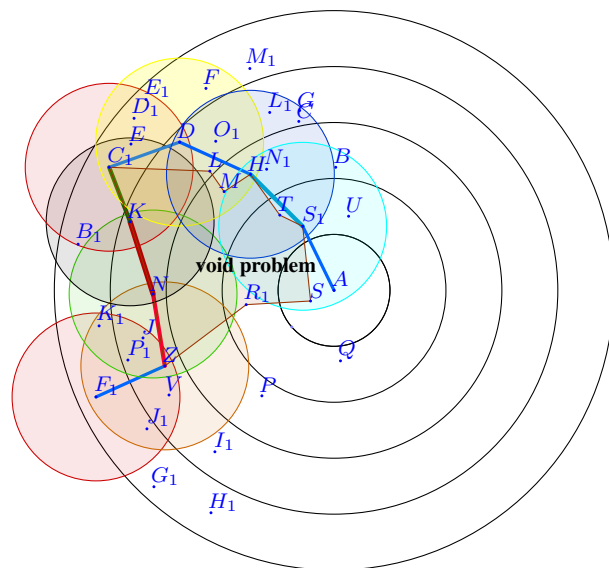


Fig. 4. Rear Forwarding Forwarding for the node K

Algorithm 4.1: When a node n receives a message data (*DataPacket*)

```

if (DataPacket.ModeForward=ANTERIOR_FORWARD)
  {Neighbor = Lookup(DataPacket, TableLevel + 1)
  if Neighbor ≠ NULL
  then {then {DataPacket.IntermediateNodes.Clear()
             Forward(DataPacket, Neighbor)
             }
        else {DataPacket.ModeForward ← EVEN_FORWARD
             }
  }
else if (DataPacket.ModeForward=EVEN_FORWARD)
  {Neighbor = Lookup(DataPacket, TableLevel + 0)
  if Neighbor ≠ NULL and Neighbor ∉ DataPacket.IntermediateNodes
  then {then {DataPacket.IntermediateNodes.add(CurrentNode)
             DataPacket.ModeForward ← ANTERIOR_FORWARD
             Forward(DataPacket, Neighbor)
             }
        else {DataPacket.ModeForward ← REAR_FORWARD
             }
  }
else if (DataPacket.ModeForward=REAR_FORWARD)
  {Neighbor = Lookup(DataPacket, TableLevel - 1)
  if Neighbor ≠ NULL and Neighbor ∉ DataPacket.IntermediateNodes
  then {then {DataPacket.IntermediateNodes.add(CurrentNode)
             DataPacket.ModeForward ← ANTERIOR_FORWARD
             Forward(DataPacket, Neighbor)
             }
        else {DELETE(DataPacket)
             }
  }
  
```

Fig. 5. Description of GRPW algorithm

level $Level_K = 3$, k seeks the next hop neighbor among the set of neighbors belonging to the level 4 but when there is no node that belongs to $L_{NA}^+(K)$ or same level as C which belongs to $L_{NA}(K)$ (the sensor C will be excluded because it belongs to the nodes already traversed defined by *DataPacket*.*IntermediateNodes*), K will route the packet to the neighbor nodes of the low level as K by selecting from the set $L_{NA}^-(K) = \{C, B, E\}$ based on the lookup function, Z will rerouting the packet to the node B.

The finite state machine shown in Figure 6 depicts various states which summarizes the three routing mode in the protocol GRPW, depending on existence of neighbors node of each logical levels.

5. SIMULATIONS AND RESULTS

Experiments were built upon the J-Sim simulator [9] dedicated to WSN simulations. It is a compositional, component-based simulation environment. It is built upon the concept of autonomous component programming model. J-Sim is developed entirely in Java. The signal attenuation due to obstacles or other factors (e.g.

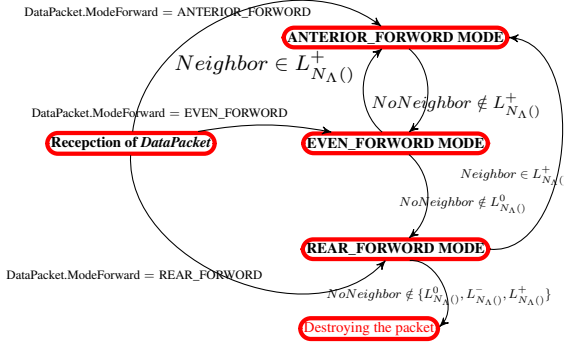


Fig. 6. Finite state machine of GRPW protocol

use of unidirectional antennas) is simulated in J-Sim. Therefore, the vicinity of a node in terms of transmission range is not necessarily spherical. Note that there several simulators in the literature such as GlomoSim[3], OMNET++[4], OPNET[2], NS-2[1]. The MAC layer is considered perfect and the transmission of messages are without loss in our simulation.

To study and analyze the performance of the protocol, three metrics are chosen: (i) average dissipated energy in sensor nodes, (ii) delivery rate, (iii) energy consumption. Average dissipated energy is the ratio of total energy dissipated in the network to the number of packets received by the sink over a given period of time. The performance of the proposed protocol is studied with respect to the above three metrics with varying network sizes. Each simulation is carried out with 10 different topologies and the final results are presented based on the average of the result sets.

The node density in the network is increased from 5 to 14 with 3 being added in every step of increment. Initially, the sensors nodes are randomly distributed on a $200m \times 200m$ area ($A = 200 \times 200$), the radio range for each sensor is kept at 15m. Firstly, to learn how to we must select the width of logic levels η , we begin by analyzing the Impact of network density δ and width on the rate of packet delivery. as cited in, for routing and forwarding the data in Gwap operates through logical levels requires that the width for each level be inversely proportional to the density of networks, which it interprets as

$$\eta = \frac{k}{\delta} \quad (1)$$

through other, for a node can communicate with the level nodes must its transmission range r is less than the width η :

$$\eta \leq r \leq 2\eta \Rightarrow 0.5 \times r \leq \eta \leq r \quad (2)$$

The density of a sensor network δ is the average number from neighbors for a node. It is obtained by the following equation:

$$\delta = \frac{n \times \pi \times r^2}{A} \quad (3)$$

which implies that:

$$r = \sqrt{\frac{A \times \delta}{n \times \pi}} \quad (4)$$

According to (1) and (2):

$$0.5 \times r \leq \frac{k}{\delta} \leq r \quad (5)$$

we replacing r by its value:

$$0.5 \times \sqrt{\frac{A \times \delta}{n \times \pi}} \times \delta \leq k \leq \sqrt{\frac{A \times \delta}{n \times \pi}} \times \delta \quad (6)$$

placing:

$$\ell = \sqrt{\frac{A \times \delta}{n \times \pi}} \times \delta \quad (7)$$

which implies that:

$$0.5 \times \ell \leq k \leq \ell \quad (8)$$

The graph show the impact of k on the performance of gwt in terms of delivery rate, it is clear that the best value is $k = 0.7 \times \ell$, it is the value that will be used later.

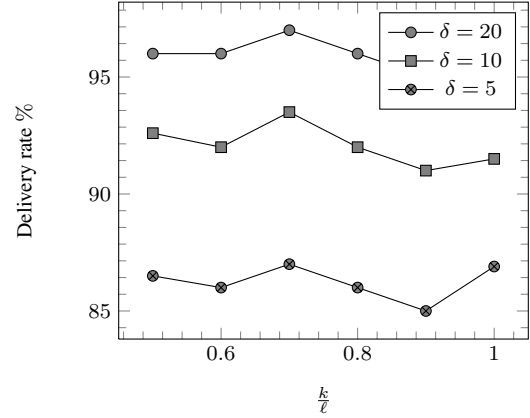


Fig. 7. density of sensor localized with Network density $\delta = 14$

5.1 Energy consumption

For each topology, we measured different parameters: the distribution of the remaining energy (average figure, and variance in) distribution of the remaining energy by partitioning the network into logic levels. (in figure)

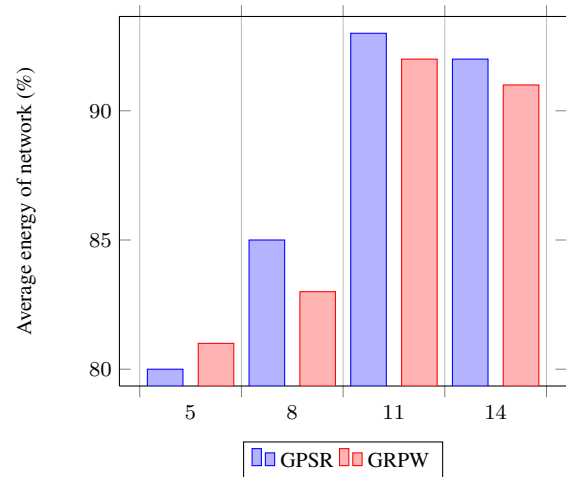


Fig. 8. Average remaining energy in different Density δ

5.1.1 The distribution of the remaining energy on the network. Because of the inflexibility of the selection of the next neighbor not localized to transmit a packet, the GPSR protocol leaves many unused nodes and uses the same nodes for transmitting a data stream. This explains the fact that the remaining energy of the entire network is just over the remaining energy in the case of the Protocol GRPW as shown in Figure 8. However, the energy distribution in the network is well distributed with GRPW compared to GPSR, as illustrated in Figure 9. However, energy is

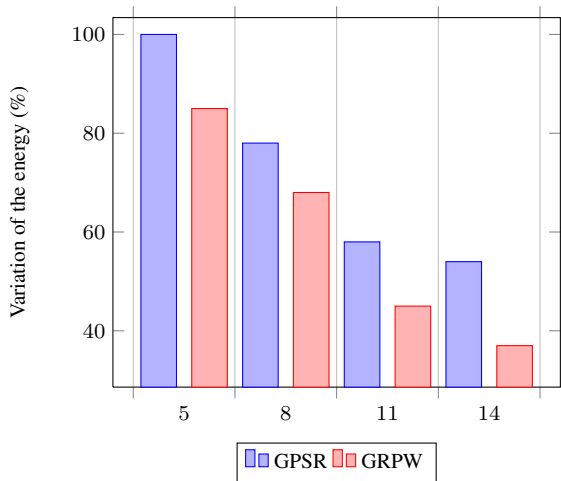


Fig. 9. Distribution of the remaining energy in different Density δ

better distributed over the network with GRPW than GPSR, and this is because in the case of the Protocol GRPW, many nodes that are not localized are used to enable efficient routing.

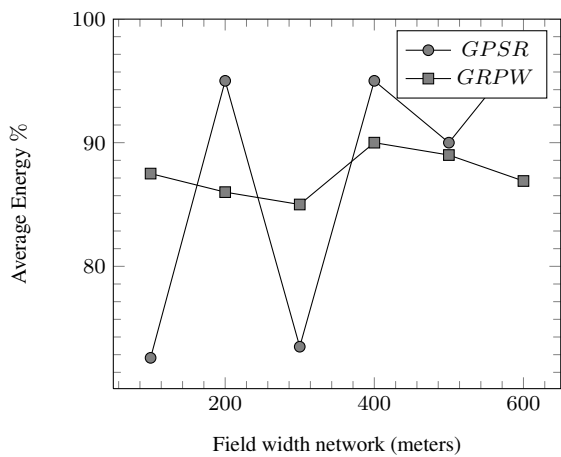


Fig. 10. Distribution of the remaining energy with network density $\delta = 14$

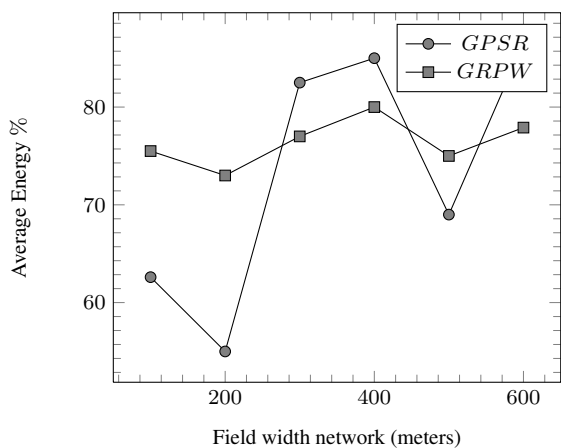


Fig. 11. Distribution of the remaining energy with network density $\delta = 5$

5.1.2 The distribution of the remaining energy in large-scale. Figure 10 and Figure 11 show the average of the remaining energy on the network partitioned on virtual logical levels. We may remark easily that the energy is uniformly distributed within the network protocol with GRPW, compared to GPSR protocol. The advantage of such functionality for a routing algorithm is that it preserves the partitioning of a network into multiple subnets completely disconnected if there are nodes that fail or non-localized nodes in the case of GPSR protocol. In this case, no path end-to-end is available.

5.2 Delivery rate and transmission time

The graphs represented in figure 12 show the impact of percentage of non-localized. In high density of localized nodes, whatever value of the delivery rate is close to 100%. In low density of localized nodes GRPW outperforms to GPSR. Through the use of several not localize node to route packets, the transmission delay, end-to-end is significantly reduced by using GRPW. The packet loss rate as a function of percentage of non-localized nodes was also decreased as shown in Figure 12. These improvements in the delivery of packets can be explained by:

- percentage of nodes not used by GPSR, especially non-localized node.
- The use of a single path to forward packets to put all flow will generate a long stay in queues leading to traffic congestion.
- Packets can be lost due to severe resource constraints (buffer size is very limited).

These results clearly demonstrate the ability of GRPW to issue binding flows and improve the quality of service compared to the protocol GPSR.

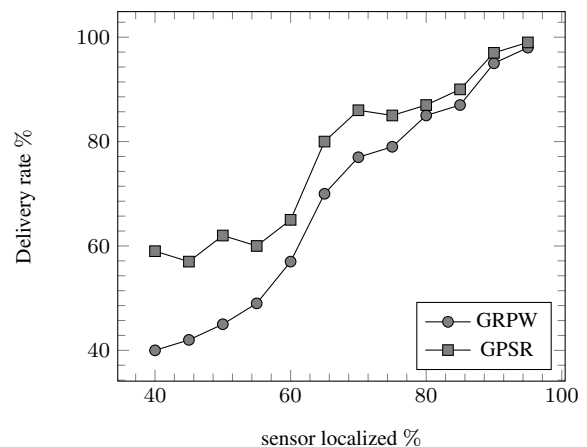


Fig. 12. density of sensor localized with Network density $\delta = 14$

6. IMPACT OF VOID SIZE

Figure 13 shows the impact of void size on the delivery rate. There is a single T void. Increasing the size of a void may not increase the number of dead ends, but will increase the number of upstream sensors whose packets fall into the dead ends. Therefore, the delivery rate increases with void size in GPSR. It also increases in GRPW but much more lowly. The performance gap between GRPW and GPSR increases when there are more voids. In Figure 14, shows the impact of void size on the rate of control messages. It is observed that the increase in the size of a void leads to an increase in the routing overhead since the rate of control messages is proportional to the number of nodes. In general, GRPW outperforms the two other studied protocols in all cases. This is expected since in both GRPW variants, we have only the

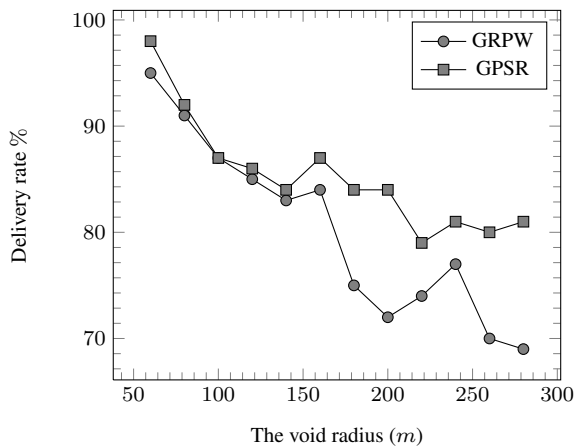


Fig. 13. Delivery rate (%) depending on the size of the void

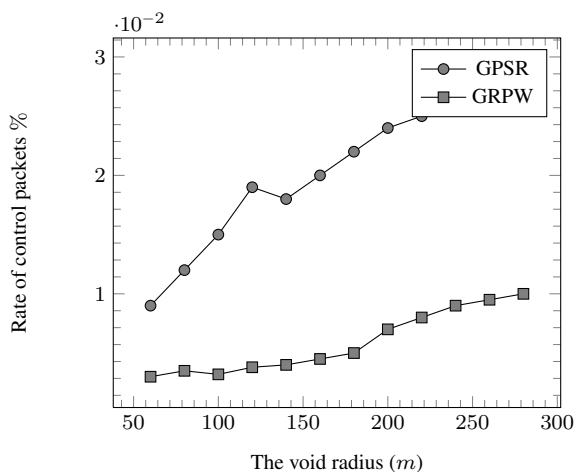


Fig. 14. Rate of control packets (%) depending on the size of the void

hello messages and the hellobelonging messages which are sent periodically and are independent of topological changes. While in GPSR, the control messages used for route discovery and route maintenance are sent frequently due to the rapidly changing topology of the network. Although GRPW uses only 'hello' and 'hellobelonging' messages as control messages, it shows higher routing overhead than GPSR. This is because GRPW does not need as many hello messages sent as GPSR to maintain its neighbouring table and its routing logical levels.

7. CONCLUSION

In this paper we present an algorithm called GRPW for assigning coordinates to nodes using the logical levels in a wireless network (to be used for geographic routing) that does not require nodes to know their location. Our key contribution is a relaxation algorithm that associates virtual logical levels to each node. These virtual logical levels are then used to perform geographic routing. Simulation results show that the success rate of greedy routing with virtual coordinates is exceedingly close to the success rate of greedy routing using true coordinates. Furthermore, in some cases such as in the presence of obstacles, greedy routing with virtual logical levels significantly outperforms greedy routing with true coordinates. Intuitively, this is because virtual coordinates reflect the network connectivity rather of the nodes' true locations which are less relevant in the presence of obstructions.

We plan to extend this work in four directions. First, we intend to

continue the study of our algorithm by simulation using more realistic link layer models and network topology. Finally, we plan to implement our algorithm on sensor nodes, and study it in real scenarios.

8. REFERENCES

- [1] The network simulator. <http://www.isi.edu/nsnam/ns/>, cited July 2010.
- [2] Opnet technologies. <http://www.opnet.com/>, cited July 2010.
- [3] About glomosim. <http://pcl.cs.ucla.edu/projects/glomosim/>, cited July 2011.
- [4] Omnet++ community site. <http://www.omnetpp.org/>, cited July 2011.
- [5] Prosenjit Bose, Pat Morin, Ivan Stojmenovic', and Jorge Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. In *WIRELESS NETWORKS*, pages 609–616, 2001.
- [6] D. S. J. De Couto and R. Morris. Location proxies and intermediate node forwarding for practical geographic forwarding. In *Proc. 4th Annual ACM/IEEE Intl. Conf. on Mobile Computing and Networking (MobiCom '98)*, Dallas, TX, 1998.
- [7] J. M. Kahn, R. H. Katz, and K. S. J. Pister. Next century challenges: mobile networking for smart dust. In *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, MobiCom '99, pages 271–278, New York, NY, USA, 1999. ACM.
- [8] Brad Karp and H. T. Kung. GPSR: greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, MobiCom '00, pages 243–254, New York, NY, USA, 2000. ACM.
- [9] Evangelos Kranakis, Harvinder Singh, and Jorge Urrutia. Compass Routing on Geometric Networks. In *Proc. 11th Canadian Conference on Computational Geometry*, pages 51–54, Vancouver, August 1999.
- [10] Fabian Kuhn, Rogert Wattenhofer, Yan Zhang, and Aaron Zollinger. Geometric ad-hoc routing: of theory and practice. In *Proceedings of the twenty-second annual symposium on Principles of distributed computing*, PODC '03, pages 63–72, New York, NY, USA, 2003. ACM.
- [11] P. Levis, A. Tavakoli, and S. Dawson-Haggerty. Overview of Existing Routing Protocols for Low Power and Lossy Networks. *IETF, Internet-Draft draft-ietf-roll-protocols-survey-07*, April 2009.
- [12] Jinyang Li, John Jannotti, Douglas S. J. De Couto, David R. Karger, and Robert Morris. A scalable location service for geographic ad hoc routing. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, MobiCom '00, pages 120–130, New York, NY, USA, 2000. ACM.
- [13] James Newsome and Dawn Song. Gem: Graph embedding for routing and data-centric storage in sensor networks without geographic information. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, SenSys '03, pages 76–88, New York, NY, USA, 2003. ACM.
- [14] Ananth Rao, Sylvia Ratnasamy, Christos Papadimitriou, Scott Shenker, and Ion Stoica. Geographic routing without location information. In *Proceedings of the 9th annual international conference on Mobile computing and networking*, MobiCom '03, pages 96–108, New York, NY, USA, 2003. ACM.

- [15] Clément Saad, Abderrahim Benslimane, and Jean-Claude König. AT-Dist: A Distributed Method for Localization with High Accuracy in Sensor Networks. *International journal Studia Informatica Universalis, Special Issue on "Wireless Ad Hoc and Sensor Networks"*, 6(1):N/A, 2008.
- [16] Ben L. Sayan, Ben Leong, Sayan Mitra, and Barbara Liskov. Path Vector Face Routing: Geographic Routing with Local Face Information. In *Proceedings of the 13th IEEE International Conference on Network Protocols, ICNP '05*, pages 147–158, 2005.
- [17] Timothy J. Shepard. A channel access scheme for large dense packet radio networks. *SIGCOMM Comput. Commun. Rev.*, 26(4):219–230, August 1996.
- [18] Timothy J. Shepard. A channel access scheme for large dense packet radio networks. In *Conference proceedings on Applications, technologies, architectures, and protocols for computer communications, SIGCOMM '96*, pages 219–230, New York, NY, USA, 1996. ACM.
- [19] Y.SABRI and N.EIKAMOUN. A Distributed Method for Localization in Large-Scale Sensor Networks based on Graham's scan. *Journal of Selected Areas in Telecommunications (JSAT)*.