Beacon-less geographic routing made practical: challenges, design guidelines, and protocols

Article *in* IEEE Communications Magazine · September 2009

DOI: 10.1109/MCOM.2009.5181897 · Source: IEEE Xplore

Juan A. Sánchez
University of Murcia

Pedro M. Ruiz
University of Murcia

Rafael Marin-Perez
University of Murcia

Beacon-less Geographic Routing Made Practical: Challenges, Design Guidelines and Protocols

Juan A. Sanchez, Pedro M. Ruiz and Rafael Marin-Perez
Dept. of Information and Communications Engineering
Fac. Informatica, Univ. of Murcia
Campus de Espinardo s/n
E-30100, Espinardo, Murcia
Email: {jlaguna, pedrom, rafael81}@um.es

***Abstract*—Geographic routing has emerged as one of the most efficient and scalable routing solutions for wireless sensor networks (WSN). In traditional geographic routing protocols, each node exchanges periodic 1-hop beacons to know the position of its neighbors. Recent studies have proven that these beacons may create severe problems in real deployments due to the highly dynamic and error-prone nature of wireless links. To avoid those problems, new variants of geographic routing protocols which do not need beacons are gaining momentum. In this paper we overview some of the latest proposals in the field of beacon-less geographic routing and introduce the main design challenges and alternatives. Additionally, using a real WSN deployment we perform an empirical study to assess the performance of some beacon-based and beacon-less routing protocols.**

I. Introduction and motivation

A wireless sensor network (WSN) consists of a set of sensing devices equipped with wireless interfaces. These devices, called sensor nodes, are able to monitor their environment and communicate with neighboring nodes. In most WSN applications, sensor nodes collaborate in a distributed way to perform some tasks such as structure monitoring, tracking, data gathering, and so on. To achieve their goal, sensor nodes can use their neighbors as data relays to communicate with other sensor nodes which are not within their radio range. In addition, given that most of the energy expenditures of sensor nodes are due to data communications, the design of highly efficient communication protocols has become one of the most active research areas within the WSN community.

In particular, geographic routing has emerged as one of the most efficient and scalable routing solutions for WSNs [1]. The key advantage of geographic routing solutions is that they do not need to know the topology of the network. Each sensor node can take its own routing decisions by just knowing its position, the position of the destination and the position of its neighbors. With that information, the current node forwards the data packet to one of its neighbors which is closer to the destination than itself according to some criteria (e.g. advance towards the destination, energy consumption, etc.). If the current node has no neighbor being closer to the destination than itself, a recovery strategy usually called *perimeter mode* or *face routing* is used [1]. All these routing decisions are fully local which means that the message overhead and processing cost of these protocols scale with respect to the number of nodes in the network.

The position of the destination is usually determined by the source node using some location service. However, for some applications the location service may not be required if the position of the destinations is known in advance. The position is embedded in the header of the data message being routed so that next hops can route the message without further information. Nodes periodically exchange short messages called *beacons* to inform all their neighbors about their identifier and position. Beacons are not flooded across the whole network, but the overhead can still be excessive and nonproductive. The reason is that every node must regularly generate a beacon even if it is not participating in any data exchange. Additionally, this issue is particularly harmful because WSNs are very limited in terms of bandwidth and energy of the nodes. As a matter of fact, beacons can also degrade the overall network performance by producing additional collisions and interferences with data packets being routed [2].

To avoid the issues produced by *beacons*, recent beacon-less geographic routing solutions have been designed to operate without beacons. Instead of sending periodic beacon messages, nodes acquire the information about positions of their neighbors reactively. That is, only when a node has a data packet to route. To do that, it asks for neighbor's positions by broadcasting a query message. Thus, nodes taking part in a routing task are the only ones spending resources. Existing protocols usually embed this discovery process into MAC layer frames, or even in next hop selection messages at the network layer. Thus, these protocols offer a better performance than traditional geographic routing solutions while still preserving their good properties (i.e. localized operation and ultimate scalability).

While beacon-less solutions are very promising, to date, most performance results available are based on simulations with very idealistic conditions and quite questionable assumptions. Authors generally assume perfect and equal transmission radii for all the nodes of the network, without interferences, losses, etc. However, the behavior of real links is quite different from the widely used Unit Disk Graph (UDG) as some studies have confirmed [3]. In addition, there is very little insight to designing efficient and effective beacon-less protocols in real WSN scenarios.

In this paper we describe the operation of the most representative beacon-less geographic routing protocols including our own proposal called Beacon-less On-demand Strategy for Sensor networks [4] (BOSS). We also present an empirical performance evaluation of beacon-based and beacon-less solutions using a real WSN testbed as well as simulations. The testbed consists of 35 Tmote-sky nodes deployed within an area of about 3000 square meters in the first floor of our Computer Science building at the University of Murcia. Our experiments show that the variability of wireless links can provoke important performance degradations if these issues are not carefully taken into account during the design of the protocols. Our results also show that, if carefully designed, beacon-less protocols are able to outperform beacon-based solutions.

The remainder of this paper is organized as follows. Section II describes and analyzes existing beacon-less geographic routing solutions. Section III studies the basic technical challenges and design recommendations for beacon-less protocols in real wireless deployments. Section IV describes the testbed, simulation setup and the results of the performance evaluation. Finally, section V summarizes our findings and describes open issues and research opportunities for further research.

II. Beacon-less geographic routing Protocols

Beacon-less geographic routing protocols have the same two modes of operation as traditional geographic routing solutions: *greedy* and *perimeter*. *Greedy mode* is the normal mode of operation. It consists of selecting as next hop one neighbor located closer to the destination than the current one. There are different schemes which differ on the strategy to select the next hop (e.g. the one closest to the destination). The routing task may eventually reach a node which has no neighbors located closer to the destination than itself. In that situation the routing protocol reaches a so-called *local minimum*. Any of the variants of *Perimeter mode* can be used to escape from that minimum by surrounding the void area until a node where greedy mode can be resumed is found. More details on existing schemes can be found in [1].

Beacon-less routing protocols are based on four different mechanisms:

Initial broadcast to all neighbors. The node currently holding the message initiates the process of selecting its next relay by broadcasting a message. Some protocols use special control messages for this purpose while others resort to broadcasting the data packet itself.

Definition of contention timers and forwarding area. Contention timers indicate neighbors when to answer the initial broadcast. In general, contention timers are defined so that nodes located closer to the destination answer first. In addition, to reduce contention, nodes cancel their timers after overhearing other responses. Finally, some protocols limit which neighbors answer the initial broacast to those located in the so-called forwarding areas. The goal is to guarantee that all the responses are received by all the candidates. This prevents forwarding inconsistencies across possible next hops.

Selection of next hop. In some protocols the next hop is selected by the sender based on the answers received by neighbors. In other cases, neighbors self-elect themselves in a distributed way and resend the data packet. Some protocols incorporate active acknowledgment using special control messages. However, passive acknowledgment is also used so that when the sender overhears the forwarding of the data packet it interprets that as an ACK from the next hop.

Perimeter operation. When no neighbor provides progress towards the destination, perimeter routing needs to be used. Traditional perimeter routing requires the sender to know all its neighbor so that it can construct a planar subgraph. So, some protocols incorporate special conditions in their contention timers to make all neighbors report their positions in that case. There exist more efficient beacon-less proposals that do not require all neighbors to answer (i.e. Kalosha et al. [5]). However, in this paper we focus our work in the greedy operation. Hence, perimeter details are not further discussed.

Below, we explain the operation of the main beacon-less routing protocols in the literature with special emphasis on how they address each of the mechanisms highlighted before.

A. Implicit Geographic Forwarding (IGF)

Implicit Geographic Forwarding (IGF [6]) is one of the first beacon-less geographic routing protocols proposed in the literature. In IGF the node currently holding the message broadcasts a Request to Send (RTS) frame and waits for the first Clear to Send (CTS) response. Each neighbor receiving the RTS frame evaluates its own suitability as next hop. The neighbor providing the largest advance towards the destination is preferred and should answer first. Finally, at the network layer, the forwarding node transmits the data message and the selected neighbor confirms the reception by answering with an Acknowledgment message (ACK).

IGF includes two optimizations to reduce the number of responses and collisions: a mechanism to avoid simultaneous responses from neighbors based on timers, and a scheme to cancel unnecessary responses when other neighbor's responses are overheard.

Upon receiving a RTS message, each neighbor sets a timer to wait before answering with a CTS message. The timer value depends on the reduction in distance towards the destination provided by the node plus a random component. Thus, neighbors located closer to the destination answer first. Besides, neighbors overhearing an earlier CTS from another neighbor cancel their own timers.

IGF defines a forwarding area so that all nodes within that area are separated by a distance lower than the theoretical radio range. That is, in theory, all nodes inside it can hear one another (see Fig. 1(a)). Only those nodes located inside the forwarding area can take part in the selection process. This is defined that way to allow neighbors to overhear other neighbors' answers. However, in practice, radio propagation can make nodes within the forwarding area not to overhear some answers. Also, as a side effect, the use of a forwarding area may neglect some neighbors providing a higher progress because of being outside that area.

B. Geographic Random Forwarding (GeRaF)

Geographic Random Forwarding (GeRaF [7]) is also a Routing/MAC beacon-less routing protocol. GeRaF's main contribution is a collision avoidance MAC scheme. In GeRaF next hop candidates are those nodes whose position is closer to the destination than the node currently holding the message. As Fig 1(c) depicts, that area is logically divided in N_p regions $A_1 \ldots A_{N_p}$ such that all points in A_i are closer to the destination than all points in A_j for $j > i,\ i = 1 \ldots N_p - 1$. The collision avoidance MAC scheme is based on the assumption that nodes can have two radios. One is used for the traditional RTS/CTS handshake and the other is used just to transmit busy tones indicating that the first radio is being used to transmit control packets.

The contention scheme works as follows. The node currently holding the message transmits a RTS frame and starts waiting for responses for a period of time called CTS slot. All nodes in the first region answer with a CTS frame and keep listening for a data packet from the transmitter of the RTS. If the transmitter successfully received a CTS message it issues a data packet containing the payload and a header indicating the identifier of the neighbor selected as next hop. That is, the one whose CTS was received first. If the transmitter does not correctly receive a CTS frame within the CTS slot then, the data packet issued indicates a collision and all the nodes in the same region decide whether to send another CTS or not with probability 0.5. If no node answers during the CTS slot, the transmitter indicates in the message that nodes in the next region must answer because there are no neighbors in the first one. In the worst case this process is repeated N_p times, one for each region.

Besides, when a node does not have any neighbor providing advance towards the destination, GeRaF's authors suggest to use Greedy-Face-Greedy [1](GFG). This scheme requires a local planarization of the neighbors. So, the node needs to know the positions of all its neighbors. To collect this information all neighbors are allowed to answer including those not providing advance towards the destination. This is done just when necessary. Thus, some kind of special message should be used, but this process is just outlined by the authors, leaving out some interesting problems for further study. More efficient solutions such as [5], allow localized planarization by knowing only a relevant subset of neighbors.

C. Contention-Based Forwarding (CBF)

In Contention-Based Forwarding (CBF [8]) there are two phases: contention process and suppression phase.

In the contention process the node currently holding the data packet forwards it and waits for its neighbors to determine themselves which one will be the next relay in a distributed contention process. During the contention process candidate neighbors compete for becoming the next relay by setting timers related to their actual positions. The neighbor providing the most advance towards the destination waits for the shortest time before forwarding the data packet. All the candidate forwarders cancel their timers when they hear the transmission from the winning neighbor.

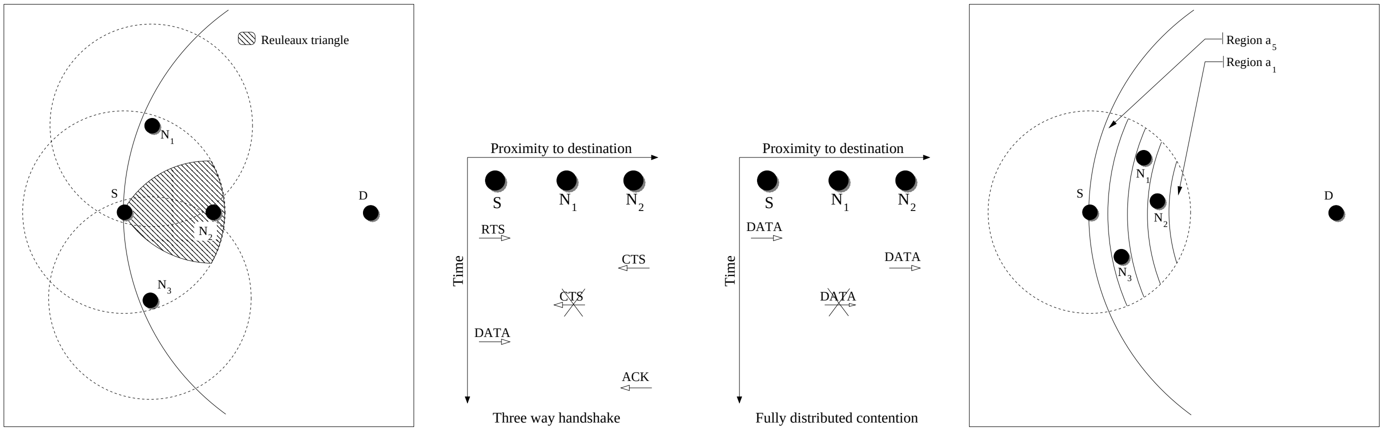
The second phase is the suppression of redundant messages. The suppression phase is used to reduce the chance of accidentally selecting more than one node as the next hop as well as to reduce the overhead of the protocol. Three different suppression schemes are proposed. The basic scheme consists just on canceling timers after hearing a transmission from another neighbor. The area based scheme consists of defining a forwarding area as in IGF. Three different areas are studied in CBF, but the one achieving the best results is the Releaux triangle. Finally, a third suppression mechanism called active suppression is defined, which is in fact the same RTS/CTS approach proposed in IGF, that allows the forwarding node to determine which neighbor to select as next forwarder among the ones whose CTS frames were received. The active selection scheme prevents some forms of packet duplication. Multiple nodes may send a CTS control packet, but only one is selected because the forwarding node acts as a central authority. Obviously, this comes at the cost of additional overhead in terms of RTS/CTS control packets. Fig 1(b) shows the differences between the second and the third schemes in terms of number of messages.

D. Beacon-Less Routing (BLR)

Beacon-Less Routing (BLR [9]) relies on a distributed contention process (see Fig. 1(b)) as the only way of determining the next forwarder. BLR selects a forwarding node in a distributed manner among all its neighboring nodes without having information about their positions or about their existence. Data packets are broadcast and the protocol takes care that just one of the receiving nodes forwards the packet. This is accomplished by computing a Dynamic Forwarding Delay (DFD) at each node depending on its position relative to the sender and the destination node.

Among all neighbors providing advance, the one in the best position forwards the data packet first. The other neighbors cancel their scheduled transmission when they overhear the data packet. To ensure that all nodes detect the forwarding, only nodes within a certain forwarding area take part in the contention to forward the packet. Furthermore, passive acknowledgments are used. That is, by detecting the transmission of the packet, the previous transmitting node concludes that it was successfully received by its next hop.

Additionally, BLR defines a recovery strategy to deal with local minima. The node broadcasts a short request and all neighboring nodes reply with a packet indicating their positions. If a node located closer to the destination than current node replies, this node is chosen as the next hop. Otherwise the actual node extracts a planar subgraph (e.g. Gabriel Graph) for its neighborhood and forwards the packet via unicast according to the right-hand rule [1].



(a) Forwarding area (b) Timing (c) Division in subareas

Figure 1. Different strategies to determine which neighbors can take part in the selection of the next forwarder process, and their response order.

E. Beacon-less On-demand Strategy for Sensor networks (BOSS)

In our previous work [4] we proposed a beacon-less protocol called BOSS (Beacon-less On-demand Strategy for Sensor networks). BOSS was designed to take into account packet losses and collisions which are common in radio communications. BOSS uses a three way (RTS/CTS/ACK) handshake (see Fig. 1(b)) and a new timer assignment function called Discrete Dynamic Forwarding Delay (DDFD). DDFD divides the neighborhood area in sub-areas according to the progress towards the destination (see Fig. 1(c)). In DDFD the forwarding delay of neighbors located in the same sub-area is computed as a shared base time plus a random number of milliseconds. Thus, neighbors in a sub-area with high progress can answer before the neighbors placed in low progress sub-areas while still avoiding collisions among nodes in the same sub-area. The goal of the DDFD is to reduce collisions among answers during the selection phase.

Additionally, when no neighbor provides advance towards the destination (i.e. no answer from positive progress areas is received), the DDFD function is also used in the next hop selection phase of the recovery process. Concretely BOSS applies the same recovery strategy as GPSR. Using the DDFD function all the neighbors not providing advance towards the destination transmit their response according to their distance from the node currently holding the message.

Moreover, the major contribution of BOSS is the inclusion of the full data payload in the RTS control packet. The reason is that, usually, bigger messages are more error-prone than short ones. So, it may happen that normally short RTS and CTS messages can traverse a link that data packets cannot. By sending the data packet first, BOSS performs the neighbor selection only among those neighbors that successfully received the data packet. This decision is justified by the results of a series of experiments [4] showing the strong relationship between the size of the message and the Packet Reception Rate (PRR).

III. Technical Challenges in Real Deployments

Most beacon-less routing protocols in the literature have been designed assuming an almost perfect network model. Their design has neglected in many cases the presence of interferences, collisions, packet losses and all typical issues which are present in almost every real deployment of a wireless network. This causes some problems such as selecting neighbors with weak or unreliable links, additional message overhead due to retransmissions, etc. Thus, in a real testbed the performance of these algorithms is unsatisfactory. We elaborate below on how the most important problems affect the operation of the beacon-less routing protocols described in the previous section.

- Next forwarder unreachability. The use of small control messages to select next forwarders, such as the case of the RTS/CTS mechanism, can lead to selecting a neighbor whose probability of reception of a much bigger message (the one containing the payload being routed) can be very low. This argument has been supported by several recent studies [4], which show the impact of the message sizes on the probability of reception at different distances.
- Unreliability. The lack of a retransmission mechanism in some protocols makes them achieve a low delivery ratio in real testbeds.
- Generation of duplicate messages. The definition and use of a forwarding area for delimiting the neighbors that can be next forwarders does not prevent neighbors from generating duplicates in realistic scenarios. Messages can be lost, and radio ranges of nodes may be different in reality due to obstacles. Therefore two or more neighbors can consider themselves as the next forwarders by not overhearing one each other even if both are in the forwarding area.
- High Contention. It is necessary to minimize the probability of two neighbors answering at the same time. Thus, the design of efficient timer assignment functions is crucial.

	Algorithm				
Problem	*IGF*	*GeRaF*	*CBF*	*BLR*	*BOSS*
Unreachability	*High*	*High*	*Med*	*Low*	*Low*
Unreliability	*Med*	*Med*	*Med*	*Med*	*Low*
Duplicates	*Low*	*Low*	*Low*	*High*	*Low*
Contention	*Med*	*Low*	*Med*	*Med*	*Low*

Table I
COMMON PROBLEMS AFFECTING BEACON-LESS ALGORITHMS WHEN CONSIDERING REAL LINKS

Table I summarizes how much each protocol is influenced by each of the problems. In particular, if we look at the *unreachability* row, we can see that protocols based on a CTS/RTS scheme (IGF, GeRaF) are the most affected, whereas those based on the idea of sending the data packet first are less affected. Regarding *unreliability*, the less affected is BOSS because it considers both active and passive acknowledgment mechanisms to reduce unreliablity without increasing the control overhead. The other protocols are only moderately affected by this problem because their retransmission schemes are less sophisticated. Regarding the creation of *duplicates*, we can see how protocols in which the sender takes a centralized selection decision (IGF, GeRaF, CBF, BOSS) are not very much affected by this problem. However, for BLR duplicates are a very serious problem because the distributed selection process may fail in realistic network conditions. We shall see clearly this effect in the experiments presented in the next section. Finally, regarding *contention*, the protocols based on forwarding areas (IGF, CBF and BLR) have medium contention because that area limits which neighbors can answer. In the case of the protocols based on forwarding subareas (GeRaF and BOSS) the contention is low because the division in subareas reduces the contention to those nodes within the same subarea, which is smaller than the whole forwarding area.

In order to evaluate the importance of taking into account the error prone nature of real wireless links during the design phase of beacon-less routing protocols, we present the results of our empirical experiments in the next section.

IV. PERFORMANCE EVALUATION

A. Experimental Setup

Fig. 2 shows our network deployment of 35 motes within the first floor of the Computer Science building at the University of Murcia. The final network covers an approximate area of 75x40 m^2 and presents some very dense areas and other less dense ones (i.e. possible void areas). This would allow us testing not only the greedy behavior but also the recovery strategy of the algorithms.

The sensors used in the deployment are the new TmoteSky sensors from *moteiv*. Each one is preloaded with information about its location, as well as the coordinates of the rest of nodes. This is done so that in our experiments there is no need for an additional location service, and we can focus on evaluating the performance of the routing itself. In each experiment we generate a log entry for each packet transmission and reception of messages, including the time, originator, destination, and size of the packet. With this information we obtain cumulative distribution functions (CDF) for the different performance metrics. Concretely, we measure the end-to-end delay, the total number of messages, the number of messages per hop, the total hop count, and the packet delivery rate (PDR).

We randomly select 15 nodes as sources and 10 as destinations. Then, we transmit 25 messages from each source to each destination. The time between data messages generated by the source has been fixed to 20 seconds to guarantee no previous messages are in the network, and the size of those messages is 120 bytes.

We select a representative protocol from each of the categories in the design alternatives. IGF is chosen among the proposals in which the sender selects the next hop based on a RTS/CTS handshake. BLR is considered among those protocols in which the sender broadcasts the data packet first, and next hops are selected in a distributed way. BOSS combines the idea of sending the data packet first, together with a next hop selection by the sender. Finally, we have evaluated the well-known GPSR [10] algorithm to compare the performance of the beacon-less proposals against a beacon-based one.

All the beacon-less protocols have been configured to wait a maximum time of 300ms before starting their recovery strategies. That is, IGF and BOSS wait for a maximum of 300ms to receive responses, and BLR listens for a retransmission (Passive ACK) for a maximum time of 300ms. In relation to the number of retries, all protocols try to chose a next forwarder up to 3 times before dropping the packet. BOSS has been configured with 5 positive progress areas. In the case of GPSR we configured the beacon interval to 5 seconds and a warm-up time of 20 seconds.

Finally, we also simulated the same protocols in the same 35 motes scenario using the TOSSIM simulator. All the protocols use the same configuration parameters as in the testbed. Additionally, we incorporated the link properties derived in [4] to take into account the relationship between the size of the messages and the Packet Reception Ratio (PRR). By doing that we tried to make the simulations as close to reality as possible.

B. Analysis of results

Fig. 3(a) shows the CDF of the end-to-end delay. That is, given a certain end-to-end value x in the X axis, the CDF represents the ratio of experiments achieving a delay lower or equal than x. For example, the figure shows that 50% of messages transmitted using GPSR reached the destination in less than 100ms.

As expected, GPSR obtains the best results in terms of end-to-end delay because it does not need any extra time to select the next hop based on the table of neighbors. All beacon-less protocols show a similar average end-to-end delay because the differences in the usage of timers among them are very

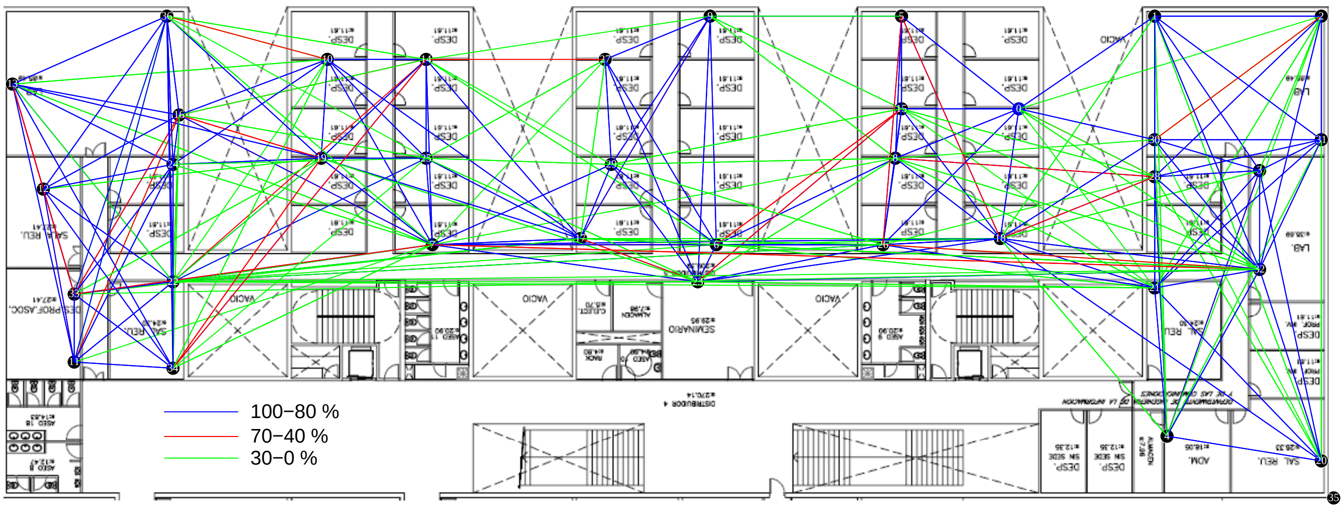
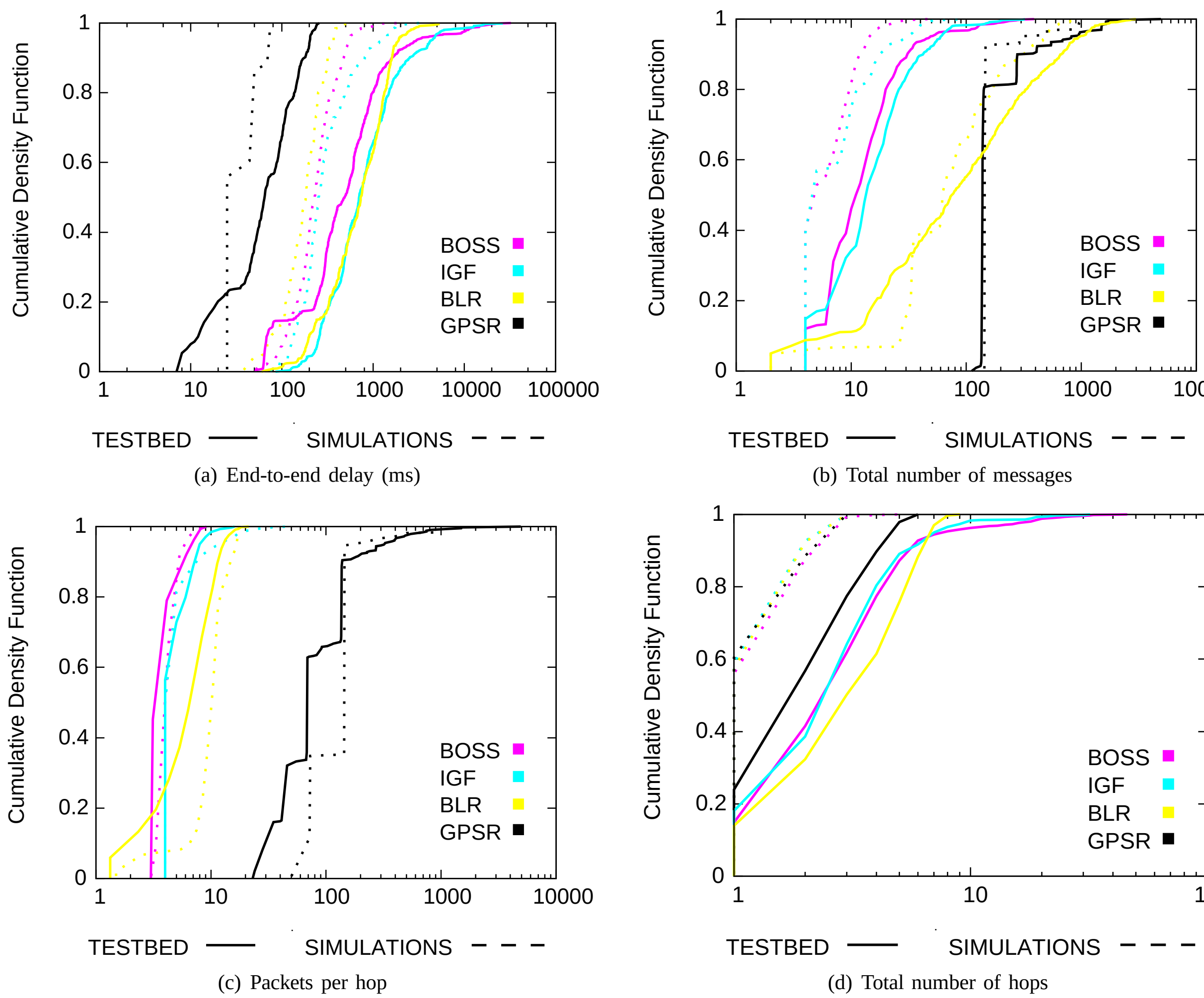


Figure 2. 35 TmoteSky deployed on to the first floor of the Department.



(a) End-to-end delay (ms)

(b) Total number of messages

(c) Packets per hop

(d) Total number of hops

Figure 3. Comparison between testbed and simulation results.

subtle and it only has a small impact in the end-to-end delay. Even though in theory BLR might be the fastest one due to not using any handshake mechanism, in the testbed it is one of the slowest. The reason is that the big amount of traffic it generates (due mainly to duplicates) creates a lot of contention at the MAC layer. Therefore, nodes need to back off very often, incrementing thus the end-to-end delay.

Moreover, all the protocols perform better in the simulator than in the real testbed. The reason is that radio link conditions in the simulator are not as tough as in the real deployment.

In wireless sensor networks, reducing the consumption of resources is far more important than having low delay communications. Thus, the overhead of the protocols is a very important metric to consider. Fig. 3(b) shows the CDF of the total number of messages transmitted by each protocol including control (RTS, CTS, ACK and beacon) and DATA messages. The figure shows that BOSS needs a lower number of messages than the other protocols to deliver the data packet. In the 90% of the experiments BOSS reaches the destination using less than 30 messages while IGF needs 40 and BLR more than 500. The reason is that BOSS, by considering only nodes which already received the data packet in its next hop selection algorithm, requires a lower number of retransmissions. That means a lower number of messages per hop. This is clearly illustrated in Fig. 3(c) when comparing the number of packets per hop required by each protocol.

The huge overhead obtained by BLR is also due to the large amount of duplicate messages which are generated in its next hop selection process. We can also see the overhead introduced by beacons needed by GPSR. It generates a minimum of 141 messages and more than 500 in 10% of the experiments. The 141 messages transmitted by GPSR are due to the 5 beacons generated by the 35 nodes every 5 seconds during the 20 initial seconds of the test plus the data message being sent. Obviously, in beacon-based protocols such as GPSR, the use of beacons generates an overhead not introduced in beacon-less protocols.

Regarding the length of the shortest path found by the protocols, Fig. 3(d) shows that, as expected, all protocols obtain a very similar performance. The reason is that the number of reasonable paths for a greedy forwarding scheme between sources and destinations in the testbed is not very high. So all protocols tend to choose very similar paths in terms of hop count. The very few cases in which BLR performs better than the other schemes is when they require perimeter forwarding. The reason is that even though BLR also requires perimeter, the huge amount of duplicate packets makes it very likely that, at least, one of them gets routed to the destination through a shorter path. Of course, as shown in Fig. 3(c), in those cases, the huge packet overhead does not pay off considering the subtle reduction in the length of the paths.

The deviations between simulation and testbed results are explained by the differences in the mean number of hops. These are caused because in the simulator radio conditions are more benign, so that radio links tend to be longer than in reality. Thus, the number of hops tends to be lower than in the real testbed, where hostile and variable radio link conditions are quite common.

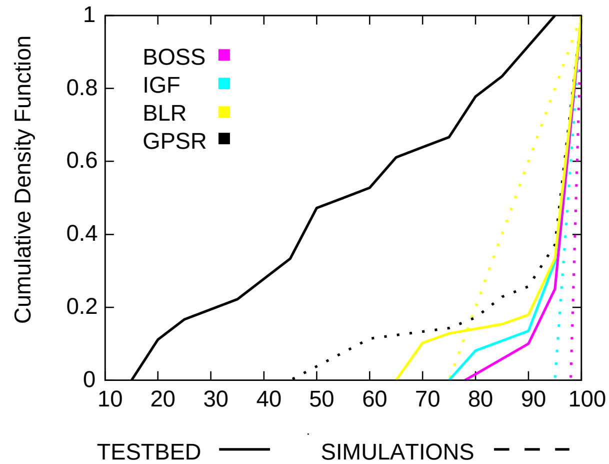


Figure 4. Packet Delivery Ratio

Finally, regarding the PDR, Fig 4 shows that beacon-less protocols are far better than GPSR. Concretely, 80% of the experiments successfully deliver more than a 90% of messages. The reason for this is mainly that the neighbor table is not able to keep up with the changes in the radio conditions. Also, by selecting the neighbor closest to the destination GPSR ends up selecting as next hop neighbors with weak radio links.

V. Conclusions and Open Issues

In this paper we describe the most important protocols in the field of beacon-less geographic routing. We evaluate the different design issues and possible problems they could suffer in a real deployment. In addition, we use simulations and a real testbed consisting in 35 TmoteSky sensors to study the performance of some representative solutions of each mayor design alternative. This study includes also a performance comparison against the well known beacon-based GPSR algorithm.

The results of our experiments show that beacon-less protocols such as BLR, in which next hops are selected in a distributed way, generate a tremendous number of duplicates degrading the performance of the protocol to unacceptable levels. On the contrary, letting the sender select the next hop based on some kind of RTS/CTS handshake approach keeps the overhead at very low levels while achieving a high Packet Delivery Ratio. Our study also confirms that the strategy of selecting next hops using exclusively a distance criteria and neighborhood tables typical of beacon-based geographic algorithms is not appropriate for real deployments. In our experiments, IGF and BOSS provide a higher packet delivery ratio and a much lower packet overhead than GPSR.

Beacon-less geographic routing is certainly a very promising research area. However, there are still some open issues.

These include dealing with energy efficiency, duty-cycle operation of sensor nodes, and also improving the operation in perimeter mode. Moreover, the operation of beacon-less geographic routing algorithms seems also a good candidate for scenarios with some mobility, where neighbor tables created with beacons can store outdated positions. Finally, extensions for beacon-less multicast routing could be of great interest for some applications which require sending the same message to multiple nodes.

ACKNOWLEDGMENT

Part of this work has been funded by the excellence groups research funds of Fundacin Seneca under grant 04552/GERM/06 and the SMART project (TIN2005-07705-C02-02) from the Spanish MEC. Authors would like to thank the valuable comments of the anonymous reviewers.

REFERENCES

[1] S. Giordano, I. Stojmenovic, and L. Blazevie, "Position Based Routing Algorithms for Ad Hoc Networks: A Taxonomy," *Ad Hoc Wireless Networking*, pp. 103–136, 2004.

[2] M. Witt and V. Turau, "The Impact of Location Errors on Geographic Routing in Sensor Networks," in *Proc. of the Second International Conference on Wireless and Mobile Communications (ICWMC'06)*, Bucharest, Romania, Jul. 2006, p. 76.

[3] J. Zhao and R. Govindan, "Understanding Packet Delivery Performance in Dense Wireless Sensor Networks," in *Proc. First International Conference on Embedded Networked Sensor Systems (SenSys '03)*. New York, NY, USA: ACM Press, 2003, pp. 1–13.

[4] J. A. Sanchez, R. Marin-Perez, and P. M. Ruiz, "BOSS: Beacon-less On Demand Strategy for Geographic Routing in Wireless Sensor Networks," in *Proc. of the 4th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS '07)*, October 2007, pp. 1–10.

[5] H. Kalosha, A. Najak, S. Ruhrup, and I. Stojmenovic, "Select-and-protest-based beaconless georouting with guaranteed delivery in wireless sensor networks," in *Proc. 27th Conference on IEEE Computer Communications (INFOCOM 08)*, April 2008, pp. 346–350.

[6] B. Blum, T. He, S. Son, and J. Stankovic, "IGF: A state-free robust communication protocol for wireless sensor networks," Department of Computer Science, University of Virginia, USA, Tech. Rep., 2003.

[7] M. Zorzi and R. R. Rao, "Geographic random forwarding (GeRaF) for ad hoc and sensor networks: energy and latency performance," *IEEE Transaction on Mobile Computing*, vol. 2, no. 4, pp. 349–365, 2003.

[8] H. Füßler, J. Widmer, M. Käsemann, M. Mauve, and H. Hartenstein, "Contention-Based Forwarding for Mobile Ad Hoc Networks," *Ad Hoc Networks*, vol. 1, no. 4, pp. 351 – 369, 2003.

[9] M. Heissenbüttel, T. Braun, T. Bernoulli, and M. Wächli, "BLR: Beacon-Less Routing Algorithm for Mobile Ad-Hoc Networks," *Elsevier's Computer Communications Journal (ECC)*, vol. 27, no. 11, pp. 1076–1086, July 2004.

[10] B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *Proc. 6th annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '00)*. New York, NY, USA: ACM Press, 2000, pp. 243–254.