The Impact of Location Errors on Geographic Routing in Sensor Networks

Matthias Witt Volker Turau
Hamburg University of Technology, Institute of Telematics
Schwarzenbergstraße 95, 21073 Hamburg, Germany

Abstract

Geographic routing in wireless sensor networks is based on the prerequisite that every node has information about its current position, for instance via GPS or some localization algorithm. This location information has a certain degree of inaccuracy in real deployments. The majority of geographic routing algorithms, however, has been designed for nodes with exact position information. We show that location errors yield bad performance or even complete failures.

Two elaborated geographic routing algorithms for sensor networks, GPSR and BGR, are evaluated with the nodes having inaccurate location information, varying the standard deviation of the position error between zero and the transmission range. Simulation studies show a vast decrease of the packet delivery ratio. To enhance both algorithms, fixes for them are presented to improve the delivery ratio and to save energy in case of location errors.

1. Introduction

Wireless sensor networks offer new possibilities in environmental and industrial monitoring. In the near future, it will be possible to spread hundreds or thousands of small sensor nodes in large areas in order to monitor data about the environment. These data will eventually be routed to a base station, the so-called *sink*. The sink may also want to send messages to nodes at a designated location.

This scenario suggests that *geographic* routing schemes should be used in sensor networks. 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. In most algorithms, the nodes only need information about their near neighborhood. Regarding the sparse storage resources available in micro devices, this is the only viable solution. Furthermore, the information about the position of a node is more important than its identity. In other words, it is not important, *which* node generated a message, but *where* the node is. Thus, location information is likely to be essential in sensor network applications.

Several geographic routing algorithms for sensor networks were proposed in the last years, most of them were evaluated using simulation tools. These simulations, however, were based on exact location information of each node, because the algorithms were designed to work for nodes with exact position information. Since this is an unrealistic assumption in most sensor networks, the simulation results cannot be directly applied to real deployments. This paper presents the evaluation of two different elaborated geographic routing algorithms, GPSR and BGR, simulated with location errors. Afterwards, fixes for GPSR and BGR are presented and evaluated that increase the packet delivery ratio and decrease the number of sent packets in the presence of location errors.

The remainder of this paper is organized as follows: Section 2 gives a short introduction to the geographic routing algorithms discussed in this paper. Section 3 covers related work. In Section 4, the impact of location errors on these algorithms is evaluated through simulation. In Sections 5 and 6, the GPSR and BGR algorithms are improved to perform better in case of location errors. Finally, Section 7 recapitulates the contribution of the paper.

2. Geographic Routing Algorithms

In this section, two elaborated geographic routing algorithms for wireless sensor networks are presented, GPSR and BGR. While GPSR forwards packets based on neighborhood information, BGR needs no topology information at all since it uses a contention-based scheme.

Geographic routing algorithms generally work in two different ways, namely greedy forwarding and face routing. When using greedy forwarding, the packet is routed to the neighbor that is closest to the destination. This scheme, however, fails if there is no neighbor that is closer to the destination than the current hop. On the other hand, face routing works on a planarized neighborhood graph (i. e., crossing edges are removed) and forwards packets along faces. Face routing has been shown to be correct if the neighborhood graph is connected [1]. Improvements of face routing can be found in [2].

In GPSR [1], each sensor node holds a table containing its direct neighbors. All nodes periodically broadcast short beacon messages so that the neighborhood tables are updated. Packets are routed using greedy forwarding. If there is no neighbor that is closer to the destination, the packet is forwarded in perimeter mode, which is in fact face routing. Greedy mode is used again as soon as the packet reaches a node that it closer to the destination than the node where the packet entered perimeter mode.

BGR [3] uses no topology information, which is useful when the topology changes rapidly or the wireless communication is unreliable (which is the case in most real deploy-

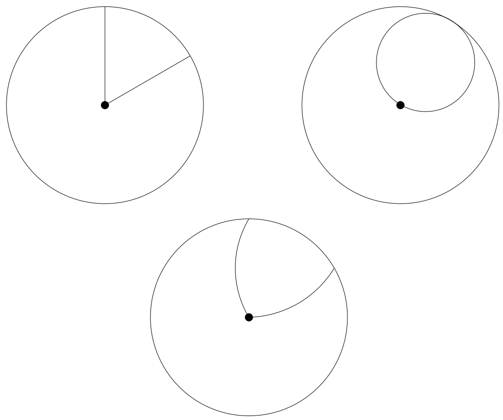

Figure 1. Forwarding areas in BGR: sector, circle, and Reuleaux triangle

ments [4]). Packets are simply forwarded via broadcast; all nodes which are located in a designated area called *forwarding area* start a timer. The timer function ensures that the timers of nodes closer to the destination expire first. The forwarding area must be small enough so that all nodes within it can communicate with each other, and large enough to contain a sufficient number of nodes. BGR proposes a 60° sector, a circle, or a Reuleaux triangle, as shown in Figure 1. The timer of the node that is closest to the destination expires first and forwards the packet again. All other nodes within the forwarding area also receive this packet and know that it has been forwarded, so they cancel their timers. If no node forwards the packet, another try is initiated by turning the forwarding area by 60°. If this also fails, the forwarding area is turned in the other direction; if this one is also empty, the packet is dropped.

While GPSR guarantees delivery if the network is connected, BGR may fail to deliver a packet if the network density is sparse. In GPSR, however, the communication overhead is much higher due to the periodically generated beacon messages.

3. Related Work

Simulation results with ns-2 [5] for both routing algorithms have been presented by the respective authors. These results, however, 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 in case of location errors can be found in [6]. Through simulation, it was found out that delivery rate and path length remain acceptable up to location errors of about 40 % of the transmission range. Other modes than the greedy mode were not investigated.

Location errors in GPSR have been studied in [7]. Greedy and perimeter mode were investigated separately. 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. It is a difference whether the path is merely 1 % or 100 % longer than the optimal path. A simple boolean value (optimal or non-optimal path) is not enough for a clear understanding.

Regarding the perimeter mode, a phenomenon called *planar graph collapse* has been studied, which means that an edge is not removed due to location errors, but it should be. Since this is not the only possible planarization error, this analysis is not sufficient either. In perimeter mode, a packet drop rate up to 28 % was observed at location errors of $0.2\,r$.

A fix for GPSR in case of location errors has been proposed by Seada et al. [8], who found out that most of the failures are due to incorrectly removed edges. Therefore, they proposed that, before a node u removes an edge (u, v), it sends a message to v, who responds only if it also sees the neighbor w. Only when u receives a positive response, the edge is discarded. This modification results in a much higher success rate in their simulations. The position error in their simulations, however, is uniformly distributed between zero and the maximum error, which is not an appropriate model. For modeling errors, Gaussian distributions should be used.

Another study can be found in [9]. Here, a geographic routing protocol is analyzed that uses greedy mode where possible and flooding to route around obstacles and voids. Hence, this protocol is very energy-consuming. Analytical computations and simulation runs reveal that performance starts dropping at location errors of about 20 % of the transmission range; when using 2-hop neighborhood information, however, this can be improved up to 40 %. Unfortunately, they also use a uniform distribution of the location error.

4. Studying the Impact of Location Errors

The insufficiency of prior investigation led to the decision to re-run the simulations of GPSR and BGR with ns-2 giving the nodes incorrect information about their "real" positions. The location errors follow a two-dimensional Gaussian distribution $\mathcal{N}(0, \sigma^2)$. 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. (Errors in the order of magnitude of the transmission range can occur in localization schemes [10].) The two-dimensional Gaussian distribution implies that the distance between real and estimated location follows a Rayleigh distribution with expected value $\sigma\sqrt{\frac{\pi}{2}}$. Every value is the average of 20 simulation runs with different randomly generated topologies. To make the results more comparable, the same 20 topologies were used for all experiments.

The scenario for the experiments is as follows: 150 nodes are randomly distributed in an area of $150\,\text{m} \times 150\,\text{m}$. The sink is placed in a corner, all nodes know the exact position of the sink. In intervals of 10 seconds, all nodes send a packet toward

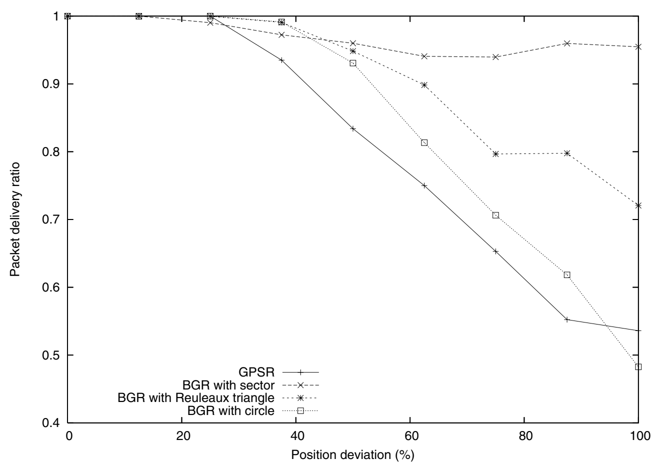

Figure 2. Packet delivery ratio in GPSR and BGR with different forwarding areas, position deviation plotted as percentage of transmission range

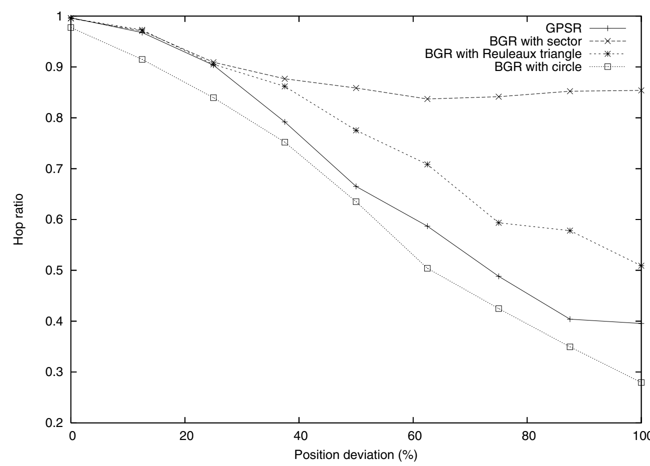

Figure 3. Optimal hop count in proportion to actual one

the sink in turn. In order to wait for building the neighbor tables in GPSR, a start phase of 100 seconds is added. The beacon interval in GPSR is 10 seconds.

Figure 2 shows the packet delivery ratio in GPSR and BGR (using different forwarding areas). The position deviation is plotted as percentage of the transmission range. Obviously, 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. BGR has a much higher delivery ratio at large position errors, because its recovery strategy is much more error-tolerant. The Reuleaux triangle outperforms GPSR at all position error ranges. The high delivery ratio of the sector at high location errors is due to massive packet duplication.

In Figure 3, the optimal hop count in proportion to the actual hop count is depicted. For instance, if the optimum is 9 hops and the packet needed 10 hops, a ratio of 0.9 is computed. For packets that did not reach the destination the value is zero. Except from the circle, which in general has a bad performance regarding the hop count (cf. [3]), BGR outperforms GPSR. Again, BGR's recovery strategy appears to be more successful than GPSR's planarization, which leads to longer paths.

5. Improving GPSR

An attempt to adopt the fix from Seada et al. [8] in the original ns-2 GPSR implementation revealed that it is not applicable because of way too many packet collisions, even with random backoff delay in the order of magnitude of one minute. Seada et al. used an ideal MAC and physical layer without packet loss, hence they did not face this problem.

Another fix that removes crossing links is suggested in [11]. To detect cross-links, each node sends probes along all of its links using the right-hand rule until they return to the originating node. This fix sends even more packets than the one cited above, hence it is just as little applicable in real deployments.

To make it applicable, the fix from Seada et al. has been modified: No packets are exchanged during the planarization phase; instead, the planarization is done using 2-hop neighborhood information. The IDs of the neighbors are added to the periodically sent beacon messages. When receiving a beacon, not the entire neighborhood of this neighbor needs to be stored, but only those which are located within the Gabriel circle. This reduces memory requirements.

Originally, a re-planarization is triggered when either a new node is detected or a node has been removed from the neighborhood. This policy is not sufficient anymore: re-planarization must be done when the neighborhood of a neighbor has changed. So the following must be added to the fix in order to make it work correctly: When the neighborhood of a node changes, a flag is set in the next three beacon messages that forces the receiving nodes to re-planarize. (The flag is sent multiple times as beacons can get lost. The number of three was chosen because this is the number of beacons that must have been missed in order to remove a node from the neighborhood table.) For this period of three beacons, the use of implicit beacons is disabled. Implicit beacons are regular data packets that are regarded as beacons so that the scheduling of the next regular beacon is delayed, but since they do not contain neighborhood information, they cannot be used during the planarization phase. This leads to a small increase of the total number of packets.

Figure 4 shows the success rate of GPSR with this fix compared to the original GPSR implementation. The success rate increases significantly at high location errors; at lower errors, however, the success rate decreases by a little amount. This is the result of more collisions due to the increase of the number of packets (see above).

The proposed fix has two shortcomings:

1) More memory space is required to store 2-hop neighborhood information.
2) The beacon messages are much longer, which leads to more energy consumption and delivery failures.

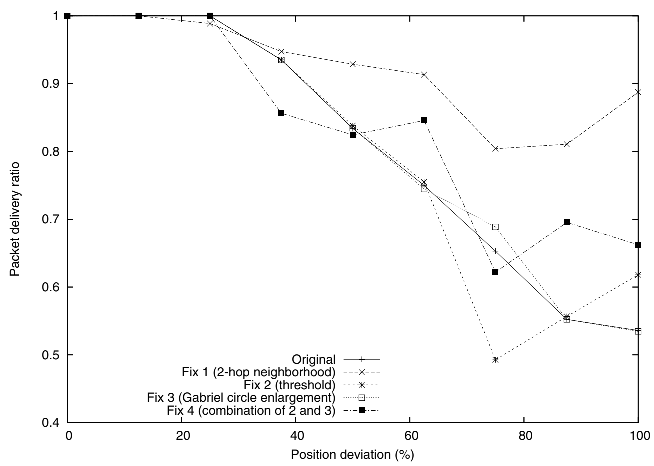

Figure 4. Success rate of original GPSR and several fixes

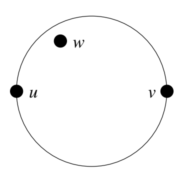

Figure 5. Node u sees node v, node w is within the Gabriel circle; can node v also see w? All nodes are shown at estimated locations.

To solve both problems, a probabilistic approach can be applied: When a node u sees a node w within the Gabriel circle (u, v), the decision whether the link (u, v) should be removed is based on the question if node v also sees node w (see Figure 5). This is the case when the real distance between v and w is at most the transmission range r. This leads to the following question: *If the real distance between two nodes is d, what is the expected value for the estimated distance e?*

For the calculation, we assume that one node has an error-free location at $(d, 0)$; the other node is located at $(0, 0)$ with its estimated position (x, y) being distributed following the sum of the two Gaussian distributions of both nodes, which is $\mathcal{N}(0, 2\sigma^2)$.

The function $g(x, y)$ calculates the estimated distance e (see Figure 6):

$$g(x, y) = \sqrt{y^2 + (d - x)^2}.$$

The random variable G describes the estimated distance e;

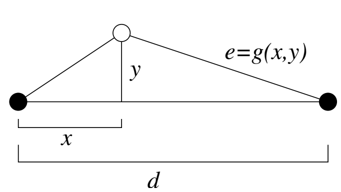

Figure 6. Real distance d and estimated distance e between two nodes (represented as black circles; the white circle denotes the estimated position of the left node)

Table 1
Numerically calculated values of $E(G)$ assuming $d = 40$

σ	5	10	15	20	25	30	35	40
$E(G)$	40.6	42.6	46.2	51.3	57.5	64.4	71.8	79.5

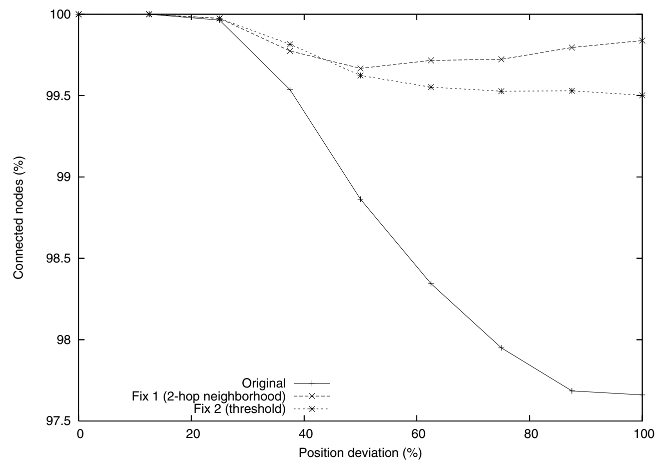

Figure 7. Average percentage of fully connected nodes after planarization

$f(x, y)$ is the two-dimensional Gaussian pdf. The expected value of G is

$$\begin{aligned} E(G) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x, y) f(x, y) \, \mathrm{d}x \, \mathrm{d}y \\ &= \frac{1}{4\pi\sigma^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \sqrt{y^2 + (d - x)^2} \ \mathrm{e}^{-\frac{x^2+y^2}{4\sigma^2}} \, \mathrm{d}x \, \mathrm{d}y. \end{aligned}$$

Unfortunately, this integral cannot be solved analytically. But if d and σ are fixed, the value can be calculated numerically. The only necessary value is the result for $d = r$, hence the result can be stored in the nodes as a constant t. If the estimated distance between two nodes is above t, the real distance is assumed to be above r, so it is assumed that the two nodes do not hear each other. Table 1 lists the numerically calculated values of $E(G)$ in case $d = 40$, which is the transmission range used in our experiments.

This leads to a new potential fix for GPSR that avoids the exchange of the 2-hop neighborhood. The decision whether to remove the link to node v is based on the estimated distance e between v and w. The link is removed if $e \leq t$. Since node u knows that node v will remove the link only if the distance between u and w is not above t, the link is kept when this distance is above t. Thus, the link is only removed if both the distance between u and w and between v and w does not exceed the threshold t. This leads to a more consistent planarization.

The evaluation of this fix, also depicted in Figure 4, is disappointing: The success rate is only marginally higher

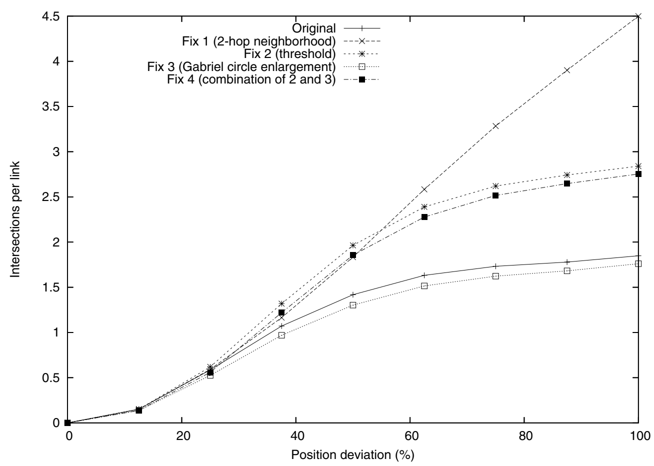

Figure 8. Average number of intersections per planar link

than in the original GPSR algorithm; at location errors of about 75 %, the delivery rate even drops considerably. Further analysis revealed that the problem of disconnection due to incorrect edge removal (unlike claimed in [8]) is *not* the main problem of location errors in face routing. To show this, the planarization of 1000 randomly generated topologies was analyzed. In Figure 7, the average percentage of fully connected nodes after the planarization phase is depicted. These results show that only a few nodes are isolated. Moreover, the two fixes increase the number of connected nodes, but since fix 2 has such a bad performance compared to the original algorithm, disconnection cannot be the main problem.

Instead, the main problem is intersection of links. Figure 8 depicts the average number of intersections per planar link. Note that the two fixes described so far result in more planar links than the original GPSR, because a condition is evaluated before a link is removed. Thus, more links are retained. To avoid the intersection of links, however, *fewer* links should be retained. This is achieved through a third fix, which enlarges the Gabriel circle (cf. Figure 5) by $\frac{\sigma}{2}$. (A further enlargement would lead to too many isolated nodes.) This fix results in fewer link intersections, as can be seen from Figure 8. Additionally, Figure 4 demonstrates that this fix leads to a higher success rate, but only for medium and very high location errors.

In fact, fix 3 has a higher success rate than the original GPSR where fix 2 has a lower one, and vice versa. This raises the question if a combination of both fixes yields a better delivery rate. Thus, this fourth fix both enlarges the Gabriel circle and removes links based on the threshold. Figure 4 reveals that this fix indeed performs better where fix 2 is better; at other position errors, however, the success rate is still worse.

This indicates that the impact of the various planarization errors varies at different position deviations; a probabilistic approach for finding the correct links to remove seems hardly to exist. A probabilistic fix that does always have a higher delivery rate than the original GPSR must therefore select the appropriate algorithm depending on the position deviation. Fix 1, however, performs much better than the original GPSR and the probabilistic fixes at medium and high location errors. Thus, this fix should be used if its drawbacks, i. e., larger memory requirements and longer beacon packets, are tolerable.

6. Improving BGR

The timer function used by BGR can be improved in order to better disperse the calculated times when the position information is incorrect. The original BGR algorithm uses the following timer function, where d is the forwarder's distance to the destination, c is the candidate's distance to the destination, and r is the transmission range:

$$t(c) = \max\left(0, Max\_Delay \cdot \left(1 - \frac{d-c}{r}\right)\right).$$

This function calculates a value of Max_Delay when the distance of the candidate node is the same as the forwarder's distance, and a value of 0 when the distances differ by r, which corresponds to maximum packet progress. However, if the location information is incorrect, the assumed distances may differ by a greater value than r, which leads to negative values. These result in the value 0 due to the maximum function, but when several nodes all calculate the value 0, they will forward the packet simultaneously, which causes at least packet duplications, if not collisions in the worst case.

The solution for this problem is to stretch the assumed transmission range, since we have shown in Section 5 that the estimated distances between the nodes grow with increasing position deviation. Hence, we replace the quotient r with the estimated value for r, which is $t = E(G)$ assuming $d = r$, as listed in Table 1 for $d = 40$. Thus, the new timer function is

$$t(c) = \max\left(0, Max\_Delay \cdot \left(1 - \frac{d-c}{t}\right)\right).$$

In Figure 9, the delivery ratio is shown with both the original and the modified timer function when using the Reuleaux triangle as forwarding area. For higher position deviations, the delivery ratio is significantly better if the new timer function is used.

Figure 10 compares the timer functions in terms of the number of packets sent altogether. Especially at large range errors, the new timer function results in great reduction of packets, as expected. This shows that the improved timer function helps to save energy while at the same time increasing the delivery ratio.

7. Conclusion

Geographic routing algorithms for sensor networks cannot rely on exact position information. Most algorithms, however, assume exact positions. The experiments presented in this paper show the performance of two geographic routing algorithms in case of location errors, GPSR and BGR. The main results are that the packet delivery ratio of GPSR decreases significantly at high location errors, according to

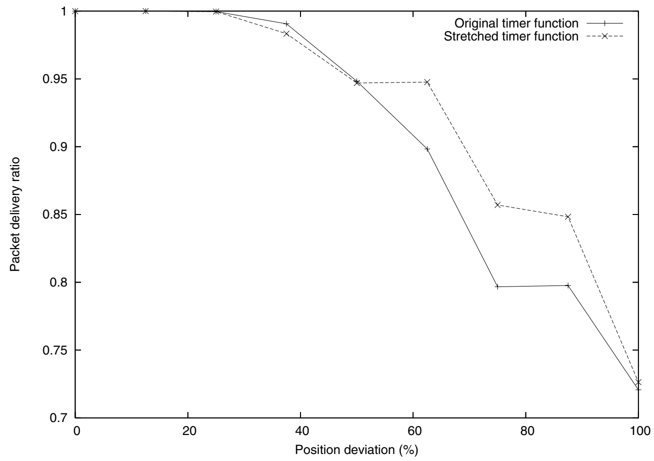

Figure 9. Packet delivery ratio of BGR with Reuleaux triangle using different timer functions

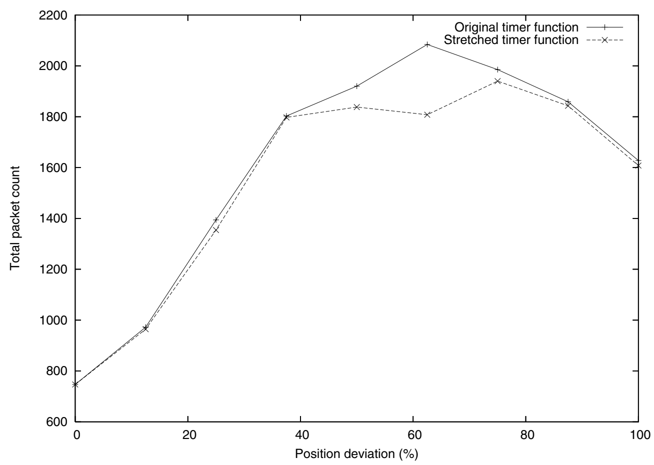

Figure 10. Number of packets sent totally in BGR with Reuleaux triangle using different timer functions

related studies, while BGR has less problems. Likewise, the path lengths are lower in BGR compared to GPSR, unless using the circle as forwarding area.

Several fixes for GPSR have been presented to improve the success rate. Results show that probabilistic approaches can slightly increase the delivery ratio; information about 2-hop neighbors, however, leads to much better results.

Furthermore, an improvement of BGR has been introduced in order to lower the number of nodes that calculate a forwarding delay of zero and thus forward the packet simultaneously. This improves the performance of BGR again, as both the success rate increases and the number of packets drops.

Future experiments will account for other problems that arise in wireless communication and are habitually idealized in simulation experiments. These are for example:

- varying transmission radii,
- anisotropic communication ranges,
- transient communication failures,
- unidirectional links,
- lost links,
- presence of obstacles.

Most of these problems are not considered in simulations of GPSR and BGR, but do occur in real sensor network deployments [4]. Especially GPSR fails frequently in these situations, as it relies on correct planarization. Future experiments will reveal the performance of geographic routing with more realistic parameters.

References

[1] B. 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*, Boston, Massachusetts, Aug. 2000, pp. 243–254.

[2] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger, “Geometric Ad-Hoc Routing: Of Theory and Practice,” in *Proc. 22nd ACM Symposium on Principles of Distributed Computing (PODC)*, Boston, Massachusetts, July 2003, pp. 63–72.

[3] M. Witt and V. Turau, “BGR: Blind Geographic Routing for Sensor Networks,” in *Proceedings of the Third International Workshop on Intelligent Solutions in Embedded Systems*, Hamburg, Germany, May 2005.

[4] V. Turau, M. Witt, and C. Weyer, “Analysis of a Real Multi-hop Sensor Network Deployment: The Heathland Experiment,” in *Proc. Third International Conference on Networked Sensing Systems (INSS 2006)*, Chicago, IL, June 2006.

[5] S. McCanne and S. Floyd, “ns Network Simulator,” `http://www.isi.edu/nsnam/ns/`.

[6] T. He, C. Huang, B. Blum, J. Stankovic, and T. Abdelzaher, “Range-Free Localization Schemes for Large Scale Sensor Networks,” in *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking*, San Diego, California, Sept. 2003, pp. 81–95.

[7] 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.

[8] K. Seada, A. Helmy, and R. Govindan, “On the Effect of Localization Errors on Geographic Face Routing in Sensor Networks,” in *Proceedings of the Third International Symposium on Information Processing in Sensor Networks*, Berkeley, California, Apr. 2004, pp. 71–80.

[9] R. Shah, A. Wolisz, and J. Rabaey, “On the performance of geographical routing in the presence of localization errors,” in *Proceedings of IEEE International Conference on Communications (ICC 2005)*, Seoul, Korea, May 2005, pp. 2979–2985.

[10] L. Lazos and R. Poovendran, “SeRLoc: Secure Range-Independent Localization for Wireless Sensor Networks,” in *Proceedings of the 2004 ACM Workshop on Wireless Security*, Philadelphia, PA, Oct. 2004, pp. 21–30.

[11] Y.-J. Kim, R. Govindan, B. Karp, and S. Shenker, “Geographic Routing Made Practical,” in *Proceedings USENIX Symposium on Networked Systems Design and Implementation*, Boston, Massachusetts, May 2005.