Area-based beaconless reliable broadcasting in sensor networks

Francisco Javier Ovalle-Martínez, Amiya Nayak and Ivan Stojmenović*

School of Information Technology and Engineering,
University of Ottawa,
Ottawa, Ontario, Canada K1N 6N5
E-mail: fovalle@site.uottawa.ca
E-mail: anayak@site.uottawa.ca
E-mail: ivan@site.uottawa.ca
*Corresponding author

Jean Carle and David Simplot-Ryl

IRCICA/LIFL, INRIA futurs,
University of Lille,
Lille, France
E-mail: Jean.Carle@lifl.fr
E-mail: David.Simplot@lifl.fr

Abstract: We consider the broadcasting problem in sensor networks where the nodes have no prior knowledge of their neighbourhood. We describe several Area-based Beaconless Broadcasting Algorithms (ABBAs). In 2D, on receiving the packet (together with geographic coordinates of the sender), each node calculates the ratio *P* of its perimeter, along the circle of transmission radius, that is not covered by this and previous transmissions of the same packet. The node then sets or updates its timeout to be inversely proportional to *P*. If the perimeter becomes fully covered, the node cancels retransmissions. Otherwise, it retransmits at the end of the timeout interval. The protocol is reliable, that is, all nodes, connected to the source, are guaranteed to receive the packet, assuming an ideal MAC layer. We also describe three 3D-ABBAs, one of them being reliable. These three protocols are based on covering three projections, covering particular points on intersection circles and covering intersection points of three spheres. Our protocols are the first reliable broadcasting protocols, other than blind flooding.

Keywords: sensor networks; beaconless broadcasting, flooding, localised algorithms.

Reference to this paper should be made as follows: Ovalle-Martínez, F.J., Nayak, A., Stojmenović, I., Carle, J. and Simplot-Ryl, D. (XXXX) 'Area-based beaconless reliable broadcasting in sensor networks', *Int. J. Sensor Networks*, Vol. X, No. Y, pp.XXX–XXX.

Biographical notes: Francisco Javier Ovalle-Martínez received a Bachelor of Science degree in Telecommunications Engineering from the National Autonomous University of Mexico (UNAM) in 2001 and an MSc in Computer Science from the University of Ottawa in 2005. Currently, he is a Software Engineer (bioinformatics algorithms) at the National Institute of Genomics Medicine, Mexico City, Mexico. His current research interests include wireless ad hoc and sensor networks and bioinformatics.

Amiya Nayak is an Associate Professor at the School of Information Technology and Engineering (SITE) at the University of Ottawa. He received a BMath degree in Computer Science from the University of Waterloo and a PhD in Systems and Computer Engineering from Carleton University, Ottawa in 1982 and 1991, respectively. He has over 15 years of industrial experience in software engineering in avionics and telecommunication applications, working at CMC Electronics and Nortel Networks prior to joining University of Ottawa in 2002. He has been an adjunct research professor at Carleton University since 1994 and the Canadian Editor and the Book Review Editor of *VLSI Design: An International Journal of Custom-Chip Design, Simulation, and Testing* since 1998. His research interests are in the areas of fault-tolerant computing, ad hoc networks and distributed computing.

Ivan Stojmenović received a PhD in Mathematics in 1985. He held regular or visiting positions in Serbia, Japan, USA, Canada, France and Mexico. He published over 200 papers in journals and conferences, edited *Handbook of Wireless Networks and Mobile Computing* (Wiley, 2002) and co-edited *Mobile Ad Hoc Networking* (IEEE Press, 2004), and Handbook of Sensor Networks (Wiley, 2005). His current research interests include wireless ad hoc, sensor and cellular networks. Currently, he is Editor-in-Chief of three journal and Editor of several journals including, *IEEE Transactions on Parallel and Distributed Systems*. He guest edited recently special issues in several journals including *IEEE Computer Magazine*, *IEEE Networks* and *Wireless IEEE TPDS Communications and Mobile Computing*.

Jean Carle is an Associate Professor in the Fundamental Computer Science Laboratory of Lille (LIFL) at the University of Lille, France. His research interests include distributed computing, data communication, mobile ad hoc networks and sensor networks. He received a PhD in Computer Science from the University of Amiens.

David Simplot-Ryl is an Associate Professor in LIFL at the University of Lille. His research interests include ad hoc networks, distributed computing, embedded operating systems and RFID technologies. He received a PhD in Computer Science from the University of Lille.

1 Introduction

In this paper, we consider sensor networks with homogeneous nodes. That is, all nodes have the same and fixed transmission radius. Wireless nodes cooperate to perform data communication tasks. In a broadcasting task, a message is to be sent from one node to all the other nodes in a sensor network, directly or via retransmissions from other nodes. Broadcasting is an important task used for paging, alarming, location updates, route discoveries or even routing in highly mobile environments. Blind flooding is a classical approach to broadcast messages across a network. It starts with the source node disseminating a message among its neighbours. Whenever a node receives the message for the first time, it transmits one copy to all of its neighbours. Despite its broad use and advantages, an increase in the density (average number of neighbours) produces contentions, collisions and redundancies that limit the scalability of blind flooding.

There exist several localised broadcasting algorithms that use intelligent techniques to reduce communication overhead during the broadcast. However, most of these protocols assume one-hop or possibly two-hop knowledge. To search their neighbours, the nodes in the network use beacons. The use of such ‘hello’ messages increases with mobility and changes in nodes’ activity status, resulting in critical bandwidth and energy losses. To address this issue, beaconless broadcasting, more intelligent than flooding, has been considered. In these protocols, there are no ‘hello’ messages, and the nodes act based on information gathered only during the broadcasting process. Beaconless broadcasting reduces energy consumption and increases broadcasting accuracy, since the protocols based on neighbour knowledge may have outdated information in dynamic ad hoc and sensor networks. This paper studies the design of beaconless broadcasting protocols. The protocols proposed here require position information of each node to be available, which is not necessary to run the blind flooding protocol.

The first beaconless broadcasting protocols (other than blind flooding) are probabilistic and location-based schemes by Ni et al. (1999). However, they do not have good performance, either leaving many nodes uncovered or requiring most nodes to retransmit, as shown experimentally by Stojmenovic et al. (2002). We identified two existing recent beaconless broadcasting protocols: Optimised Broadcast Protocol (BPS) by Durresi et al. (2005) and Geoflood by Arango et al. (2004). BPS is based on selecting the forwarding nodes, which are the closest to the imaginary vertices of a honeycomb network. The message contains the location of the previous and the last senders. This enables the receiver to determine the locations of two vertices of a honeycomb network, with its side being equal to the transmission radius, and with one direction corresponding to the edge direction determined by the last two senders. On receiving such a message, each node finds its distance to the two locations (vertices of the hexagon), and sets a timeout proportional to the minimum distance. This allows closer nodes to retransmit sooner, which also suppresses other retransmissions. The authors claimed that their algorithm minimises the number of necessary transmissions and that it outperforms all the other variations of flooding. However, BPS lacks reliability. Even for highly connected networks, BPS can fail to deliver to all the recipients connected to the source.

In the Geoflood algorithm (Arango et al., 2004), each node defines a Cartesian plane with its location as the origin. On receiving the first message copy, the node records the quadrant from where the message was sent and sets a timeout. If the node also receives at least one copy of the message from each of the other quadrants before the timeout expires, it does not retransmit. As with OFP, the lack of reliability is the principal problem that Geoflood has. Its authors recognised and showed this problem; however, they claimed that in high-density networks this is not a serious issue. However, our measurements show that Area-based Beaconless Broadcasting Algorithms (ABBAs) have lower retransmission counts than Geoflood, while preserving the reliability. In later sections, we will show how to modify Geoflood to convert it to a reliable beaconless broadcasting algorithm. However, the retransmission count worsens further.

The principal objective of this work is to develop several beaconless broadcasting algorithms. These protocols do not use 'hello' messages. More precisely, we propose four versions of ABBAs, one for Two-Dimensional (2D) and three for Three-Dimensional (3D) spaces. The idea behind ABBAs is very simple. We assume that the nodes do not have information about their neighbours. On receiving the first copy of a message, each node sets a timer. Before the timeout expires, the node may receive more copies of the same message. Every node has a transmission coverage, which is assumed to be a circle in 2D and a sphere in 3D. If a node *A* receives a message from different sources and these sources cover the transmission circle or the sphere of *A*, then the node *A* has no need to retransmit the message. This means that each of its possible neighbours has received the message already from one or more nodes that cover it. If the node is covered before the timeout expires, then it decides not to transmit and ends the timer. Otherwise, the timer will continue to run, and, when it expires, the node transmits. 2D and one of 3D-ABBA versions are reliable.

We simulated our proposed broadcast algorithms, measuring the number of transmissions done for different network densities. We implemented OFP, three versions of Geoflood and compared them with 2D-ABBA. We showed that OFP and Geoflood are not reliable algorithms. To become reliable, the nodes in Geoflood have to divide their transmission area in six angular regions instead of four. In our tests, with 500 nodes and different average degrees, OFP failed considerably. In connected networks with average degree 7, OFP failed to complete a broadcast in 97% of the tested networks. Even in connected networks with high average degrees, OFP failed occasionally. This shows that OFP is not a good broadcasting protocol due to its frequent lack of reliability. Although Geoflood was experimentally successful in every test except one, it is shown that the versions with four and five angles division can be unreliable. In addition, Geoflood, even with four angles division, had higher retransmission counts than ABBAs. The reliable version of Geoflood (with division into six angles) leads to further increase in retransmissions. 2D-ABBA with coverage-based timeout used on average 40% less messages than the reliable version of Geoflood, whereas the random 2D-ABBA had on average 28% less messages. Compared to Geoflood with four and five angles division, the original 2D-ABBA had on average 12% and 34% less messages, respectively.

Among our 3D proposals, 3D-ABBA1, based on covering three 2D projections, had the best results in terms of number of messages to complete the broadcast. However, this version lacked reliability. The second best 3D version (3D-ABBA2) performed very close to 3D-ABBA1 and had 100% delivery rate during the tests, although this version in theory is also unreliable. 3D-ABBA2 is based on covering (by other transmitting spheres) six equidistant points on the intersecting perimeter of considered sphere and any transmitting neighbouring sphere. 3D-ABBA3 is a reliable broadcasting protocol. It is based on a covering theorem. A sphere *A* is covered by other spheres if every intersection point of three spheres from the set, one of them being *A*, is located inside another sphere from the covering set. Any three spheres determine two such common intersection points to be covered.

Protocol versions that used random timeout function were very competitive to the original optimised timeout values. Therefore, we believe that our protocols will preserve good performance if implemented under a realistic MAC layer such as in IEEE 802.11 based networks. Recall that IEEE 802.11 use random timeouts in its backoff scheme.

The rest of this work is organised as follows. Section 2 presents the related work for the broadcasting problem. In Section 3, we describe four versions of ABBA. Section 4 gives performance evaluation of the proposed 3D broadcasting algorithms. Section 5 compares OFP and Geoflood with 2D-ABBA. Section 6 concludes this paper.

2D-ABBA has been originally proposed in summer 2003 by the last three authors, and its one paragraph description was published by Simplot-Ryl et al. (2005), Ingelrest et al. (2005) and Li and Stojmenovic (2006). The first three authors developed generalisations to 3D-ABBAs in spring of 2004. This paper summarises the master thesis work of the first author, defended in summer 2005.

2 Literature review

Williams and Camp (2002) classified the broadcast protocols into: simple (blind) flooding, probability-based, area-based and neighbour knowledge methods. Stojmenovic and Wu (2004) reclassified area-based methods within other groups while neighbour knowledge methods were divided into clustering-based, selecting forwarding neighbours and internal node-based methods. They presented a comprehensive taxonomy of broadcasting schemes for the one-to-all model (where a message sent by one node is received by all its neighbours). All schemes can be classified following the taxonomy consisting of five categories: determinism, network information, reliability, 'hello' message content and broadcast message content.

In the blind flooding scheme, whenever a node receives the message for the first time, it retransmits that message. The method disseminates information quickly in a network with enough bandwidth and no loss prone links (Heinzelman et al., 1999). Nevertheless, it has disadvantages prohibiting its use in networks with dynamic topology behaviour or where the requests for information are frequent and must be known by all the participants. The main disadvantage of blind flooding is its extensive use of the bandwidth in dense networks. At the node level, each message retransmission consumes power, processing time and the available bandwidth on the incoming and the outgoing links.

It has been recognised that *scalability* in wireless networks cannot be achieved by relying on solutions where each node requires global knowledge about the network.

To achieve scalability, the concept of *localised* algorithms was proposed, as distributed algorithms where simple local node behaviour, based on local knowledge (one- or two-hop neighbours), achieves a desired global objective.

Ni et al. (1999) studied the broadcast storm problem. Blind flooding is usually very costly and will result in serious redundancy, contention and collision. They identified this broadcast storm problem by showing how serious it is through analyses and simulations. Several beaconless schemes (probabilistic, counter-based, distance-based and location-based) were proposed by Ni et al. (1999) to reduce redundant rebroadcasts and differentiate the timing of rebroadcasts to alleviate this problem. In the probabilistic scheme (Ni et al., 1999), each node rebroadcasts the first copy of a receives message with a given probability p. In the counter-based scheme (Ni et al., 1999), each node rebroadcasts the message if and only if it received the message from less than C neighbours. In the distance-based scheme (Ni et al., 1999), the message is retransmitted if and only if the distance to each neighbour that already retransmitted the message is greater than D. In the location-based scheme (Ni et al., 1999), the message is retransmitted if and only if the additional area that can be covered if the node rebroadcasts the message (divided by the area of circle with transmission radius) is greater than the threshold A. A simplified version of the method is to rebroadcast the message if the node is not located inside the convex hull of neighbouring nodes that already retransmitted the message. However, these methods are not reliable. Further, the experimental data (Ni et al., 1999; Stojmenovic et al., 2002) indicate low saved rebroadcasts for a high reachability, compared to other methods. We therefore did not include them in our experiments.

Lin and Shrivastava (2003) considered small-scale auctions and proposed a broadcast counter-based broadcast algorithm from Ni et al. (1999) for use in the environment. In this protocol, each node receiving a message for the first time sets a timeout interval, which can be modified if more messages arrive while it waits. If a node receives a certain threshold number of messages (suggested value is 3), the message is not forwarded. Otherwise, it is forwarded on the expiration of timeout. As noted already by Stojmenovic et al. (2002), the protocol does not guarantee delivery, and therefore is not suitable in applications for auctions, which should provide access to *all* buyers.

A survey of broadcasting protocols which use beacons to acquire neighbour knowledge is given by Stojmenovic and Wu (2004). These methods include distance estimate (Cartigny and Simplot, 2003), multipoint relay (Calinescu et al., 2001; Sun and Lai, 2001, 2002a,b), neighbour elimination (Stojmenovic et al., 2002) and broadcasting based on connected dominating sets (Stojmenovic et al., 2002) (they are based on the concepts introduced by Wu and Li (2001)). We elaborate here on the methods that are more relevant to the presented contributions.

Gerla et al. (2000) proposed a combined clustering and broadcasting algorithm that has no communication overhead for either maintaining a cluster structure or updating the neighbourhood information. In their passive clustering algorithm, the cluster structure is updated with existing traffic by adding two bits to each of the ongoing message. The source S of a broadcasting task transmits the message to all of its neighbours. S declares itself a Clusterhead (CH), for the timeout period that is a parameter in the method, if it has no neighbouring active CH. Upon receiving the message, each node A declares itself a CH using the same criterion as the source S. Otherwise, A checks the ratio of neighbouring CHs and neighbouring gateway nodes and declares itself a gateway if that ratio is above a certain threshold, which is also a parameter of the method. If A decides to be a gateway, it retransmits the message. Otherwise A decides to be an ordinary node and does not retransmit the message. The method is not reliable, there are pathological cases of poor delivery ratio and it has global parameters.

Yi et al. (2003) modify the passive clustering protocol from Gerla et al. (2000) to achieve broadcasting without periodic, background control packet exchange. Passive clustering maintains clusters using an implicit timeout. A node assumes that the nodes it had previously heard from, have died or are out of its locality if they have not sent any data within timeout duration. Thus instead of using CH to gateway ratio (Gerla et al., 2000), the protocol (Yi et al., 2003) uses a cluster timeout interval. A node that belongs to two or more clusters at the same time is eligible to be a gateway node. Only one gateway for each pair of neighbouring gateways is allowed. Distributed gateways are used to connect neighbouring cluster heads three hops away. The problem with the protocol (Yi et al., 2003) is that no description is given on how the process works at the very beginning. Assume, for example, that there was no traffic at all in the network for the timeout interval, and therefore the complete cluster structure is lost. How then broadcasting proceeds from a node? According to the algorithm (Yi et al., 2003), a source node transmits the message, and all its neighbours then become gateway candidates and simply hold the message without retransmitting it. Another problem is that the selection of a proper value of timeout interval is highly dependent on the mobility speed. High node mobility may 'destroy' the cluster structure before the expiration of the timeout interval, which of course affects the delivery rate for broadcasting.

Bergonovo et al. (2003) described applications of broadcasting in inter-vehicle communications on the highway, with a large and variable number of mobile terminals, an extremely dynamic network topology and stringent requirements related to the broadcast warning messages. Their proposed multihop broadcast protocol is based on the reservation ALOHA MAC protocol. The protocol appears to be a variant of the neighbour elimination scheme (Stojmenovic et al., 2002), which requires the maintenance of the neighbours list. This may not be an easy task in a highly mobile environment, where ongoing traffic has to be mixed with beacon messages for maintaining the neighbourhood information. The solution should be beaconless, and with guaranteed delivery, since missing to warn a single car may be fatal.

Sun et al. (2003) proposed a reliable multicasting where RTS signals are followed by CTS signals from selected neighbours instead of all neighbours. The set of neighbours for sending back the CTS signal is selected so that they cover the same area as all the neighbours. Two versions are proposed, with and without using geographic information. Therefore, Sun et al. (2003) also use area coverage, but for a different purpose. The broadcast or multicast protocol on the network layer is not changed.

Beaconless position-based routing has been recently studied, and several protocols were proposed: BLR (Heissenbuttel and Braun, 2004), CBF (Fussler et al., 2003) and IGF (Blum et al., 2003). The sender node transmits a packet looking for help in forwarding a message. Nodes closer to the destination respond sooner by using a timeout function based on distance. The neighbour closest to the destination therefore responds first. The protocols are obviously not appropriate for a broadcasting task.

Durresi et al. (2005) proposed a beaconless flooding algorithm named Optimised Broadcast Protocol (BPS). It is based on a hexagonal tiling of a plane, with the transmission radius as the edge length of the hexagons. Each broadcast packet contains two location fields $L1$ and $L2$, in its header. Whenever a node transmits a packet, it sets $L1$ to the location of the node from which it received the packet and sets $L2$ to its own location. The source node S sets both $L1$ and $L2$ to its location and transmits the packet. Each receiving node B first determines if the packet can be discarded. A packet can be discarded if the node has transmitted the same packet or if a node, which is very close has already transmitted this packet. If the packet is not discarded, B determines if the packet came from the source S. If the message came from the source node S, B finds the nearest vertex V of a hexagon with centre coordinates at S and with $(Sx + R, Sy)$ as one of its verticexes, where $S = (Sx, Sy)$. It computes its distance l from V and then delays the rebroadcast by a delay l/R. If the message came from other node K, then B selects the nearest hexagon vertex, and the rebroadcast is delayed by $l/20 \times R$. After the delay elapses, B determines if the packet can be discarded (for the same explained reasons). If the packet cannot be discarded, B sets $L1$ to the location of the node from which it received the message and $L2$ to its own location, and transmits. The authors of BPS propose avoiding retransmissions by having node B keep track of its distance dB to the nearest node that has retransmitted the packet. If this distance is greater than a threshold value Th, then B will retransmit. BPS authors proposed a value $Th = 0.4$ R to ensure high delivery ratio while keeping the number of transmissions low.

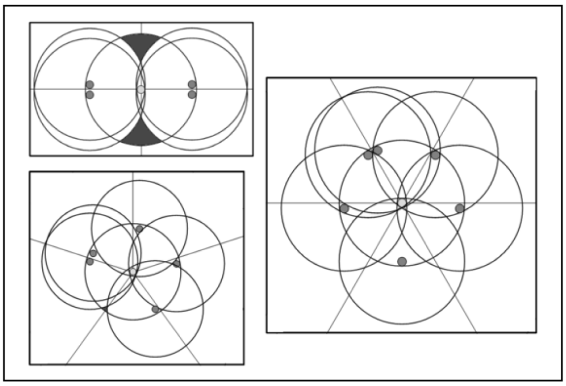
As observed by Kim and Maxemchuk (2003), the protocol may repeat flooding if neighbours do not agree on the choice of node near common ideal point. Authors (Kim and Maxemchuk, 2003) introduced a stopping rule to prevent that, and applied this type of flooding for route discovery. A more serious problem with BPS is its lack of reliability. As we will show in our experimental results, OFP can fail even in highly dense connected networks.

Arango et al. (2004) proposed Geoflood, another beaconless broadcasting protocol. The algorithm is based on setting timers before retransmitting the message. Each node A defines a Cartesian plane with its location as the origin. On receiving the first copy of a message, node A records the quadrant from where the message was sent and sets a timeout. Authors (Arango et al., 2004) proposed that receiving nodes furthest away from the local sender should select smallest holding times. This, timeouts increase as the distance to the sender decreases. They also introduced a random component to avoid contention that could arise between nodes located at the same distance from the sender. If node A receives more copies of the same message, at least one from each of other quadrants, before the timeout expires, it does not retransmit. The authors (Arango et al., 2004) recognised that their algorithm could fail in some situations. They showed the worst-case scenario in which part of the transmission area was not covered even when a node received messages from all four quadrants. They claimed that this area was relatively small and that other nodes in high-density networks would likely cover it. However, our experiments do not support such optimism.

Geoflood divides the transmission area into four quadrants or into four angles. This division causes the unreliability of Geoflood. To ensure reliability, Geoflood needs to divide into six angles instead of four. However, six transmissions from neighbours may not be necessary to cancel a retransmission, even in some cases three would be enough. This will bring unnecessary transmissions leading to unnecessary use of energy. In other words, reliability comes with the price of increased average retransmission cost.

Figure 1 shows a case where a node divides its transmission area into four angles, and since the node received a message from the four angles it will not retransmit. However, it can be observed that there is still significant uncovered area (shaded area in the first example). Also in Figure 1, we show the cases where a node divides its transmission area into five and six angles. As the reader may observe, even with five angles the algorithm could fail (small shaded area in the south–west direction). Geoflood can guarantee reliability only by dividing the transmission area in six angles.

Figure 1 Receiving nodes dividing their transmission area into four, five and six angles. Only with six divisions, Geoflood guarantees reliability



Recently Heissenbüttel et al. (2004, 2006) proposed a beaconless broadcasting protocol called: Dynamic Delayed Broadcasting (DDS). The idea behind DDS is basically the same as the one behind 2D-ABBA for nodes in two dimensions with position information available. Instead of taking into account perimeters as 2D-ABBA does, DDS takes into account the area left to be covered when a transmission arrives and decide not to transmit if the area left to be covered is less than a threshold value *th*. This protocol, however, does not guarantee the delivery of messages. The authors only proposed a 2D beaconless protocol, while we describe one 2D and three 3D beaconless protocols.

3 Area-based beaconless broadcasting algorithm in 2D

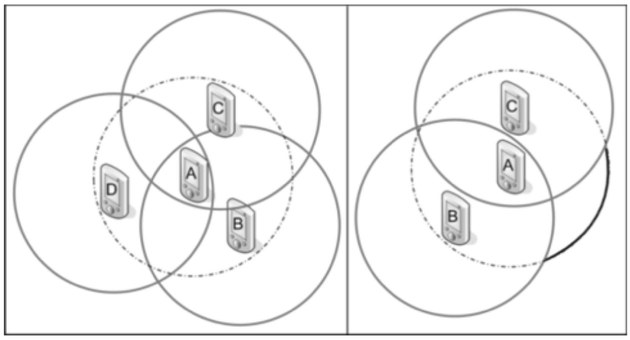
3.1 Assumptions

We assume that every node in the network knows its geographic position. This geographic position can be obtained, for example, by the Global Positioning System (GPS). Our second assumption is that all the nodes have the same transmission radius and use omnidirectional antennas; in other words, their transmission patterns are circumferences. Finally, nodes do not know where the neighbouring nodes are, so no beacons or hello messages are used.

3.2 Description of 2D-ABBA

The idea behind 2D-ABBA is very simple. We assume that the nodes do not have information about their neighbours. On receiving the first copy of a message that is broadcast, each node sets a timer. Before the timer expires, the node may receive more copies of the same message. Every node has a transmission coverage, which is assumed to be a circle. If a node *A* receives a message from different sources and these sources cover the transmission circle of *A,* then node *A* has no need to retransmit the message. This means that each of its possible neighbours has received the message already from one or more nodes that cover it. If the node is covered before the timeout expires, then the node decides not to transmit and ends the timer. However, as long as the node is uncovered, the timer will continue to run, and when the timer expires the node transmits. Since all circles are of the same radius *R*, and each node *A* is aware only of the intersections with circles whose centre *B* is inside its own circle (i.e. $|AB| \leq R$), the coverage criterion can be simplified. Instead of covering the whole circle centred at *A*, only the perimeter of that circle needs to be covered. Further, when this perimeter is intersected by several circles whose centres are inside *A*'s circle, there can be up to two segments on the perimeter that are not yet covered. This property further simplifies the implementation. In Figure 2, we show two examples of how the transmission radius of a node could be covered by other transmitting nodes. When a node is completely covered, it sets its timeout to 0 and can decide immediately not to retransmit.

Figure 2 On the left: transmission area of node *A* is covered by transmissions of nodes *B*, *C* and *D*, so node *A* does not transmit. On the right: Nodes *B* and *C* retransmit the same message received by other sources, node *A* processes the messages and updates its perimeter covered by the other two nodes. After its timeout expires, node *A* will transmit



We propose two possible timeout functions for 2D-ABBA. The first one is a function that increases when the total length of uncovered portions of the circles with transmission radius around the node decreases. The function chosen was *timeout* = *degreesCovered*. The variable *degreesCovered* refers to the angle in degrees of the circle that has been covered. The second function that we considered is a random function with values between 0 and 1. We considered the second function to test the viability of 2D-ABBA if used in conjunction with already deployed IEEE 802.11 networks. This broadcasting protocol is reliable independently on the selected timeout function.

The pseudo code for 2D-ABBA is as follows:

```
BEGIN 2D-ABBA
A source node S broadcasts a message M
Each intermediate node L runs as follows with respect to the message M initiated by S
{
        Set perimeter covered by the transmission of M;
        Set timeout;
        REPEAT
                Wait until another copy of M is received or timeout expires;
                IF another copy is received THEN
                        Update perimeter covered;
                        update timeout;
                ENDIF
        UNTIL timeout expires;
        IF still uncovered THEN
                Retransmit;
        ENDIF
        Ignore further copies of M
}
END 2D-ABBA
```

4 Area-based beaconless broadcasting algorithms in 3D

4.1 Assumptions

We consider the transmissions patterns as spheres. In 3D, a similar simplification can be applied. All spheres are of the same radius, and each node *A* is only aware of transmissions from neighbours at distance ≤*R*. Instead of covering the whole sphere centred at *A*, one can consider only covering its 3D perimeter (set of nodes at distance = *R* from *A*). All versions described here refer to covering that 3D perimeter.

4.2 3D-ABBA1

The idea behind our 3D-ABBA1 protocol is to use the three projections planes *XY*, *XZ* and *YZ*. We used these planes to try to observe the intersections of the transmission spheres of the nodes in the network. By projecting the transmission spheres into the three planes, the nodes are able to apply 2D-ABBA in each plane. Node *A* will not transmit if its corresponding transmission circles in all three planes have been covered. This in turn is simplified to covering the corresponding 2D perimeters. Again we define two different timeout functions. The first one is directly proportional to the three perimeters (one for each plane) covered by other nodes. In other words we considered

$$\text{timeout} = \frac{degsCovXY + degsCovXZ + degsCovYZ}{3}$$

The variables *degsCovXY*, *degsCovXZ* and *degsCovYZ* refer to the angles in degrees of the circles that have been covered in the three corresponding planes. The second function considered is a random function with values between 0 and 1. We considered the second function to test the viability of 3D-ABBA1 in IEEE 802.11 networks.

Figure 3 shows an example of how a node receives two transmissions in 3D and how it updates its transmission pattern on the three planes, *XY, XZ* and *YZ*. When a node gets completely covered, it sets its timeout to 0 so it can decide immediately not to transmit.

3D-ABBA1 is a simple proposal of a 3D beaconless algorithm. However, it can fail in some cases. Figure 4 shows a specific example where 3D-ABBA1 fails; the three nodes transmit the same message. The central blue node receives the transmissions and updates its transmission sphere. We could believe that the central blue node is completely covered if we look at the projections; however, we can observe that part of the central blue transmission sphere still is not covered. Unfortunately, this can question the viability of 3D-ABBA1. However, as we will show in the simulation results section, 3D-ABBA1 was reliable in almost all the tests.

Figure 3 Example of 3D-ABBA1. The receiving node (blue) updates its transmission pattern perimeter in the three planes *XY*, *XZ* and *YZ* after receiving two transmissions

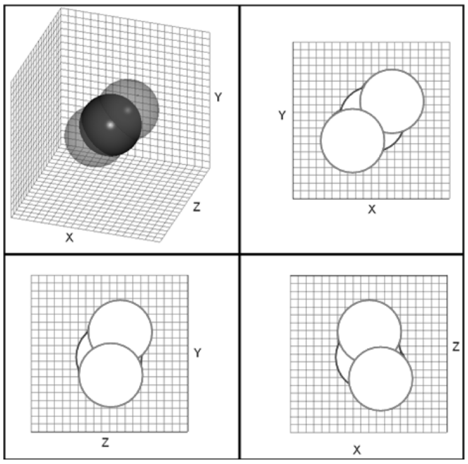
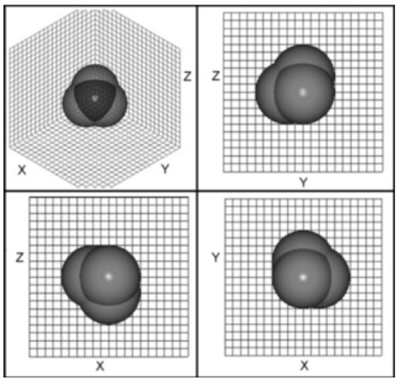


Figure 4 The three plane projections of the transmission patterns. Observe for this special case that the central blue node believes its transmission sphere has been covered

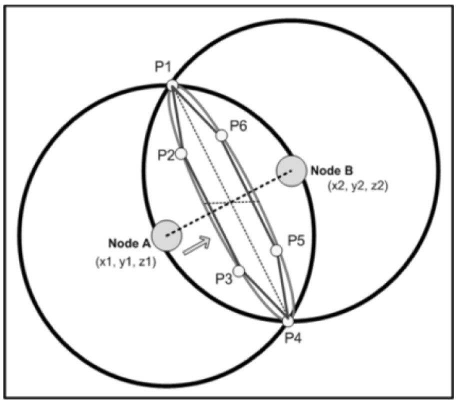


4.3 3D-ABBA2

For this version, we considered the circles of intersection between two transmission spheres. When a node *A* receives a transmission, a circumference of intersection is created. A circumference is a circle on its 3D perimeter. Instead of covering the 3D perimeter, we simplify the criterion to cover only such circumferences. The support for this simplification is given later in Theorem 1, where even covering these circumferences is reduced to covering certain points on them.

If after receiving several transmissions, each of the created circumferences is covered by other transmission spheres (defined by the nodes that transmitted the same message), then the transmission sphere of node *A* is completely covered by others and it will not transmit. We further simplify computationally this scheme, in two ways, leading to two protocols 3D-ABBA2 and 3D-ABBA3. They both consider certain points for coverage instead of the mentioned whole circumference. In 3D-ABBA2 variant, we propose to select six points forming a hexagon that resides in the intersection circle (circumference). Thus, instead of trying to cover the whole circumference, we propose to cover six points that reside on it. Figure 5 shows the circumference of intersection of two spheres and six selected points that reside on it: $P1$, $P2$, $P3$, $P4$, $P5$ and $P6$. These points form a regular hexagon. Receiving node A is located at ($x1$, $y1$, $z1$), and transmitting node B is located at ($x2$, $y2$, $z2$). We apply the following criterion at each node A. If, for every such intersection with other transmitting sphere B, each of six considered points $P1$–$P6$ is located inside another transmitting sphere (different from A and B), then the considered sphere A is covered in full. More precisely, let (d, e, f) be the coordinates of an intersection point of spheres centred at A and B. There must exist another transmitting node C with centre coordinates (xi, yi, zi) such that $(d - xi)^2 + (e - yi)^2 + (f - zi)^2 < R^2$. Note that different selected points may be covered by different transmitting nodes.

Figure 5 The circumference of intersection formed by the two transmission spheres of nodes A and B



Therefore a receiving node keeps a list of all the selected points. Whenever a new transmission is received, the following actions are performed:

- some existing selected points are inside the new transmission sphere. These points are eliminated. If no point remains, the sphere is completely covered
- the new sphere is considered, precisely the new six selected points. Each such point is tested whether it is inside another existing sphere (spheres of previous transmitters). Those that are inside another one are ignored. Those that are outside any existing sphere are entered into the list of points to be covered.

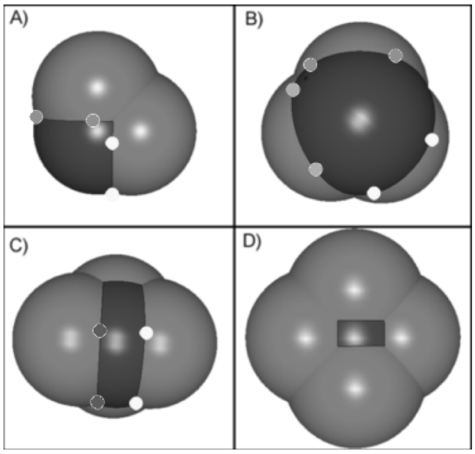
4.3.1 Timeout function for 3D-ABBA2

The timeout function in 3D-ABBA2 is fundamental. Estimating intersecting volumes might lead to a more precise timeout function, but the calculation is too complex to be really used. A timeout that only depends on the number of uncovered selected points will lead to simultaneous retransmissions, because of a few discrete possible values for such counter. 3D-ABBA2 uses an additional parameter to have a unique timeout function. This parameter is directly related to the volume covered by the transmission. That volume depends directly on the distance between the sender and the receiver. For this reason, we decided that each selected point has a weight d/R. The parameter d is the distance between the transmitter that created the point and the receiver, while R is the transmission radius. In conclusion, we propose a timeout function: *Timeout* = (number of received transmissions) + (sum of the weights of the selected points still to be covered).

4.3.2 Reliability of 3D-ABBA2

In all our experiments, 3D-ABBA2 was always reliable. However, this is not proof that it will always be the case. Unfortunately, there exist pathological cases where 3D-ABBA2 may fail to deliver to all the nodes connected to the source. In Figure 6 we show an example when this occurs. In this example, all the selected points are covered although part of the transmission sphere of the receiver is not completely covered.

Figure 6 Example when 3D-ABBA2 fails (intersection points of the same colour are created from the same sphere). Section A shows how two transmissions cover part of the sphere; the receiving node has some points left to be covered from each sphere. Sections B and C show three and four transmissions, respectively, with some points still to be covered. Section D shows that all points are covered, however, part of the receiving sphere still is not covered



One way of trying to achieve reliability would be to select more than six points; however, no matter how many points we select, the reliability cannot be guaranteed. One way to guarantee reliability would be to consider covering the whole circumference, instead of a finite selected number of points on it. This idea is explored further, and discretised, to develop 3D-ABBA3. The problem is that the six points are selected for coverage independently on other transmitting nodes. In 3D-ABBA3, we show how to select points to guarantee reliability.

4.4 3D-ABBA3

To design protocol 3D-ABBA3, we consider the intersections of three spheres (more precisely, their 3D perimeters) rather than two. One of these spheres is centred by node A making the retransmission decision. Thus the intersection points are on their 3D perimeter. The other two spheres are centred at two neighbouring nodes B and C that transmitted the message already. The intersection (if it exists) consists of two points.

If each such intersection point is located inside another transmitting sphere *D*, then the considered sphere is covered in full.

Therefore, each receiving node *A* keeps a list of all such uncovered intersection points. Whenever a new node *D* transmits a message in the neighbourhood, then the following events occur:

- some existing intersection points are inside the sphere centred at *D*. These points are eliminated
- the new sphere centred at *D* is considered, with the current fixed sphere centred at *A*, for intersections with any other transmitting sphere, to find new intersection points. Each such point is tested whether it is inside another transmitting sphere. Those that are inside are ignored. Those that are outside any existing sphere are entered into the list of intersection points to be covered.

When a transmission from a node *D* does not generate new intersection points, then the receiving node *A* adds node *D* to a list of transmitters that do not generate intersection points. We named this list as a list of 'singular' spheres. Every time a message is received, node *A* checks its list of singular spheres to see if these nodes now create intersection points with the last received transmission. When an intersection is created, the corresponding node is removed from the list of singular spheres and the generated points are inserted in the list of intersection points. If the timeout for node *A* expires and both lists (the list of intersection points and the list of singular spheres) are empty, then node *A* is fully covered, and decides not to transmit. We again propose two possible timeout functions. The first one involves using the intersections created by the transmitting nodes with the receiving nodes (transmission patterns).

Each time a receiving node *A* receives a transmission from a node *B*, node *A* applies the following algorithm:

BEGIN 3D-ABBA3 timeout Algorithm
IF the transmission done by *B* creates *n* valid intersection points THEN

$i = 1$;

WHILE $i \le n$ DO

C_i = node that created intersection point with nodes *A* and *B;*

$$timeout = timeout + \frac{\text{dist}(A,B) + \text{dist}(A,C_i)}{2R};$$

i++;

ENDWHILE

ELSE IF the transmission done by *B* does not intersect with other spheres THEN

$$timeout = timeout + \frac{\text{dist}(A,B)}{R};$$

ENDIF
END 3D-ABBA3 timeout Algorithm

The second function that we considered is a random timeout function with values between 0 and 1. As in the previous versions, we considered the second function to test its viability in already deployed IEEE 802.11 networks.

4.4.1 Reliability of 3D-ABBA3

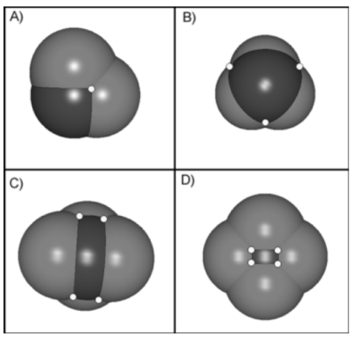
A node in 3D-ABBA3 will not transmit only when it is covered by other transmissions. In order to test coverage, a receiving node *A* looks its list of singular nodes (transmitting spheres that do not intersect with other transmitting spheres) and its list of intersection points (points generated by the receiving sphere and two transmitting spheres). If both lists are empty when the timeout expires, then node *A* will not transmit. The reliability follows from the following theorem.

Theorem 1: *Suppose that node A received transmissions from nodes C_1, C_2, …, C_m. Consider all the intersection points X of the 3D perimeters of the spheres centred at the three nodes: A, C_i and C_j. If there exists at least one such intersection point and every such intersection point X is located (strictly) inside at least of one of the remaining spheres, centred at C_k then the sphere centred at A is fully covered by the spheres centred at C_1, …, C_m, and node A does not need to retransmit, without affecting the reliability of broadcasting*.

Proof: Note that all the considered intersection points and all the intersection circumferences (between the sphere centred at *A* and any of the spheres centred at C_i for any *i*) are located on the 3D perimeter of the sphere centred at *A*. The conditions of the theorem are then topologically equivalent to an analogous theorem for the plane, applied, for example, in Zhang and Hou (2005) for sensor area coverage problems. Both theorems use circles as closed curves that are intersected, but the circles are located in the plane and on a 3D perimeter (sphere), respectively. The proof, in its simplified form, is by contradiction. Assume that the condition of the theorem is satisfied, but there is an area (on the considered 3D perimeter) that is still not covered. Let *Y* be a point in this area. Find the closest curve (circle) to *Y* from the set. This curve separates the covered and the uncovered regions. Follow this curve until another curve is met. The meeting point is an intersection point from the theorem. It is easy to see that this intersection point is not strictly inside other curve (circle), which is a contradiction.

Figure 7 shows an example of how the intersection points are generated. It is easy to observe from this example that as long as there is an uncovered surface, there will be uncovered intersection points. This guarantees that 3D-ABBA3 achieves complete reliability.

Figure 7 Intersection points created during 3D-ABBA3. The coverage of these points guarantees 3D-ABBA3 reliability



5 Comparison of 2D-ABBA with BPS and Geoflood

In this section, we describe the results of our comparison of 2D-ABBA with the Optimised Broadcast Protocol and with Geoflood. We implemented BPS with a threshold value $Th = 0.4\ R$ (as it was suggested by the authors). Geoflood was also implemented. In the literature review section we showed that Geoflood was unreliable. We also showed that, in order to be reliable, Geoflood has to divide the transmission area into six angles. For this comparison with 2D-ABBA, we implemented three versions of Geoflood: one dividing the transmission area into four angles (as the originally proposed by Arango et al. (2004)), other with divisions into five angles and finally one with division into six angles (the reliable version of Geoflood). For the tests, we considered a network of $n = 500$ static nodes, randomly distributed over an area of 100×100. In order to control the average node degree *d,* we sorted all $n(n - 1)/2$ (potential) edges in the network by their length, in an increasing order. The transmission radius R is equal to the $nd/2$th edge in the sorted array. We used Dijkstra's Shortest Path (SP) algorithm to test whether a graph was connected. We generated a total of 100 connected graphs for each of the following network degrees: $d = 7$, 8, 9, 10, 15, 20, 25, 30, 35, 40, 45, 50, 60, 70, 80, 90, 100 and 125.

After creating each of the connected graphs, we started a transmission from a randomly selected node, and we measured the following characteristics to broadcast a single message to the entire network using the original and the random 2D-ABBA, the beaconless version of OFP and the three mentioned versions of Geoflood:

- average number of transmissions
- average percentage of nodes receiving the message during the broadcast.

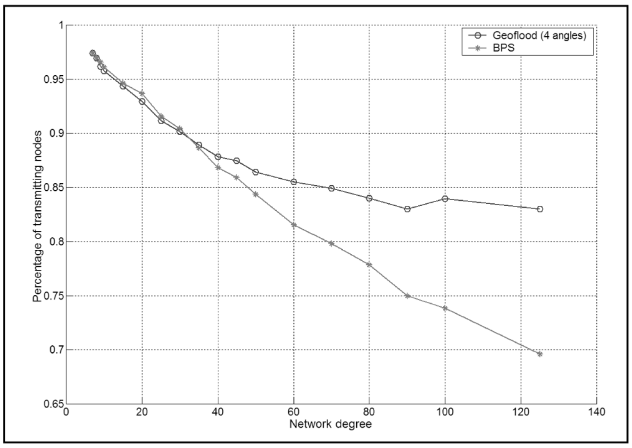
It is important to remark that BPS is not reliable. Even with high degree networks, BPS can fail as given in Table 1. For the case of Geoflood, almost all the tests were successful; in other words, the average percentage of nodes receiving the message during the broadcast was 100% in every test except one; only for a network with average degree 7, Geoflood with four angles division failed (once).

In Figure 8 we show the percentage of transmitting internal nodes. We define internal node as the node whose transmitting area or volume is covered by its neighbour nodes. In other words, if before performing a broadcast the transmission pattern of a node A is covered by the transmission patterns of its neighbours, then node A may or may not retransmit during the broadcast; however, if the transmission pattern of a node B is not covered by the transmission patterns of its neighbours then node B must retransmit during the broadcast. Such type of nodes will be called as external nodes.

Table I Percentage of unsuccessful broadcasts and average percentage of the nodes receiving the message when the broadcast failed using OFP in 100 tests

Net degree	*BPS unreliability*	
	% of failed broadcasts (number of tests out of the total 100 where a broadcast failed)	*Average % of nodes receiving the message when the broadcast was unsuccessful*
7	21	98.67
8	13	98.92
9	11	99.00
10	8	99.00
25	1	99.00

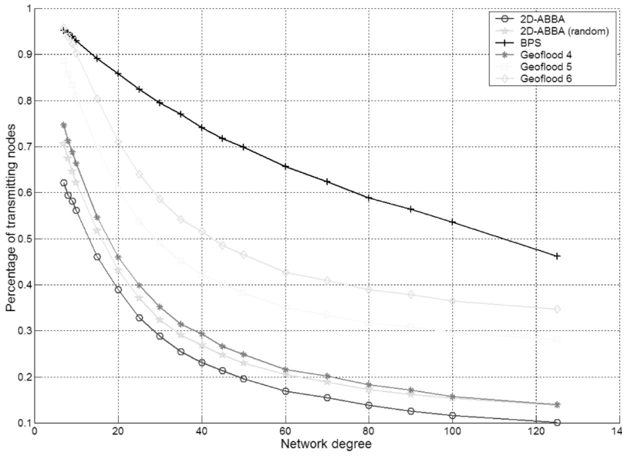
Figure 8 Percentage of transmitting nodes (internal nodes)



The average percentage difference between the two versions of 2D-ABBA was 4.55%, making us believe that the random version is a viable option to be deployed in IEEE 802.11 networks. The difference between the original 2D-ABBA and BPS was 44.24%. Here besides BPS performs worst, it also suppresses transmissions from external nodes, reaffirming its unreliability. The difference between 2D-ABBA and Geoflood of four, five and six angles was 6.83%, 20.74% and 29.22%, respectively.

In Figure 9 we show the percentage of transmitting external nodes. For this case the percentage was 100% for 2D-ABBA and for Geoflood of five and six angles. In this figure we show in other way how unreliable BPS is. Here we can observe that BPS causes that a considerable percentage of external nodes will not transmit, which can cause broadcast failures (if there were more neighbours in areas solely covered by such nodes). On an average BPS caused that 13.29% of the external nodes did not transmit during the broadcast. Even for high-connected networks (degree = 125) 30.41% of the external nodes did not transmit. These results support, that BPS can fail even in high-connected networks. In the case of Geoflood of four angles the percentage of external transmitting also was considerable. In average, 10.56% of external nodes did not transmit.

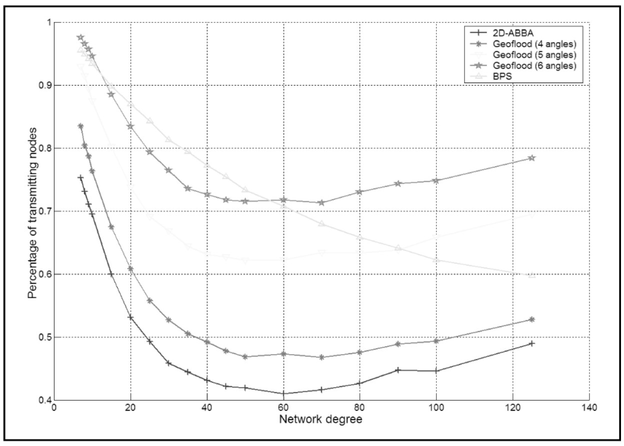
Figure 9 Percentage of transmitting nodes (external nodes)



5.1 *Impact of a realistic MAC layer*

In order to quantify the benefits of our new protocols and its impact in real environments, we repeated our experiments on a more realistic MAC layer, which considers message collisions. We have added a contention window of size CW and a timeout for each node before it can send any message. Any node randomly picks up an integer value between 0 and CW. Then, a node can neither receive two messages at the same time nor receive a message while transmitting. We selected a contention window of size 32, in accordance with the IEEE 802.11 standard. Figure 10 shows the percentage of internal transmitting nodes when the MAC layer was taken into account. As we can see from Figure 10 2D-ABBA performed better than Geoflood and BPS. This clearly shows great potential of 2D-ABBA to be deployed in IEEE 802.11 networks. Although BPS was the most unreliable algorithm, it presented less percentage of collisions as shown in Figure 11.

Figure 10 Percentage of transmitting nodes (internal nodes) with MAC layer



For low degree networks, the collisions contributed to the broadcast failure in all the algorithms. From degrees between 7 and 10, all algorithms failed in several occasions; however, for higher degrees only BPS failed at least in one occasion. Table 2 gives these results.

Figure 11 Percentage of collisions from transmitting nodes

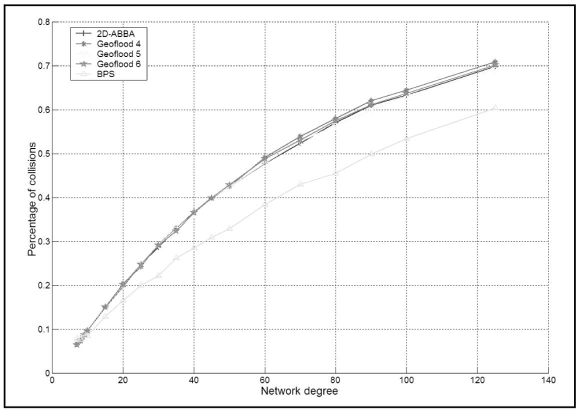


Table 2 Percentage of unsuccessful broadcasts

Degree	*BPS*	*Geoflood (4)*	*Geoflood (5)*	*Geoflood (6)*	*2D-ABBA*
7	15	34	38	32	36
8	12	21	25	21	19
9	8	17	16	13	14
10	6	13	6	12	9
15	0	2	0	1	0
20	0	0	0	0	0
25	1	0	0	0	0

6 Performance evaluation of the 3D-ABBAs

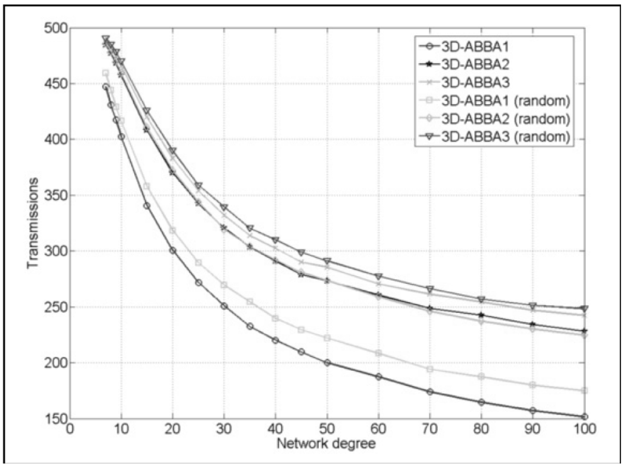
We considered a network of $n = 500$ static nodes, randomly distributed over a volume of $100 \times 100 \times 100$. In order to control the average node degree d, we sorted all $n(n-1)/2$ (potential) edges in the network by their length, in increasing order. The transmission radius R is equal to the $nd/2$th edge in the sorted array. We used Dijkstra's SP algorithm to test whether a graph was connected. We generated a total of 100 connected graphs for each of the following network degrees: $d = 7$, 8, 9, 10, 15, 20, 25, 30, 35, 40, 45, 50, 60, 70, 80, 90 and 100. We considered ideal MAC and physical layers.

After creating each of the connected graphs we started a transmission from a randomly selected node, and we measured the following characteristics to broadcast a single message to the entire network using 3D-ABBA1, 3D-ABBA2 and 3D-ABBA3:

- average number of transmissions
- average percentage of nodes receiving the message during the broadcast.

The percentage of nodes receiving the message during the broadcast was 100% in every test for all ABBAs, except for 3D-ABBA1 where we found two unsuccessful tests, and the average number of nodes that did not get the message during these failures was only one. This shows that although not all versions of ABBA are 100% reliable (as shown in previous sections), all versions performed extremely well on the reliability criterion. Figure 12 shows the average number of messages during the broadcast using all 3D versions of ABBA. We observe that among the 3D versions, 3D-ABBA1 performed the best in terms of number of messages needed to complete a broadcast. However, since its reliability is not 100%, we cannot immediately conclude that 3D-ABBA1 is the best protocol. The second best version was 3D-ABBA2; however, this is again, in theory, an unreliable protocol. The difference between 3D-ABBA1 and 3D-ABBA3 was on an average about 25% of the messages for the original timeout function and about 20% for the random timeouts.

Figure 12 Average number of transmissions of all 3D-ABBAs



Between 3D-ABBA3 and 3D-ABBA2, the difference was on an average about 3% of the messages for the original timeout function and for the random timeout versions, the difference was about 5% of the messages. The original timeout function in 3D-ABBA1 had about 10% fewer retransmissions than the random timeout function in 3D-ABBA1. However, when we applied random timeouts to 3D-ABBA2, we surprisingly obtained a minor improvement. In case of 3D-ABBA3, the difference was about 2% in favour of the original timeout function.

Figure 13 shows the percentage of transmissions done by the internal nodes in the network. Figure 14 shows the percentage of transmissions done by external nodes in 3D-ABBA1 and 3D-ABBA2. It is obvious that for reliable broadcasting algorithms, the percentage of transmissions of external nodes has to be 100%. This was the case for 3D-ABBA3 but not for 3D-ABBA1 and 3D-ABBA2.

Figure 13 Percentage of transmitting nodes (internal nodes)

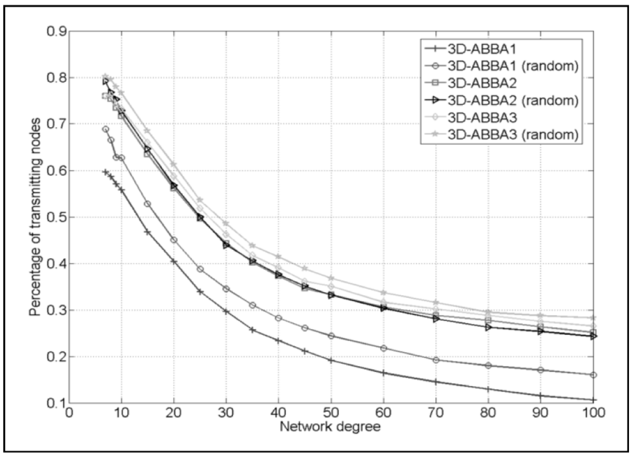
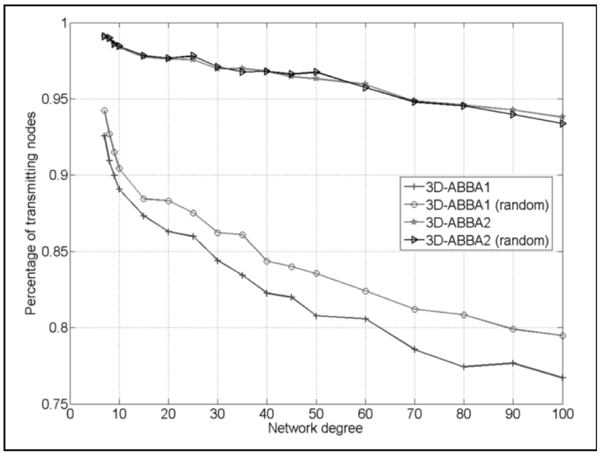


Figure 14 Percentage of transmitting nodes (external nodes)



Comparing the percentage of transmissions from internal nodes, 3D-ABBA1, 3D-ABBA2 and 3D-ABBA3 had a difference of 5.68%, 0.34% and 2.32% between its two timer functions, respectively. The difference between 3D-ABBA1 and 3D-ABBA2 was 15.11% when original timers were used and of 9.77% when random timers were tested. 3D-ABBA2 and 3D-ABBA3 had differences of 1.47% and 3.45%, respectively. When high degree networks were tested, only 30% of the internal nodes transmitted in 3D-ABBA3. This shows that 3D-ABBA3 achieves considerable energy savings while guaranteeing the reliable completion of a broadcast.

The average difference of the percentage of transmitting external nodes was 10.80% between 3D-ABBA1 and 3D-ABBA2 (original timer functions). When using random timers, the difference was of 12.88%. The average percentage of transmitting nodes (external) shows that although 3D-ABBA1 and 3D-ABBA2 are not reliable, they both guarantee that almost all external nodes will transmit. In 3D-ABBA1 the minimum percentage of transmitting external nodes was 77% but in 3D-ABBA2 this minimum value was 94%.

6.1 *Impact of a realistic MAC layer in the 3D-ABBAs*

As in the 2D algorithms, we simulated the effect of the MAC layer using discrete random timeouts between 0 and 32 (as in IEEE 802.11). If any packet was transmitted at the same time, then a collision occurred and the packet was lost. In Figure 13, we show the average percentage of internal transmitting nodes for all 3D-ABBAs. As it can be observed, the percentage of internal transmitting increases considerably when collisions affect the broadcasting process; however, we still achieve energy savings while we were able to complete a successful broadcast.

This shows that our 3D proposals have potential to be deployed in IEEE 802.11 networks.

7 Conclusions, future work and open ideas

ABBA was proposed as a broadcasting protocol for ad hoc networks. This algorithm has fewer assumptions than almost all broadcasting algorithms. Localised broadcasting algorithms normally assume the use of 'hello' messages to provide the list of *k*-hop neighbours. The use of such beacons can cause large communications overhead and delivery failures due to outdated information. Our protocols, however, find neighbours only when they are really needed, and rely on geographic location of the nodes.

We presented four beaconless broadcasting algorithms, one for networks that reside in the plane (2D), and three for networks that reside in the space (3D). All versions were based in setting timeouts before retransmitting; these timeouts depend on the transmission area still to be covered. We also tested the algorithms by setting their timeouts using a discrete random function, resembling the IEEE 802.11 functionality and simulating the MAC layer impact in our algorithms.

Several beaconless routing protocols were proposed by Ni et al. (1999) but they were not competitive in comparisons made in other papers. We propose 2D-ABBA as a new beaconless broadcasting algorithm in 2D, and compare it with two recent such protocols, BPS and Geoflood. We showed that BPS is not the best broadcasting protocol since it frequently lacks reliability. We showed that the Geoflood protocol, in order to be reliable, has to divide the transmission area into six angles instead of four. We implemented BPS and three versions of Geoflood in order to compare them with 2D-ABBA. 2D-ABBA had less message count than all versions of Geoflood and BPS.

The 3D-ABBA versions presented in this work are the first beaconless broadcasting algorithms proposed, other than blind flooding, to work in 3D networks. Although only one 3D-ABBA version (3D-ABBA3) is fully reliable in all scenarios, the other two 3D versions performed almost perfectly in the experimental tests. More precisely 3D-ABBA1 failed only in two tests, while 3D-ABBA2 was always reliable in our experiments. 3D-ABBA1 performed better than 3D-ABBA2, and 3D-ABBA3, in terms of the number of messages to complete a broadcast.

We also measured the percentage of external and internal transmitting nodes during a broadcast. We obtained that BPS suppressed a considerable percentage of the external transmitting nodes, while in fact, the percentage of the external transmitting nodes has to be 100% to guarantee reliability. This 100% was the case for 2D-ABBA, for Geoflood of five and six angles, and for both versions of 3D-ABBA3.

It is an interesting extension of this research to evaluate how the physical layer affects the performance of all beaconless algorithms. Recall that we used only ideal physical layer. That is, the message sent by any node is correctly received by all the neighbours located within the transmission radius. In reality, however, it is impossible for any antenna to have a perfect sphere as its transmission pattern. We believe that our 3D proposals set the basis for future research that takes into account the different shapes found in real 3D patterns. Another important open issue is the development of beaconless broadcasting protocols for networks whose nodes have directional antennas. Perhaps, ABBA versions could be modified to work with another transmission patterns, instead of circumferences and spheres or to work in networks where nodes have different transmission radii.

We were able to obtain excellent results with discrete random timeout variants. We believe that these show a very good promise to deploy our beaconless algorithms in IEEE 802.11 networks. The reason is that IEEE 802.11 uses discrete random timeouts, and the implementation of the random timeout ABBAs would be straightforward.

Finally, we note that, besides serving for broadcasting purposes, our algorithms could also be a viable option for routing purposes. Almost every routing protocol requires the interchange of routing tables or at least the interchange of neighbour information. When great mobility exists, a lot of energy is wasted only for these tasks. For this type of environments with significant node mobility, and small or perhaps medium number of nodes, one can apply ABBA as a routing algorithm.

Acknowledgements

This paper was supported by grants from NSERC, CONACyT (projecto 37017-A) and INRIA. The first author was supported by CONACyT and SEP scholarships during the development of this work.

References

Arango, J., Degermark, M., Efrat, A. and Pink, S. (2004) 'An efficient flooding algorithm for ad-hoc networks', *Proceedings of the Second Workshop on Modeling and Optimizations in Mobile Ad Hoc and Wireless Networks (WiOpt 2004)*, March.

Bergonovo, F., Capone, A., Cesana, M., Fratta, L., Coletti, L., Moretti, L. and Riato, N. (2003) 'Inter-vehicles communication: a new frontier of ad hoc networking', *Proceedings of the Medhoc*, Tunisia, June.

Blum, B., He, T., Son, S. and Stankovic, S. (2003) 'IGF: a state-free robust communication protocol for wireless sensor networks', *Technical Report CS-2003-11*, Department of Computer Science, University of Virginia, USA.

Calinescu, G., Mandoiu, I., Wan, P.J. and Zelikovsky, A. (2001) 'Selecting forwarding neighbors in wireless ad hoc networks', *Proceedings of DIAL M*.

Cartigny, J. and Simplot, D. (2003) 'Border node retransmission based probabilistic broadcast protocols in ad hoc networks' *System Sciences, Proceedings of the 36th Annual Hawaii International Conference*, 6–9 January, p.10.

Durresi, A., Paruchuri, V.K., Iyengar, S.S. and Kannan, R. (2005) 'Optimized broadcast protocol for sensor networks', *IEEE Transactions on Computers*, Vol. 54, No. 8, August.

Fussler, H., Widmer, J., Kasemann, M., Mauve, M. and Hartenstein, H. (2003) 'A novel forwarding paradigm for position-based routing (with implicit addressing)', *Computer Communications, CCW 2003, Proceedings of 2003 IEEE 18th Annual Workshop*, 20–21 October, pp.194–200.

Gerla, M., Kwon, T.J. and Pei, G. (2000) 'On demand routing in large ad hoc wireless networks with passive clustering', *Proceedings of the IEEE WCNC*, September.

Heinzelman, W., Kulik, J. and Balakrishnan, H. (1999) 'Adaptive protocols for information dissemination in wireless sensor networks', *MOBICOM*, Seattle, pp.174–185.

Heissenbuttel, M. and Braun, T. (2003) 'A novel position based and beaconless routing for mobile ad hoc networks', *Third IEEE Workshop on Applications and Services in Wireless Networks ASWN03*, Bern, Switzerland, 2–4 July.

Heissenbuttel, M. and Braun, T. (2004) 'BLR: beacon-less routing algorithm for mobile ad-hoc networks', *Computer Communications Journal*, Vol. 27, No. 11, pp.1076–1086.

Heissenbüttel, M., Braun, T., Wälchli, M. and Bernoulli, T. (2004) 'Broadcasting in wireless multihop networks with the dynamic forwarding delay concept', *Technical Report, IAM-04-010*, University of Bern, Switzerland, December.

Heissenbüttel, M., Braun, T., Wälchli, M. and Bernoulli, T. (2006) 'Optimized stateless broadcasting in wireless multi-hop networks', *IEEE Infocom (Infocom 2006)*, Barcelona, Spain, 23–29 April.

Ingelrest, F., Simplot-Ryl, D. and Stojmenovic, I. (2005) 'Energy-efficient broadcasting in wireless mobile ad hoc networks', in Cardei, M. Cardei, I. and Du, D-Z. (Eds). *Resource Management in Wireless Networking*, Springer, pp.543–582.

Kim, D. and Maxemchuk, N.F. (2003) 'A comparison of flooding and random routing in mobile ad hoc network', *Third New York Metro Area Networking Workshop*, September.

Li, X.Y. and Stojmenovic, I. (2006) 'Broadcasting and topology control in wireless ad hoc networks', in Boukerche A. (Ed). *Handbook of Algorithms for Wireless Networking and Mobile Computing*, Chapter 11, Chapman & Hall/CRC, pp.239–264.

Lin, N. and Shrivastava, S. (2003) 'System support for small-scale auctions', *Proceedings of the IFIP Medhoc*, Tunisia, June.

Ni, S.Y., Tseng, Y.C., Chen, Y.S. and Sheu, J.P. (1999) 'The broadcast storm problem in a mobile ad hoc network', *Proceedings of MOBICOM*, Seattle, August, pp.151–162.

Simplot-Ryl, D., Stojmenovic, I. and Wu, J. (2005) 'Energy efficient backbone construction, broadcasting, and area coverage in sensor networks', in I. Stojmenovic (Ed). *Handbook of Sensor Networks: Algorithms and Architectures*, Wiley, pp.343–379.

Stojmenovic, I., Seddigh, M. and Zunic, J. (2002) 'Dominating sets and neighbor elimination based broadcasting algorithms in wireless networks', *IEEE Transactions on Parallel and Distributed Systems*, Vol. 13, No. 1, pp.14–25.

Stojmenovic, I. and Wu, J. (2004) 'Broadcasting and activity scheduling in ad hoc networks', in S. Basagni, M. Conti, S. Giordano and I. Stojmenovic (Eds). *Mobile Ad Hoc Networking*, IEEE/Wiley, pp.205–229.

Sun, M., Huang, L., Wang, S., Arora, A. and Lai, T.H. (2003) 'Reliable MAC layer multicast in IEEE 802.11 wireless networks', *Wireless Communications and Mobile Computing*, Vol. 3, pp.439–453.

Sun, M.T. and Lai, T.H. (2001) 'Location aided broadcast in wireless ad hoc network systems', *Proceedings of IEEE Symposium on Ad Hoc Wireless Networks*, GLOBECOM, November.

Sun, M.T. and Lai, T.H. (2002a) 'Computing optimal local cover set for broadcasting in ad hoc networks', *Proceedings of IEEE ICC*, pp.3291–3295.

Sun, M.T. and Lai, T.H. (2002b) 'Location aided broadcast in wireless ad hoc network systems', *Proceedings of IEEE WCMC*.

Williams, B. and Camp, T. (2002) 'Comparison of broadcasting techniques for mobile ad hoc networks', *Proceedings of MobiHoc*, Lausanne, Switzerland, June.

Wu, J. and Li, H. (2001) 'A dominating set based routing scheme in ad hoc wireless networks', *Proceedings of DIAL M*, Seattle, August 1999, pp.7–14; *Telecommunication Systems*, Vol. 18, Nos. 1–2, pp.13–36.

Yi, Y., Gerla, M. and Kwon, T.J. (2003) 'Efficient flooding in ad hoc networks using on-demand (passive) cluster formation', *Proceedings of IFIP Medhoc*, Tunisia, June.

Zhang, H. and Hou, J.C. (2005) 'Maintaining sensing coverage and connectivity in large sensor networks', *Ad Hoc and Sensor Wireless Networks: An International Journal*, Vol. 1, Nos. 1–2, pp.89–124.