

Performance Evaluation of Gradient Routing Strategies for Wireless Sensor Networks

Fadila Khadar and Tahiry Razafindralambo

INRIA Lille - Nord Europe, IRCICA/LIFL, CNRS UMR 8022, Univ. Lille 1
Parc Scientifique de la Haute Borne
50, avenue Halley - BP 70478
59658 Villeneuve d'Ascq, France
tahiry.razafindralambo@univ-lille1.fr, fadila.khadar@inria.fr

Abstract. We consider Wireless Sensor Networks (WSN) applications in which sensors have to send data to a unique sink in a multi-hop fashion. Gradient routing protocol is a scalable way to route data in these applications. Many gradient routing protocols exist, they mainly differ in their performances (delay, delivery ratio, etc.). In this paper, we propose an extensive performance evaluation study of some gradient routing protocols in order to give guidelines for WSN developers.

Keywords: Sensor Network, Routing protocol, Gradient.

1 Introduction

A Wireless Sensor Network (WSN) is a set of autonomous objects with limited processing and storage capabilities which cooperate in order to perform a common task. They are receiving more and more attentions due to their potential applications in various areas such as monitoring, security and data gathering. The communication paradigm in WSN is very characteristic since in most applications nodes send data only to a sink. Furthermore, they operate on limited capacity batteries. As a result, WSN protocols' design, especially routing protocol, is a challenging task.

The routing problem in WSN has been the subject of intense studies. One important difference between wired and wireless networks is the use of location for routing purposes. Position awareness improves the efficiency and scalability of routing protocols as it helps reducing the number of messages used for route discovery. This information could be obtained by GPS or by an internal service. GPS is not a practical solution in WSNs, as sensors would have to be equipped with additional hardware, which would increase both cost and energy consumption. Yet, other solutions exist, such as localization protocols based on trilateration and triangulation ([1],[2]). Other solutions include the use of virtual coordinates, i.e., a coordinate system set up for routing purposes only. This is how work gradient routing protocols. They create a one-dimensional virtual coordinate system where the position of a node corresponds to its hop distance to the sink. This information is then used to efficiently route packets to the sink in

a multi-hop fashion. When receiving a packet, a node forwards it if it is closer to the sink than the previous sender. Since each sensor receiving the packet decides locally whether it should forward it, no state information about its neighborhood has to be kept, thus reducing routing complexity.

Many variants of gradient routing protocols have been proposed in the literature. They mostly differ in the way a node decides whether it should forward a received packet. As a result, their performances concerning the message delivery ratio, the end-to-end delay, and the number of nodes involved in the transmission of a message also differ.

We propose an extensive performance evaluation of different gradient routing algorithm strategies for wireless sensor networks. We identified three basic schemes: basic, probabilistic and unicast schemes. The evaluation of these strategies shows that there exists a trade-off between the four considered metrics (delay, delivery ratio, overhead and energy consumption).

This paper is organized as follows. In Section 2 we give a brief review of gradient-based routing protocols. In Section 3 we describe the assumptions we make. We then describe the algorithm we consider in Section 4. The simulation results are discussed in Section 5. Finally we conclude in Section 6.

2 Related Work

A number of routing protocols have been proposed for wireless sensor networks [3]. They aim at providing energy efficient, low delay and low overhead routing process. In this section we briefly describe data centric routing protocol in which the sink sends a request to certain regions (or all the network) and waits for data from the sensors which are located in these regions. For a more complete review of existing routing protocols for sensor networks please refer to [3] or other surveys.

The most classical protocols are flooding and gossiping [4] [5] [6] [7]. In flooding, each sensor broadcasts the packet to all its neighbors. The basic gradient protocol described in this paper is equivalent to flooding except that only a subset of sensors are allowed to broadcast packets. Flooding is very easy to implement and delivery ratio can be very high. However, it has several drawbacks, the most important one being the overhead generated by the transmission of a single message. The probabilistic flooding described in this paper is very similar to gossip algorithm especially the one described in [6]. Gossiping algorithms alleviate the problem of overhead compared to flooding algorithm. However delivery ratio can be very low.

Direct diffusion [8] is one of the first routing protocols that introduces gradient. In this protocol, an interest is flooded by the sink. The gradient is a reply link to a neighbor from which the interest was received. Rumor routing [9] is a variant of direct diffusion where flooding is not used.

In Gradient-based routing protocol [10], the authors propose a variant of Directed diffusion. The authors keep the number of hops when the interest is flooded into the network. Therefore each sensor can evaluate the number of hops

to the sink. The difference between a node's height and the one of its neighbor is considered as the gradient on that link. A packet is forwarded on a link with the largest gradient. In this paper, we use this version of gradient routing protocol as a basis. We do not use the version proposed in [10] in this paper since we do not consider data aggregation. Moreover, we mainly focus on forwarding scheme and consider that sensors do not have any information on possible next hops for the forwarding scheme.

3 Background and Assumptions

We assume a large number of quasi-static wireless sensor nodes and time-driven applications: the main task of each sensor node is to periodically collect data from the sensing devices. These data are then sent to the sink in a multi-hop fashion due to a limited radio range. The data are sent toward the sink using a geographic routing based on gradient. The sensor nodes are assumed to use CSMA/CA MAC protocol such as 802.15.4 [11]. Acknowledgement scheme is used for unicast packet but broadcast packets are not acknowledge.

Each periodically collected data is stored into a packet P and broadcasted by the node. This packet also contains the depth of the node ($P.depth = node.depth$), the source of the packet ($P.src = node.id$) and a sequence number ($P.seq = node.seq$) which is generated by the node.

Each node also keeps the list of sequence numbers of packets it has already forwarded. It is worth noting that our application does not need to store neighbors' list nor exchange discovery message (after the initialization phase) due to the stateless aspect of the presented protocols.

In this paper, the term "broadcast" stands for message propagation in a node's neighborhood and the term "flooding" refers to network-wide message propagation. In the sequel, given a node n , we use the denotation $n.x$ to refer to the variable x at the node n and given a message M and a name x that identifies a field, we use the denotation $M.x$ to refer to the field x in message M .

4 Algorithm Description

In this section, we describe the gradient establishment and the three algorithms that we evaluate. The gradient establishment or gradient construction is the scheme used to help routing process. The three algorithms are all based on the same gradient construction. The only difference between the algorithms is the way sensors forward each received packets to the sink.

4.1 Gradient Construction

A BUILD message is flooded in the network starting from the sink s . The BUILD message contains the source of the message src which is the id of the node and the $depth$ of the node. Initially, the sink sets $BUILD.depth = s.depth = 0$ and $BUILD.src = s$. Each node n that receives the BUILD message set its own $n.depth$

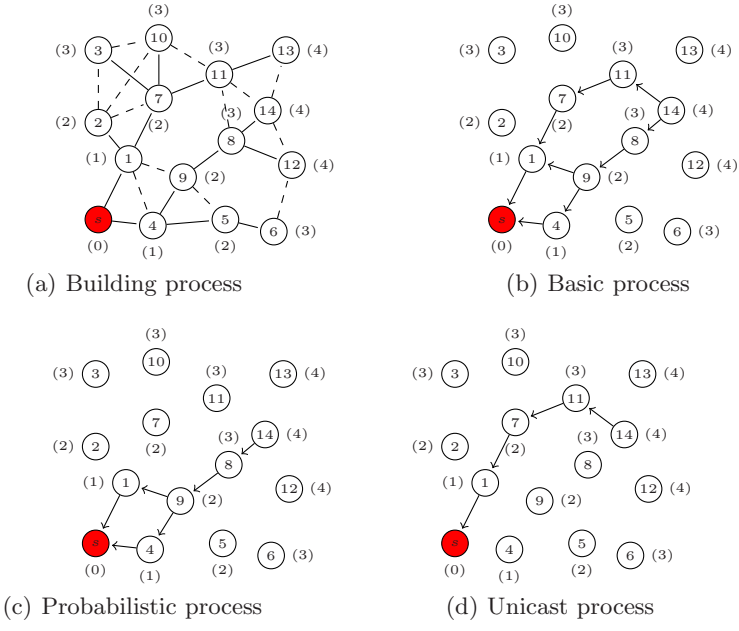


Fig. 1. Illustration of the gradient routing. The sink is node s . In Figure 1(a) Lines represent BUILD message flooding used to set node depth. The number in brackets represents the node depth. Dashed lines represent communication links. Figure 1(b), 1(c) and 1(d) are examples of basic, probabilistic and unicast process based on the depth of each node.

to $BUILD.depth + 1$ and broadcasts the BUILD message with its own $depth$ and also changes the BUILD.src field. A node can receive a BUILD message more than once. In this case, the node keeps the smallest $depth$ received among all the BUILD messages but does not send the BUILD message if it has already sent one. See Figure 1(a) for the resulting process.

4.2 Basic Gradient Routing Protocol

Upon receiving a packet P which is not a BUILD message a node computes algorithm 1. It first checks if the depth in the packet field is greater than its own depth and if the sequence number of the packet is not in its packet sequence number list. If these conditions (Line 2 of Alg. 1) are met, the packet depth is changed to the actual node depth and the sequence number of the packet is added into the sequence list. If the node n is not the sink the packet is broadcasted.

Figure 1(b) gives an example of the paths obtained by the basic gradient routing protocol. Here, node 14 broadcasts a message, only nodes 8 and 11 forward it since their depths are smaller than the depth of node 14.

Algorithm 1. Basic Gradient routing protocol – Forwarding process

Message forwarding on node n :

```

1: Reception of packet P
2: if ( $P.depth > n.depth$ ) && ( $P.seq \notin n.seqlist$ ) then
3:   if ( $n \neq sink$ ) then
4:      $P.depth = n.depth$  ;
5:      $n.seqlist \leftarrow P.seq$ 
6:     broadcast packet P ;
7:   else
8:     Process packet P ;
9:   end if
10: else
11:   Drop P;
12: end if

```

4.3 Probabilistic Gradient Routing Protocol

The forwarding policy is the only difference between the basic gradient routing protocol and the probabilistic one. Indeed, in the probabilistic version, a packet P is forwarded based on a probability p . Line 3 of Alg. 1 is changed to:

[3-1] **if** ($n \neq sink$ && $rand(0,1) \leq p$) **then**

The idea behind the probabilistic version of the gradient protocol is to reduce the overhead due to message forwarding while maintaining a good delivery ratio. Figure 1(c) gives an example of the paths obtained by the probabilistic gradient routing protocol. Here, node 14 broadcasts a message and only node 8 forwards it due to the probability and since its depth is smaller than the depth of node 14.

4.4 Unicast-Based Gradient Routing Protocol

The unicast version of the gradient routing is built to reduce the overhead due to forwarding while maintaining a good delivery ratio. However this version is resource costly for nodes that are in the forwarding path. It is also worth noting that this version strongly relies on gradient construction.

In this version of gradient routing, each node needs to store the packet source of the BUILD message it considers for its depth. This information is stored at each node in a variable called *hop* ($n.hop = BUILD.src$). The forwarding process described in Alg. 1 is modified on Line 6 to implement the unicast version by the following line:

[6-1] unicast packet P to $n.hop$;

Figure 1(d) gives an example of the path obtained by the unicast-based gradient routing protocol. Here, the message follows a single path based on the depth of each node.

5 Performance Evaluation

We evaluate the performances of the gradient routing protocols through simulations using WSNet¹. The performance metrics we use are delays, delivery ratio and overhead (number of duplicated messages received by the sink).

5.1 Simulation Settings

Network Topology. We consider scenario where static nodes are randomly deployed in a 1000 meters \times 1000 meters flat square. The radio range is set to be 100 meters. The physical layer is modelled by the unit disk graph model. The total number of nodes varies from 100 to 400 and only connected networks are considered. We assume only one sink (at position (0,0)) initiates the gradient construction.

Node Settings. Sensors have to send data every 10 seconds with a random jitter to avoid synchronization. Each sensor has a buffer B of size 10 to keep packets it has to forward. FIFO policy is used for packet transmission. It has also a table S of size 20 in which it stores sequence numbers of packets it has already forwarded.

Metrics of Interest. We consider three metrics:

Delay Time between the sending of a packet and the first reception of it at the sink node.

Delivery ratio number of distinct received packets on number of distinct sent packets.

Overhead average number of times the same packet is received by the sink.

Gradient Algorithms Settings. For the probabilistic version of gradient routing, the probability of packet forwarding is set to 0.5. For the unicast version, data packets are acknowledged and retransmitted in case of failure following the scheme described in the 802.15.4 standard. Broadcast packets are also sent using the 802.15.4 standard.

As our focus is to evaluate the routing process we do not try to optimize the gradient construction. It is worth noting that some optimizations avoiding unstable or long links increase the performance of the routing protocol. These optimizations on gradient construction are left to future work.

The simulation results are divided into three parts presented in Figure 2 for part I, Figures 3, 4, and 5 for part II and Figure 6 and 7 for part III.

The first part (Section 5.2) shows a performance comparison between the three routing strategies depending on the number of nodes in the network. None of the studied schemes outperforms the other ones in all the metrics. The second part (Section 5.3) studies the behavior of each protocol according to nodes' depth. As expected, it shows that the delay increases when the depth increases. In the third part we evaluate the energy consumption of each algorithm and

¹ <http://wsnet.gforge.inria.fr>

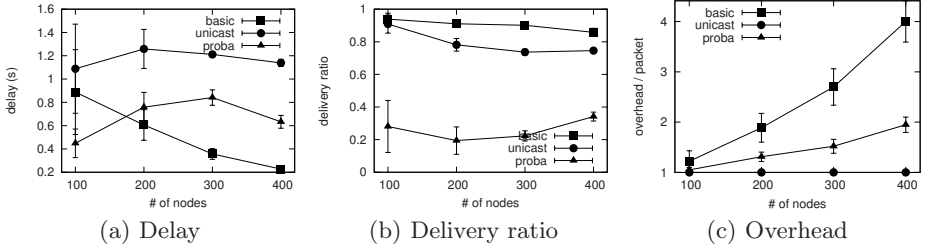


Fig. 2. Simulation results

plot the energy map at different simulation time. The results of this part show that the probabilistic version has the best performance among the three versions (Section 5.4).

5.2 Simulation Results: Density Effect (Part I)

Delay. Figure 2(a) plots the mean end-to-end delay for different number of sensors in the network. Surprisingly, when the number of nodes increases, the mean delays for the basic version of the gradient decreases. This is mainly due to the multiple path followed by each packet. This is confirmed by Figure 2(c) which shows the duplication ratio for each packet. Indeed, when the number of nodes increases, the number of possible forwarding nodes also increases. Therefore, packets can follow different paths using nodes that are not overloaded, decreasing the delay of each packet. The unicast version performs badly in term of delay. The mean delay remains constant when the number of nodes increases. This can be explained by the double effect increasing density has. On the one hand it increases the collision rate, leading to higher delays as packets must be retransmitted. On the other hand, delay decreases due to shorter paths in hop count. As a result, the end-to-end delay of the unicast version is stable.

We can notice that the probabilistic version is a shift of the basic version. Indeed, the mean delay for 400 nodes in *proba* is close to the one for 200 nodes of *basic* and the mean delay for *proba* 300 nodes is close to the one for 150 nodes of *basic*.

Delivery ratio. The Figure 2(b) depicts the delivery ratio for the three strategies. The basic scheme is more efficient than the others. The delivery ratio for *basic* is always above 85%. This is explained by the multiple paths a packet can follow. The delivery ratio is decreasing when the number of nodes increases. This is due to the increasing number of collisions when the node density increases. *proba* gives a low but increasing delivery ratio. When the number of nodes is low there may exist only one single path from a sensor to the sink. In this case, the probability that a packet is successfully received by the sink is $P_r = (p)^{hops}$ where p is the forwarding probability, a parameter of *proba* and $hops$ is the number of hops from a given sensor to the sink. When the number of nodes increases,

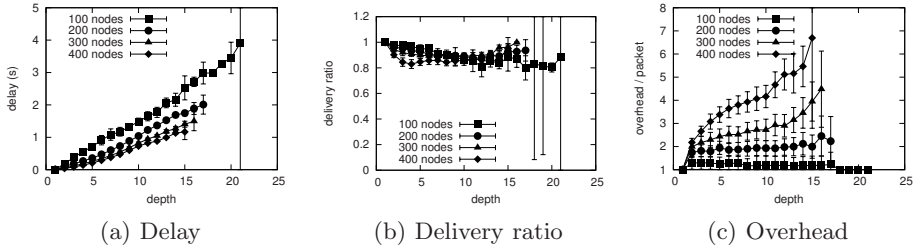


Fig. 3. Simulation results basic gradient

the probability that multiple paths exist increases and thus the delivery ratio increase for *proba*.

The delivery ratio of *unicast* is around 80%. This high value is due to the retransmission used at the MAC layer for unicast packets. However, when the number of nodes increases, the number of collisions also increases, which then reduces the delivery ratio.

Overhead. The overhead of each strategy is plotted in Figure 2(c). We can see that the overhead of *unicast* is 1, which means that each packet is received only once by the sink. This is the perfect behavior.

The overhead of *basic* is increasing with the number of nodes. This is mainly due to the multipath effect produced by the forwarding scheme. The overhead *proba* is half of the overhead of *basic* when the number of nodes is 300 and 400. It is also interesting to notice that when the number of nodes is low (100), the overhead is close to 1 for the three strategies, reflecting the fact that there is no multiple disjoint path from a sensor to a sink.

5.3 Simulation Results: Depth Effect (Part II)

Basic. The simulations for the basic gradient routing protocol are presented in Figure 3. The delay, delivery ratio and the overhead are plotted depending on the depth of the node initiating te message. Figure 3(a) shows that the delay

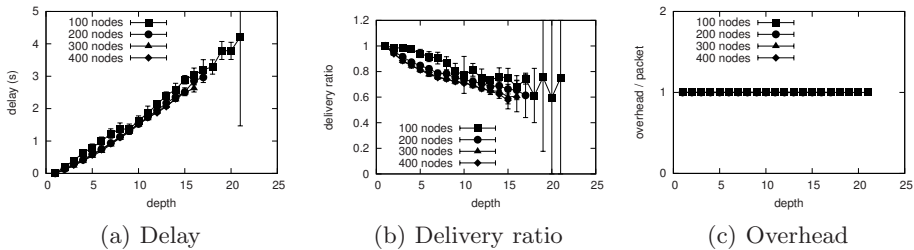


Fig. 4. Simulation results unicast gradient

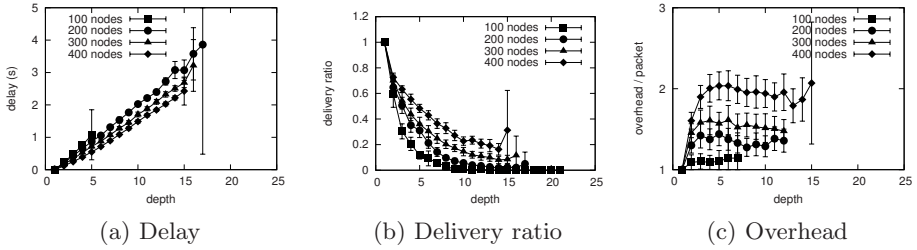


Fig. 5. Simulation results probabilistic gradient

increases when the depth increase, as expected. We can also notice that the maximum depth is lower when the number of nodes increases (21 for 200 nodes, and 15 for 400 nodes). This first result also indicates that for any depth the delay decreases when the number of nodes increases.

Figure 3(b) plots the delivery ratio depending on the depth of the originating nodes. The depth does not seem to strongly affect the delivery ratio. The latter remains roughly constant. This behavior is explained by the multiple paths followed by each packet. The overhead for each packet depending on the depth is drawn in Figure 3(c). We can see that for 100 nodes, the overhead is decreasing along with the depth. This is probably due to the small number of paths available when the source node has an higher depth. For the other densities, the overhead is increasing when the number of nodes increases. Again, this is due to the multiple paths a packet can follow when the depth increases. For 200 nodes, we can see that the overhead remains stable. For this density, only a constant number of multiple paths can be found.

Unicast. Figure 4 shows the simulation results for the unicast gradient routing protocol. The mean delay increases in the same way for each density when the depth increases (Figure 4(a)). This is due to the fact that all packets follow only one path and that the delay for a one hop communication is constant. Figure 4(b) plots the delivery ratio depending on the depth. We can see that the delivery ratio decreases when the depth increases. This is mainly due to the buffer size of each node. In the unicast version, packets follow the same path which overloads the buffer. Packets may then be dropped. When the number of hops increases (depth), the probability for a packet to be dropped also increases, which explains this behavior. In Figure 4(c), we can see that the overhead is equal to 1 as unicast routing is used.

Probabilistic. The simulation results for probabilistic gradient routing are shown in Figure 5. The results are similar to the basic gradient results (Figure 5(a)). However, the maximum depth changes depending on the number of nodes. When the number of node is 100, the maximum depth is only 5. This means that for the probabilistic version, messages coming from higher depth do not reach the sink. This also confirms the fact that for 100 nodes the

probability of having multiple paths is very low. A simple way to avoid this problem is to modify the forwarding probability depending on the node depth. With the probabilistic approach, the delivery ratio (see Figure 5(b)) decreases when the depth increases. This is foreseeable since for a given path the delivery probability decreases when the number of hops increases.

Figure 5(c) shows the overhead for each packet depending on the depth of the originating nodes. For all number of nodes, the overhead increases and then remains stable. This is due to the number of paths reduction caused by the probability of retransmission. It is worth noting that the shape of the *proba* curve of 400 nodes is similar to the shape of *basic* the curve for 200 nodes. This indicates that the effect of the probability is the same as the effect of reducing the number of nodes.

5.4 Energy Consumption (Part III)

In the previous simulation battery lifetime was set to infinity to evaluate the performance of each protocol. In this part we introduce results about energy consumption for the different algorithms. The energy model we use is very simple and linear: the energy cost of a transmission is 2, the energy cost of a reception is 1. Each sensor has initially 200000 units of battery.

The graph in Figure 6 shows the evolution of the percentage of dead nodes over time for the three algorithms. The unicast version seems to perform better than the other two versions after 160 seconds. Yet, one must be cautious with these results as they do not indicate the geographical repartition of dead nodes. Indeed, if all dead nodes are located near the sink, the network becomes useless. For better understanding the repartition of dead nodes across the network we use energy maps.

The energy maps of each algorithm at different step of the simulation (20s, 30s, 100s and 200s) for 200 sensors in a field of $1000m \times 1000m$ are drawn on Figure 7. The sink is at the bottom-left corner of each figure and has the

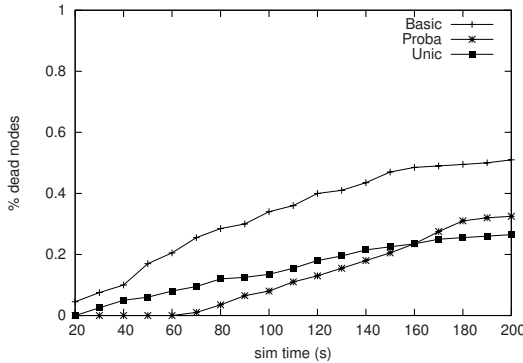


Fig. 6. Percentage of dead nodes over time for the 3 algorithms. After 160s the unicast version performs better, but the mean delay is higher (see Figure 2(a)).

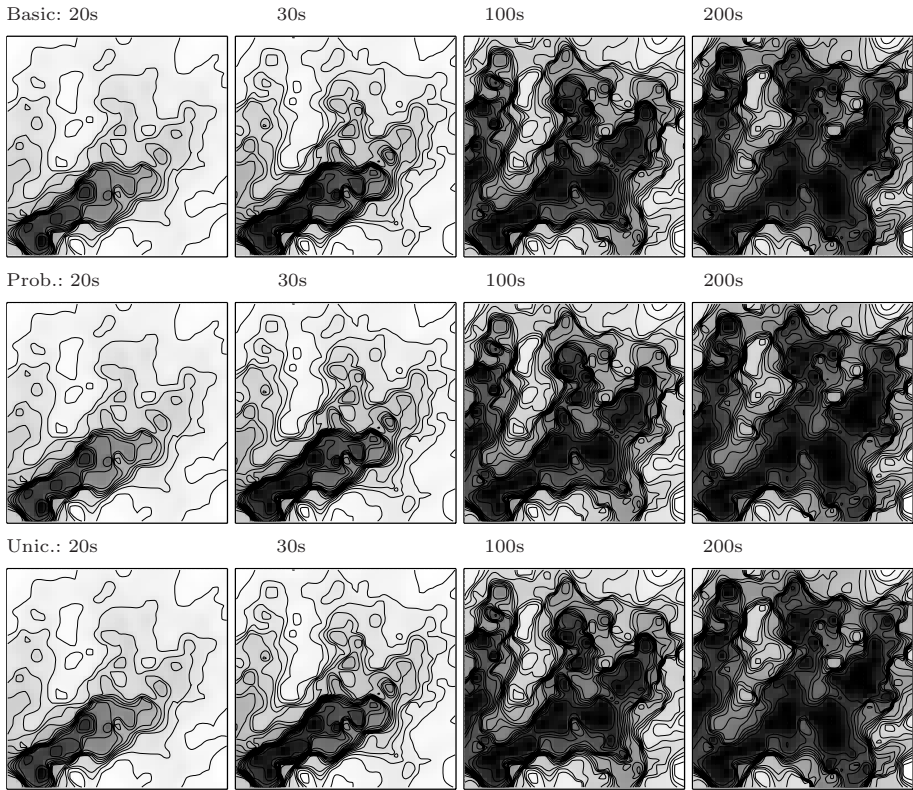


Fig. 7. Energy map after of each algorithm at 20, 30, 100, and 200 seconds of simulation. Black (blank) color means that sensors within this area have 0% (100%) battery left.

geographical coordinate (0,0). All other parameters are left unchanged. Black color means that sensors within the area have no battery left (0%). Blank color means that battery is full (100%).

We can conclude from Figure 7 that the worst algorithm is the basic version. Indeed, after 20s most of the nodes within the sink’s neighborhood have 0% battery left. This is due to the high number of sensors, especially sensors close to the sink, involved in the reception and (re)transmission of packets. For the unicast version, after 30s, most of the nodes around the sink have 0% battery left. After 100s and for the unicast version, nodes within the paths followed by the packets have almost 0% battery left. The probabilistic version exhibits better performances compared to both unicast and basic versions. We can see that for the probabilistic version, most of the nodes within the sink neighborhood are dead after 200s.

These results demonstrate that there exists a trade-off between the network lifetime and the performances of the gradient routing algorithm. Note that the results may be different depending on the energy model used but the conclusion

are still the same. It is also worth noting that the performance of the probabilistic algorithm can be enhanced by modifying the forwarding probability according to the remaining battery of each sensor such as in [12]. However, modifying this probability may affect other performance metrics.

6 Conclusion and Future Work

In this paper we evaluated three routing algorithms for wireless sensor networks. The three algorithms are based on a gradient that helps sensors to forward the data they receive to the sink. We first evaluated an algorithm where packets are forwarded by each node using local broadcast. The second algorithm is based probabilistic forwarding of the packet and the third one is based on unicast forwarding of the packet. We evaluated these algorithms based on three metrics namely the delay, the delivery ratio and the overhead. Our simulations show that the performances of each scheme are very different and that none of the studied scheme is better than the other. We also evaluate the energy consumption of each algorithm and these results confirm that there exists a tradeoff between delivery ratio and energy consumption. The choice of the algorithm must be based on what is expected from the network. If reliability is the main criteria, the basic scheme should be used at the expense of energy consumption. If the lost of some messages is not critical, the unicast version is a good compromise. The next step of this work is to evaluate the impact of having a better gradient construction and maintenance on the performance of each routing protocol. Indeed, in the gradient construction we use gradient may not be optimal due to message loss. We also want to evaluate the effect of different parameters especially for the probabilistic version of the gradient or evaluate the combination of different strategies depending on the depth of the originating sensor.

References

1. Niculescu, D., Nath, B.: Ad hoc positioning system (aps). In: GLOBECOM 2001, vol. 5, pp. 2926–2931. IEEE, Los Alamitos (2001)
2. Bulusu, N., Heidemann, J., Estrin, D.: Gps-less low-cost outdoor localization for very small devices. *Personal Communications* 7, 28–34 (2000)
3. Akkaya, K., Younis, M.: A survey on routing protocols for wireless sensor networks. *Elsevier Ad Hoc Network Journal* 3, 325–349 (2005)
4. Haas, Z., Halpern, J., Li, L.: Gossip-based ad hoc routing. *IEEE/ACM Trans. Netw.* 14(3), 479–491 (2006)
5. Maroti, M.: Directed flood-routing framework for wireless sensor networks. In: Jacobsen, H.-A. (ed.) *Middleware 2004*. LNCS, vol. 3231, pp. 99–114. Springer, Heidelberg (2004)
6. Barrett, C.L., Eidenbenz, S.J., Kroc, L., Marathe, M., Smith, J.P.: Parametric probabilistic sensor network routing. In: *WSNA 2003*, New York, NY, USA, pp. 122–131. ACM, New York (2003)
7. Hedetniemi, S.M., Hedetniemi, S.T., Liestman, A.L.: A survey of gossiping and broadcasting in communication networks. *Networks* 18(4), 319–349 (1988)

8. Intanagonwiwat, C., Govindan, R.D., Estrin, D.: Directed diffusion: a scalable and robust communication paradigm for sensor networks. In: *MobiCom 2000*, pp. 56–67 (2000)
9. Braginsky, D., Estrin, D.: Rumor routing algorithm for sensor networks. In: *Proceedings of WSNA 2002*, pp. 22–31. ACM, New York (2002)
10. Schurgers, C., Srivastava, M.: Energy efficient routing in wireless sensor networks. In: *IEEE MILCOM 2001*, vol. 1, pp. 357–361 (2001)
11. De Nardis, L., Di Benedetto, M.G.: Overview of the iee 802.15.4/4a standards for low data rate wireless personal data networks. In: *4th Workshop on Positioning, Navigation and Communication, WPNC 2007*, pp. 285–289 (March 2007)
12. Jaffres-Runser, K., Comaniciu, C.: A probabilistic interference and energy aware gradient broadcasting algorithm for wireless sensor networks. In: *3rd International Symposium on ISWPC 2008*, pp. 1–5 (May 2008)