BGR: Blind Geographic Routing for Sensor Networks

Matthias Witt[1] and Volker Turau[1]

[1]Department of Telematics,
Hamburg University of Technology, Hamburg, Germany
{matthias.witt,turau}@tuhh.de

Abstract — *This paper introduces BGR, a new beacon-less geographic routing algorithm for wireless sensor networks. Data packets are forwarded toward the destination, and nodes which hear the packet compete for becoming the next hop. A recovery strategy is provided for the case of empty forwarding areas. The main innovation is a strategy to avoid simultaneous forwarding situations, which would otherwise cause packet failures. It is confirmed by simulation that BGR sends very few packets and is reliable in fields with sufficiently high node density. Furthermore, BGR is compared with similar algorithms after developing a general framework for beacon-less algorithms with an integrative terminology. Different kinds of forwarding areas are compared.*

1 Introduction

Wireless sensor networks are of increasing interest in different application areas. The deployment of large-scale sensor networks will become possible in the near future. Since the transmission range of small sensor devices is limited and energy resources are sparse, new routing algorithms have to be developed that send preferably few packets.

There are basically two kinds of routing algorithms for ad-hoc networks: geographic and non-geographic algorithms. The latter can further be divided into table-driven (proactive) and on-demand (reactive) routing algorithms, whereat both of these let the nodes keep information about routes. In contrast to that, geographic routing can be stateless, because the next hop is chosen using the geographic location of the destination, which is stored in the packet header. To accomplish this, nodes must know their geographic position, for instance via GPS or explicitly programmed. An overview of geographic routing algorithms can be found in [1].

A well-known geographic routing algorithm is GPSR [2]. In GPSR, each node maintains a neighbor table which is updated by periodically sending beacon messages. However, these beacon messages represent the largest amount of packets that are sent, which is contradictory to the requirement that few packets should be sent. Thus, it is essential to develop a routing algorithm without the use of beacons. Basic beacon-less algorithms have neither routing tables nor neighbor tables; they are completely stateless.

This paper introduces *Blind Geographic Routing* (BGR), a novel beacon-less routing algorithm that is very reliable if the node density is sufficiently high. The algorithm is "blind" in the sense that packets are forwarded without knowing which node will be the next hop. The nodes which receive the packet set a timer depending on parameters like distance to the destination; the node whose timer expires first will forward the packet, whereas the other nodes cancel their timers.

The remainder of this paper is organized as follows. Section 2 provides a general framework and an overview of existing beacon-less routing algorithms. In Section 3, BGR is described in detail. Simulation results are discussed in Section 4. Finally, Section 5 summarizes the contribution of this paper and gives an outlook on future work.

2 Beacon-Less Routing Algorithms

In the following, a general framework for beacon-less routing algorithms is provided and an integrative terminology is established.

2.1 General Framework

The concept of a *forwarding area* is fundamental for beacon-less geographic routing. It consists of a set of sensor nodes that can mutually communicate with each other. Forwarding areas are described using geographical constraints, such that a node can determine whether it belongs to a forwarding area or not using only its location. The basic routing algorithm can be described as follows:

1. A forwarder f selects a forwarding area which includes f.
2. Node f broadcasts the packet including a description of the forwarding area.
3. Nodes which hear the packet and have a timer running for it cancel the timer.
4. All nodes in the forwarding area receive the packet (nodes outside the forwarding area discard the packet).
5. All nodes in the forwarding area start a timer, if the destination has not been reached.
6. The first node where the timer expires continues with 1.

A specific algorithm must define

- how the forwarding area is formed and how to guarantee that it is not empty and
- the duration of the timers.

All nodes within the forwarding area must hear each other, so that they can cancel their timers when the next hop is forwarding the packet. So the forwarding area cannot consist of all nodes that are nearer to the destination than the last hop. This implies that the forwarding area is included in a circle of diameter r (r is the transmission range, which is supposed to be fixed and known). Apart from this, the forwarding area should also meet the following conditions:

(A) It should be sufficiently large (and contain at least one node),

(B) it should contain many nodes that make big progress toward the target (compared to the forwarder), and

(C) it should contain few nodes close to the forwarder.

There are three reasonable possibilities for the forwarding area: a $60°$ sector, a circle with diameter r (transmission range), and a Reuleaux triangle with a width of r.[1] As the circle has the largest surface area, it meets condition (A) best; the Reuleaux triangle is the best choice for (B), and the sector fulfills (C) best.

The timer function t controls the next hop. The selection may be based on different criteria, like distance to the destination, remaining energy, load, previous usage, fault rate, random values, or some combination of these. The criteria for the timer function are:

(D) The timer should select a good forwarder (e. g., based on the progress).

(E) The timer should differentiate the length of times at the different nodes in order to avoid simultaneous or almost simultaneous selection.

When using the progress toward the destination in the timer function, different values can be comprised (see Figure 1):

- c (the distance between candidate node and destination)
- d (the distance between forwarder and destination)
- p (the distance between forwarder and the projection of the candidate node's position on the straight line from forwarder to destination)
- r (the transmission range)

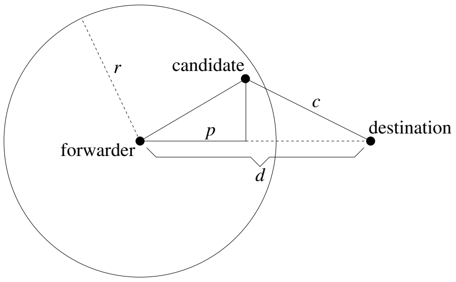

Figure 1: Values for computing the time interval

To allow for a recovery strategy if the forwarding area turns to be empty, there has to be a maximum waiting time Max_Delay. The choice of Max_Delay is crucial for the packet delivery time. There is a trade-off between delivery time and packet collisions: The forwarding latency depends on Max_Delay, but if it is set too low, packet collisions will arise frequently, because many nodes will try to forward the same packet almost simultaneously. Max_Delay could be set dependent on the node density.

There are three geographic routing algorithms for sensor networks that are not based on beacon messages, namely BLR [3], CBF [4], and IGF [5]. They have been developed independently from each other and use different forwarding areas, timer functions, and recovery strategies. However, problems that arise when the timers of two nodes expire almost simultaneously are considered by neither of them. In the following, these algorithms are briefly reviewed.

[1] The Reuleaux triangle is the intersection of the three circles with radius r and center at the corners of an equilateral triangle with side length r.

2.2 BLR: Beacon-Less Routing

BLR proposes all three above-mentioned forwarding areas. For the timer t, three different functions are proposed, which have influence on the latency in different ways. One of these functions is:

$$t = Max\_Delay \cdot \left(\frac{r-p}{r}\right) \tag{1}$$

The denoted function causes a linear decrease of the computed delay toward the destination. An exponential decrease could also be used. This penalizes nodes close to the forwarder, as their delays are significantly longer, which could bring about faster delivery rates; however, nodes far away from the forwarder compute similar delay intervals, which causes more simultaneous forwarding situations.

Due to the fact that a node hears which node has forwarded the packet, subsequent packets are directly sent to this node via unicast. Admittedly, to accomplish this, a routing table has to be stored, which has a negative impact on the scalability. To account for new nodes, a beaconing interval is introduced; when it expires, packets are sent via broadcast again. Optionally, there is a promiscuous mode, in which nodes within the forwarding area process unicast packets that are not sent to them and start a timer for the case that the receiver of the packet is no longer available.

If the forwarder does not hear another node forwarding the packet within the time interval Max_Delay, the forwarding area is considered to be empty, and the backup mode is triggered: The node broadcasts a request, and all neighbor nodes reply and send their positions. If one of the neighbor nodes is closer to the destination, it is chosen as the next hop; otherwise the forwarding continues according to the right-hand-rule on a planarized graph, like in GPSR.

2.3 CBF: Contention-Based Forwarding

CBF operates in the greedy mode exclusively and does not have a strategy for the case that it fails. The forwarding process consists of two phases: *contention* and *suppression*. In the former phase, a node is determined as the next hop; in the latter, it suppresses the other candidates from also forwarding the packet.

The contention phase works timer-based using the following timer function:

$$t = Max\_Delay \cdot \left(1 - \frac{d-c}{r}\right) \tag{2}$$

The timer is only started when $d > c$. Note that the computed time is non-negative and not greater than Max_Delay, because $0 \leq d - c \leq r$.

For the second phase, the suppression phase, two alternatives are proposed: *area-based suppression* and *active selection*. Area-based suppression operates with a forwarding area. The areas that are discussed are the circle and the Reuleaux triangle; the latter is favored due to condition (B). If the Reuleaux triangle turns to be empty, the forwarder sends up to two more broadcasts with different forwarding areas; these are the areas left and right to the Reuleaux triangle with positive progress (i. e., where nodes are closer to the destination than the forwarder). If these areas are also empty, a recovery strategy is necessary, but details on it are left open in the paper.

In contrast to that, active selection works like follows: The forwarder broadcasts a *Request To Forward* (RTF), whereupon all nodes with positive progress start a timer and send back a *Clear To Forward* (CTF) when the timer expires. The forwarder chooses the next hop among all nodes from which it has received a CTF. It sends the data packet via unicast to this node. Other nodes cancel their timers when hearing a CTF. Through active selection, all forms of packet duplication are excluded. The cost is a packet overhead of the factor three.

2.4 IGF: Implicit Geographic Forwarding

IGF is a combined routing/MAC protocol that shifts the timer-based selection of the next hop to the MAC layer. The MAC protocol IEEE 802.11 is modified in the sense that instead of an RTS an *Open Request To Send* (ORTS) is broadcasted. Nodes in the forwarding area start a timer and send a CTS on expiration, if no other node has sent a CTS. The remaining process is identical to the original 802.11 protocol: The forwarder unicasts the data packet to the selected hop, which acknowledges with an ACK packet.

As forwarding area, the sector is used. The node density is assumed to be high enough so that the forwarding area contains at least one node. An optional enhancement is the movement of the forwarding area, if the forwarder has not received any CTS.

The scheduled time is computed by an advanced function using three parameters:

- distance to the destination
- available energy of the node
- additional random delay

Nodes with little remaining energy resources increase the time interval, so that nodes with more energy are favored.

3 The BGR Routing Algorithm

In BGR, the forwarding area is an implementation-dependent choice. If no node forwards the packet, the forwarding area is assumed to be empty, and up to two more retries are started with different forwarding areas. If these also fail, the packet is dropped because no route was found to the destination. – The main innovation of BGR is a technique called *Avoidance of Simultaneous Forwarding* (ASF), which solves problems that arise when two nodes forward the packet almost simultaneously. The whole process is described in detail in the following.

3.1 The Basic Algorithm

The forwarder initiates the forwarding process by broadcasting the packet in the mode BLINDBROADCAST. It then starts a *recovery timer*, which expires after Max_Delay seconds plus a short additional delay[2].

If a node that receives this packet is located within the forwarding area and has not yet received the packet before[3], it becomes a *candidate node* and starts a *contention timer* for

[2] for the case that a node forwards the packet very shortly before Max_Delay

[3] To check this, the packet ID is compared with the last n IDs, which have to be stored. When every node has a unique ID, unique packet IDs consisting of node ID and packet sequence number can be generated.

this packet. For the measurements evaluated in the next section, the timer function (2) is used.

When the contention timer expires, the candidate node becomes the next hop and initiates a new forwarding step by sending the packet in the mode BLINDBROADCAST again. The other candidates hear this packet and cancel their contention timers. The original forwarder also hears the packet and cancels its recovery timer. Now the same procedure takes place with the new forwarder setting a recovery timer etc.

All nodes which receive a packet first check if they are a suitable destination node. This is done by comparing the destination position with their own position. If the difference lies within a certain limit of tolerance, the node passes the data packet to the application. At the same time, it broadcasts a simple CANCEL packet with the consumed packet's ID. All nodes cancel their timers when receiving this packet.

3.2 Recovery

When the recovery timer of the forwarder expires, no node has forwarded the packet, so there is presumably no node within the forwarding area. The forwarder now initiates a recovery strategy. The forwarding area is turned by $60°$ to the left or to the right by broadcasting the packet in the mode FW_AREA_LEFT or FW_AREA_RIGHT[4]. A new recovery timer is started; when it expires, the third option for the forwarding area is given a try. If this also fails, the packet is dropped.

Note that using the shifted forwarding areas, the value resulting from (2) can be greater than Max_Delay, because the candidate nodes can be farther away from the destination than the forwarder. So the recovery timer must be set to a time greater than Max_Delay. The time is computed with (2), where c is set to the maximum distance between the destination and any node within the shifted forwarding area (see Figure 2). The resulting values are listed in Table 1. The exact value for the Reuleaux triangle is rather complicated to compute; however, the upper boundary is sufficient for practical usage.

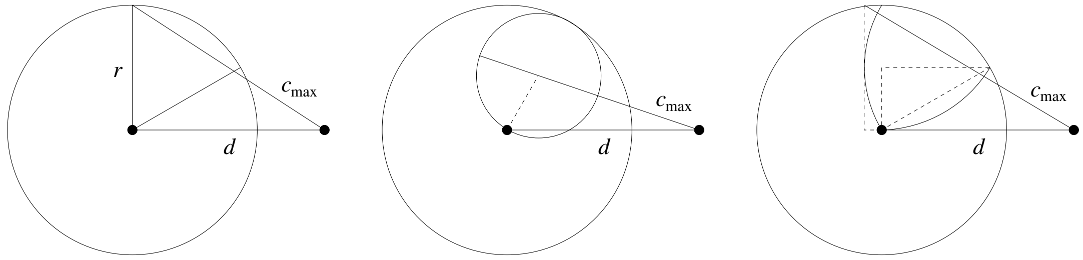

Figure 2: Maximum distance between the destination and any node within the shifted sector, circle, and Reuleaux triangle (upper boundary only)

If the forwarder does not hear the forwarding of the packet by the next hop due to a transmission failure, the recovery mode will be triggered although the packet has been forwarded; however, this does not lead to a failure of the algorithm, because the packet will be transmitted at two different places in the network simultaneously. The second packet will be dropped, because nodes will simply ignore a packet which they already received.

[4] This is chosen randomly in order to avoid routing around empty areas always in the same direction [4].

Forwarding Area	Sector	Circle	Reuleaux Triangle (upper boundary)
$c_{\max}$	$\sqrt{d^2 + r^2}$	$\frac{r}{2} + \sqrt{d^2 + \frac{r^2}{4} - dr\cos 60^\circ}$	$\sqrt{\left(d + \left(1 - \frac{\sqrt{3}}{2}\right) r\right)^2 + r^2}$

Table 1: Maximum distance between the destination and any candidate node within the shifted forwarding areas

3.3 ASF

There is another problem that arises when two nodes within the forwarding area with similar distances to the destination forward the packet almost simultaneously. The forwarder cancels its recovery timer, because it hears the forwarding of the packet. Nodes within the intersection of the two new forwarding areas start contention timers when they receive the packet from the first node; however, they cancel them very shortly afterwards, because they assume that the second node (from which they receive the same packet again) has forwarded it already. The same applies for the recovery timers of the two forwarders, so that now all timers have been canceled and the process has been stopped without the packet reaching the destination.

To solve this problem, the number of hops is stored in the packet header. When a node hears the forwarding of a packet for which it has a timer running, it compares the hop count of the stored packet with the hop count of the received one. If they are equal, the timer is not canceled. This technique is called *Avoidance of Simultaneous Forwarding* (ASF).

4 Simulation Results

To evaluate BGR in a realistic environment, the algorithm has been implemented in the network simulator *ns-2* [6], which allows detailed simulation of mobile nodes with an IEEE 802.11 MAC and physical layer.

For the simulation runs, a square topology with 100 randomly placed nodes was used. To decrease the influence of one special topology on the results, each experiment was repeated 20 times with different topologies; for the evaluation, the mean values of these 20 runs were taken. To use all possible ways in these topologies, packets are sent from every node to every other node by sending one packet every 10 seconds. So, a total of 9900 data packets are generated.

The node density has been varied between 150 m^2 and 500 m^2 per node. This has been done by changing the area stepwise from 122×122 m^2 to 224×224 m^2, which corresponds approximately to 6.6 to 2 nodes per 1000 m^2. The most important simulation parameters are summarized in Table 2.

Simulation runs were done for all three forwarding areas, each with and without recovery, and finally with recovery and ASF. Additionally, GPSR was simulated with the same simulation parameters. To do so, the original implementation of GPSR [2] has been ported to the current version of *ns-2* and provided with some functionality to compute the

Number of Nodes	100
Transmission Range	40 m
Simulation Time	100,100 s
Max_Delay	0.5 s
Additional Delay (for Recovery Timer)	0.05 s

Table 2: Simulation Parameters

necessary statistical data. For the beaconing interval, a relatively broad-minded range of 10 seconds was chosen. To give all nodes enough time to detect their neighbors, the first data packet is sent after 100 seconds.

4.1 Packet Delivery Ratio

The average packet delivery ratio for the experiments with the Reuleaux triangle as forwarding area is shown in Figure 3. The basic mode fails to deliver up to 40 % of the packets in areas with little node coverage. Using recovery, the success rate can drastically be increased. This is the delivery rate that BLR achieves. When also using ASF, the delivery ratio goes up to 100 % when every node has enough neighbors.

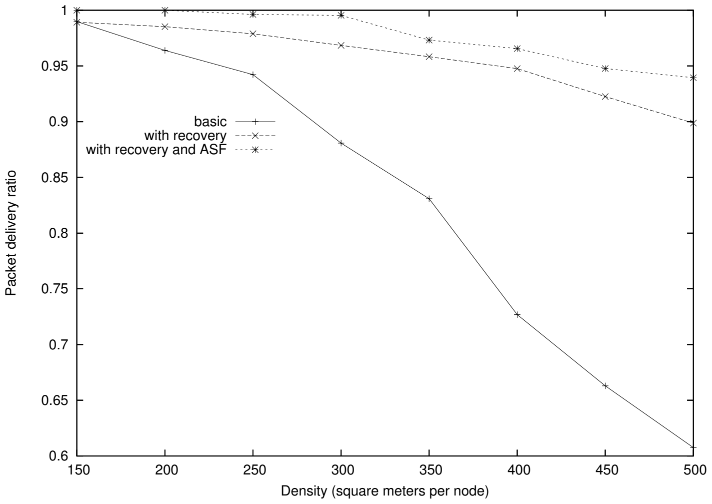

Figure 3: Average packet delivery ratio (Reuleaux triangle)

Figure 4 shows the average packet delivery ratio for all forwarding areas and GPSR. Here, recovery and ASF are always enabled. GPSR always has a delivery ratio near 100 %, because its perimeter mode guarantees the delivery in an ideal environment where the neighborhood graph is connected. Failures are solely due to isolated nodes and collisions with beacon packets.

However, in fields with high node density, BGR is also nearly failure-free. When the density decreases, the different forwarding areas swap the role of being the best one: In middle-density fields, the sector is the best choice; in fields with very low density, only the Reuleaux triangle provides acceptable results. This is because the surface area of the

Reuleaux triangle is larger than the sector's. The circle's is even larger, but the circle does not meet condition (B) stated in Section 2.1 well, which apparently turns to be fatal in low-density fields.

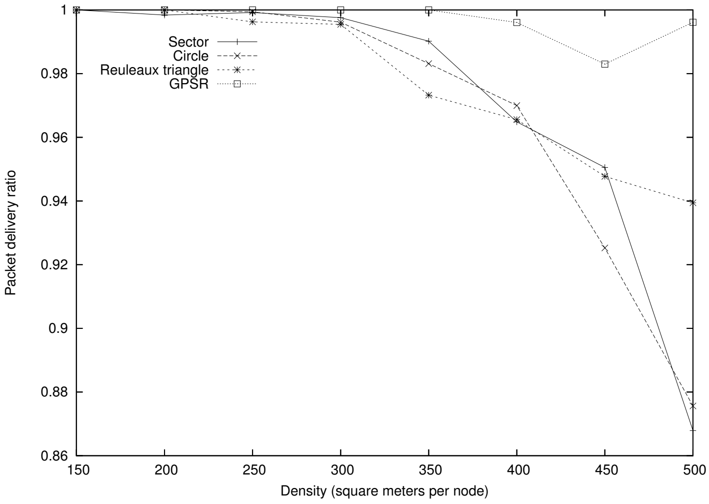

Figure 4: Average packet delivery ratio

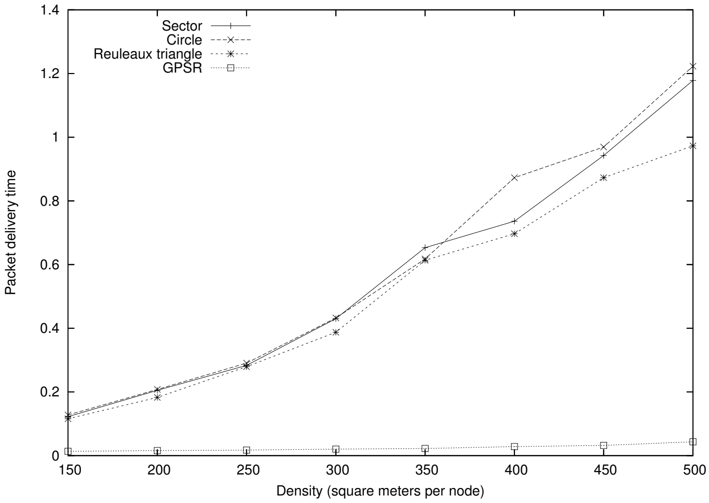

Figure 5: Average packet delivery time

4.2 Packet Delivery Time

The average delivery time of data packets is depicted in Figure 5. Only the successfully delivered packets are included. Not surprisingly, GPSR has very short delivery times because it is not based on timers, but always forwards the packets immediately. The results for the three forwarding areas in BGR are comparable to each other. The Reuleaux triangle is the fastest option, because the recovery mode has to be triggered less than using the other forwarding areas. The circle has most problems in fields with low node density.

It should be noted that the delivery time can significantly be improved by setting down the values for Max_Delay and the additional delay in Table 2. However, when setting these values too low, packet collisions and ASF situations will arise frequently. But the values chosen here can assuredly be decreased without too many negative ramifications.

4.3 Hop Count

Figure 6 shows the average hop count. The optimum has been computed using Floyd's algorithm. Once again, the circle yields the worst results. The sector curtly has the smallest hop count. This is a little surprising, since it has the smallest surface area of all forwarding areas and the last subsection shows that the Reuleaux triangle delivers the packets faster. The reason is the following: Consider the case that the sector is empty, but the Reuleaux triangle is not. Using the latter as forwarding area will lead to a smaller latency, since the sector must trigger the recovery mode. However, using the moved sector in the recovery mode, a node closer to the destination may be found that is not within the Reuleaux triangle. So the next hop will be closer to the destination than the hop chosen by the Reuleaux triangle, although taking more time to find it.

In fields with low node density, GPSR has a significantly higher hop count than BGR. This is due to its perimeter mode, which is triggered when no neighbor node is closer to the destination than the forwarder. In many of these cases, BGR fails to find a route to the destination, so the higher hop count of GPSR is a result of its ability to always find a route if it exists. In the perimeter mode, GPSR forwards the packet on a planarized subgraph of the neighborhood graph, so it can be the case that some links on the optimal path have been discarded.

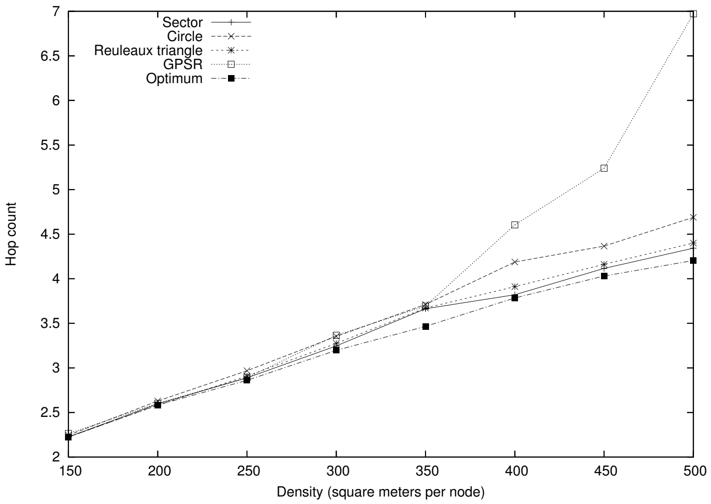

Figure 6: Average hop count

4.4 Total Packet Count

For the total amount of packets that are sent by any node, no graph is provided here, because the values of BGR and GPSR differ so immensely that they cannot be visualized accurately in one coordinate system. In fact, the values in BGR range from about 33,000 to 70,000, whereas in GPSR they go from about 1,012,000 to 1,039,000. This is due to the fact that GPSR sends nearly a million beacon packets during the whole simulation time. This is even a broad-minded value, because the beacon interval was set to 10 seconds, which may be too high in many cases with frequent node failures or movements. In the original implementation of GPSR, the beacon interval is set to 0.5 s, which would increase the packet count again by the factor 20.

This significantly lower packet count of BGR is its great advantage: Much more energy can be saved than in GPSR, because the amount of packets that are actually sent is decreased to a minimum. It cannot be done effectively with fewer packets, and in fields with a sufficiently high node density, the delivery rate is close to 100 %.

5 Conclusion and Outlook

The contribution of this paper is BGR, a geographic routing algorithm that is reliable in fields with high node density and sends significantly fewer packets than most other approaches. BGR is based on ideas from other beacon-less routing algorithms; however, unlike these, its behavior is studied by extensive simulation experiments, using not only the basic mode, but also implementing a recovery strategy. For solving problems that arise frequently when nodes try to forward packets almost simultaneously, a novel strategy called ASF has been introduced.

Future work will focus on the nodes remembering delivery failures and through this finding better routes. Apart from that, topology information will be exchanged between the nodes in order to avoid routing into dead ends. For the timer function, the remaining energy and other parameters will be taken into account.

The advantages of the sector and the Reuleaux triangle (see Subsection 2.2) can be combined to a routing method with an adaptive forwarding area. The advantage of the sector is that it meets condition (C) better; the Reuleaux triangle, on the other hand, fulfills (B) better. If the forwarder is close to the last hop, most of the forwarding area is known to be empty (otherwise a node farther away from the last hop would have become the next hop). So, the Reuleaux triangle is the better choice in this case. If the forwarder is far away from the last hop, the sector is better on average.

References

[1] I. Stojmenovic. Position-Based Routing in Ad Hoc Networks. *IEEE Comm. Magazine*, 40(7):128–134, July 2002.

[2] B. Karp and H. T. Kung. GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. In *Proceedings Mobicom 2000*, pages 243–254, Boston, Massachusetts, August 2000.

[3] M. Heissenbüttel, T. Braun, T. Bernoulli, and M. Wälchli. BLR: Beacon-Less Routing Algorithm for Mobile Ad-Hoc Networks. *Elsevier's Computer Comm. Journal*, 27(11):1076–1086, July 2004.

[4] H. Füßler, J. Widmer, M. Mauve, and H. Hartenstein. A Novel Forwarding Paradigm for Position-Based Routing (with Implicit Addressing). In *IEEE Computer Comm. Workshop (CCW 2003)*, pages 194–200, Dana Point, California, 2003.

[5] B. Blum, T. He, S. Son, and J. Stankovic. IGF: A state-free robust communication protocol for wireless sensor networks. Technical Report CS-2003-11, Department of Computer Science, University of Virginia, 2003.

[6] S. McCanne and S. Floyd. ns Network Simulator. `http://www.isi.edu/nsnam/ns/`.