Broadcasting in Wireless Multihop Networks with the Dynamic Forwarding Delay Concept*

Marc Heissenbüttel

Torsten Braun

Markus Wälchli

Thomas Bernoulli

Technical Report
IAM-04-010

Institute of Computer Science and Applied Mathematics (IAM)
University of Bern
Switzerland

December 2004

*The work presented in this paper was supported (in part) by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 5005-67322.

Contents

Abstract

In this report we present a simple and stateless broadcasting protocol called Dynamic Delayed Broadcasting (DDB) which allows locally optimal broadcasting without any knowledge about the neighborhood. As DDB does not require any transmissions of control messages, it conserves critical network resources such as battery power and bandwidth. Local optimality is achieved by applying a principle of Dynamic Forwarding Delay (DFD) which delays the transmissions dynamically and in a complete distributed way at the receiving nodes such that nodes with a higher probability to reach new nodes transmit first. An optimized performance of DDB over other stateless protocols is shown by analytical results. Furthermore, simulation results show that, unlike stateful broadcasting protocols, the performance of DDB does not suffer in dynamic topologies caused by mobility and sleep cycles of nodes. These results together with its simplicity and the conservation of network resources, as no control message transmissions are required, make DDB especially suited for sensor and vehicular ad-hoc networks.

1 Introduction

Mobile wireless ad-hoc networks consist of a collection of wireless host which are free to move randomly. Nodes can communicate directly with all neighbor nodes within transmission range. As transmission ranges are limited, nodes have to cooperate to provide connectivity. Sensor networks and vehicular ad-hoc networks can considered as special types of ad-hoc network with some distinct characteristics. A very important characteristic of those networks is that the topology may change frequently and unpredictably either due to mobility or sleep cycles of nodes. Furthermore, in sensor networks the number of nodes may be significantly higher and nodes may be more prone to failures and have more strict constraints in terms of battery power, processing capabilities, and bandwidth. These two types of ad-hoc network also differ in their communication paradigm from a general ad-hoc network where traffic flows are almost uniformly distributed. In sensor networks, the information flow is mainly from and to one or few specific sinks, whereas in vehicular ad-hoc networks the packet may simply sent into a specific direction, e.g. along the road. In the remainder of this paper, we use ad-hoc network as an umbrella term including sensor networks and vehicular ad-hoc networks.

Broadcasting in ad-hoc networks is different from broadcasting in wired networks for several reasons. The network topology may change frequently caused by mobility or changes in the activity status of nodes. Broadcast protocols have also to cope with limited system resources such as bandwidth, computational and battery power. Unlike in wired networks where the total cost of the broadcast is normally just the sum of all link costs, ad-hoc networks can make use of the broadcast property of the wireless medium which allows all neighbor nodes to receive a packet with one single transmission. Thus, the costs are typically not associated with the links between nodes but with the nodes themselves.

Broadcasting in mobile ad-hoc networks is most simply and commonly realized by flooding where nodes broadcasts each received packet exactly once. Duplicated packets are detected e.g. by the source node ID and a sequence number. Assuming a completely connected network, there may be up to as many transmissions as nodes in the network. Especially in dense networks, flooding generates a large number of redundant transmissions where most of them are not required to deliver the packets to all nodes. Nodes in the same area receive the packet almost simultaneously such that the timing of the retransmissions is highly correlated. This excessive broadcasting causes heavy contention and collisions, commonly referred to as the *broadcast storm problem*, which consumes unnecessarily scarce network resources.

Two important objectives of any broadcast algorithms in ad-hoc networks are reliability and the optimization of resource utilization. First, reliability deals with the successful delivery of a packet to all nodes in the network. Even in a completely connected network, the packet may often not be delivered to all nodes since broadcast packets are normally not acknowledged and the broadcast storm makes the one-hop transmissions highly unreliable. Second, the use of network resources should be minimized without that reliability suffers. Interestingly, these objectives are often complementary. Minimizing the number of transmissions may also help reliability and decrease delay as it alleviates the broadcast storm.

It is impossible, or possible with a prohibitive amount of control traffic only, to broadcast network-wide optimally a packet. For example, to minimize the number of transmissions would require determining the minimal connected dominating set. Thus, most practical broadcast algorithms for ad-hoc networks try to approach network-wide optimal by locally optimal broadcasting of packets. This is commonly achieved by the proactive exchange of hello messages between neighbors such that nodes are aware of the network topology in their local neighborhood. However, this statefulness raises many critical issues such as the proactive use of network resources for control messages and the scalability in dynamic topologies. Another kind of broadcast protocols has also been proposed, which are stateless and do not require any knowledge about the neighborhood. They were shown to perform well in specific scenarios but very poorly in others, e.g. for varying node densities and traffic loads.

In this paper, we introduce the protocol DDB (Dynamic Delayed Broadcasting). DDB is stateless and completely localized. Thus, it does not cause any overhead and is highly scalable in dynamic networks. However, it does neither suffer from the drawbacks of other stateless broadcast algorithms, which is achieved by the use of the dynamic forwarding delay (DFD) concept. DFD allows nodes making locally optimal rebroadcasting decisions. Nodes decide whether to rebroadcast a message solely based on information available at the node itself and the information given in the broadcast packet, which are also used to compute a short delay before rebroadcasting packets by applying a DFD function. The concept of DFD supports the optimization for different metrics such as the number of retransmitting nodes, end-to-end delay, network lifetime, etc. and can take different parameters as input such as distance to other nodes, incoming signal strength, etc. We explicitly propose and evaluate in more detail DDB with four different DFD functions. The first two DFD functions aim at reducing the number of overall transmissions to deliver the packet to all nodes in the network. The first uses the distance to the previous transmitting node which allows estimating the additionally covered area, whereas the second uses the distance itself. The third DFD function has the same objective of minimizing the number of rebroadcasting nodes, but assumes that no location information is available and instead uses the power level of the incoming signal to approximate the distances between nodes. The fourth function addresses the problem of power consumption and aims at extending the network lifetime by favoring nodes which have more residual battery energy. We refer to DDB with one of these four specific DFD functions as DDB_{AC}, DDB_{DB}, DDB_{SS}, and DDB_{RB}, respectively. (AC, DB, SS, and RB stand for "Additional Coverage", "Distance-Based", "Signal Strength", and "Residual Battery", respectively.) DDB without subscript refers to the general DDB protocol without any explicit DFD function.

The remainder of this paper is organized as follows. An overview of related work is given in section 2. We describe the details of DDB in section 3. Analytical and simulation results are provided in section 4 and 5. Finally, section 6 concludes the paper.

2 Related Work

Many broadcast protocols have been proposed in order to cope with the broadcast storm problem and optimize broadcasting in ad-hoc networks. We first provide a brief taxonomy of existing broadcast algorithms for mobile ad-hoc networks. In a second step, we discuss some characteristics and encountered problems of broadcasting protocols and summarize the conclusions from several comparison studies. The concept of dynamic forwarding delay (DFD) used to achieve optimized broadcasting in this paper has also been applied to unicast routing protocols. At the end of the related work section, we briefly describe how DFD is used in these protocols.

2.1 Taxonomy

2.1.1 Simple Flooding

It was argued in [1] that flooding might be the only way to reliably deliver a message to every node in highly dynamic or very sparse networks. This does not only hold for broadcast transmissions, but also for multicast and unicast packets. In such environment, the overhead of other protocol may be even higher than of simple flooding, or they are not able to delivery the packet at all. Simple flooding may also be used just because for reasons of simplicity.

2.1.2 Probability-based approaches

In [2], each node rebroadcasts a message with a certain probability p and drops the packet with probability $1 - p$. If the probability to forward a packet is 1, this scheme is identical to simple flooding. [2] also proposed a counter-based scheme, where a node only rebroadcasts a message if it has received the message less frequently than a fixed threshold. In [3], the threshold is no longer fixed but adapts to the number of neighbors. [4] evaluated probabilistic broadcasting in more depth and proposed several extensions to the protocol of [2] based on the obtained results. The authors were able to improve the performance of their optimized protocols by accounting for nodes' neighbor counts and local congestion levels. These modifications require however additional transmission such as hello packets, called beacons, and an adaptive random delay. They also noted that these improvements have several drawbacks. First, the big advantage of these protocols, their simplicity, is negated. Secondly, if beacons are transmitted, the information about the local neighborhood could be employed in a more intelligent way like in the neighbor knowledge schemes as argued in [5]. In [6], the authors proposed to adjust the probability with which a node rebroadcasts a message depending on the distance to the last visited node. The distance between nodes is approximated by comparing the neighbor lists. Probability-based schemes were evaluated theoretically and by simulations in [7].

Several extensions have been proposed for these protocols to account for these circumstances by trying to dynamically adapt the threshold parameters depending on the encountered network conditions. For example, [4] proposes to compare the number of received messages to the number of neighbors. However, as stated in [5], this would require again knowledge of the neighborhood at the cost of additional transmissions. And still, the problem remains how many neighbors should rebroadcast in order to avoid a dying packet which is still depending on the global density of the network. Furthermore, if neighbor knowledge is available, protocols should not only use this information to adapt thresholds, but make more intelligent use of this information.

2.1.3 Location-based approaches

In the location-based schemes proposed in [2], the forwarding decision is solely based on the position of the node itself and the position of the last visited node as indicated in the packet header. Nodes wait a random time and only forward a message if the distance to all nodes from which they received the message is larger than a certain threshold distance value. The random waiting time is required to give nodes sufficient time to receive redundant packets and to avoid simultaneous rebroadcasting at neighbor nodes. The rational behind this is that only nodes which cover significantly large additional area rebroadcast the message. Instead of using the distance of nodes as a measure for the additional area covered, they also proposed an area-based method, which directly determines the possible covered area from the distances between nodes. In a second scheme, it was proposed to use signal strength to approximate distances.

2.1.4 Neighbor-designated approaches

Neighbor-designated schemes are characterized by the fact that nodes are aware of their neighborhood. The basic idea in all proposed approaches is that each node selects a set of forwarders among its one-hop neighbors such that the two-hop neighbors can be reached through the forwarders. A node only forwards packets from the set of neighbors out of which it was selected as forwarder thus reducing the total number of transmitted messages. In multipoint relaying (MPR) as described in [8], all two-hop neighbors should be covered by the selected one-hop forwarder. MPR is the broadcast mechanism used in the OLSR routing protocol as defined in RFC 3626 [9]. In [10], the set of forwarders also comprises all one-hop neighbors, which are not at least covered by two other forwarders. In [11] and [12] the set of forwarders was reduced by excluding the one-hop neighbors that were already covered by the node from which the broadcast packet was received. In [13], two-hop neighborhood information is piggybacked on packets and permits to eliminate the two-hop neighbors already covered by the last visited node. In [14], the set of forwarding nodes is selected from all neighbors with higher priority.

2.1.5 Self-pruning approaches

Unlike in the neighbor-designated method, each node decides for itself on a per packet basis if it should rebroadcast the packet. In [11], a node piggybacks a list of its one-hop neighbors on each broadcast packet and a node only rebroadcasts the packet if it can cover some additional nodes. Several of these approaches are based on (minimal) connected dominating sets. As the problem of finding such a set is proven to be NP-hard [15], several distributed heuristics are proposed. [16] proposed an algorithm, which only requires two-hop neighborhood information. A node belongs to the dominating set, if two unconnected neighbors exist. Furthermore, two rules are proposed to reduce the size of the connected dominating set, which requires an order on the IDs of the nodes. This idea was further improved in [17], where the degree of a node was used as primary metric instead of their IDs. The protocol proposed in [18] also relies on two-hop neighborhood information and assigns a priority to nodes proportional to the number of neighbors. Nodes with higher priority rebroadcast a packet first. A generic scheme was proposed in [19] based on two conditions, namely on neighborhood connectivity and history of the already visited nodes. In [20], it was shown that minimum latency broadcasting is also NP-hard and an algorithm was proposed where latency and the number of transmissions are bounded by a factor of the optimal values. To be able to cope more efficiently with mobility, [21] proposed to use two different transmission ranges for the determination of forwarders and for the actual broadcast process. In [22], connected dominating sets and the concept of planar subgraphs are combined to reduce the communication overhead for broadcast message in a one-to-one network model where each transmission is directed only towards one neighbor. A comprehensive performance comparison of various of these broadcast protocols based on self-pruning is given in [23].

2.1.6 Energy-efficient approaches

The problem of transmitting a message energy-efficiently to all nodes in the network where node have adjustable transmission radii was considered in several papers. [24] proposed an incremental power algorithm, which constructs a tree starting from the source node and adds in each step a node not yet included in the tree that can be reached with minimal additional power from one of the tree nodes. [25] considered the minimum energy broadcasting problem and proposed a localized protocol, where each node requires only the knowledge of the position of itself and the neighboring nodes. [26] showed the NP-completeness of minimal power broadcast. In [27] it was shown that that minimizing the total transmit power does not maximize the overall network lifetime. Note that energy efficiency is not necessarily directly related to network lifetime. If always the same nodes forward packets, broadcasting may be energy-efficient, but the battery at these nodes deplete quickly. In [27], the algorithm constructs a static routing tree, which maximizes network lifetime by accounting for residual battery energy at the nodes. A static tree does not change after the tree has been setup and, thus, does not really maximize the possible network lifetime, if nodes are mobile and routing can be dynamically adjusted. [28] presented a distributed topology control algorithm, which extracts network topologies that increase network lifetime by reducing the transmission power. A comparison of several power-efficient broadcast routing algorithms is given in [29].

2.1.7 Directional antenna-based approaches

Directional antennas can be used to improve the performance of broadcasting by reducing interferences, contention, etc. It was shown in [30] that MAC protocols, which utilize directional antennas can improve the performance of broadcast traffic in ad-hoc networks. In [31], nodes broadcast packets only into the opposite direction from which they were received. In [32], directional antennas are used to transmit in a one-to-one fashion broadcast packet to all neighbors. A comparison study of the performance of various directional antennas algorithms is provided in [33].

2.2 Discussion of Related Work

The probability- and location-based schemes, as well as simple flooding belong to the category of stateless algorithms as they do not require any neighbor knowledge. The neighbor-designated, the self-pruning, and the energy-efficient schemes all belong to the stateful protocols. They require at least knowledge about their one-hop neighbors, sometimes even global network knowledge is required. Comprehensive comparison studies were conducted in [5], [4], [23], and [29]. Their main conclusions can be summarized as follows:

Stateful protocols were found to be barely affected by high traffic loads and collisions. However, their performance suffers significantly in highly dynamic networks as the frequent topology changes induce an excessive, or even prohibitive, amount of control traffic, which occupies a large fraction of the available bandwidth. Furthermore, stateful algorithms may also never converge and reach a consistent state, if changes occur too frequently. Topology changes can not only be caused by mobility of the nodes but also by energy saving mechanisms, where nodes toggle between sleep and active modes. Their inability to cope with frequent topology changes together with the proactive transmissions of control messages, which wastes network resources, make stateful protocols unsuitable for certain kind of ad-hoc networks such as sensor and vehicular networks. The authors of [5] concluded that stateful protocols, more precisely neighbor-designated schemes, should be only used in semi-static or extremely congested networks.

On the other hand, it was shown that stateless algorithms are almost immune to frequently changing network topologies. Among the stateless schemes, the location-based methods performed best overall. The main drawbacks of stateless protocols and the reason why they were not recommended in [5] were found to be twofold. First, the number of rebroadcasting nodes is disproportionately high in networks with a high node density. Secondly, the random delay introduced at each node before rebroadcasting a packet is highly sensitive to the local congestion level. The main reason for this is that these stateless protocols use fixed parameters, e.g. the probability- or distance-threshold whether to rebroadcast a packet. They are highly sensitive to the chosen value and may perform well in some scenarios, and very poorly in others. For example, packets may either die out in sparse networks or the number of transmissions may not be reduced significantly in dense networks for too low and high parameters values, respectively. Energy-efficient schemes may not be suited for mobile networks with frequently changing topologies. They require a large computational and communication overhead to construct a power-efficient network structure. The overhead may be beneficial in a static network, where this structure has to be determined only once. In a mobile network, it may either not be possible to maintain this structure at all or only with a prohibitive amount of energy consumption.

We may conclude that stateless protocols would be a preferred choice for sensor networks, vehicular ad-hoc networks, and other ad-hoc networks with dynamic topology and/or strictly limited resources, if they could achieve nearly the same performance of stateful protocols. The

DDB protocol introduced in this paper is stateless and thus has all the aforementioned advantages of stateless protocols. DDB is not affected by changing topologies and does not require the proactive transmission of control messages, which saves scare network resources such as bandwidth and battery power. Unlike other stateless protocols, DDB allows making locally optimal rebroadcasting decisions by applying the concept of DFD such that "better" nodes rebroadcast first and suppress the transmissions of other neighbors. This dynamic process of distributed neighbor selection enables DDB to cope with a wide range of network conditions. In other stateless protocols, the sequence of rebroadcasting neighbors is random such that transmissions occur which are not necessary.

Our work is different in the following way from the work in [2] which also used location information for designing a broadcast algorithm: First, the timing of the rebroadcasting in DDB is not randomly, but nodes apply the concept of DFD to determine when to forward the packet which allows taking locally optimal rebroadcasting decisions without knowledge about the neighborhood. In [2], location information is used only to decide whether or not to rebroadcast. Second, DDB is designed with a cross-layer perspective in mind by coupling the MAC and network layer. This allows taking advantage of information only available at the network layer to more optimally schedule packets at the MAC-layer. Third, a common problem of broadcast protocols based on fixed parameters values and thresholds, i.e. which also occurs in [2] and other stateless protocols, is that they hardly can adapt to changing network conditions. Even though we also use a threshold in DDB to determine whether to rebroadcast a packet, we propose a different forwarding threshold policy which almost completely eliminates the drawbacks of fixed parameter. Forth, DDB is less sensitive to local congestion level which is an immediate consequence of the dynamic adjusted rebroadcasting. The motivation and justification for these changes are discussed in more detail below and will become evident in the simulation section. Fifth, DDB may be improved to extend the network lifetime by accounting also for the battery level of nodes in the forwarding decision. A further contribution of this report is the energy-based scheme DDB_{RB} which is to the best of our knowledge the first completely localized schemes which aims at extending the network lifetime. Most other energy-efficient protocols aim at reducing the energy to deliver the broadcast packet to all nodes in the network and/or adjust transmission power. However this may be complementary to the network lifetime in most scenarios [27].

2.3 Dynamic Forwarding Delay for Unicast Routing

Lately, several unicast routing protocols for ad-hoc and sensor networks have been proposed which adopt a new paradigm for position-based routing [34][35][36][37]. The next hop is not determined at the sender, but in a distributed way at the receivers. Nodes do not rely on information about neighbors anymore and allow disposing beaconing completely. These beaconless routing protocols exploit the broadcast property of the wireless medium to determine in a completely distributed way the next node after the packet has been transmitted. Any data packet is just broadcasted and all receiving nodes compete to forward the packet. Each node calculates an additional Dynamic Forwarding Delay (DFD) before forwarding the packet based on its position relative to its neighbors and the destination. The first node, which succeeds to transmit, suppresses the others. These protocols eliminate a lot of drawbacks of conventional position-based routing protocols which need beacons such as GFG [38], GOAFR [39], GPSR [40], but also cause new problems. For example, to assure mutual reception of relayed packets, only nodes within a certain area are potential forwarders.

3 Dynamic Delayed Broadcasting Protocol (DDB)

3.1 Introduction

We assume that nodes are either aware of their absolute geographical location by means of GPS or virtual coordinates as proposed lately in several papers, e.g. [41]. Many applications in sensor and vehicular ad-hoc networks already require per se location information. Thus, this location information available for free can be used to optimize lower network operations such as routing and broadcasting. In DDB, the last broadcasting node stores its current position in the header of the packet. This is the only external information required by other nodes in order to calculate when and whether to rebroadcast. Location information may not always be available. DDB can also operate without location information and use incoming signal strength to approximate the distance to other transmitting nodes.

For reasons of simplicity we do not consider the altitude, i.e. nodes are located in a two-dimensional plane. However it is not required that a node has any information about its neighborhood. Thus, no hello messages have to be transmitted periodically which saves scares resources like bandwidth and battery power. Even though, one may argue if location information is available, it will also be used for unicast routing and most position-based routing algorithm (e.g. GFG/GPSR [40], GOAFR [39]) require the periodical transmission of hello messages and knowledge about one-hop neighbors anyway. Thus this information can also be used without any additional overhead for broadcasting algorithms. First, lately there were several unicast routing protocols proposed which do no longer rely on neighbor information for forwarding as described in section 2.3. Furthermore, we may think of several applications where unicast routing is never required, and only broadcast and geocast communication occurs. For example in sensor networks or vehicular networks, messages often only need to be distributed to all nodes are some nodes within a certain area, called geocasting [42].

3.2 DDB_{AC} for Minimizing the Number or Transmissions

The objective of the first scheme DDB_{AC} is to minimize the number of transmissions and at the same time to deliver the packet reliably to all nodes. Nodes that receive the broadcasted packet use the concept of dynamic forwarding delay (DFD) to schedule the rebroadcasting and do not forward the packet immediately. From the position of the last visited node stored in the packet header and the node's current position, a node can calculate the estimated additional area that it would cover with its transmission. Depending on the size of this additionally covered area, the node introduces a delay before relaying the packet, where the delay is longer for a smaller additional area. In this way, nodes that have a higher probability to reach additional nodes broadcast the packet first. Note that this is achieved without nodes having knowledge about their neighborhood. Unlike in stateful broadcast algorithms, the "best" nodes for rebroadcasting are chosen in a completely distributed way at the receiving nodes and not at the senders. If a node receives another copy of the same packet and did not yet transmit its scheduled packet, i.e. the calculated DFD timer did not yet expire, the node recalculates the additional coverage of its transmission considering the previous received transmissions. As usually, a node is able to detect copies of a broadcast packet by their unique source ID and a sequence number. From the remaining additional area, the DFD is recalculated which is reduced by the time the node already delayed the packet, i.e. the time between the reception of the first and the second packet. For the reception of any additional copy of the packet, the DFD is recalculated likewise. A node does not rebroadcast a packet if the estimated additional area it can cover with its transmission is less than a *rebroadcasting threshold*, denoted as RT, which also may be zero. Obviously, DDB_{AC} can "only" take locally optimal rebroadcasting decisions as nodes do only receive transmissions from their immediate one-hop neighbors and thus have no knowledge about other more distant nodes possibly already covering partially the same area.

To illustrate the complete procedure of the algorithm, consider the example given in Fig. 1, where we assume a rebroadcasting threshold $RT = 0$. Furthermore, we do not account for propagation and processing delay. They are typically in the order of μs and negligible compared to the transmission delay and the delay introduced by DFD which are several orders of magnitude higher.

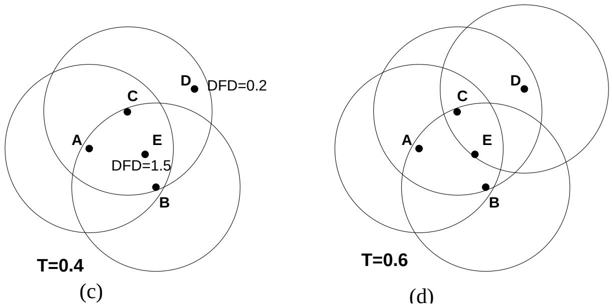


Figure 1: Example of the broadcast algorithm

Node A broadcasts a packet at time $T = 0.0\,ms$. The packet is received at neighbors B, E, C in Fig. 1(a). These nodes determine the size of the additional area they cover and introduce the additional delay accordingly. Say, node B, E, C calculate a DFD of $0.1\,ms, 0.2\,ms$ and $0.3\,ms$, respectively. Note that node C has no knowledge that there are two other neighbors which are located at a better position, i.e. calculate a smaller DFD. Similarly, neither have node B nor E. As node B introduces the shortest additional delay and consequently rebroadcasts the packet first after $0.1\,ms$ which is also overheard at node E, C in Fig. 1(b). Upon the detection of this transmission, they determine a new DFD depending on the remaining additional coverage. Thus, the new DFD of C will now be smaller than of E unlike before the transmission of node B. Assume that node E and C calculate a new DFD of $0.7\,ms$ and $0.4\,ms$ minus the $0.1\,ms$ they already delayed the transmission. Consequently, node C will rebroadcast the packet $0.3\,ms$ later in Fig. 1(c) already at time $T = 0.4\,ms$. Node D and E receive the packet and calculate the DFD as $0.2\,ms$ and $1.5\,ms$, respectively. Node D received the packet for the first time only now, but it still schedules the rebroadcasting much earlier, i.e. after $0.2\,ms$ than node E, which waits $1.5\,ms$ minus $0.4\,ms$ passed since the reception of the first copy of this packet. After node D transmitted the packet in Fig 1(d), node E drops the packet because it cannot cover any additional area. The dynamic calculation and recalculation of the DFD assures that always nodes that have a higher probability to reach new neighbors transmit first. As these nodes are located close to the transmission boundary, the calculated delay is short and the packet should be disseminated quickly within the network. In section 4, we will give some analytical results about the expected delay and additionally covered area by DDB_{AC}.

3.2.1 DFD function

The explicit DFD function is crucial to the performance of DDB_{AC} and should fulfill certain requirements in order to operate efficiently. The function should yield larger delays for smaller additional coverage and vice versa, if the objective is to minimize the number of transmissions. We assume the unit disk graph as the network model and thus a transmission range scaled to 1.

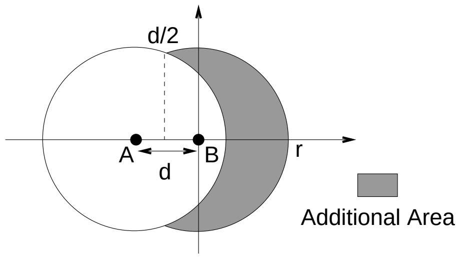


Figure 2: Additional covered area

Considering Fig. 2, we can determine the size of the additionally covered area AC of a node B's transmission if it is at a distance $d \in [0, 1]$ from the previous transmitting node A as follows.

$$AC(d) = 2 \cdot \left(\int_{-\frac{d}{2}}^{1} \sqrt{1 - x^2}\, dx - \int_{-\frac{d}{2}}^{-d+1} \sqrt{1 - (x + d)^2}\, dx \right)$$

which immediately yields

$$AC(d) = \frac{d}{2}\sqrt{4 - d^2} + 2 \arcsin\left(\frac{d}{2}\right) \tag{1}$$

The size of the additional covered area is maximal if node B is located just at the boundary of the transmission range of node A, i.e. if $d = 1$.

$$AC_{MAX} = \left(\frac{\sqrt{3}}{2} + \frac{\pi}{3} \right) \simeq 1.91$$

Consequently, one transmission can cover a maximum of $\frac{AC_{MAX}}{\pi} \simeq 61\%$ additional area which was not yet covered by the transmission of other nodes, i.e. at least already 39% were covered by other nodes' transmissions.

Taking into account this maximal AC_{MAX}, we propose a DFD function which is exponential in the size of the additional covered area as it was shown in [43] that exponentially distributed random timers can reduce the number of responses. Let AC denote the size of the additionally covered area, i.e. $AC \in [0, 1.91]$,

$$Add\_Delay = Max\_Delay \cdot \sqrt{\frac{e - e^{\left(\frac{AC}{1.91}\right)}}{e - 1}} \tag{2}$$

where Max_Delay is the maximum delay a packet can experience at each node. The DFD function is depicted graphically in Fig. 3 for a $Max\_Delay = 1$. We see that when nodes have a higher AC, the calculated DFD timers are distributed over a larger interval. Thus, the probability that a collision occurs at the first transmitting nodes, i.e. the ones close to the transmission boundary, is lower. The timers of nodes with only a small AC are closer to each other. However, as they transmit much later, they have received multiple transmission of other nodes and may not require to retransmit at all because $AC < RT$.

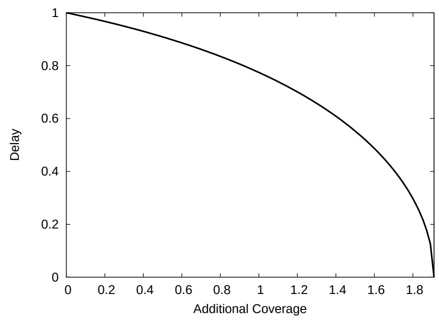
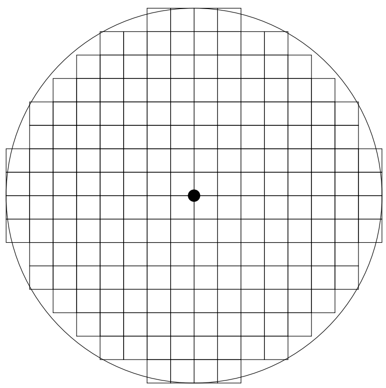
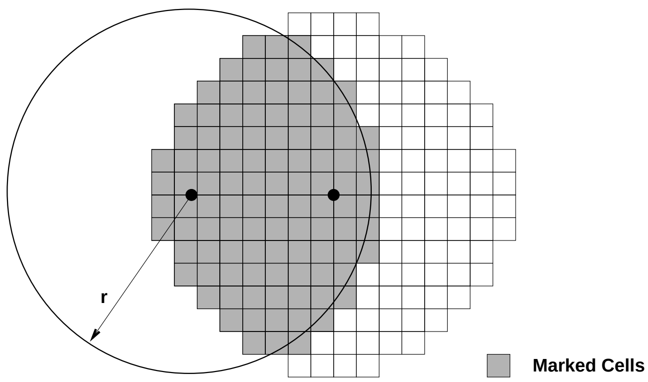


Figure 3: Delay introduced by the DFD function

Calculation of Additional Coverage The derivation of the additional area AC a node can cover with its transmission is easy to calculate for just one received packet. However, it gets more and more complicate when the node has to calculate AC after having overheard several copies which require to determine the intersection of several circles. We approximate AC in the following way. The transmission range is covered with a grid of square cells as depicted in Fig. 4(a).



(a) Grid of squares over the transmission range



(b) Marking of grid cells

Figure 4: Grid cells

The size of the square cells determines the accuracy of the approximation. Each node considers itself located at the origin of the coordinate system. When a node A receives a packet from a node B, it calculates that node's position relative to its position (x_r, y_r) and uses the circle disk inequality given in (3) to determine which of its grid cells are covered with B's transmission and marks the corresponding cells as shown in Fig.4(b).

$$(x - x_r)^2 + (y - y_r)^2 \leq r^2 \quad (3)$$

A node proceeds analogously for each subsequent received copy of the same packet and marks the unmarked cells which are covered by that transmission. Thus, each node can now easily determine AC by dividing the number of marked cells by the total number of cells. With a typical transmission radius of $250\,m$ and a grid square sizes of 5x5 m, the divergence is in the order of 1%.

3.2.2 DDB_{DB} based on Distances

Instead of using the additional covered area, which can be computational expensive, the distance between the transmitting nodes is used as an approximation of the likelihood to cover additional area. Each node keeps track of the minimal distance $d_{min} \in [0, 1]$ to all nodes from which it received a broadcast packet. After the first reception of a broadcast packet, d_{min} is just the distance to the last transmitting node. d_{min} is used to calculate the DFD similar like with the additional covered area. Unlike in the area-based variant of DDB_{AC}, the DFD is recalculated for each redundant received copy from a node which is closer than the currently stored d_{min}, i.e. each packet is considered separately.

$$Add\_Delay = Max\_Delay \cdot \sqrt{\frac{e - e^{\frac{d_{min}}{r}}}{e - 1}} \quad (4)$$

The *rebroadcasting threshold RT* is accordingly based on distances. A node with a d_{min} smaller than the rebroadcasting threshold does not rebroadcast the packet.

3.2.3 DDB_{SS} based on signal strength

Dynamic Forwarding Delay (DFD) may also be applied to optimize broadcasting in sensor and ad-hoc networks where nodes are not location aware. Instead of using the distance to the previous transmitting node as the input to the DFD function, nodes use the incoming signal strength. Packets received at higher power levels are delayed more as one may assume that the sender is located close by, i.e. for a higher signal strength, the DFD should calculate a larger additional delay as we may assume that we are close to the transmitting node, i.e. only cover few additional area. Signals can only be decoded if they are received above the receiver sensitivity. If the signal strength just equals the receiver sensitivity, the transmitting node is at the boundary of the transmission range. Thus, we may assume that it has a large additional coverage area and should retransmit quickly. For an attenuation factor a, a receiver sensitivity S_r, and a received power of P_r measured in dBm, we propose the following DFD function.

$$Add\_Delay = Max\_Delay \cdot \sqrt{\frac{e - e^{\sqrt[A]{10^{\left(\frac{S_r - P_r}{10}\right)}}}}{e - 1}} \quad (5)$$

Basically, (5) corresponds to (2) of the distance based DDB_{DB}, respectively. Typical IEEE 802.11b WLAN card have a transmission power P_t of about $15\,dBm$ and a receiver sensitivity S_r of $-81\,dBm$. These values are just exemplary and are not fixed. The transmission power is normally subject to regulatory limitations and may vary in different countries. The receiver sensitivity depends on the modulation scheme, i.e. on the data rate used, where lower data rates normally use more robust modulation schemes which can still be decoded at lower power levels, i.e. at higher distances.

Analogously, the *rebroadcasting threshold* is set to some signal strength value and a node only transmits a packet if it has not received any packet at a power level above this threshold. As the attenuation factor is normally not known, it has to be estimated. The more accurate the estimation of the attenuation factor is, the better will the performance be. An advantage of DDB_{SS} based on signal strength is that it is less sensitive to non-isotropic transmission ranges. If a node very close to the transmitting node receives a packet at a very low power level, we may nevertheless assume that it is at the boundary of the transmission range, e.g. due to a very high attenuation factor or a very power limited sender. Furthermore, nodes do not need to store their position in the packet header. This not only reduces the size of the packet and, thus, the energy to transmit and receive it, but also allows faster processing as packets remain unaltered through the whole broadcasting. Thus, no overhead and external information is required at all.

3.3 DDB_{RB} for Maximizing Network Lifetime

The objective of extending the network lifetime can be complementary to the objective of minimizing the number of transmissions to reach all nodes [27]. It may be beneficial that more nodes with a lot of residual battery energy broadcast a packet instead of fewer nodes with an almost depleted battery. In scenarios, where the source of the broadcast message is almost uniformly distributed over all nodes in the network or mobility is high and movement patterns are random, we may expect that the traffic load is also uniformly distributed over all nodes, and thus the battery will deplete roughly at the same time at all nodes. However, in many network environments, nodes rarely move and traffic flows are highly directed. This especially applies to sensor networks where all traffic is normally originating from or directed to one or few designated sinks and the mobility is rather low. If a deterministic algorithm is applied in such a scenario, which does not take into account the battery level at nodes, always the same nodes rebroadcast the packet. Consequently, some nodes will deplete much quicker than others.

In DDB_{RB}, the calculated delay by DFD depends solely on the residual battery level of a node and does not take into account the additionally covered area and the signal strength. They are only used to determine whether to rebroadcast a packet, i.e. whether they are smaller as RT. Nodes with an almost depleted battery schedule the rebroadcasting of the packet with a large delay whereas nodes with a lot of remaining battery power forward the packet almost immediately. Consequently, energy is conserved at almost depleted nodes, which increases their lifetime and in turn extends the connectivity of the network. Therefore, we simply adapt the DFD function to favor nodes with a lot of residual battery energy for rebroadcasting of packets. The DFD function introduces a small delay for nodes with a lot of battery energy whereas nodes with an almost depleted battery add a large delay. This is again done similar as in (2).

$$Add\_Delay = Max\_Delay \cdot \sqrt{\frac{e - e^{E_B}}{e - 1}} \tag{6}$$

E_B is the remaining battery power of a node as percentage of the total battery capacity. The possible benefit of such an energy-based scheme is highly depending on the MAC protocol and the ratio between the energy consumption of sending/receiving/idle listening. If idle listening consumes a substantial amount of energy compared to actual sending and receiving, all nodes spend their energy almost independently whether they forward packets or not. In scenarios, where either the MAC protocol puts a node into sleep mode to save energy or sending/receiving consume substantial more energy than idle listening, it is essential that the task of forwarding packets is fairly distributed among the nodes to maximize network lifetime even if traffic flows are spatially constant.

3.4 Effects of Irregular Transmission Range

Until now, we have just assumed simple propagation model such as the two-ray ground reflection model, which yields isotropic transmission ranges. This model is an oversimplification of the real world as transmission ranges are always isotropic and links are bidirectional. However, it is used in most paper including this one as it allows deriving some general properties of protocols. Other more complex network models probability closer match the real physical characteristics of a wireless network, but due to their complexity are often difficult to handle theoretically. Simple flooding, probabilistic schemes, and neighbor knowledge schemes should not suffer too much from non-isotropic transmission ranges. Neighbor knowledge methods are topology aware and the other algorithms do not rely on any external information at all. However, the efficiency of the DDB protocol, as well as the location-based protocols of [2], may be affected severely. The performance of these protocols depends on the accurate determination of the distance to the previous nodes and the additional coverage area. Consider the example given in Fig. 5 with three nodes where node A broadcasted a packet which was successfully received at node B and C. The lines indicate an equal mean power density which here just equals the receiver sensitivity.

If the three nodes assume a circular transmission range, and this is the only thing they can reasonably assume, node B rebroadcasts before node C. Taking into account the irregular transmission ranges, node C covers a larger additional area and thus should rebroadcast first which results in suboptimal rebroadcasting decisions of the algorithm. In case of irregular transmission ranges, the fixed transmission range value r used for the calculation of the DFD in DDB may be smaller than actual distance between the nodes. Thus, if the DFD functions yield a value greater as Max_Delay and smaller than 0, the values are simply set to Max_Delay and 0, respectively. In this paper, we also use the two-ray ground reflection model, but also have simulated the protocols with a more realistic propagation models to assess the impact on the protocols performance. This model yields highly irregular transmission ranges by taking into account non-isotropic path losses, continuous variation, and heterogeneous signal sending power.

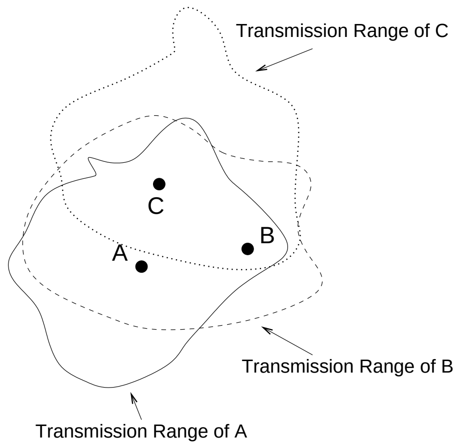


Figure 5: Irregular transmission ranges

3.5 Optimizations

3.5.1 "First Always" Forwarding Policy

A common problem of broadcast protocols based on fixed parameter values is that they are not able to cope with varying network conditions such as node density and traffic load [4]. DDB also uses a rebroadcasting threshold and thus would be susceptible to the same problem. A minor modification to the forwarding policy eliminates the problem almost completely. Nodes do always forward a packet, which is received exactly once after the DFD expires *independent* of the additional coverage, i.e. even if $AC < RT$. That means, the rebroadcasting threshold is only applied from the second received packet on. Especially in sparse networks, a node even with only very few additional area, may still be the only one to connect to other nodes and serve as the bridge to other node clusters. With this "first always" forwarding policy of DDB, the packet will be forwarded almost always in such scenarios and thus reduces the risk of packets dying out. At the same time there is only a small increase in the number of "unnecessary" transmissions compared to the case when the threshold is applied to all packets, including the first received packet. Particularly in dense networks, nodes overhear more than one copy and thus apply the threshold criterion, which prevents packets from being rebroadcasted.

3.5.2 Cross-Layer Information

Only the network layer, where DDB logically resides on, is able to interpret the payload of the packet such as source ID and sequence number, and thus detects that a just received packet is a redundant packet. As long as the packet has not yet been passed down to the MAC layer, this does not create a problem. The node simply either drops the packet if the threshold RT is exceeded or recalculates a new DFD for that packet. However, it may frequently happen that the packet is already forwarded to the MAC-layer. Two neighboring nodes normally receive the same broadcast packet almost simultaneously and may calculate nearly the same additional delay before rebroadcasting, i.e. because the have the same additional coverage. Thus, the packet is handed down to the MAC layer at about the same time and both nodes try to send the packet. The MAC layer is responsible to serialize the two transmissions. In this situation, a network layer protocol has normally no influence on the further processing anymore and thus, cannot prevent the second actually "unnecessary" of the two transmissions. DDB is able to access packets on the MAC layer, more precisely in the queue of the wireless interface and to reprocess them accordingly, i.e. either drop the packets or schedule their transmission for a later time.

3.5.3 Directional Antennas

As we have seen in section 3.2.1, already at least 39% of the transmission range of a node was covered by previous transmissions, often much more. Consequently, a transmission with an omnidirectional antenna radiates a lot of power unnecessarily into directions where no additional area can be covered. Directional antennas may mitigate this drawback by forming the beam only in directions of uncovered areas. Furthermore, for certain scenarios, the packet does not need to be broadcasted to all nodes in the network but only in some specific directions. In sensor networks, a request is sent into the network to collect some data from a specific region, thus, nodes distant from the target region broadcast the packet only to nodes in the corresponding direction and not to all neighbors. DDB could be further improved, if nodes are equipped with directional antennas, which is discussed at the end of this paper. Implementing DDB with directional antennas and a comparison with broadcast protocols, which make use of directional antennas, are outside of the scope of this paper and left for future work.

4 Analytical Assessment

We want to calculate the expected size of the additional area AC that is covered by a node's transmission with DDB_{AC}, i.e. nodes which cover more additional area broadcast the packet first to minimize the number of transmissions. We assume again a transmission radius of 1. In order to simplify the calculation, we compute the Taylor series expansion of the additional coverage $AC(d)$ as given in (1) with respect to the variable d about the point 0. The Taylor series expansion of a function $f(x)$ about a point $x = a$ is given by

$$f(x) = f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f^{(n)}(a)}{3!}(x-a)^3 + \dots$$

Thus, we obtain for $AC(d)$

$$AC(d) = d - \frac{1}{8}d^3 + \ldots + d + \frac{1}{24} + \ldots \simeq 2d \tag{7}$$

Let n indicate the number of neighbors and $X_i \in [0,1]$ be a random variable indicating the Euclidean distance of a neighbor $i \leq n$. We assume that nodes are independently and randomly distributed according to a two dimensional Poisson point process with constant spatial intensity. Thus, the X_i are identically and independently distributed and have the same cumulative distribution function (cdf) and probability density function (pdf). The cdf of the X_i can simply derived by dividing the area of the circle with radius x by the size of the whole transmission range, which is π. Thus, we obtain for the cdf F_X and pdf f_X with $0 \leq x \leq 1$.

$$F_X(x) = P(X \leq x) = x^2 \qquad f_X(x) = 2x$$

From probability theory, we know that for a random variable $V = g(U)$ as a function of a random variable U, the pdf f_V of V can be derived from g and the pdf f_U of U as follows

$$f_V(x) = f_U[g^{-1}(x)]\frac{d}{dx}g^{-1}(x)$$

Thus, for a random variable Y, which indicates the additional area covered by a node's transmission, given as $Y = g(X) = 2X$ by the approximation of the distance (7), the pdf f_Y of Y can be calculated as follows.

$$f_Y(x) = f_X[g^{-1}(x)]\frac{d}{dx}g^{-1}(x) = \frac{x}{2} \qquad \text{with } 0 \leq x \leq 2 \tag{8}$$

Thus, the cdf the additional coverage of a nodes transmission is simply

$$F_Y(x) = \frac{x^2}{4} \qquad \text{with } 0 \leq x \leq 2 \tag{9}$$

In order to derive the expected additional coverage of each of the n neighbors, we sort their additional coverage Y_i such that $Y_{(1)} \leq Y_{(2)} \leq \ldots \leq Y_{(n)}$. Thus Y_i is only the same as $Y_{(i)}$ with probability $\frac{1}{n}$ and the sample maximum and minimum are $Y_{(n)}$ and $Y_{(1)}$, respectively. Obviously, the k-most distant neighbor has also the k-largest expected additionally covered area. The general cumulative distribution function cdf $F_{Y_{(k)}}(x)$ for all $Y_{(k)}$ is given by

$$\begin{aligned} F_{Y_{(k)}}(x) &= P(Y_{(k)} \leq x) \\ &= \sum_{j=k}^{n} P(Y_{(j)} \leq x) \\ &= \sum_{j=k}^{n} P(\text{Exactly } j \text{ of the } Y_i \leq x) \\ &= \sum_{j=k}^{n} \binom{n}{j} [F_Y(x)]^j [1 - F_Y(x)]^{n-j} \end{aligned}$$

where $F_Y(x)$ is the cdf of the Y_i as given in (9).

The derivation $f_{Y_{(k)}}$ of $F_{Y_{(k)}}$ with respect to x can be calculated straightforward.

$$f_{Y_{(k)}}(x) = \frac{d}{dx}F_{Y_{(k)}}(x) = \frac{d}{dx}\left[\sum_{j=k}^{n} \binom{n}{j} [F_Y(x)]^j [1 - F_Y(x)]^{n-j}\right] =$$

$$\frac{d}{dx}\left[\binom{n}{k}[F_Y(x)]^k[1-F_Y(x)]^{n-k}+\binom{n}{k+1}[F_Y(x)]^{k+1}[1-F_Y(x)]^{n-k-1}+\ldots\right.$$
$$\left.+\binom{n}{n-1}[F_Y(x)]^{n-1}[1-F_Y(x)]^1+\binom{n}{n}[F_Y(x)]^n[1-F_Y(x)]^0\right]$$

and thus

$$\begin{aligned}
&f_{Y_{(k)}}(x) = \\
&\binom{n}{k}\left[kF_Y(x)^{k-1}f_Y(x)\left(1-F_Y(x)\right)^{n-k}-(n-k)F_Y(x)^k\left(1-F_Y(x)\right)^{n-k-1}f_Y(x)\right]+ \\
&\binom{n}{k+1}\left[(k+1)F_Y(x)^k f_Y(x)\left(1-F_Y(x)\right)^{n-k-1}-(n-k-1)F_Y(x)^{k+1}\left(1-F_Y(x)\right)^{n-k-2}f_Y(x)\right] \\
&+\ldots+\binom{n}{n-1}\left[(n-1)F_Y(x)^{n-2}f_Y(x)\left(1-F_Y(x)\right)-F_Y(x)^{n-1}f_Y(x)\right]+ \\
&\binom{n}{n}\left[nF_Y(x)^{n-1}f_Y(x)\right]
\end{aligned}$$

Expanding the terms yields

$$\begin{aligned}
f_{Y_{(k)}}(x) \quad = \quad & \frac{n!}{k!(n-k)!}kF_Y(x)^{k-1}f_Y(x)(1-F_Y(x))^{n-k}- \\
& \frac{n!}{k!(n-k-1)!}F_Y(x)^k(1-F_Y(x))^{n-k-1}f_Y(x)+ \\
& \frac{n!}{(k+1)!(n-k-1)!}(k+1)F_Y(x)^k(1-F_Y(x))^{n-k-1}f_Y(x)+\ldots \\
& n(n-1)F_Y(x)^{n-2}f_Y(x)(1-F_Y(x))- \\
& nF_Y(x)^{n-1}f_Y(x)+nF(x)^{n-1}
\end{aligned}$$

what eventually simply yields

$$f_{Y_{(k)}}(x)=\binom{n}{k}kF_Y(x)^{k-1}f_Y(x)\left(1-F_Y(x)\right)^{n-k}$$

From (9), we have $F_Y(x)=\frac{x^2}{4}$ and obtain

$$f_{Y_{(k)}}(x)=\binom{n}{k}k\left(\frac{x^2}{4}\right)^{k-1}\frac{x}{2}\left(1-\frac{x^2}{4}\right)^{n-k} \qquad \text{with } 0\le x\le 2$$

It is well-known that the expected value of a random variable Z can be calculated from its pdf f_Z by

$$E_Z=\int_{-\infty}^{\infty}xf_Z(x)\,dx \tag{10}$$

Therefore, we obtain the expected value $E_{AC}^{Y_{(k)}}$ for the additional coverage for the k-most distant neighbor solely depending on the number of neighbors n as follows.

$$\begin{aligned}
E_{AC}^{Y_{(k)}} &= \int_0^2\binom{n}{k}k\,\frac{x}{2}\,x\left(\frac{x^2}{4}\right)^{k-1}\left(1-\frac{x^2}{4}\right)^{n-k}dx \\
&= 2\binom{n}{k}k\int_0^2\left(\frac{x^2}{4}\right)^{k}\left(1-\frac{x^2}{4}\right)^{n-k}dx
\end{aligned} \tag{11}$$

In order to calculate this integral, we use the beta function $B(p, q)$, which is defined by

$$B(p,q) = \frac{\Gamma(p)\Gamma(q)}{\Gamma(p+q)}$$

and can be expressed as

$$B(p,q) = \int_0^1 u^{p-1}(1-u)^{q-1}\,du$$

To put it in the form we need it, let $u = \frac{x^2}{4}$ and $du = \frac{1}{2}x\,dx$, and

$$\begin{aligned}
\frac{\Gamma(p)\Gamma(q)}{\Gamma(p+q)} &= \int_0^1 u^{p-1}(1-u)^{q-1}\,du \\
&= \frac{1}{2}\int_0^2 \left(\frac{x^2}{4}\right)^{p-1}\left(1-\frac{x^2}{4}\right)^{q-1} x\,dx \\
&= \int_0^2 \left(\frac{x^2}{4}\right)^{p-\frac{1}{2}}\left(1-\frac{x^2}{4}\right)^{q-1} dx
\end{aligned}$$

Together with (11), this yields

$$E_{AC}^{Y_{(k)}} = 2\binom{n}{k} k\,\frac{\Gamma(n-k+1)\Gamma\left(k+\frac{1}{2}\right)}{\Gamma\left(n+\frac{3}{2}\right)}$$

and by using $\Gamma(n) = (n-1)!$, we finally obtain

$$E_{AC}^{Y_{(k)}} = \frac{2\Gamma(n+1)\Gamma\left(k+\frac{1}{2}\right)}{\Gamma(k)\Gamma\left(n+\frac{3}{2}\right)} \tag{12}$$

We compare this result with the expected additional coverage E_{AC}^* of other stateless broadcasting schemes where the sequence of neighbors' transmission is independent of their additional coverage, e.g. as in the location-based and probability-based schemes. Clearly, the pdf f_Y of the additional coverage for a single node is the same as derived before in (8). However, the expected additional coverage is independent of the number of neighbors n and the same for all neighbors $k \leq n$ and, thus, is constant. Again with (10), we obtain

$$E_{AC}^* = \int_0^2 x\,\frac{x}{2}\,dx = \frac{4}{3}$$

In Fig. 6, the graph is plotted for $E_{AC}^{Y_{(k)}}$ of DDB_{AC} and E_{AC}^* of other stateless broadcasting algorithms depending on the number of neighbors for $n = 1 \ldots 30$. Again, $k \leq n$ denotes the k-most distant neighbor, i.e. the node with the k-largest additional coverage. E_{AC}^* is simply the plane at $\frac{4}{3}$. Already for very few neighbors, the "best" node, i.e. $k = n$, already covers almost the maximum size of additional area of 1.91. Furthermore, also the next $k \leq n$-best nodes cover normally more than $\frac{4}{3}$ what would be covered by a node's transmission with other stateless broadcasting schemes. Assuming the same rebroadcasting threshold RT for DDB_{AC} and the other location- and probability-based schemes, we can conclude that we might expect an improved performance up to $43\% = \frac{1.91}{4/3}$ in terms of transmissions. However, the advantage of DDB_{AC} is not only the reduction in number of transmissions, but also that the delay can be reduced as distant node which transmit first almost add no delay. From the expected additional

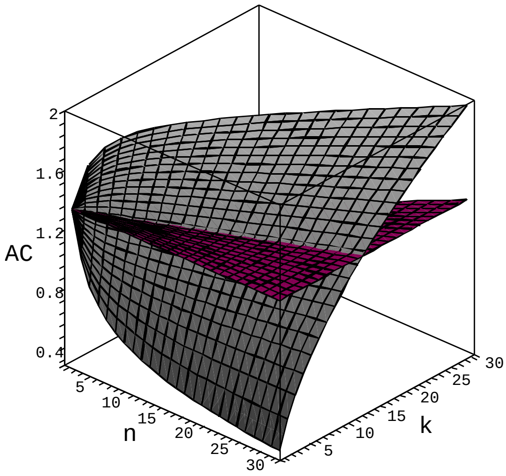


Figure 6: Expected additional coverage

coverage in (12) and the DFD function (2), we can easily determine the expected additional delay introduced at the nodes.

$$E_{AD}^{Y_{(k)}} = Max\_Delay \cdot \sqrt{\frac{e - e^{\left(\frac{\frac{2\Gamma(n+1)\Gamma\left(k+\frac{1}{2}\right)}{\Gamma(k)\Gamma\left(n+\frac{3}{2}\right)}}{1.91}\right)}}{e-1}}$$

The results are depicted in Fig. 7 for a Max_Delay of 1. Most nodes, which broadcast, i.e. where k is in the order of n, delay the transmission only by a small fraction of the total Max_Delay.

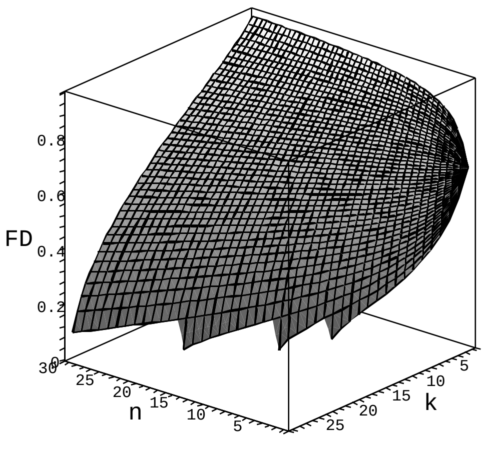


Figure 7: Expected delay through DFD

Furthermore with DDB_{AC} we know that nodes which cover more additional area broadcast first and thus can design the DFD accordingly, which allows reducing the number of collisions.

In other stateless schemes, the delay has to be much larger to have the same number of collisions than in DDB_{AC} as neighbors transmit randomly. As it is difficult to asses the exact influence of the MAC layer and to take into account the dependencies between neighboring nodes when their transmission ranges overlap, these analytical only provide a kind of rough bound for the performance. Obviously, the values only are only correct when rebroadcasting neighbor nodes' transmission ranges do not overlap. However, this will never be the case as there are a maximum of three non-overlapping neighbor node's transmission ranges, namely when the nodes are located at the boundary of the source node's transmission range at a distance of 2. Thus, only the expected additional coverage for the first transmitting node is is exact. The derived values are slightly too high for the subsequent transmitting nodes. The values are more and more overestimated when more node transmit and more transmission ranges will overlap. Still, the general conclusions of the analytical results hold and will be validated in the next section by simulations.

5 Simulations

5.1 Protocols

The DDB protocol was implemented with two optimizations proposed in section 3.5, namely the "first always forwarding policy" and the "cross-layer information". However, we did not use directional antennas for DDB as the other protocols were not optimized for the use with directional antennas. The performance of DDB is compared to three protocols described in section 2: The location-based broadcasting protocol [2], which is abbreviated by LBP in the following, the multipoint relay MPR [8], and simple flooding as the most simple broadcasting protocol. LBP and MPR were chosen as representatives for the categories of stateless and stateful broadcast protocols, respectively. However, we did not use any energy-efficient and directional antenna based algorithms for comparison as they either use adjustable transmission power and transmission directions, respectively.

The parameters of LBP and MPR are set as suggested in [5] and in RFC 3626 [9], respectively. Specifically, the random delay at each node for LBP is set to $10\,ms$ and the rebroadcasting threshold to 40% of the maximal additional covered area. The hello message interval and neighbor hold time are $2\,s$ and $6\,s$ respectively for MPR. For the flooding, the packets are jittered $2\,ms$ to avoid that all neighbors transmit simultaneously.

5.2 Simulation Parameters and Quantitative Performance Metrics

We implemented and evaluated the protocols in the Qualnet network simulator [44]. The results are averaged over 10 simulation runs and given with a 95% confidence interval, which is sometimes very small and barely visible. The payload of the packets is 64 bytes and the interface queue length is set to 1500 bytes. Radio propagation is modeled with the isotropic two-ray ground reflection model. The transmission power and receiver sensitivity are set corresponding to a nominal transmission range of 250m. We use IEEE 802.11b on the physical and MAC-layer operating at a rate of 2 Mbps. The simulations last for 900s and data transmission starts at 180s and ends at 880s such that emitted packets arrive at the destination before the end of the simulation.

The performance of the different protocols was measured in terms of quantitative metrics. The delivery ratio is measured in the number of nodes which actually received a specific broadcast packet divided by the total number of nodes in the network. The second metric is the number of rebroadcasting nodes. Since each node transmits a received packet exactly once, the number of rebroadcasting nodes corresponds exactly to the total number of transmissions in the network. This is unlike for unicast transmissions where retransmission might occur on the MAC layer if a packet is lost during transmission. Obviously, many control messages are additionally required in MPR. These broadcast transmissions are not counted. The end-to-end delay is simply the latency between the moment when the packet is transmitted from the source and the time the last node received it.

5.3 Evaluating different versions of DDB

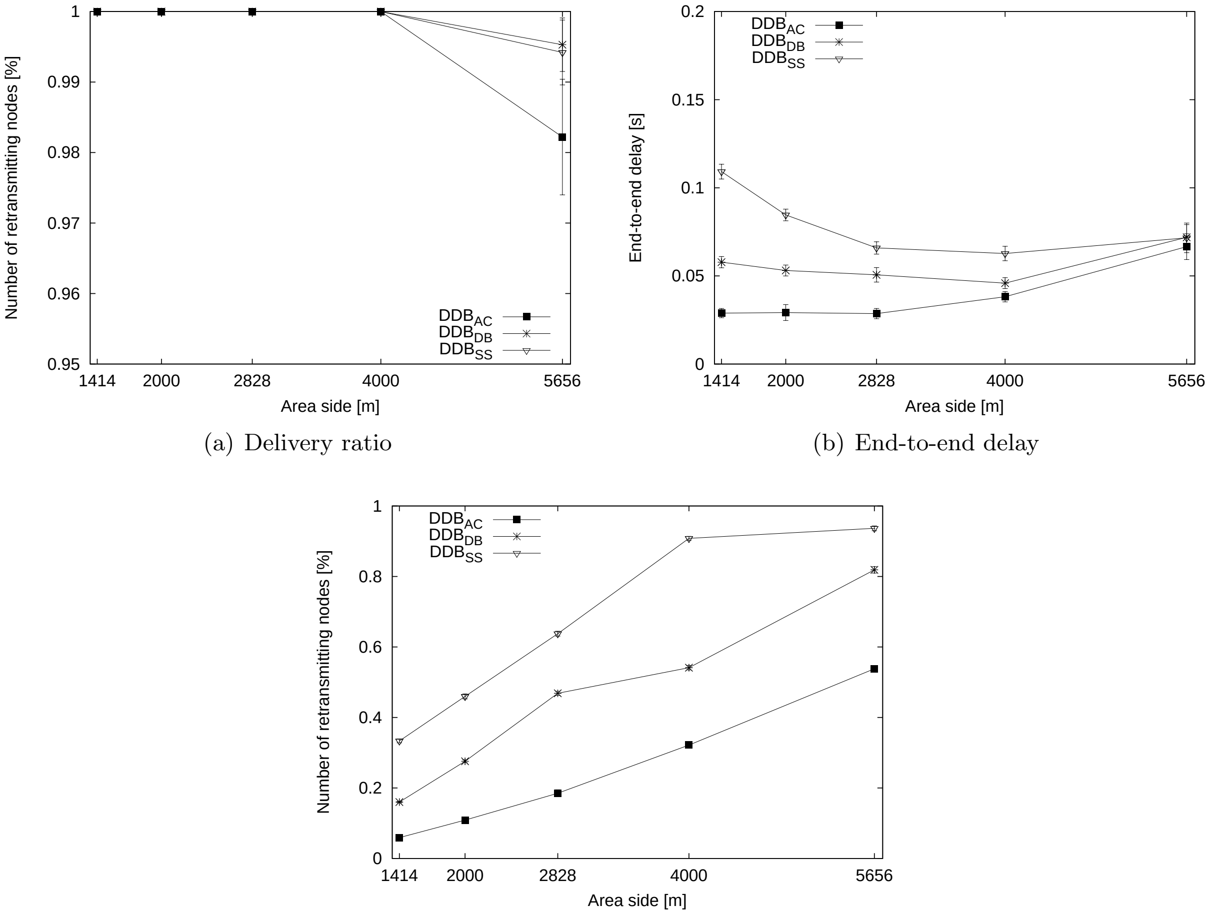
We simulated DDB in various scenarios to study the effect of the different components and also to determine appropriate values for the rebroadcasting threshold RT and the Max_Delay.

5.3.1 The versions to minimize the number of transmissions

We proposed three versions of DDB, which all have the same objective of reducing the number of rebroadcasting nodes, namely $\mathrm{DDB_{AC}}$, $\mathrm{DDB_{DB}}$, $\mathrm{DDB_{SS}}$. In this subsection, we compare their relative performance. The rebroadcasting thresholds RT are set to 40% of the maximum used for the respective DFD functions. The selection of such a high value will be motivated in the subsection 5.3.4. The thresholds for the area- and distance-based versions are easy to determine. For $\mathrm{DDB_{AC}}$ and $\mathrm{DDB_{DB}}$, RT was set to 40% of the maximal area a node can cover and 40% of the maximal transmission radius, respectively. For a transmission radius of $250\,m$ this yields $0.4 \cdot 1.91 \cdot (250\,m)^2 \simeq 47750\,m^2$ for $\mathrm{DDB_{AC}}$ and $0.4 \cdot 250\,m = 100\,m$ for $\mathrm{DDB_{DB}}$. The threshold for the signal strength version requires some calculations and further assumptions about the typical attenuation factor. The attenuation factor in real physical environments is about 2 for free space and may raise up to 6 for indoor environment. We choose an average attenuation factor of $a = 3$ to roughly estimate the distance between nodes in the signal strength version $\mathrm{DDB_{SS}}$. We set the threshold RT to $S_r + 12\,dBm$. The value is motivated by the fact that nodes with 40% additional coverage are at a distance of approximately $100\,m$. This is 2.5 times closer to the source than a node at $250\,m$, which receives a packet just at S_r and has the maximal additional covered area. Assuming an average attenuation factor of 3, this immediately yields that the signal strength at a distance of $100\,m$ is $10 \cdot \log_{10}\left(2.5^3\right) = 12\,dBm$ stronger than S_r. Obviously, we could derive exact distances from signal strengths as the underlying propagation model is known. Thus, the performance of the $\mathrm{DDB_{DB}}$ and $\mathrm{DDB_{SS}}$ would be the same. In reality, the attenuation factor is not known and can only be estimated. Therefore, we did also not use the exact attenuation factor used in the two-ray ground reflection propagation model which is 2 until a certain distance and 4 afterwards.

The simulations were conducted in a static network without any congestion as we wanted to compare the efficiency of the core algorithms and excluded any external influences. Thus, only one source broadcasts one packet per second. We placed 1000 nodes randomly over a square area with side lengths of $1414, 2000, 2828, 4000, 5656\,m$ to obtain different node densities. The density is always doubled for the next smaller area size and equals approximately 6,12,24,49, and 98 neighbors per node. The least node density of 6 neighbors was chosen as results from percolation theory have shown [45] that 6 neighbors is is just about the minimal required density for a completely connected network. For lower node density, the network is almost always disconnected. However, it may still happen that the network is not completely connected with only 6 neighbors and that a packet cannot be delivered to all nodes. To eliminate this bias of the results, we implemented an algorithm to determine the size of the maximal connected cluster which includes the source node. The delivery ratio and the number of rebroadcasting nodes are calculated relatively to the size of that cluster.

The delivery ratio is almost always 100% as shown in Fig. 8(a), except for very sparse networks where all protocols suffer slightly. $\mathrm{DDB_{AC}}$ has the lowest delivery ratio, even though



(a) Delivery ratio

(b) End-to-end delay

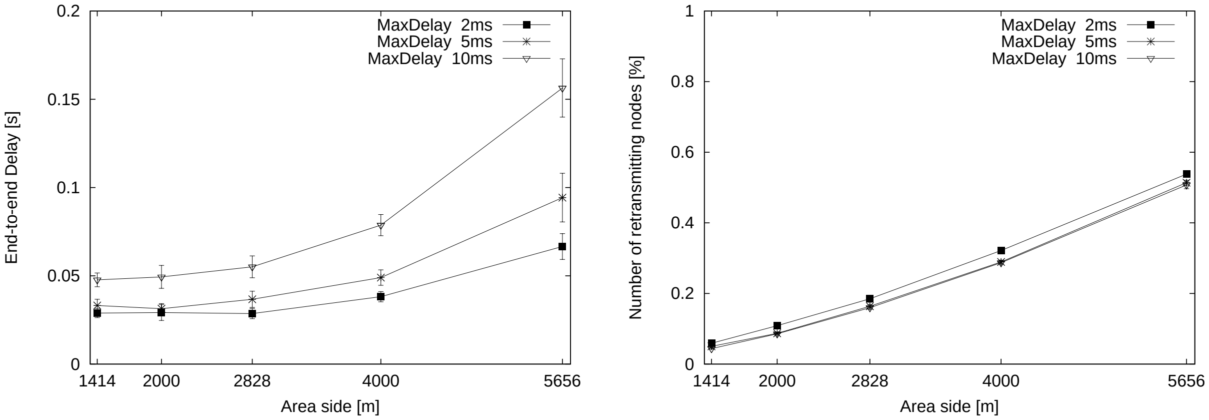
(c) Ratio of rebroadcasting nodes

Figure 8: Comparison of different DFD functions

still higher than 98%. This is due to the fact that the metric of the additionally covered area is additive. That means that even none of the neighbors covers by itself more than the threshold of 40%, all together still may cover more than than the threshold. On the other hand, $\mathrm{DDB_{DB}}$ and $\mathrm{DDB_{SS}}$ are not additive. As long as no node is below the 40% threshold, the node will rebroadcast, independent on how many nodes perhaps cover already 39%. Due to the same reason however, the number of rebroadcasting nodes for the $\mathrm{DDB_{AC}}$ is smaller than for $\mathrm{DDB_{DB}}$ as depicted in Fig. 8(c). As expected the $\mathrm{DDB_{SS}}$ performed not as well as the $\mathrm{DDB_{AC}}$ as signal strengths only allows to approximate distances and transmission ranges are perfectly circular in our simulations. The situation may completely look different in case of irregular transmission ranges, where distances do not match the additionally covered area anymore. Also when considering the delay in Fig. 8(b), $\mathrm{DDB_{AC}}$ outperforms the other two versions. This is to due the reduced number of transmitting nodes. Thus, in the following we normally only evaluate $\mathrm{DDB_{AC}}$ in more detail. The general observations should however still hold for the other two versions as well.

5.3.2 Impact of Max_Delay

The delivery ratio was similar to the results in the previous subsection, almost 100% for all scenarios, and independent of the Max_Delay, and is, thus, not depicted. In Fig. 9(b), the delay of for Max_Delay of 2, 5, and 10 ms is given. A smaller Max_Delay has a significant smaller delay in sparse networks. The difference is reduced for denser networks. On the other hand, we can see in Fig. 9(a) that the number of rebroadcasting nodes is basically not affect of different values for Max_Delay. As we will seen in subsection 5.3.4, the reason that a shorter Max_Delay does not increase the rebroadcasting nodes is due to the "Cross-Layer Information" optimization.



(a) End-to-end delay (b) Rebroadcasting nodes

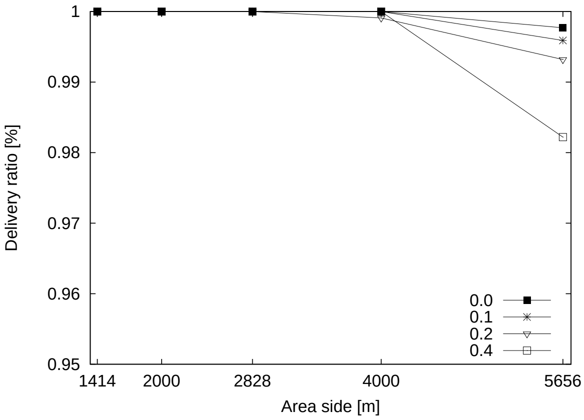
Figure 9: Impact of Max_Delay

5.3.3 Impact of rebroadcasting threshold RT

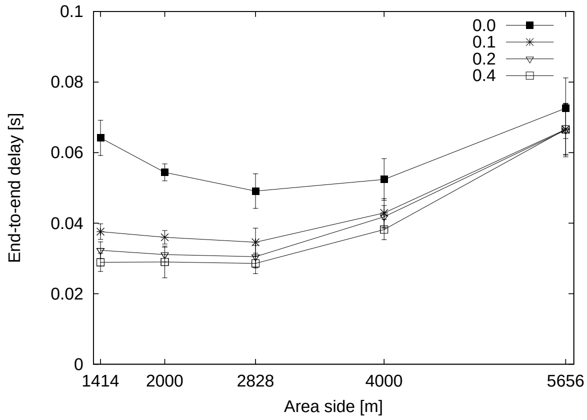
The values of RT are given as ratio of the maximal additionally covered area, i.e. 0% signifies that no area must be left uncovered. The results in Fig. 10(a) show that the delivery ratio does not suffer significantly from a higher rebroadcasting threshold RT even in sparse networks. The reason is the "first always" forwarding policy as shown in the next subsection 5.3.4. On the other hand, we observe that a higher RT has a major impact on the delay and the rebroadcasting nodes as depicted in Fig. 10(b) and Fig. 10(c). Especially, the raise from 0 to 10% of the maximal additional area yield much better values, whereas a further raise only marginally improves the results further.

5.3.4 Impact of the different components

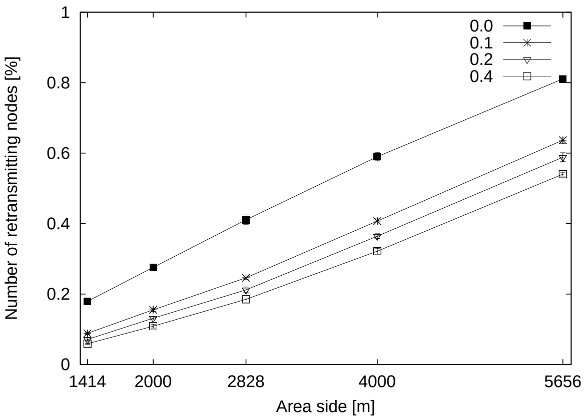
In this section, we evaluate the impact of the two different optimizations proposed in section 3.5, namely of the "first always" forwarding policy and the "cross-layer information", which allows DDB to drop packets stored in the queue of the MAC layer. We compare the performance of the DDB_{AC} with both optimizations to two slimmed versions, each one only comprising one of the optimizations. In the DDB_{AC} version without the "first always" optimization, the rebroadcasting threshold is also already applied if only one packet is received and not only if two or more redundant packets are received. If the cross layer information is not enabled, DDB_{AC} does not have the ability to access packets on the MAC layer. Thus, as soon as DDB_{AC} passes the packet down to the MAC layer, the packet will be sent and cannot be cancelled anymore. In Fig. 11(b) and Fig. 11(c), we can observe that the delay remains unaffected by the "first always" forwarding policy and that the number of rebroadcasting nodes is increased very slightly. On the other hand, we have in Fig. 11(a) that the delivery ratio sharply drops in sparse networks, if



(a) Delivery ratio



(b) End-to-end delay



(c) Ratio of rebroadcasting nodes

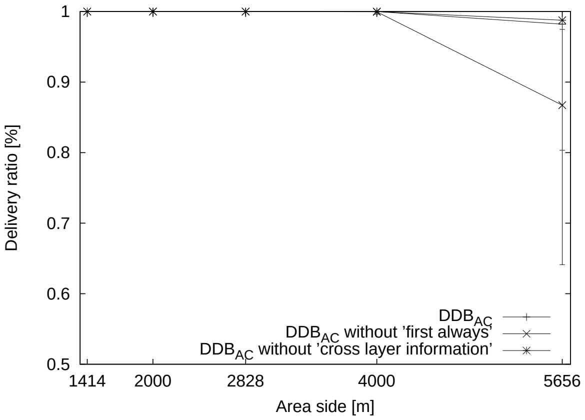
Figure 10: Impact of rebroadcasting threshold RT

the "first" always option is not enabled. This optimization allows DDB to efficiently cope with varying node densities. These results correspond to our prior considerations in section 3.5.

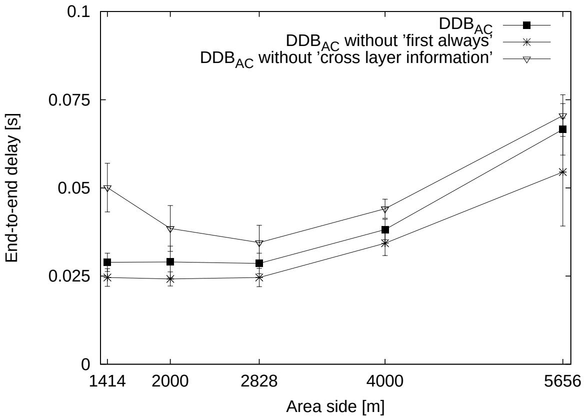
The performance of DDB_{AC} without the cross layer information suffers drastically, especially in numbers of rebroadcasting nodes in Fig. 11(c). The ratio to the DDB_{AC} is about 2 for sparse networks, but then increases to more than 10 for denser networks. As more nodes transmit almost simultaneously, the ability to access packets on the MAC layer is more beneficial in denser networks. The increased delay in Fig. 11(b) is a consequence of the higher number of transmitting nodes. However, if we simply increase the Max_Delay to $10\,ms$ then the performance without the cross layer information optimization almost equals again to the "original" DDB as shown in Fig. 11(d). With this longer Max_Delay, nodes keep the packet longer before passing to the MAC layer, this in turn increases the probability to receive redundant packets such that the rebroadcasting threshold is passed. Thus, we may conclude that this optimization allows us to have a short Max_Delay which decreases the end-to-end delay.

5.3.5 Conclusions

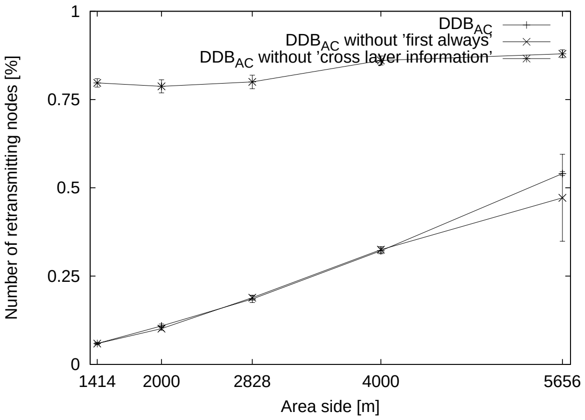
As the simulations showed a superior performance of DDB_{AC} in most scenarios, we uniquely used DDB_{AC} for the comparison with other broadcast protocols in the following sections. Even though, the situation may look different, if transmission ranges are highly irregular. In such a



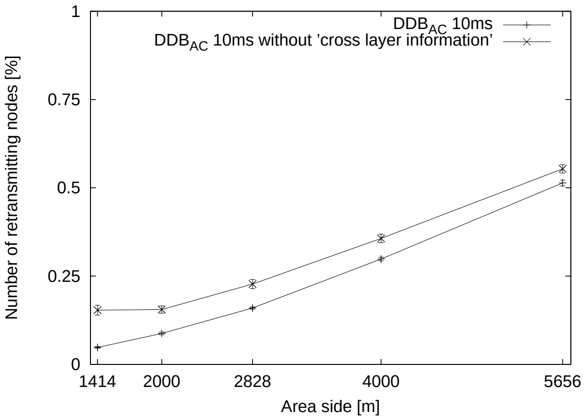
(a) Delivery ratio



(b) End-to-end delay



(c) Ratio of rebroadcasting nodes



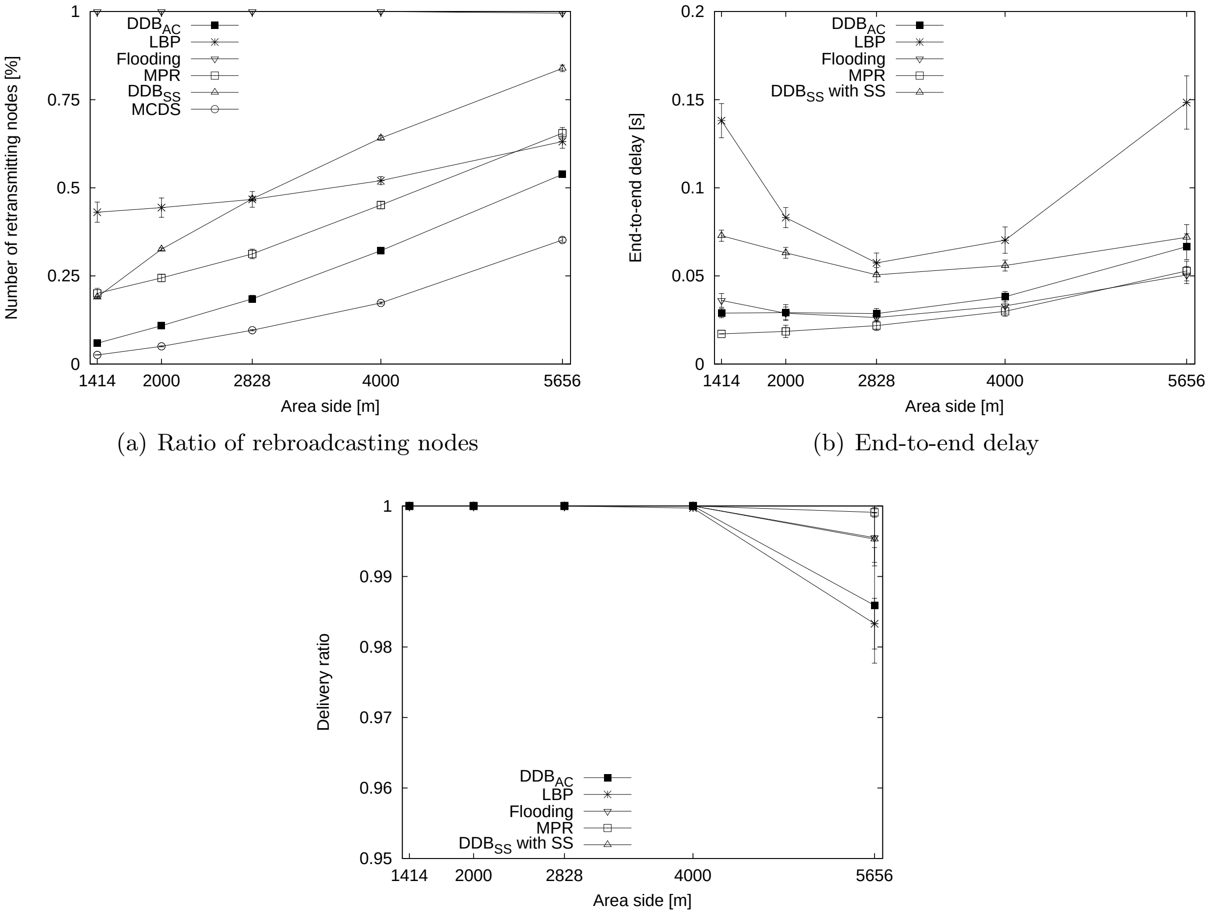
(d) Ratio of rebroadcasting nodes with 10ms

Figure 11: Impact of the different components

scenario, the performance of DDB_{SS} may improve even better than of the other two schemes. The two parameters are set to values which were found to have the best average performance over those scenarios, i.e. $Max\_Delay = 2\,ms$ and the rebroadcasting threshold to 40% of the maximal additionally covered area. Interestingly, this is the same rebroadcasting threshold as also proposed for LBP in [5] as for lower values LBP was not able to reduce significantly the number of retransmitting node. However as we will see later from the simulation results, the performance of LBP suffers in sparse networks and not all packets could be delivered. This is just the typical behavior of stateless algorithms that was discussed at the beginning in section 2.2. DDB_{RB} was not evaluated in this section, as we only used it in the simulations where we consider network lifetime.

5.4 Efficiency

The compare the performance in terms of rebroadcasting nodes of the different protocols with a theoretical optimum, we implemented additionally an algorithm that constructs the minimal connected dominating set (MCDS), which provides a lower theoretical bound for the number of rebroadcasting nodes. In Fig. 12(a), the number of transmissions of DDB_{AC} is about twice as high as for the MCDS for all network densities. As expected from the analytical results in section 4, the number constantly decreases for DDB_{AC} with higher node densities, whereas LBP



(a) Ratio of rebroadcasting nodes

(b) End-to-end delay

(c) Delivery ratio

Figure 12: Algorithm Efficiency

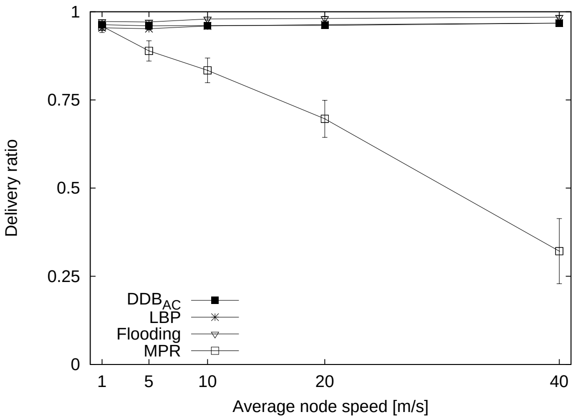
remains around 45%. This is due to the fact that the expected additional coverage of LBP is constant and increases for DDB_{AC} for higher node densities. Thus, the more neighbors a node has, the more additional coverage the rebroadcasting nodes have and the less transmissions are required. MPR performs significantly better than LBP. This is in accordance with [5], which observed that stateful protocols perform better than stateless protocols in dense networks. However, due to the locally optimal and dynamical rebroadcasting decision, the stateless DDB_{AC} outperforms even MPR. The ratio of MPR is also decreasing for higher node densities, however always remains significantly above the ratio of DDB_{AC}. The results in Fig. 12(b) show that the delay of DDB_{AC} first drops and then remains almost constant. For low node densities, a node has few neighbors which often do not cover a substantial additional area, but need to transmit anyway as no other neighbors do. These nodes add a non-negligible delay through the DFD function (2). For higher node densities, the delay is much shorter as the "best" nodes are close the the transmission range boundary and, thus, calculate a short DFD. DDB_{AC} performs always much better than LBP for two reasons. Nodes delay packets independently of the additional coverage in LBP and the delay has to be chosen much higher to avoid collisions. These facts are again supported by the analytical results. The delay for LBP increases because the number of retransmitting node is not reduced for higher node densities, which causes more and more collisions. Thus, nodes may not receive the actually first packet due to these collisions and have to "wait" for another copy which increases the delay. Even though, MPR relays packets immediately, the delay was only slightly lower than of DDB_{AC}, especially in denser networks. This is again because the "best" nodes in DDB_{AC} rebroadcast first and add lower delays for higher node densities. As expected, DDB_{SS} performs acceptable, but not as well as DDB_{AC} based on the additionally covered area because signal strength only allows a rough estimation of the distance The delivery ratio as shown in Fig. 12(c) is always 100%, expect for the case of very sparse networks with about 6 neighbors per node where the ratio drops marginally to approximately 99%.

5.5 Mobile Networks

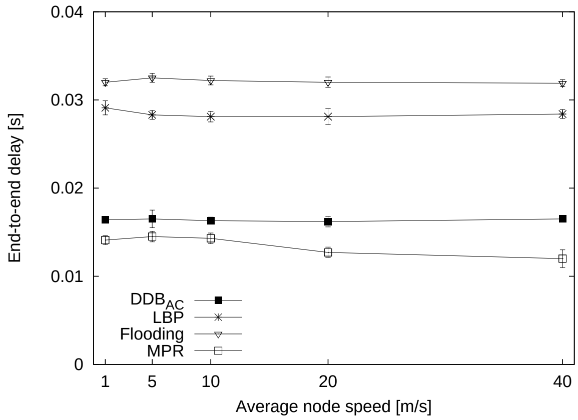
These simulations were computationally expensive and required a lot of memory. Therefore, we could only run simulations with 80 nodes. The size of the simulation areas were adapted accordingly to yield on average 9, 19, 49 neighbors per node similar to the node densities used in the previous subsection, but omitting the sparsest and densest networks with 6 and 98 neighbors respectively. We did not conduct simulations with 98 neighbors as then all nodes could be covered just by one transmission and the results are not meaningful anymore. The reason for excluding 6 neighbors is that it is hardly possible to the determine reliably the size of the maximal cluster in a mobile network for every point in time when a packet is transmitted and received. To obtain results without network partition, the minimal node density was increased to 9 neighbors. Packets are generated at a rate of 10 packets per second and nodes move according to the random waypoint mobility model. As the stationary distribution of the random waypoint mobility model is not a uniform distribution [46], the number of neighbors is higher for nodes in the center and lower for nodes at the border of the simulation area. The pause time is set to $0\,s$ and the minimal and maximal speeds are set to ±10% of an average speed. The average speed was varied over $1, 5, 10, 20, 40\,m/s$. We also run the simulation with the rather high speed values of $20, 40\,m/s$ as we consider speed as a proxy for any kind of topology changes, caused either by mobility, sleep cycles, etc.

The delivery ratio is depicted in Fig. 13(a) for an average network density of 9 neighbors. The three stateless protocols are not affected and the performance remains constant independent of the mobility. The reason for their delivery ratio slightly below 100% is due to the temporarily partition of the network caused by mobility even for an average of 9 neighbors. As expected, only the performance of the stateful MPR suffers under mobility because its view on the network topology may be inconsistent, i.e. the known one- and two-hop neighbors do not correspond to the actual physical neighbors. This causes also a not correct calculation of the forwarding nodes, i.e. either nodes which should rebroadcast the packet based on the physical network topology do not, or vice versa. As the network density is low, already few wrong rebroadcast decisions may prevent the packets from being delivered to all nodes in the network. Obviously, the number of rebroadcasting nodes drops for the MPR for higher node densities as shown in Fig. 13(c) because the delivery ratio also decreases significantly and because the percentage of retransmitting nodes is calculated relative to the total number of nodes and not the number of nodes which received the packet. The delays of MPR and DDB are about twice as low as for flooding and the LBP and are stable for all protocols over all average speeds as shown in Fig. 13(b).

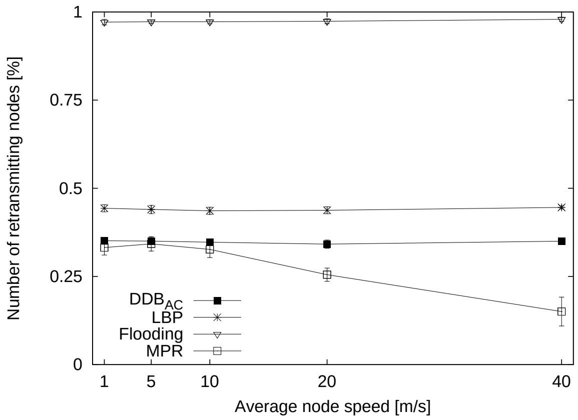
For a higher node density with 19 neighbors as shown in Fig. 14, the results are basically the same as for the sparser network. There are only two notable differences. First the performance of MPR did not decrease that strongly for a higher average speed. The reason is that the inconsistent view has a smaller impact as packets can be still delivered due to the high connectivity, even if the "wrong" nodes rebroadcast the packets. And second, the delivery ratio of the stateless protocols increased to 1 over all simulations. The high node density keeps the network connected all the time. The simulations could not be conducted for MPR with the highest node density of 49 neighbors as each topology change requires additional computation to determine the forwarding nodes, which turned out to be too resource consuming. Thus, the results are only given for the three stateless protocols in Fig. 15. The results are basically the same as before only that the number of rebroadcasting nodes is further decreased. Again due to the same reasons as already mention before, the $\mathrm{DDB_{AC}}$ yields the shortest delay among the three stateless protocols followed by LBP and flooding because of the higher number of rebroadcasting nodes.



(a) Delivery ratio



(b) End-to-end delay



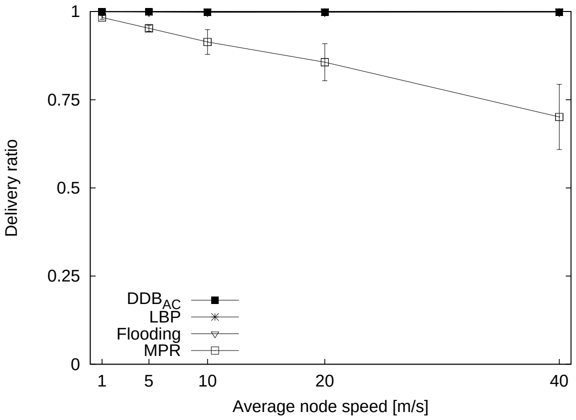
(c) Ratio of rebroadcasting nodes

Figure 13: Mobility with 9 neighbors

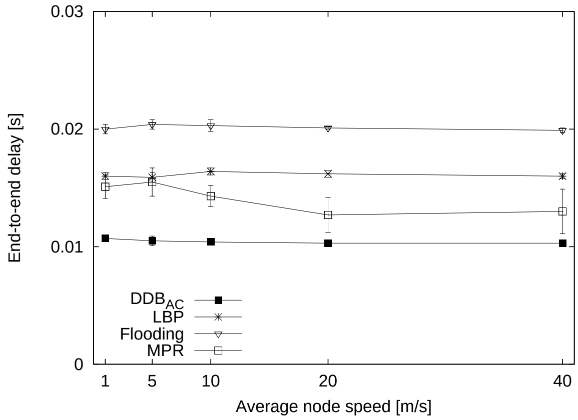
5.6 Congested Networks

The simulation parameters are the same as in the mobile network in the previous subsection, i.e. 80 nodes over different simulation areas, however without mobility as it is the objective of these simulations to evaluate solely the effect of congestion. One randomly chosen node broadcasts packets at different rates from 20 to 100 packets per second.

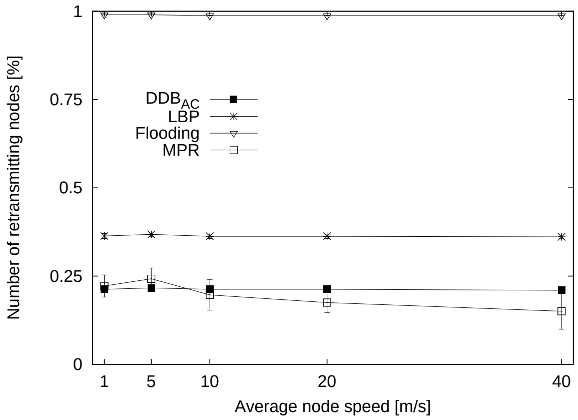
As depicted in Fig. 16(a) for an average of density 19 neighbors, the delay and the delivery ratio of all protocols suffer in congested networks due to collisions and queue overflows. MPR outperforms the other protocols in these scenarios yielding almost always 100% delivery ratio and very short delays. Two facts contribute to this superior performance. First, packets are



(a) Delivery ratio



(b) End-to-end delay

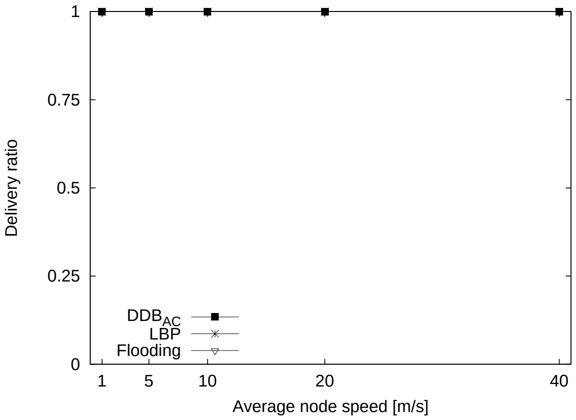


(c) Ratio of rebroadcasting nodes

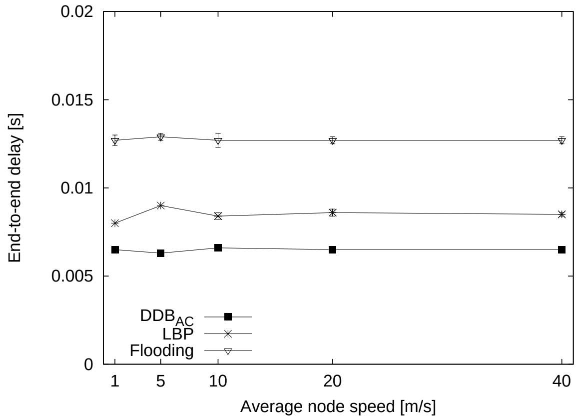
Figure 14: Mobility with 19 neighbors

rebroadcasted at nodes immediately and, second, nodes only have to forward packets received from specific nodes, namely the ones which selected them as forwarding nodes. Thus, the queues do not fill up that quickly. The stateless protocols add delay to each packet and also have to buffer at first all packets received from any neighbor. Among the stateless protocols, DDB$_{AC}$ performs best by far and even only lags behind MPR for the highest chosen congestion level. The delay of DDB$_{AC}$ remains very short and only increases for the highest traffic load. It is a factor of five and more better for highly congested networks than the other stateless protocols LBP and flooding. They show an increased delay already for lightly loaded networks. With DDB, only few packets need to be buffered at the nodes because of the short DFD in dense networks, the fewer rebroadcasting nodes, and the high RT threshold which allows dropping a lot of packets quickly. Flooding and LBP suffer under their inability and limited ability respectively to reduce the number of retransmitting nodes. The worse performance of LBP than simple flooding is due to the required long buffering time of $10\,ms$ which causes more queue overflows. The number of rebroadcasting nodes are depicted in Fig. 16(b). Only MPR and DDB$_{AC}$ remain unaffected by the packet generation rate, expect that DDB$_{AC}$ increases slightly for the highest rate. This is reflected by the increased delay and decreased delivery ratio in Fig. 16(a). Clearly, the number of retransmitting nodes of LBP and flooding decreases at least with the delivery ratio.

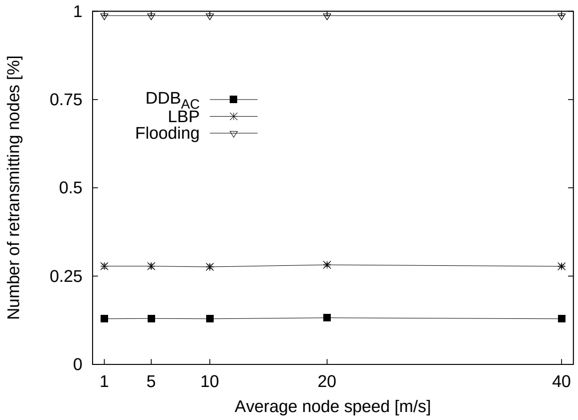
The results are similar for other simulated node densities in Fig. 17(a) and Fig.17(b). Only



(a) Delivery ratio

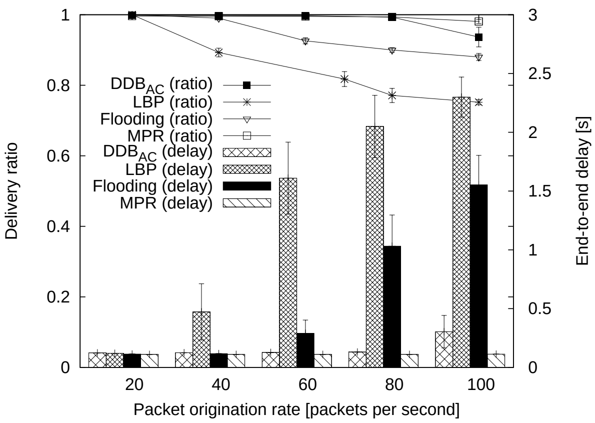


(b) End-to-end delay

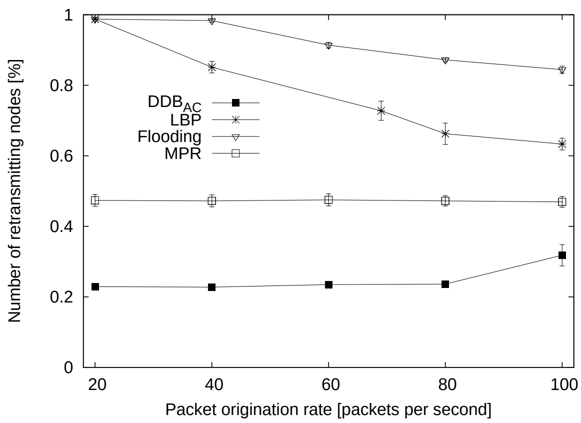


(c) Ratio of rebroadcasting nodes

Figure 15: Mobility with 49 neighbors



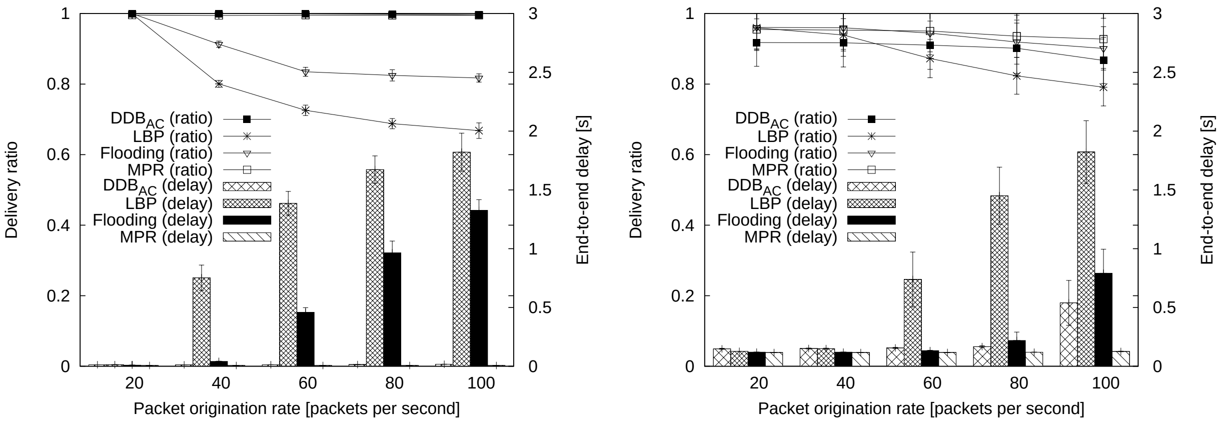
(a) Delivery Ration and end-to-end delay



(b) Ratio of rebroadcasting nodes

Figure 16: Congested Network with 19 neighbors

three significant differences can be observed. First, in a rather sparse network with only 9 neighbors, none of the protocols was able to deliver all the packets. Nodes are connected only over few links and, thus, if packets are dropped at some nodes due to congestion, the packet



(a) Delivery Ration and end-to-end delay for 44 neighbors (b) Delivery Ration and End-to-end delay for 9 neighbors

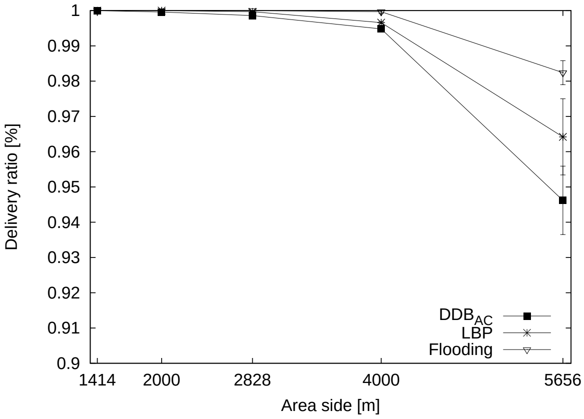
Figure 17: Congestion in sparse and dense networks

can no longer be delivered to all nodes. Second, the flooding improved in terms of delay and delivery ratio and was similar to DDB$_{\text{AC}}$ in the sparse network because the smaller number of neighbors also reduces the number of collisions of flooding. And third, the delivery ratio of DDB$_{\text{AC}}$ raised to almost 1 in denser networks with 44 neighbors over all congestion levels. At the same time, the delay was reduced to the same value as for MPR.

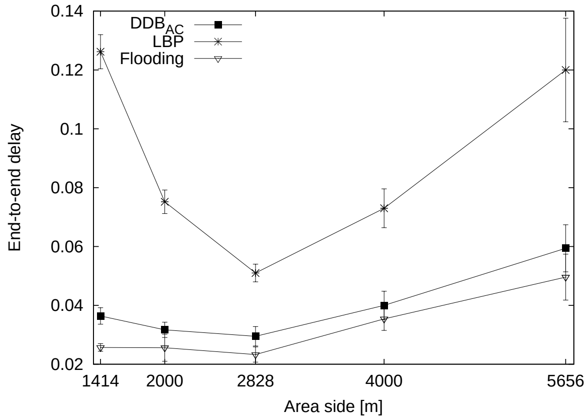
Obviously, the CSMA-based 802.11 MAC protocol has a major influence and results may differ for other MAC protocols. The MAC protocol has definitely also an impact on the lightly load static network in the previous subsection, however should be very small and almost negligible.

5.7 Irregular Transmission Range

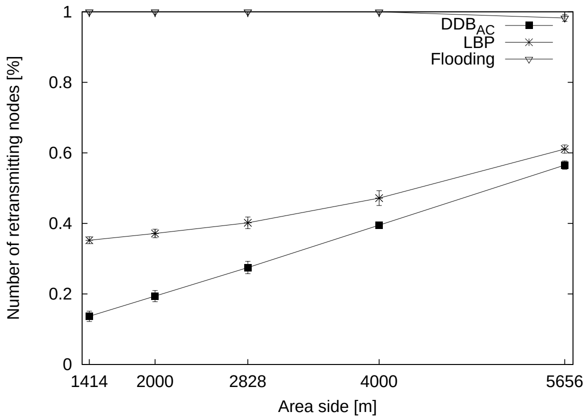
Basically all papers on broadcasting conducted simulations only with isotropic propagation models which do not accurately reflect real radio propagation characteristics. Especially for position-based broadcasting protocols, the irregularity of transmission ranges may have a strong impact on the performance. We use the radio irregularity model (RIM) of [47] to evaluate the performance under non-isotropic transmission ranges. The RIM is an extension to an underlying isotropic radio model and accounts for main properties of devices and radio signals such as non-isotropic path losses, continuous variation, and heterogeneous signal sending power. As the underlying model, we use the two ray ground reflection model. Two parameters are used to control the degree of irregularity (DOI) of the transmission range and the variance of sending power (VSP) which are set in our simulations to 0.1 and 0.5, as also suggested in [47]. When these parameters values are set to 0, the RIM is reduced to the two ray model. Please note, that even if the transmission ranges are highly irregular with RIM, the mean power density at a distance d from the transmitter is the same as in an the underlying two ray model. The rest of the simulation parameters are set as in the section 5.4 where the algorithms efficiencies was evaluated, i.e. 1000 static nodes over different simulations areas. In Fig. 18, we can see that the performance of DDB$_{\text{AC}}$ suffers under irregular transmission ranges if we compare the results with Fig. 12. Especially, the delivery ratio for sparse networks drops quite a bit. Unfortunately, it is not clear whether this is due to the protocols inability to deliver the packet or due to the partition of the network. This partition cannot be detected reliably due to the irregular transmission ranges. Most probably, both contribute to the decrease of the delivery ratio. On the other hand, in dense networks the number of rebroadcasting nodes cannot be reduced that efficiently anymore with DDB$_{\text{AC}}$ as it assumes isotropic transmission ranges to calculate the additionally covered area. Thus, with irregular transmission ranges the calculation of the DFD and the application of the rebroadcasting threshold RT is suboptimal. However, the performance is still very good when compared to the other protocols.



(a) Delivery ratio



(b) End-to-end delay



(c) Delivery ratio

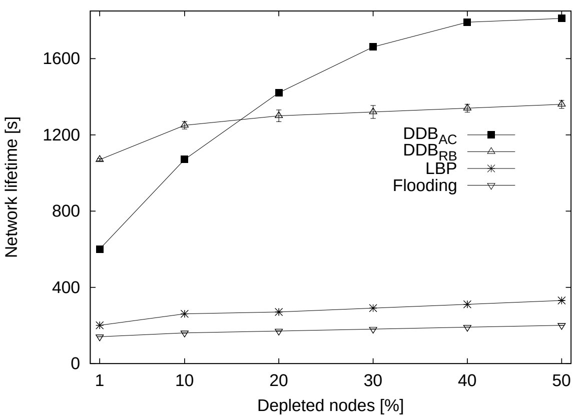
Figure 18: Impact of irregular transmission range

5.8 Network Lifetime

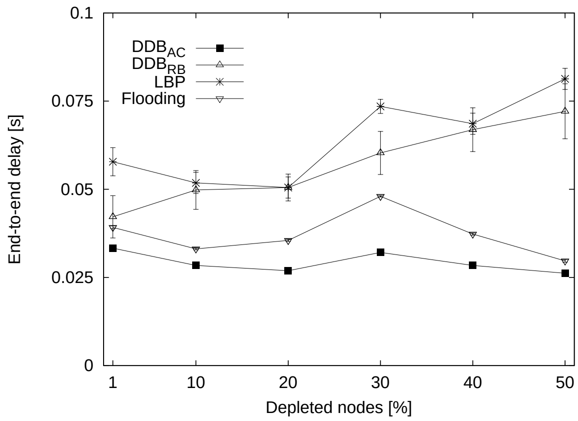
In many network contexts, where batteries of nodes cannot be recharged or replaced, the network lifetime may be of higher importance than other performance metrics. We define the network lifetime as the time until a certain number of nodes fail due to battery depletion similar to [28] and [48]. Other definitions of network lifetime are used in [49] and [50], which measure the mean expiration time and the time of the first node failure respectively.

The network lifetime strongly depends on the consumed energy while sending, receiving, and idle listening. If the ratio between these three modes is small, then obviously, which and how many nodes broadcast, does not have any effect and almost all nodes deplete at the same time. The interesting scenarios occur if the ratio is large enough as we then may expect that nodes which transmit more frequently deplete earlier. For our simulations, the ratio of sending/receiving/idle listening was set to 10/1/0.01. These values are justified by recent technology advances, cp. e.g. [51], which also allow even higher ratios. The transmission delay of one packet is approximately $\frac{\simeq 2000bit}{2Mbps} \simeq 1\,ms$ such that most of the energy would still be consumed for idle listening as a node is idle for $\simeq 99.9\%$ of the time. However, if we broadcast packet at such a high rate that nodes are transmitting most of the time, i.e. more than 500 packet per second, we would encounter severe congestion and the results would be misleading. Thus, packets are still broadcasted at only a rate of one packet per second, but we modified the energy model in a way that the energy consumption in idle mode is no longer time depending, but is simply decreased by 0.1 between two broadcast packets. This would equal a situation where a node is idle 50% of the time and transmits and receives the other 50% of the time, but without congestion. We place 1000 nodes over an area of 2000x2000 m. Furthermore, we assume that nodes have equal battery level at the beginning, and all nodes consume the same amount of energy for transmission. As DDB_{RB} always favors nodes with more residual power, the power level of all nodes are kept more or less at the same level and, thus, which in turn also increases the probability of simultaneous transmissions. Similar to simple flooding, we also jitter the transmissions at node for $2\,ms$ to reduce collisions. Assuming that sending and receiving of a hello message consumes about the same energy than a data packet, the lifetime of MPR will only be a very small fraction of the other stateless protocols. In our scenario with 1000 nodes and a hello message interval of $2\,s$, 500 hello messages are broadcasted per second which will deplete the nodes' batteries very quickly. Thus, the MPR protocol is not depicted.

As shown in Fig. 19(a), the second scheme DDB_{RB} where rebroadcasting decisions are solely based on residual battery power exhibits by far the longest time until the first percentage of nodes fail and outperforms significantly LBP and DDB_{AC}. This is achieved even under the fact that the number of rebroadcasting nodes is about the same for DDB_{RB} as for LBP, because the rebroadcast decision is independent of the additional covered area and, thus, much higher than for DDB_{AC}. However, the initially longer lifetime of DDB_{RB} comes at the cost of a longer delay as depicted in Fig. 19(b). For a higher percentage of depleted nodes, DDB_{AC} shows longer network lifetimes than DDB_{RB} due to the smaller number of rebroadcasting nodes and, thus, smaller total amount of energy consumed for each packet. With DDB_{RB}, the remaining nodes deplete quickly after the first one fail because nodes with more residual power normally rebroadcast packets. Thus, all nodes have all the time similar residual energy levels.



(a) Network lifetime until a certain percentage of nodes fail



(b) End-to-end delay after a certain percentage of nodes failed

Figure 19: Network lifetime

6 Conclusion

In this paper we presented the simple stateless broadcasting protocol DDB, which uses the dynamic forwarding delay (DFD) concept to optimize broadcasting in wireless multi-hop networks. With DFD, nodes are able to take locally optimal rebroadcasting decisions without any neighbor knowledge.

We compared the performance of DDB with one specific DFD function, which reduces the number of transmissions, to another stateless broadcasting protocol LBP and a state-of-the-art stateful protocol MPR, which uses neighbor knowledge obtained through hello messages. LBP was not able to perform well over a wide range of network conditions, namely the performance degrades under heavy traffic load and high node density, as also observed in [5]. However, DDB did not suffer from these drawbacks of other stateless protocols such as LBP. Actually, quite the contrary is true. The performance of DDB even improved for those scenarios of high traffic load and high node density.

MPR performed well in most scenarios, expect in highly dynamic networks where the delivery ratio collapsed. The delay of MPR was the shortest in all simulated scenarios closely followed by DDB whose delay was approximately 10% higher, expect in the case of highly congested networks. On the other hand, DDB outperformed MPR significantly considering the efficiency of the algorithm. DDB only required about half of the transmissions to deliver the packet reliably to all nodes compared to MPR. Furthermore, as DDB is stateless, its performance was completely unaffected in highly dynamic networks. However, the biggest advantage of DDB over MPR is its simplicity and economical use of network resource because no control messages are transmitted. These costs of proactively transmitting hello messages in MPR, which occur even if no data packets are broadcasted, makes their use in certain kind of networks with strict resource constraints even more in appropriate, e.g. sensor networks.

The four main advantages of DDB can be summarized as follows:

- DDB is stateless, thus, the performance is unaffected even under very frequent topology changes.
- DDB does neither generate any transmissions of control messages nor require proactive computation at nodes. This saves scares network resources such as battery power and bandwidth.
- DDB almost eliminates completely the problems of other stateless or location-based protocols as described in [5], namely the poor performance in dense and congested networks. The dynamic forwarding delay concept allows DDB to cope efficiently with a wide range of network conditions.
- DDB is able to extend the network lifetime by taking the energy level of nodes into forwarding decision.

We believe that these characteristics make DDB a valuable broadcast protocol for wireless multi-hop networks with either frequently changing topology and/or very strict power limitations such as vehicular and sensor networks. For future work, we envision the integration with directional antennas as already proposed in this paper. Furthermore, Max_Delay may also be adjusted dynamically according to the encountered network conditions, such as congestion and node density, and more sophisticated DFD functions, which may combine location information, signal strength, signal-to-noise ratio, bit error rate, etc., could also help to further improve performance.

References

[1] K. Obraczka, K. Viswanath, and G. Tsudik, "Flooding for reliable multicast in multi-hop ad hoc networks," *Wireless Networks*, vol. 7, no. 6, pp. 627–634, Nov. 2001.

[2] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," in *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM '99)*, Seattle, USA, Aug. 1999, pp. 151 – 162.

[3] Y.-C. Tseng, S.-Y. Ni, and E.-Y. Shih, "Adaptive approaches to relieving broadcast storms in a wireless multihop mobile ad hoc network," *IEEE Transactions on Computers*, vol. 52, no. 5, pp. 545–557, May 2003.

[4] Z. J. Haas, J. Y. Halpern, and L. Li, "Gossip-based ad hoc routing," in *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '02)*, New York, USA, June 2002, pp. 1707 – 1716.

[5] B. Williams and T. Camp, "Comparison of broadcasting techniques for mobile ad hoc networks," in *Proceedings of the 3rd ACM International Symposium on Mobile and Ad Hoc Networking and Computing (MobiHoc '02)*, Lausanne, Switzerland, June 2002, pp. 194–2002.

[6] J. Cartigny and D. Simplot, "Border node retransmission based probabilistic broadcast protocols in ad-hoc networks," *Telecommunication Systems*, vol. 22, pp. 189–204, Apr. 2003.

[7] Y. Sasson, D. Cavin, and A. Schiper, "Probabilistic broadcast for flooding in wireless mobile ad hoc networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '03)*, New Orleans, Louisiana, USA, Mar. 2003, pp. 1124–1130.

[8] A. Laouiti, A. Qayyum, and L. Viennot, "Multipoint relaying: An efficient technique for flooding in mobile wireless networks," in *Proceedings of the 34th Annual Hawaii International Conference on System Sciences (HICSS-34)*, Hawaii, USA, Jan. 2001.

[9] T. Clausen and P. Jacquet, "Optimized link state routing protocol (OLSR)," RFC 3626, Internet Engineering Task Force IETF, Oct. 2003. [Online]. Available: http://www.ietf.org/rfc/rfc3626.txt

[10] W. Lou and J. Wu, "Double-covered broadcast (dcb): A simple reliable broadcast algorithm in manets," in *Proceedings of the 23st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '04)*, Hong Kong, China, Mar. 2004.

[11] H. Lim and C. Kim, "Multicast tree construction and flooding in wireless ad hoc networks," in *Proceedings of the 3rd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems (MSWIM 2003)*, Boston, Massachusetts, United States, Aug. 2000, pp. 61–68.

[12] W. Peng and X. Lu, "Ahbp: An effcient broadcast protocol for mobile ad hoc networks," *Journal of Science and Technology*, vol. 16, no. 2, Mar. 2001.

[13] W. Lou and J. Wu, "On reducing broadcast redundancy in ad hoc wireless networks," *IEEE Transactions on Mobile Computing*, vol. 1, no. 2, pp. 111–122, Apr. 2002.

[14] W. Peng and X.-C. Lu, "Efficient broadcast in mobile ad hoc networks using connected dominating sets," *Journal of Software*, vol. 10, no. 7, Apr. 1999.

[15] M. Marathe, H. Breu, H. Hunt III, S. Ravi, and D. Rosenkrantz, "Simple heuristics for unit disk graphs," *Networks*, vol. 25, no. 59-68, 1995.

[16] J. Wu and H. Li, "On calculating connected dominating set for efficient routing in ad hoc wireless networks," in *Proceedings of the 3th International ACM Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM '99)*, Seattle, USA, Aug. 1999, pp. 7–14.

[17] I. Stojmenovic, M. Seddigh, and J. Zunic, "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, Jan. 2002.

[18] J. Susec and I. Marsic, "An efficient distributed network-wide broadcast algorithm for mobile ad hoc networks," Center for Advanced Information Processing (CAIP), Rutgers University, New Jersey, USA, Tech. Rep. TR-248, July 2000.

[19] J. Wu and F. Dai, "Broadcasting in ad hoc networks based on self-pruning," in *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '03)*, San Francisco, USA, Mar. 2003, pp. 2240– 2250.

[20] R. Gandhi, S. Parthasarathy, and A. Mishra, "Minimizing broadcast latency and redundancy in ad hoc networks," in *Proceedings of the 4th ACM International Symposium on Mobile and Ad Hoc Networking and Computing (MobiHoc '03)*, Annapolis, Maryland, USA, June 2003, pp. 222–232.

[21] J. Wu and F. Dai, "Mobility management and its applications in efficient broadcasting in mobile ad hoc networks," in *Proceedings of the 23st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '04)*, Hong Kong, China, Mar. 2004.

[22] M. Seddigh, J. S. Gonzalez, and I. Stojmenovic, "Rng and internal node based broadcasting algorithms for wireless one-to-one networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 5, no. 2, pp. 37–44, Apr. 2001.

[23] F. Dai and J. Wu, "Performance analysis of broadcast protocols in ad hoc networks based on self-pruning," *IEEE Transactions on Parallel and Distributed Systems*, vol. 15, no. 11, Nov. 2004.

[24] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "On the construction of energy-efficient broadcast and multicast trees in wireless networks," in *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '00)*, Tel Aviv, Israel,, Mar. 2000, pp. 585–594.

[25] J. Cartigny, D. Simplot, and I. Stojmenovic, "Localized minimum-energy broadcasting in ad-hoc networks," in *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '03)*, San Francisco, CA, USA, Mar. 2003, pp. 2210–2217.

[26] M. Cagalj, J.-P. Hubaux, and C. Enz, "Energy-efficient broadcasting in all-wireless networks," *ACM/Baltzer Mobile Networks and Applications*, p. to appear, 2004.

[27] I. Kang and R. Poovendran, "Maximizing static network lifetime of wireless broadcast adhoc networks," in *Proceedings of IEEE International Conference on Communications (ICC) 2003*, Achorage, Alaska, USA, May 2003.

[28] R. Wattenhofer, L. Li, P. Bahl, and Y. Wang, "Distributed topology control for power efficient operation in multihop wireless ad hoc networks," in *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '01)*, Anchorage, USA, Apr. 2001, pp. 1388–1397.

[29] I. Kang and R. Poovendran, "A comparison of power-efficient broadcast routing algorithms," in *Proceedings of IEEE Global Telecommunications Conference (Globecom 2003)*, San Francisco, CA, USA, Dec. 2003.

[30] Y. Wang and J. J. Garcia-Luna-Aceves, "Broadcast traffic in ad hoc networks with directional antennas," in *Proceedings of IEEE Global Telecommunications Conference (Globecom 2003)*, San Francisco, CA, USA, Dec. 2003, pp. 210–215.

[31] C. Hu, Y. Hong, and J. Hou, "On mitigating the broadcast storm problem with directional antennas," in *Proceedings of IEEE International Conference on Communications (ICC 2003)*, Anchorage, Alaska, USA, May 2003, pp. 104–110.

[32] J. Cartigny, D. Simplot, and I. Stojmenovic, "Localized energy efficient broadcast for wireless networks with directional antennas," in *Proceedings of the Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET'2002)*, Sardegna, Italy, Sept. 2002.

[33] I. Kang and R. Poovendran, "Power-efficient broadcast routing in adhoc networks using directional antennas: technology dependence and convergence issues," University of Washington, Washington, USA, Tech. Rep. UWEETR-2003-0015, July 2003.

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

[35] H. Füssler, J. Widmer, M. Käsemann, M. Mauve, and H. Hartenstein, "Contention-based forwarding for mobile ad-hoc networks," *Elsevier's Ad Hoc Networks*, vol. 1, no. 4, pp. 351–369, Nov. 2003.

[36] M. Zorzi and R. R. Rao, "Geographic random forwarding (GeRaF) for ad hoc and sensor networks: Multihop performance," *IEEE Transactions on Mobile Computing*, vol. 2, no. 4, pp. 337–348, Oct. 2003.

[37] M. Heissenbüttel and T. Braun, "A novel position-based and beacon-less routing algorithm for mobile ad-hoc networks," in *Proceedings of the 3rd IEEE Workshop on Applications and Services in Wireless Networks (ASWN' 03)*, Bern, Switzerland, July 2003, pp. 197–210.

[38] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks," *ACM/Baltzer Wireless Networks*, vol. 7, no. 6, pp. 609–616, Nov. 2001.

[39] F. Kuhn, R. Wattenhofer, and A. Zollinger, "Worst-case optimal and average-case efficient geometric ad-hoc routing," in *Proceedings of the 4th ACM International Symposium on Mobile and Ad Hoc Networking and Computing (MobiHoc '03)*, Annapolis, Maryland, USA, June 2003, pp. 267 – 278.

[40] B. Karp and H. T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM '00)*, Boston, USA, Aug. 2000, pp. 243–254.

[41] T. Moscibroda, R. O'Dell, M. Wattenhofer, and R. Wattenhofer, "Virtual coordinates for ad hoc and sensor networks," in *Proceedings of the ACM Joint Workshop on Foundations of Mobile Computing (DIALM-POMC'04)*, Philadelphia, Pennsylvania, USA, Oct. 2004.

[42] C. Maihöfer, "A survey of geocast routing protocols," *IEEE Communications Surveys & Tutorials*, vol. 6, no. 2, pp. 32–42, Apr. 2004.

[43] J. Nonnenmacher and E. W. Biersack, "Scalable feedback for large groups," *IEEE/ACM Transactions on Networking*, vol. 7, no. 3, pp. 375–386, June 1999.

[44] (2004, Nov.) Qualnet. Scalable Network Technologies (SNT). [Online]. Available: http://www.qualnet.com/

[45] O. Dousse, P. Thiran, and M. Hasler, "Connectivity in ad-hoc and hybrid networks," in *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '02)*, New York, USA, June 2002.

[46] W. Navidi and T. Camp, "Stationary distributions for the random waypoint mobility model," *IEEE Transactions on Mobile Computing*, vol. 3, no. 1, pp. 99–108, Jan. 2004.

[47] G. Zhou, T. He, S. Krishnamurthy, and J. A. Stankovic, "Impact of radio asymmetry on wireless sensor networks," in *Proceedings of the 2nd International Conference on Mobile Systems, Applicationns, and Services (MobiSys '04)*, Boston, USA, June 2004.

[48] Y. Xu, J. S. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing," in *Proceedings of the 7th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM '01)*, Rome, Italy, July 2001, pp. 70–84.

[49] S. Singh, M. Woo, and C. S. Raghavendra, "Power-aware routing in mobile ad hoc networks," in *Proceedings of the 4th annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM '98)*, Dallas, USA, Oct. 1998, pp. 181–190.

[50] J.-H. Chang and L. Tassiulas, "Energy conserving routing in wireless ad-hoc networks," in *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '00)*, Tel Aviv, Israel,, Mar. 2000, pp. 22–31.

[51] A. El-Hoiydi and J.-D. Decotignie, "Wisemac: An ultra low power mac protocol for multihop wireless sensor networks," in *First International Workshop on Algorithmic Aspects of Wireless Sensor Networks ALGOSENSORS 2004*, Turku, Finland, July 2004.