Optimized Stateless Broadcasting in Wireless Multi-hop Networks

Marc Heissenbüttel, Torsten Braun, Markus Wälchli, Thomas Bernoulli
Institute of Computer Science and Applied Mathematics
University of Bern, Switzerland
Email: {heissen, braun, waelchli, bernoull}@iam.unibe.ch

***Abstract*— In this paper we present a simple and stateless broadcasting protocol called Dynamic Delayed Broadcasting (DDB) which allows locally optimal broadcasting without any prior knowledge of 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 completely distributed way at the receiving nodes ensuring 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.**

***Index Terms*— Ad-hoc networks, broadcast, simulations**

I. INTRODUCTION

Broadcasting is most simply and commonly realized by flooding whereby nodes broadcasts each received packet exactly once. Duplicated packets are detected e.g. by the source node ID and a sequence number. Assuming we have 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 so 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. This may become especially critical in sensor networks where nodes have even more strict power, communication, and computation constraints.

Two important objectives of any broadcast algorithm in ad-hoc networks are reliability and optimized resource utilization. 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. The use of network resources should be minimized without effecting suffers. Interestingly, these objectives are often complementary. Minimizing the number of transmissions also may 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 optimally broadcast a packet network-wide. For example, to minimize the number of transmissions would require to determine the minimal connected dominating set. Thus, most practical broadcast algorithms for ad-hoc networks try to approach network-wide optimalality by locally optimal broadcasting of packets. This is commonly achieved by the proactive exchange of hello messages between neighbors so 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. These protocols are stateless and do not require any knowledge of 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 Dynamic Delayed Broadcasting (DDB). 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 to make locally optimal rebroadcasting decisions. Nodes decide whether to rebroadcast a message based solely 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. We explicitly propose and evaluate in more detail DDB with two different DFD functions. The first DFD function aims at reducing the number of overall transmissions to deliver the packet to all nodes in the network. The second DFD function addresses the problem of power consumption and aims at extending the network lifetime. We refer to DDB with one of these two specific DFD functions as DDB 1 and DDB 2, respectively.

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.

The remainder of this paper is organized as follows. An overview of related work is given in section II. We describe the details of DDB in section III. Analytical and simulation results are provided in section IV and V. Finally, section VI concludes the paper.

II. 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.

A. Taxonomy

Probability-based approaches: In [1], each node rebroadcasts a message with a certain probability p and drops the packet with a probability of $1-p$. If the probability to forward a packet is 1, this scheme is identical to simple flooding. [1] 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 [2], the threshold is no longer fixed but adapts to the number of neighbors. [3] evaluated probabilistic broadcasting in more depth and proposed several extensions to the protocol of [1] based on the obtained results. In [4], 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 [5].

Location-based approaches: In the location-based schemes proposed in [1], 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. Instead of using the distance of nodes as a measure for the additional area covered, [1] 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.

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 a forwarder thus reducing the total number of transmitted messages. In multipoint relaying (MPR) as described in [6], 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 [7]. In [8], the set of forwarders also comprises all one-hop neighbors, which are not covered by at least two other forwarders. In [9] and [10] 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 [11], two-hop neighborhood information is piggybacked on packets and permits to eliminate the two-hop neighbors already covered by the last visited node. A special class of neighbor-designated approaches are based on connected dominating sets, where only nodes of the dominating set rebroadcast a packet [12], [13].

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 [9], 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 [14], several distributed heuristics are proposed. [15] proposed an algorithm, which only requires two-hop neighborhood information. A node belongs to the dominating set, if two unconnected neighbors exist. This idea was further improved in [16], where the degree of a node was used as primary metric instead of their IDs. The protocol proposed in [17] 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. In [18], 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, [19] proposed to use two different transmission ranges for the determination of forwarders and for the actual broadcast process. In [20], 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 [21].

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. [22] 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. [23] 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. [24] showed the NP-completeness of minimal power broadcast. Note that energy efficiency is not necessarily directly related to network lifetime. If the same nodes always forward packets, broadcasting may be energy-efficient, but the battery at these nodes deplete quickly. In [25], 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. [26] 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 [27].

B. 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 of their one-hop neighbors, sometimes even global network knowledge is required. Comprehensive comparison studies were conducted in [28], [3], [21], and [27]. Their main conclusions can be summarized as follows:

Stateful broadcast 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. 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 were found to be twofold. First, the number of rebroadcasting nodes is disproportionately high in networks with 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 is that these stateless protocols use fixed parameters, e.g. the probability- or distance-threshold whether to rebroadcast a packet or not. This makes these algorithms not flexible enough to cope with a wide range of network scenarios. 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 do not significantly reduce the number of transmissions in dense networks for too low and too high parameter 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 however, 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 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 scarce 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 allowing ”better” nodes to rebroadcast first and suppress the transmissions of other neighbors. 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 [1], 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 deterministic concept of DFD to determine when to forward the packet which allows the making of locally optimal rebroadcasting decisions without knowledge of the neighborhood. In [1], 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 [1] and other stateless protocols, is that they can hardly 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 parameters. Forth, DDB is less sensitive the 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 should 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 $\mathrm{DDB}_{\mathrm{RB}}$ which is to the best of our knowledge the first completely localized scheme which aims at extending network lifetime. Most other energy-efficient protocols aim at reducing the energy to deliver thea 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 [25].

III. DYNAMIC DELAYED BROADCASTING PROTOCOL (DDB)

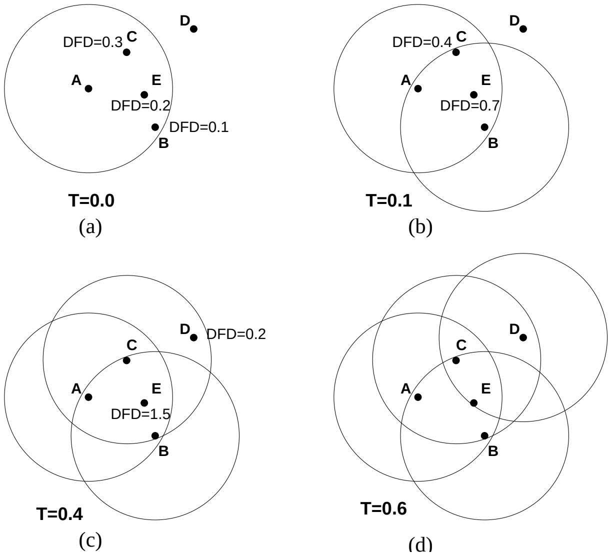
A. Introduction

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. [29]. 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 however. DDB can also operate without location information and use incoming signal strength to approximate the distance to other transmitting nodes.

B. DDB 1 for Minimizing the Number or Transmissions

The objective of the first scheme DDB 1 is to minimize the number of transmissions and at the same time to deliver the packet reliably to all nodes. Nodes that receive the broadcast 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 of 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 previously received transmissions. As usually done, 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 1 can "only" take locally optimal rebroadcasting decisions as nodes receive only transmissions from their immediate one-hop neighbors and thus have no knowledge of other more distant nodes which possibly already partially cover the same area.

To illustrate the complete procedure of the algorithm, consider the example given in Fig. 1, where we assume a rebroadcasting threshold $RT = 0$. 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. Let us assume 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, nodes B and E are not aware of their neighbors. As node B introduces the shortest additional delay and consequently rebroadcasts the packet first after $0.1\,ms$ which is also overheard at nodes E and C in Fig. 1(b). Upon the detection of this transmission, they determine a new DFD depending on the remaining additional coverage. Unlike before the transmission of node B, C calculates now a smaller delay than E. Assume that node E and C calculate a new DFD of $0.7\,ms$ and $0.4\,ms$ minus the $0.1\,ms$ they have already delayed the transmission. Consequently, node C will rebroadcast the packet $0.3\,ms$ later in Fig. 1(c) already at time $T = 0.4\,ms$. Nodes 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 schedules the rebroadcasting much earlier than E. D rebroadcast after $0.2\,ms$ while E waits $1.5\,ms$ minus $0.4\,ms$ passed since the reception of the first copy of this packet. After node D transmits 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 always assures that 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.



(a) (b) (c) (d)

Fig. 1. Example of the broadcast algorithm

DFD function: The explicit DFD function is crucial to the performance of DDB 1 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.

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

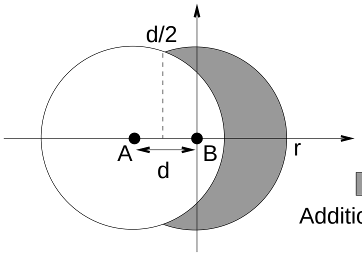


Fig. 2. Additional covered area

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, a node can cover a maximum of $\frac{AC_{MAX}}{\pi} \simeq 61\%$ additional area, which was not yet covered by the transmission of other nodes.

Taking into account this maximal AC_{MAX}, we propose a DFD which is exponential in the size of the additionally covered area, as it was shown in [30], 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 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, e.g. d_1 for two nodes with 0.2 difference in the additional coverage. 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 and e.g. only differ by d_2 for the same difference of 0.2 in the additional coverage. However, as they transmit much later, they have received multiple transmission of other nodes and may not require to retransmit at all because $AC < RT$.

DDB 1 with signal strength: Location information may not always be available. In order to minimize the number of transmissions, nodes can use the incoming signal strength as input to the DFD function instead of the additional covered area. 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 little additional area. 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 the performance will be. An advantage of DDB 1 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. Thus, no overhead and external information is required at all.

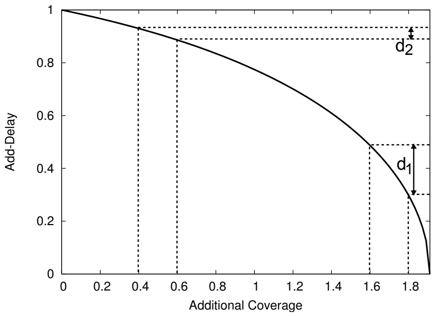


Fig. 3. Delay introduced by the DFD function

C. DDB 2 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 [25]. 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 a 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 the nodes,the same nodes will always rebroadcast the packet. Consequently, some nodes will deplete much quicker than others.

In this scheme DDB 2, the calculated delay by DFD depends solely on the remaining 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 than 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).

$$Add\_Delay = Max\_Delay \cdot \sqrt{\frac{e - e^{E_B}}{e - 1}} \tag{3}$$

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.

D. Optimizations

"First Always" Forwarding Policy: A common problem of broadcast protocols based on fixed parameters is that they are not able to cope with varying network conditions such as node density and traffic load [3]. 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 always forward a packet, which is received exactly once *independent* of the additional coverage, i.e. even if $AC < RT$. That means that the rebroadcasting threshold is only applied from the second received packet onwards. Especially in sparse networks, even a node with only very little 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 almost always be forwarded in such scenarios thus reducing 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.

Cross-Layer Information: Only DDB on the network layer is able to interpret the payload of the packet such as source ID and sequence number and, thus, detects that a newly 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 has already been 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. 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 normally has no longer influence on the further processing and, thus, cannot prevent the second "unnecessary" transmission. DDB should be able to access packets on the MAC layer, more precisely in the queue of the wireless interface, and to reprocess them accordingly by either dropping the packets or scheduling their transmission for a later time.

Directional Antennas: As we have seen in section III-B, 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. Implementing DDB with directional antennas and a comparison with broadcast protocols, which make use of directional antennas [31], [32], [33], [34], are outside of the scope of this paper and left for future work.

IV. ANALYTICAL ASSESSMENT

We want to calculate the expected size of the additional area AC that is covered by a node's transmission if we use DDB 1. In that case, the nodes that cover a larger 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.

$$AC(d) = d - \frac{1}{8}d^3 + \ldots + d + \frac{1}{24} + \ldots \simeq 2d \tag{4}$$

Let $X_i \in [0, 1]$ 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). The cdf is given simply 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 and probability density function (pdf)

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

for $0 \leq x \leq 1$.

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 of a node's transmission, given as $Y = g(X) = 2X$ by the approximation of (4), the pdf f_Y of Y is calculated as follows.

$$f_Y(x) = f_X[g^{-1}(x)]\frac{d}{dx}g^{-1}(x) = \frac{x}{2}$$

for a node at distance $0 \leq x \leq 2$. Thus, the cdf is simply

$$F_Y(x) = \frac{x^2}{4} \qquad \text{with } 0 \leq x \leq 2 \tag{5}$$

In order to derive the expected additional coverage E_{AC} 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. The general cumulative distribution function cdf $F_{Y_{(k)}}(x)$ for all $Y_{(k)}$ is given by

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

where $F_Y(x)$ is the cdf of the Y_i. Thus, we have

$$F_{Y_{(k)}}(x) = \sum_{j=k}^{n} \binom{n}{j} \left(\frac{x^2}{4}\right)^j \left(1 - \frac{x^2}{4}\right)^{n-j}$$

It is well-known that the expected value of a random variable Z can be calculated from its cdf $F_Z(x)$ by

$$E(Z) = \int_0^\infty (1 - F_Z(z))\,dz - \int_{-\infty}^0 F_Z(z)\,dz \tag{6}$$

Let $k \leq n$ denote the k-most distant neighbor, i.e. $k = n$ and $k = 1$ yield the most distant and the closest neighbor respectively. Obviously, the k-most distant neighbor has also the k-largest additionally covered area.

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.

$$E_{AC}^{Y_{(k)}}(x) = \int_0^2 \left(1 - F_{Y_{(k)}}(x)\right)\,dx = \frac{2\Gamma(n+1)\Gamma(k+\frac{1}{2})}{\Gamma(k)\Gamma(n+\frac{3}{2})} \tag{7}$$

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 cdf F_Y of the additional coverage for a single node is the same as derived before in (5). However, the expected additional coverage is independent of the number of neighbors n and the same for all neighbors and therefore is constant. Again with (6), we obtain

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

In Fig. 4, the graph is plotted for $E_{AC}^{Y_{(k)}}$ of DDB 1 and E_{AC}^* of other stateless broadcasting algorithms depending on the number of neighbors $n = 1\ldots30$. Again, $k \leq n$ denotes the k-most distant neighbor. 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, 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 1 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 1 is not only the reduction in number of transmissions, but also that the delay can be reduced as distant nodes which transmit first add almost no delay. From the values given in Fig. 4 and the DFD function (2), we can determine the additional delay introduced at the nodes which broadcast first. This delay is indeed small as it is basically inversely ”proportional” to the expected additionally covered area $E_{AC}^{Y_{(k)}}$, i.e. also the delay per rebroadcast is up to 43% shorter than with other stateless protocols.

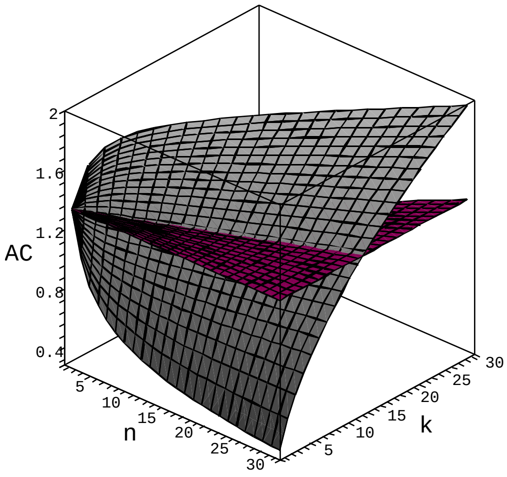


Fig. 4. Expected additional coverage

Furthermore with DDB 1 we know that nodes which cover a larger 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 1 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 analyses only provide a rough kind of boundary for the performance. We validate the general conclusions of the analytical results in the next section by simulations.

V. SIMULATIONS

A. Protocols

The DDB protocol was implemented with two optimizations proposed in section III-D, 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 use with directional antennas. The performance of DDB is compared to three protocols described in section II: The location-based broadcasting protocol [1], which is abbreviated by LBP in the following, the Multi-Point Relay MPR [6], and simple flooding as a the most simple broadcasting protocol. LBP and MPR were chosen as representatives for the categories of stateless and stateful broadcast protocols, respectively.

The parameters of LBP and MPR are set as suggested in [28] and in RFC 3626 [7], 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. With flooding, the packets have a jitter of $2\,ms$ to avoid that all neighbors transmit simultaneously. We simulated DDB in different scenarios to determine appropriate values for the rebroadcasting threshold RT and the Max_Delay. These two parameters are set to values which were found to have the best average performance over those scenarios. Max_Delay was set to $2\,ms$ because a short Max_Delay also decreases the delay significantly for low node densities. On the other hand the number of rebroadcasting nodes is only marginally lower with a longer Max_Delay even for dense networks where we expect many simultaneous transmissions. This is mainly due to the possibility of DDB to access and drop packets queued on the MAC layer. On the other hand, the rebroadcasting threshold RT should be as high as possible to reduce the number of retransmitting nodes as long as the packet is still reliably delivered to all nodes for all network conditions. But a higher RT increases at the same time the probability that in sparse network the packet is no longer delivered to all nodes. The ”first always forwarding policy” allows DDB using a very high RT. RT was set to 40% of the maximal area a node can cover, i.e. $0.4 \cdot 1.91 \simeq 0.76$. Without this ”first always” optimization, the delivery ratio dropped for the same RT to around 80% in sparse networks, opposed to the more than 99% when the ”first always forwarding” was used.

We used DDB 1 in all simulations. DDB 2 was only used in the simulation where we consider network lifetime.

B. Simulation Parameters

We implemented and evaluated the protocols in the Qualnet network simulator. The results are averaged over 10 simulation runs and given with a 95% confidence interval, which is sometimes very small and barely visible. In particular for high node densities, the performance among different simulations runs does almost not vary because nodes are static. 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 where data transmission starts at 180s and ends at 880s such that emitted packets arrive at the destination before the end of the simulation.

C. Efficiency

The first 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. With a square of 5656x5656 m^2, a node has approximately 6 neighbors which is just about the minimal required density for a completely connected network as results from percolation theory have shown [35]. We implemented a simple algorithm to determine the minimal connected dominating set (MCDS), which provides a lower theoretical bound for the number of rebroadcasting nodes.

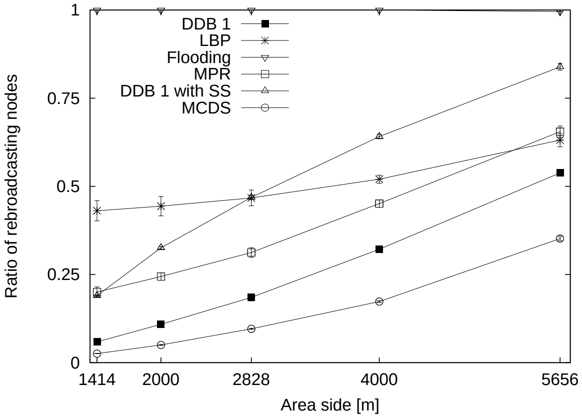


Fig. 5. Ratio of rebroadcasting nodes

In Fig. 5, the number of transmissions of DDB 1 is about twice as high as for the MCDS for all network densities. As expected from the analytical results in section IV, the ratio constantly decreases for DDB 1 with higher node densities, whereas LBP remains around 45%. This is due to the fact that the expected additional coverage of LBP is constant and increases for DDB 1 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 [28], which observed that stateful protocols perform better than conventional stateless protocols in dense networks. However, due to the locally optimal and dynamic rebroadcasting decisions, the stateless DDB 1 outperforms even MPR. Although the ratio of MPR also decreases for

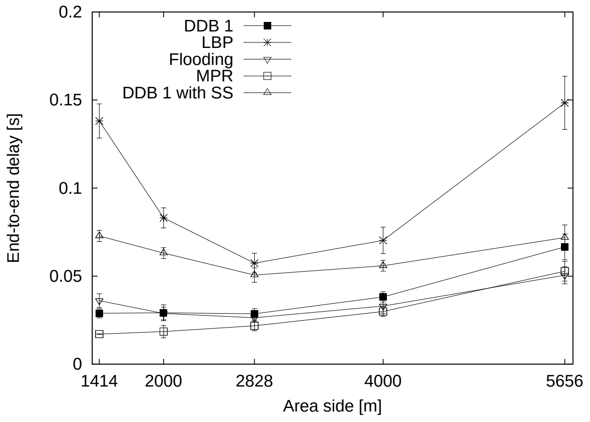


Fig. 6. End-to-end delay

higher node densities, it always remains significantly above the ratio of DDB 1. The results in Fig. 6 show that the delay of DDB 1 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 because of the ”first always” forwarding. 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 to the transmission range boundary and therefore calculate a short DFD. DDB 1 always performs 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 nodes is not reduced for higher node densities, which causes more and more collisions. Thus, nodes may not receive the actual 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 that of DDB 1, especially in denser networks. Again this is because the ”best” nodes in DDB 1 rebroadcast first and add lower delays for higher node densities. The delivery ratio was more than 99% for all protocols in all scenarios and is not shown.

D. Congested Networks

The objective of this simulation is to evaluate the effect of congestion. One randomly chosen node broadcasts packets at different rates from 20 to 100 packets per second. 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 the same node densities as in the previous subsection. In these simulations, the confidence intervals are larger than before because the high traffic volume causes fluctuations in the performance, most notably in sparse network. As depicted in Fig. 7 for an average density of 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 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 too quickly. The stateless protocols add delay to each packet and also first have to buffer all packets received from any neighbor. Among the stateless protocols, DDB 1 performs by far the best and lags behind only MPR for the highest chosen congestion level. The delay of DDB 1 remains very short and only increases for the highest traffic load. It is by a factor of five times or more lower for highly congested networks than the other stateless protocols LBP and flooding. They show a increased delay already for lightly loaded networks. Flooding suffers from its inability, and LBP from its limited ability, to reduce the number of retransmitting nodes. LBP performs worse than simple flooding because of the required long buffering time of $10\,ms$ which causes more queue overflows. The number of rebroadcasting nodes are depicted in Fig. 8. Only MPR and DDB 1 remain unaffected by the packet generation rate, except that DDB 1 increases slightly for the highest rate. This is reflected by the increased delay and decreased delivery ratio in Fig. 7. Clearly, the number of retransmitting nodes of LBP and flooding decreases at least with the delivery ratio.

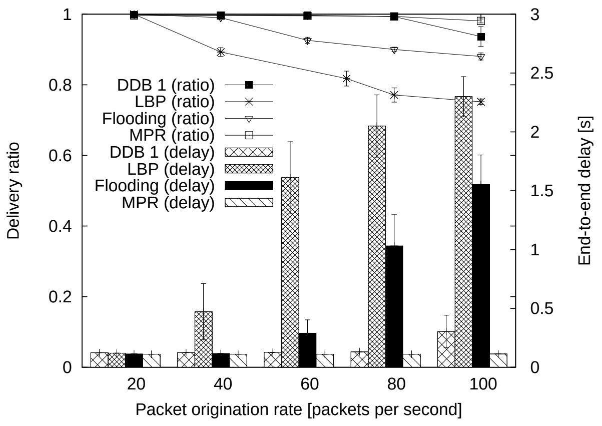


Fig. 7. Delivery ration and end-to-end delay for 19 neighbors

The results for an average of nine neighbors are given in Fig. 9. None of the protocols were able to deliver all the packets. Nodes are connected only over a few links and, thus, if packets are dropped at some nodes due to congestion, the packet can no longer be delivered reliably to all nodes. Second, the flooding improved in terms of delay and delivery ratio and was similar to DDB 1 because the smaller number of neighbors also reduces the number of possible collisions. Due to lack of space, the results are not shown for higher node densities. The only significant difference observed was that the delivery ratio of DDB 1 raises to almost 1 for all congestion levels. At the same time, the delay was reduced to similar values as for MPR.

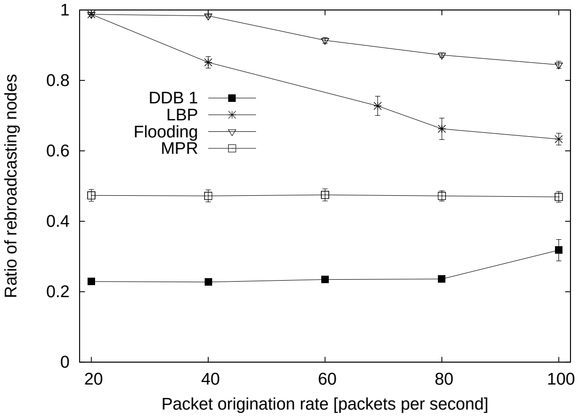


Fig. 8. Ratio of rebroadcasting nodes for 19 neighbors

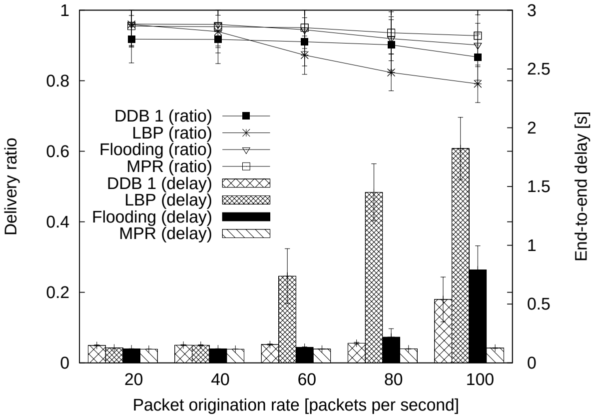


Fig. 9. Delivery ration and end-to-end delay for 9 neighbors

E. Mobile Networks

The simulation parameters are the same as in the congested network, i.e. 80 nodes over different simulation areas. Packets are generated at a rate of 10 packets per second. Nodes move according to the random waypoint mobility model. The pause time is set to $0\,s$ and the minimal and maximal speeds are set to $\pm 10\%$ of an average speed. The average speed was varied over $1, 5, 10, 20, 40\,m/s$. We also ran 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. 10 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 being slightly below 100% is due to the temporary partition of the network caused by mobility. 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 also causes an incorrect calculation of the forwarding nodes and the wrong neighbor rebroadcast. If the network density is low, already a few wrong rebroadcast decisions may prevent the packets from being delivered to all nodes in the network. For higher node densities, which is not shown in this paper due to lack of space, the performance of MPR did not decrease that significantly because the inconsistent view has a smaller impact. Packets can be still delivered due to the high connectivity, even if the ”wrong” nodes rebroadcast the packets. The delay is given in Fig. 11. Again due to the same reasons as already mentioned, DDB 1 yields the shortest delay among the three stateless protocols followed by LBP and flooding because of the higher number of rebroadcasting nodes. The confidence intervals are small for the stateless protocols as they are unaffected by mobility.

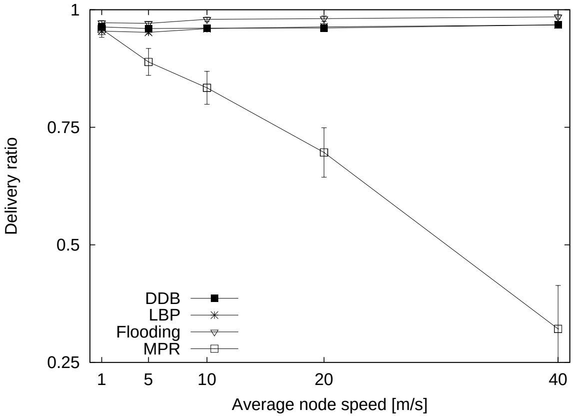


Fig. 10. Delivery ration

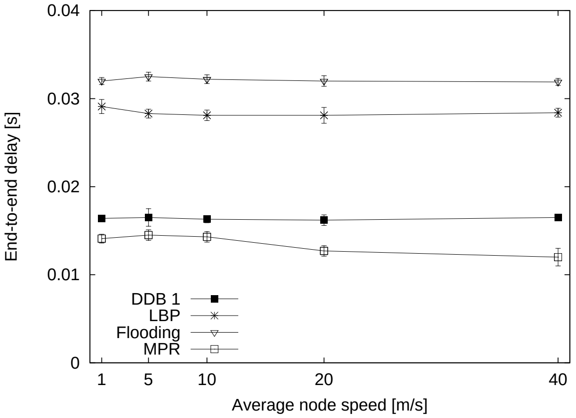


Fig. 11. End-to-end delay

F. Network Lifetime

In many network scenarios, 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. The network lifetime strongly depends on the consumed energy during 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 so that we may then expect nodes that transmit more frequently deplete sooner. For our simulation, the ratio of sending/receving/idle listening was set to 10/1/0.01. These values are justified by recent technology advances, cp. e.g. [36], which also allow even higher ratios. To avoid congestion, packets are again sent at a rate of one packet per second. We place 1000 nodes over an area of 2000x2000 m. Assuming that sending and receiving of a hello message consumes about the same energy as 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. 12, the second scheme DDB 2 where rebroadcasting decisions are solely based on residual battery power exhibits by far the longest time until the first nodes fail and outperforms significantly LBP and DDB 1. For a higher percentage of depleted nodes, DDB 1 shows longer network lifetimes than DDB 2 due to the smaller number of rebroadcasting nodes leading to a smaller total amount of energy consumed for each packet. This is achieved even under the fact that the number of rebroadcasting nodes is about the same for DDB 2 as for LBP, because the rebroadcast decision is independent of the additional covered area and, thus, much higher than that of DDB 1. However, the initially longer lifetime of DDB 2 comes at the cost of a longer delay as depicted in Fig. 13.

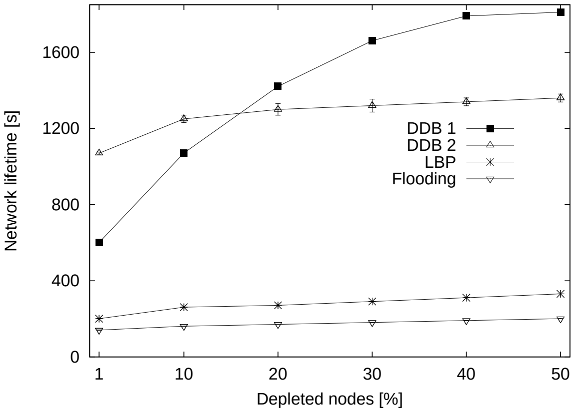


Fig. 12. Network lifetime until a certain percentage of nodes fail

VI. CONCLUSION

In this paper we presented the simple stateless DDB protocol, 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 to another stateless broadcasting protocols 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 [28]. 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.

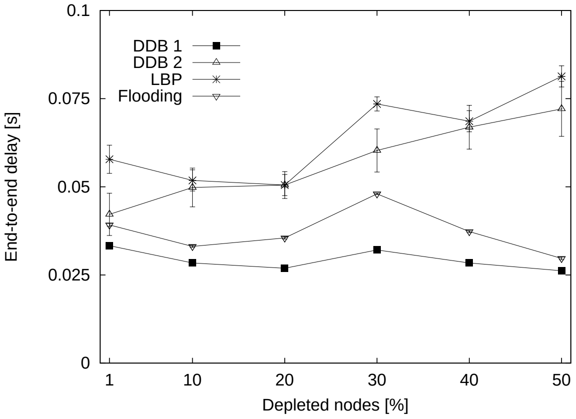


Fig. 13. Delay after a certain percentage of nodes failed

MPR performed well in most scenarios, except 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% longer, except in the case of highly congested networks. On the other hand, DDB outperformed MPR significantly considering the efficiency of the algorithm. DDB compared to MPR only required about half of the transmissions to deliver the packet reliably to all nodes. 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.

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, more sophisticated DFD functions, which may combine location information, signal strength, signal-to-noise ratio, bit error rate, etc., could help to further improve performance.

REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] F. Kuhn, T. Moscibroda, and R. Wattenhofer, "Initializing newly deployed ad hoc and sensor networks," in *Proceedings of the 10th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM '04)*, Philadelphia, USA, Sept. 2004.

[13] J. Gao, L. J. Guibas, J. Hershburger, L. Zhang, and A. Zhu, "Discrete mobile centers," in *Proceedings of the 17th ACM Symposium on Computational Geometry*, Boston, MA, USA, June 2001, pp. 188–196.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

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

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

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