2008

3D Geographical routing in wireless sensor networks

Mohamed Watfa
University of Wollongong, mwatfa@uow.edu.au

Alaa Al Tahan
American University of Beirut

Publication Details

Watfa, M. & Al Tahan, A. 2008, '3D Geographical routing in wireless sensor networks', International conference on wireless netowrks (ICWN'08), pp. 1-7.

3D Geographical Routing in Wireless Sensor Networks

Mohamed Watfa[1] **and Alaa al Tahan**[1]

[1]Department of Computer Science, American University of Beirut, Beirut, Lebanon

Abstract - *In this paper, we present a novel 3D geographical routing algorithm (3DGR) that makes use of the position information to route packets from sources to destinations with high path quality and reliability. The locality and high scalability of this algorithm make it suitable for wireless sensor networks. It provides high adaptability to changes in topology and recovery of link failures which increases its reliability. We also incorporate the battery-aware energy efficient schemes to increase the overall lifetime of the network. To reduce latency, a method of keeping a small record of recent paths is used. We also show that location errors still result in good performance of our algorithm while the same assumptions might yield to bad performance or even complete failures in others. Simulation results show that the power consumption and delay using 3DGR are close to optimal obtainable based on full knowledge of the network.*

Keywords: 3D Routing, geographical, Void Problem, adaptability, recovery.

1 Introduction

The use of wireless sensor networks forms a major part in next generation technology. Characteristics of sensor nodes make them suitable for use in many different fields like intrusion detection, environmental monitoring, and military applications. However, characteristics of sensor nodes require the design of new protocols that take into consideration resources scarcity in sensor nodes like memory and computing power. Another essential criterion that should be taken into consideration while designing a protocol for wireless sensor nodes is power consumption. Since sensor nodes are battery powered, energy becomes a limiting factor. In most cases, changing or recharging the battery might cost more than deploying a new node. Hence, extending the network lifetime is a critical metric in the evaluation of wireless sensor network protocols. These factors make traditional routing algorithms like distance vector and link state not suitable for the use in wireless sensor networks. In an attempt to overcome these issues, new routing algorithms have been proposed using different approaches like greedy forwarding and geographical routing [1-6]. These new approaches handle sensor nodes restrictions by using local information about neighbor nodes. However, they have their own and they make their own assumptions which limit the use of such algorithms to specific environments that satisfy these assumptions. One of the major assumptions made by geographical algorithms is assuming that nodes are deployed in a 2D plane. Such an assumption is invalid in real life scenarios and hence these algorithms cannot be applied in most situations. Three-dimensional modeling of the sensor network would reflect more accurately the real-life situations. Some applications of the results presented in this paper are disaster recovery, mapping topographical properties, space exploration [7], and undersea monitoring [8].

Also, most geographic routing algorithms for sensor networks that were proposed in the last years were evaluated using simulation tools that were based on exact location information of each node. Since this is an unrealistic assumption in most sensor networks, the simulation results cannot be directly applied to real deployments.

These unrealistic assumptions make the need of routing algorithms that work in three dimensional spaces a necessity to fit real applications. The algorithm proposed in this paper provides a new approach that works in three dimensional spaces and takes into consideration all restrictions imposed by the nature of sensor nodes. The rest of the paper is organized as follows. Related research work is summarized in Section 2. In Section 3, 3DGR is analyzed and compared with GPSR analytically. Simulation results are presented in Section 4. We conclude this paper in Section 5.

2 Related Work

In WSNs routing, approaches that depend on either proactive routing, like dynamic Destination Sequenced Distance Vector DSDV[1], Optimized Link State Routing OLSR[2] or reactive routing, like Ad-hoc On demand Distance Vector AODV [3] still have significant problems with resources scarcity and communication overhead when the topology changes frequently due to mobility of nodes. Although approaches that are based on flooding or directional flooding like DREAM [4] have high robustness, they also have significant overhead resulting from flooding and may still fail when there are no nodes in the area in the direction of flooding within flooding angle. Ideas based on random walking like Rumor Routing [5] are limited in use to specific situations where events and queries occurrences are within a specific range. Rumor Routing uses flooding as a recovery strategy which leads to overhead in communication and paths chosen are not always optimal.

Another approach, geographical routing [6], has been proposed to be used as an alternative to routing algorithms in WSNs. Messages are not sent to designated devices identified by some sort of network address, but rather to geographic locations. Every node knows its position (*either through GPS or after running a localization algorithm*) and tries to route packets in the direction toward the destination location. Geographic routing has the advantage that it is more scalable due to the lesser need for routing information. A common problem that faces this kind of algorithms is localization errors in addition to other problems related to the specific approach used. Some of these location based algorithms use restricted directional flooding like DREAM [4], Hierarchical approach like Terminodes [9] and Grid routing, Quroum system like HomeZone and GLS [10], greedy approaches like Most Forward within R (MFR), Nearest with Forward Progress (NFP), and Compass Routing [11]. The greedy approaches provide efficient communication complexity of $O(\sqrt{n})$ where n is number of nodes in the network. The main problem that faces greedy approaches is the void problem. Void problem arises when there is no node closer to destination than the sender and thus results in failure of the greedy approach in finding a path to the destination although one might exist. Some algorithms like GPSR [12] solve the void problem by using the right hand rule; however, GPSR shares with all location based algorithms proposed so far the assumption that all nodes are roughly in plane (i.e. the use of planer graphs). Such assumption is not valid in real applications where nodes are distributed in three dimensional spaces [13]. Moreover, GPSR needs to build a planar graph before the routing algorithm can be applied. Also, it has been shown in [14] that the performance of GPSR decreases significantly with the increase in localization errors. Another drawback of GPSR is that packets follow boundary edges while traversing holes in the network which causes nodes on the boundary to be depleted quickly. Funke and Milosavljevic propose MGGR algorithm [15] which is macroscopic variant of geographic greedy routing. MGGR performs better than GPSR with imprecise node locations. However, MGGR introduces the use of land marks in addition to the need to form planar sub-graphs. MGGR also has a higher average communication cost per message than GPSR. Kim et al [16] proposed another approach to remove non-planarities using cross link detection protocol CLDP.

An approach to solve void problem without planarization, GDSTR, is suggested in [17]. In this approach the algorithm handle void problem by switching to route on a spanning tree that is likely to make progress toward the destination until it reaches a node where greedy routing can be continued. Although the build of planar graph is avoided, GDSTR needs to build spanning tree and each node needs to maintain information about the area covered by the tree below each of its tree neighbor.

To overcome some of the problems due to the use of actual coordinates like localization errors, the use of virtual coordinates has been proposed [18]. In virtual coordinates nodes' locations are specified relative to some reference fixed nodes. This reduces localization errors problem but requires flooding at initialization from the reference nodes in order for other nodes to compute their positions. On the other hand, this makes the system vulnerable to signal fading in initialization phase. Also the conventional void problem is replaced by another void problem of the same nature when the node is closer to destination than all its neighbors even using relative coordinate's measures. Moreover, some nodes may have identical virtual coordinates although they may be far apart. In 3DGR, the algorithm proposed in this paper, geographical routing is applied to three dimensional spaces. Although we assume, as in previous algorithms, that links between nodes are bidirectional, we do not assume radio ranges are uniform and that they cover unit disks. Hence, it overcomes problems and restrictions with previous algorithms and there is no need to build a planar graph as in GPSR or MGGR. It also provides a higher successful delivery ratio with high tolerance to localization errors. In addition, this algorithm provides a way of shortcutting where a path to the same destination is found at an intermediate node. When such a path exists, the algorithm switches to another mode where there is no need for routing anymore. It rather follows the already existing path and therefore the overhead incurred in the routing process is avoided.

3 Algorithm Description and Analysis

In this section, we start by explaining the techniques we used in our algorithm which are geocasting, recent path, and battery awareness measures. Then, we provide a general description of 3DGR for the initialization phase as well as the sending and receiving phases. A detailed flowchart is presented. Finally, the complexity analysis of 3DGR is discussed.

3.1 Geocasting Technique

The purpose of geocasting is to send a message to nodes in a specific geographic region. The problem is reduced to checking whether a point belongs to a specific region in three dimensional spaces. To simplify the problem, we take advantage of the fact that projection preserves order i.e. if a point (x,y,z) is in the region bounded by any three dimensional shape, then its projection on any plane belongs to the projection of the shape on that plane. Without any loss of generality, our 3D region is the intersection of the sphere representing the range of the source and the cone whose head is the source node and head angle α is specified to suit the application as depicted in Fig.1. A node $P(x_p,y_p,z_p)$ belongs to the set of nodes within the geocasting area if it satisfies the conditions:

- P is in the same direction of the destination D according to the sender S. This can be verified by checking that P and D are on the same side of the line perpendicular to SD and passing thourgh S.
- $d(PP') \leq d(SP') \times \tan(\alpha)$ (1)

where S and D are the locations of the source and destination nodes respectively and P' is the projection of P on (SD). If there aren't any nodes in the targeted region, the source node will receive no reply and it will increase the head angle α

and resends the request. This will increase the targeted region to include more nodes.

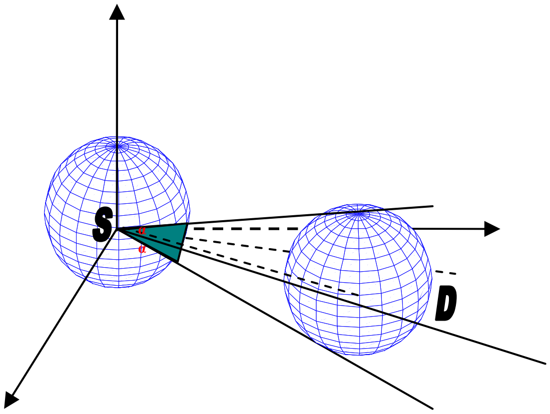

Fig. 1 3D view of source S and destination D. Geocasting region is the intersection of source range sphere and the cone whose head is the source and its head angle α (shaded area).

If again no node replies, the angle α is increased again and if the angle exceeds a threshold, the request is locally broadcasted. The choice of α and the method of increments depends mainly on the density of the network. If the density is high then α should be chosen to be small to conserve energy. If the density is low then α should be chosen to be large to get more nodes to respond. The same strategy applies for the method of increment. If the density is high, a small increment in the angle will include a significant number of new nodes whereas when the density is low a large increment is needed to include enough new nodes.

3.2 Recent Path

To decrease routing overhead, recent paths to destinations are maintained locally and temporally. Initially nodes have no recent paths for any destination. When a node wants to forward a packet, it uses the algorithm to pick the next node to which the packet will be forwarded, forwards the packet to it and add a recent path specifying it as the next node on the path to the destination. Thereafter, whenever the sender wants to forward a packet to the same destination, the packet is forwarded directly to next node on the path without the need of applying any algorithm and hence saving a significant overhead and delay. Storage of recent paths is done using a dynamic link list. Recent paths are added dynamically for each destination whenever a node forwards a packet intended to that destination and the path is removed from the recent path list when it expires after a specific time. This makes the size of the list very efficient where it will be limited to the number of destinations during recent path expiry interval. Hence, even if source-destination pairs are chosen randomly in the network, the size of recent path's list is limited to the number of destinations during one expiry interval. The size of the list is reduced more when the network topology is made of a limited number of destination sinks.

Recent path's expiry interval can be updated dynamically by the routing algorithm depending on several factors. Monitoring neighbors list is one of the factors that can be used in updating expiry interval. When the neighbor's list is updated frequently as in highly mobile networks, the interval is reduced to accommodate for the dynamicity of the network. Also, the size and the rate of data packets are included in updating expiry interval in order to take into consideration the energy and battery state of neighbor nodes. When data packets are large and the rate is high, nodes on the path are depleted quickly and the interval is reduced to result in a better load distribution over the network.

3.3 Battery Awareness Optimization

Recent study in battery technology helps us better understand the battery behavior [19]. Unlike what we used to believe, the energy consumed from a battery is not equivalent to the energy dissipated in the device. When discharging, batteries tend to consume more power than needed, and can reimburse the over-consumed power later. The process of the reimbursement is often referred to as battery recovery. This behavior is due to chemical characteristics of batteries. So if the battery is given time to recover, its energy can be used more efficiently.

Based on a discrete time battery model, we present an optimization to 3DGR protocol to dynamically schedule routing in sensor networks. Our algorithm is aware of the battery status of network nodes and schedules recovery to extend their lifetime. We evaluate the performance of our routing algorithm with and without the battery awareness optimization in the simulation results. The nature of network traffic as packets allows us to assume a discrete model for the battery life time. Several battery models have been discussed in literature [20]. In this paper, we use the battery state to refer to the recovery state of the battery and battery energy to refer to the energy still stored in the battery. Also, we create a simple model considering that battery state decreases with each sending or receiving and recovers when the node is idle. We also consider that discharging and recovering have the same rate. Using more sophisticated battery models is part of an extended version of this paper. The battery recovery state is given by:

$$B_{t+t_0} = B_t \pm (\frac{E}{E_0} \times t_0)\% \qquad (2)$$

where B is battery recovery state, E is battery energy at current time, E_0 is the initial battery energy, and t_0 is the timer interval. To incorporate battery awareness in our routing algorithm, we use the following formula for path evaluation:

$$\min((w_1 D + w_2 \frac{1}{B} + w_3 \frac{1}{E}), P) \qquad (3)$$

where w_1 is weight assigned to distance D factor and w_2 is weight assigned to battery state B calculated from equation (2). w_3 is the weight assigned to the energy still stored in the battery. P the set of available paths received from nodes that have replied to the request. Our goal will be to minimize D and maximize B and E.

3.4 Initialization Phase

When the nodes are initially deployed, each node will broadcast one HELLO packet which includes their position information and will schedule another HELLO packet to be sent at a random time. This random scheduling of the second HELLO packet is to reduce collisions of HELLO packets during the initialization interval where all nodes will be sending HELLO packets. When a node receives a HELLO packet, it checks if the sender is already in its list of neighbors. If it is not, it adds the sender to its neighbor list. It then checks if it is within the random time scheduled for the second HELLO packet (which means the node is still in the initialization phase) then it does not reply with a HELLO packet. If the node is not in the time scheduled for the second HELLO packet, then the HELLO packet received is from a new node added to the network; hence the node broadcasts a HELLO packet to inform the new node about itself. If the sender is already in the neighbors' list, it silently drops the packet.

3.5 Sending and Receiving Phase

When a source wants to send a packet to some destination, it starts by checking if it has a recent path to that destination. If such a path exists, the packet is forwarded to the next node in the path. Otherwise, it geocasts a small request packet that includes the coordinates of the destination and setting a timer (R_t).

When a node receives a request packet, it checks if the sender is already in its neighbors' list. If not, it assumes that it has missed the HELLO packet sent by this neighbor and adds it to its neighbors' list. This improves the discovery of nodes but does not eliminate the need of HELLO packets because each node needs to know its neighbors to respond to requests. Also, some of its neighbors may never want to send packets and hence will never send requests. So, if no HELLO packets are exchanged, they will not be discovered. Then, each node that has heard the request, checks if it is in the intended region specified by the request packet. If not, it silently drops the packet. Otherwise it checks for a recent path to the requested destination (*the time interval in which a path is considered recent is specified to suit the application and environment conditions*). If a recent path exists, it sends a response for the request indicating that. Otherwise, it checks its neighbors' list and picks the closest one to the requested destination (*or the best in terms of an evaluation function incorporating some other metrics such as battery levels*) then sends a response for the request specifying the closest distance to the destination it can reach, the estimated cost of energy to reach there and the status of its battery (*only nodes that have heard the request packet will reply hence node failure will be detected automatically*).

When the timer R_t expires, the node checks if the replies it received contain a recent path leading to the packet being forwarded on that direction. If there is no recent path then it checks if there is a path to a node closer than itself to the destination. If such a path exists, it forwards the packet to the neighbor who can reach the closest node to the destination with minimal cost and taking into consideration the battery status. If there does not exist any reply message that indicates a path to a node closer to destination than the source (*either there are no neighbors in that direction or those neighbors suffer a void problem*), then the geocasting angle is increased and the process is repeated again. This gives one more chance to nodes that where included in the previous casting in case there were some difficulties during the last transmission.

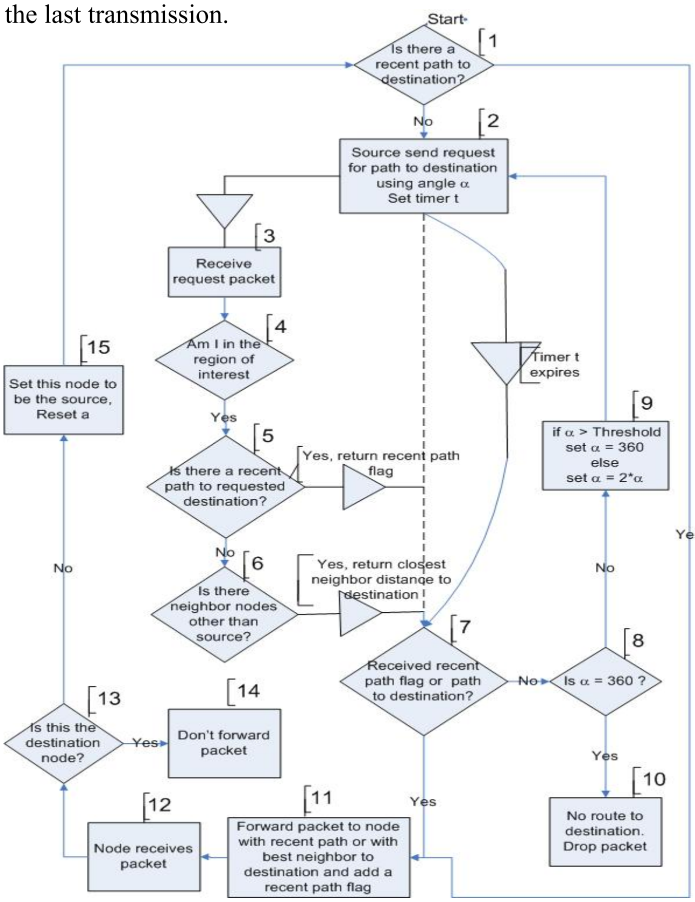

Fig. 2 The flowchart of the sending algorithm

When the geocasting angle reaches 2□(*i.e. broadcasting*), the source will forward the packet to the neighbor than can reach closest to the destination even if that location is farther than the source itself (*this forces the algorithm to try all possible paths in this case starting from the best one available*). This allows backtracking where the packet may be forwarded to the old source. However, to prevent routing loops, this option will be the last one after using the 2□ geocasting option (*i.e. if there are no neighbors except the original source*). When a node receives a packet, it checks if it is the destination node. If it is, then no forwarding is needed. Otherwise, it repeats the sending process described above. The flowchart of the routing algorithm is provided in Fig. 2.

3.6 Preventing looping and pingponging

When a packet is forwarded, the last option for a node will be to backtrack to the source from which it has received the

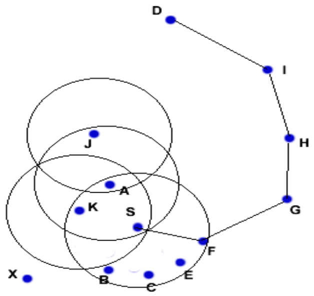

Fig. 3 A scenario illustrating indirect looping.

packet. Hence, when the packet is backtracked, another path will be tried automatically and this will prevent pingponging. To prevent indirect looping, we take advantage of the fact that to have a loop; the packet must be forwarded in the direction opposite to the greedy choice at some point i.e. away from the destination. Fig. 3 provides an example of such a case. In this case, when node S sends the request for the first time, nodes K and A will reply with possible paths while node J will not reply since it cannot forward the packet to any node. Since node A has received the packet from node S, it will make S as its last choice. Hence, the packet is forwarded to K which is farther to the destination from A itself so A will put a backtrack flag indicating that it has forwarded the packet in a backwards direction. When K receives the packet, it will send a request and since it received the packet from A it will make A its last choice. This forces the packet to be forwarded to S. When S receives the packet, it will notice that A has put a backtrack flag and this will cause it to black list A. The fact that S has received the packet from K makes it mark K as its last choice, the packet is forwarded to F and the recent path from A to D is updated to indicate that F is the next node. The packet is then forwarded from F to G→H→I→D. This operation is done only during path discovery afterwards the recent path will indicate the right direction directly.

3.7 Complexity Analysis

In this section, we analyze the computational complexity of our proposed algorithm. The analysis is divided to two parts:

Sending Phase: The complexity of the inner loop is related to the initial angle of geocasting and the method of incrementing it. In our case, each increment doubles the angle, the complexity will be $O(log_2\left(\frac{2\pi}{\alpha}\right))$. The complexity of the outer loop is related to the topology of the network and the distribution of nodes. Assuming uniform distribution, the complexity of algorithm is linearly dependent on the diameter of the network. This gives complexity $O(\sqrt[3]{n})$, where n is number of nodes in the network. Hence, the overall complexity of the sending phase will be in worst case $O(\sqrt[3]{n} \times log_2\left(\frac{2\pi}{\alpha}\right))$ where n is number of nodes and α is the initial geocasting angle.

Receiving Phase: The only parts that contain considerable computation are: i) checking if the source of the request is already in neighbors' list; ii) choosing the closest of the neighbors to some destination. Hence, the receiving complexity is related to number of communication neighbors of a node. Hence, the complexity of the receiving phase $O(N)$ where N is number of nodes within the communication ranges.

4 Simulation Results

Network Simulator NS-2 is used. We decided to do the simulations on 2D regions since there is no three dimensional geographical routing algorithm to compare with. Most of the results presented were also obtained on 3D regions using 3DGR. Sensor nodes were randomly deployed on square fields with the source and destination chosen randomly. Also, we test the tolerance of our algorithm for localization errors. Delay, power consumption, packet delivery ratio and network lifetime are evaluated and the results from applying 3DGR are compared with the results obtained from using GPSR. Also, a path optimality measure is defined and 3DGR is compared with GPSR using this optimality measure. Simulation is done on networks of 100 nodes deployed in 300 x 300 area. We adopt byte division for sending and receiving energy. Idle listening also consumes some energy that is significantly lower than sending and receiving energy. We also consider that the node has 50 joules initially. Five source-destinations pairs are chosen randomly and results are based on the average of five simulation runs of the algorithm. Metrics used in comparison are: end-to-end delay, energy consumption, tolerance to localization errors, and path optimality.

End-to-end delay: As shown in Fig. 4, results of simulations show that end-to-end delay for 3DGR is larger than that for GPSR in the initialization phase; however, 3DGR delay becomes much smaller. In this simulation the recent path record is not updated. If the recent record updating parameter is activated, delay graph shows a pulse every time the recent record is updated; however, the overall effect on average end-to-end delay is negligible. Recent path updating interval can be set to suit application noting that there is an inverse relationship between recent path interval and adaptability to topology changes.

Energy consumption: Energy is taken as the average energy per node calculated over intervals of 10 seconds. As shown in Fig. 5, simulation results show that 3DGR conserve significant energy compared to GPSR. 3DGR has a small increase consumption when paths are built then power conservation is more with the use of recent path. On the other hand, GPSR's energy consumption per node continues to decrease linearly.

Network lifetime: The metrics used in evaluating system lifetime is the number of active nodes. The overall lifetime is the continuous operational time of the system before the percentage of active nodes drops below a specified threshold (for example 90%). For evaluating the battery awareness in our algorithm, we use formula (3) in choosing the best path and we assign equal weights for distance and battery factors. As can be seen in Fig. 6, 3DGR can effectively incorporate battery and energy awarness and a major increase in the system life time (about 80%) is obtained.

Localization errors: Results from earlier research work in the field of geographical routing are based on the assumption that each node has knowledge about its exact location. This assumption is inappropriate in real deployments, since location information is gained either through GPS signals or some localization algorithm, both of which are error-prone. An evaluation of greedy forwarding algorithms and GPSR in case of location errors can be found in [14, 21]. In plain greedy mode, a high packet drop rate due to false dead ends was observed. The drop rate increases with higher network density. Values up to 50% were observed at location errors of 0.2 *r* in dense networks (*r* is the transmission range). Furthermore, the impact on the optimal path rate was investigated. The simulations showed that up to 53% of the paths were non-optimal; these results, however, are not very significant, since they say little about the actual path lengths.

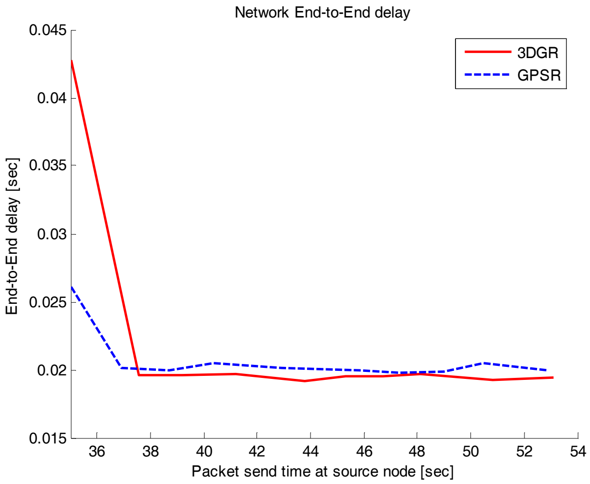

Fig. 4 End-to-End delay using 3DGR and GPSR. 3DGR takes longer time in initialization phase then it outperforms GPSR.

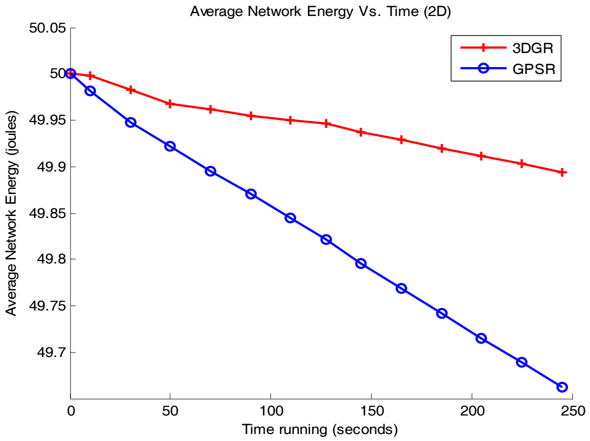

Fig. 5 Average energy per node as a f unction of time using 3DGR and GPSR.

To measure path optimality, we measure the ratio of number of hops in the path picked by our algorithm to number of hops in an optimal path. For this purpose, we propose the following metric:

$$opt_A(s,d) = \frac{O_p(s,d)}{A_p(s,d)} \qquad (4)$$

where $O_p(s, d)$ is number of hops in the optimal path from *s* to *d* and $A_p(s, d)$ is number of hops in the path resulting from algorithm *A*. Then we take our value to be:

$$\overline{opt(s,d)} = \frac{1}{k}\sum_{i=1}^{k} opt_A(s,d) \qquad (5)$$

The location errors follow a two-dimensional Gaussian distribution $N(0, \sigma 2)$. The standard deviation σ is varied in steps of 5 meters between 0 (which means no location errors) and 40 m, which is the transmission range. The two-dimensional Gaussian distribution implies that the distance between real and estimated location follows a Rayleigh distribution with expected value $\sigma\sqrt{\frac{\pi}{2}}$.

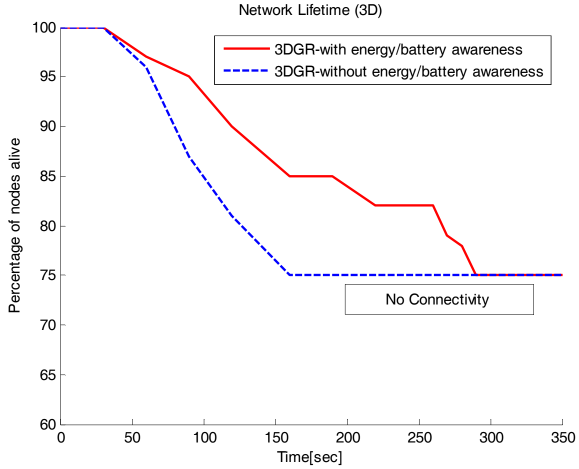

Fig. 6 The system lifetime using battery awareness feature

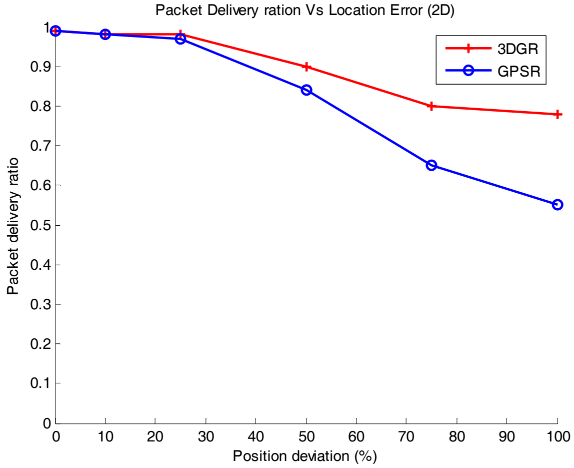

Fig. 7 Packet delivery ratio as a function of the location error

In Fig. 7, the packet delivery ratio is plotted as we vary the position deviation as a percentage of the transmission range. GPSR has very bad delivery performance when the position deviation is high. This is mainly the result of incorrect planarization, which leads to loops on the perimeter. 3DGR has a much higher delivery ratio at large position errors, because its recovery strategy is much more error-tolerant.

In Fig. 8, the optimality measure of both 3DGR and GPSR are analyzed. At low position errors (0-25%), both algorithms perform close to optimal; however as the percentage of deviation increases (25-100%), GPSR's performance decreases drastically until it even fails. 3DGR, on the other hand, continues to perform close to optimal due to its resilience to location errors. 3DGR's recovery strategy outperforms GPSR's planarization which leads to longer paths.

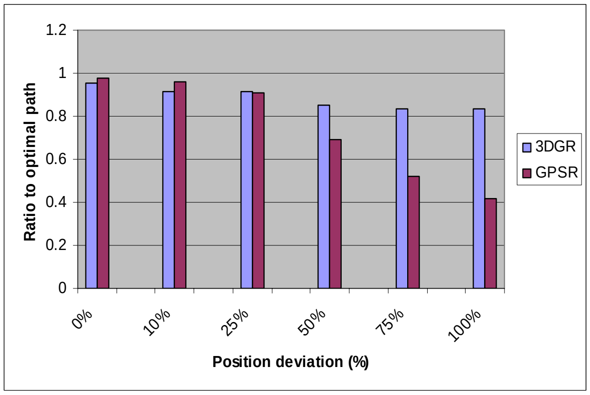

Fig. 8 Optimality measure as a function of the location error

5 Conclusions

In this paper, we propose a novel 3D geographical routing algorithm that takes into consideration special characteristics of wireless sensor networks and eliminate assumptions made by algorithms proposed before. We show that 3DGR (*with the ability of operating in 3D spaces*) has better or comparable results to other algorithms like GPSR. Although 3DGR gets the benefits of geographical information to route packets, has a relatively high tolerance for localization errors and close to optimal path. The incorporation of battery model leads to the extension of network life time and better distribution of loads. In future work, our focus will be on optimizing parameters in the evaluation function and incorporating another parameter for cost to optimize power consumption. Experimenting with other battery models and optimizing battery function will form another side of our future work.

6 References

[1] C. Perkins, and P. Bhagwat, "Highly-dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," proceedings of the SIGCOMM '94 Conference on Communications, Architectures, Protocols, and Applications London, UK, pp 234–244, September 1994.

[2] Z. Ren, Y. Zhou, and W. Guo, "An adaptive multi-channel OLSR routing protocol based on topology maintenance," proceedings of the IEEE International Conference on Mechatronics & Automation, Niagara Falls, Canada, pp 2222-2227, July 2005.

[3] C. Perkins, and E. Royer, "Ad-Hoc on-demand distance vector routing," proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, LA, USA, pp 90–100, February 1999.

[4] S. Basagni, I. Chlamtac, V. Syrotiuk, and B. Woodward, "A Distance routing effect algorithm for mobility (DREAM)," proceedings of the MobiCom, TX, USA, pp 76–84, October 1998.

[5] D. Braginsky, and D. Estrin, "Rumor routing algorithm for sensor networks," proceedings of First Workshop on Sensor Networks and Applications (WSNA), GA, USA, pp 22-31, September 2002.

[6] M. Mauve , J. Widmer, and H. Hartenstein, "A Survey on position-based routing in mobile ad-hoc networks," IEEE Network, pp 30–39, November/December 2001.

[7] X. Hong, M. Gerla, R. Bagrodia, P. Estabrook, T. Kwon, and G. Pei, "The mars sensor network: Efficient, power aware communications," Proceedings of IEEE Military Communications Conferences (MILCOM 2001), McLean, VA, October 2001.

[8] F. Akyildiz, D. Pompili, and T. Melodia, "Underwater acoustic sensor networks: Research challenges." Ad Hoc Networks (Elsevier), vol.3, pp. 257-279, May 2005.

[9] L. Blazevic, L. Buttyan, S. Capkun, S. Giordano, J.-P. Hubaux, and J.-Y. Le Boudec, "Self-organization in mobile ad-hoc networks: the approach of terminodes," in IEEE Communications Magazine, pp 39-45, June 2001.

[10] J. Li, J. Jannotti, D. Decouto, D. Karger, and R. Morris, "A Scalable location service for geographic ad hoc routing," proceeding of MobiCom, MA, USA, pp 120–30, August 2000.

[11] E. Kranakis, H. Singh, and J. Urrutia. "Compass routing on geometric networks," proceedings of the 11th Canadian Conference on Computational Geometry, Vancouver, BC, pp 51–54, August 1999.

[12] B. Karp and H. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," in Proceedings of Mobicom, Boston, MA, USA, pp 243-254, August 2000.

[13] S. Commuri and M. Watfa, "Coverage strategies in 3D wireless sensor networks," International Journal of Distributed Sensor Networks, pp 333-353, October 2006.

[14] M. Witt and V. Turau, "The impact of location errors on geographic routing in sensor networks," proceeding of International Conference on Wireless and Mobile Communications, pp 76-81, July 2006.

[15] S. Funke and N. Milosavljecvic, "Guaranteed delivery geographic routing under uncertain node locations," IEEE INFOCOMM,pp 1244-1252, Anchorage, AK, USA, May 2007.

[16] Y. Kim, R. Govindan, B. Karp, and S. Shenker, "Geographic Routing Made Practical," in Proceedings of NSDI 2005, pages 217- 230, May 2005.

[17] B. Liang, B. Liskov, and R. Morris, "Geographic Routing without Planarization," in Proceedings of USENIX Symposium on Networked Systems Design and Implementation, Apr. 2006.

[18] A. Caruso, S. Chessa, S. De and A.o Urpi, "GPS Free Coordinate Assignment and Routing in Wireless Sensor Networks," INFOCOM 2005

[19] R. Rao, S. Vrudhula, and D. Rakhmatov, "Battery modeling for energy aware system design," IEEE Computer Society, pp. 77-87, December 2003.

[20] C. Ma and Y. Yang, "Battery aware routing for streaming data transmissions in wireless sensor networks," Mobile Networks and Applications, pp 757-767, May 2006.

[21] Y. Kim, J.-J. Lee, and A. Helmy, "Modeling and Analyzing the Impact of Location Inconsistencies on Geographic Routing in Wireless Networks," ACM SIGMOBILE Mobile Computing and Communications Review, vol. 8, no. 1, pp. 48–60, Jan. 2004.