DRB and DCCB: Efficient and Robust Dynamic Broadcast for Ad Hoc and Sensor Networks

Alireza Keshavarz-Haddad[†] Vinay Ribeiro[§1] Rudolf Riedi[†‡]

Department of Electrical and Computer Engineering[†] and Department of Statistics[‡]
Rice University, 6100 Main Street, Houston, TX 77005, USA
Department of Computer Science and Engineering, I.I.T. Delhi, New Delhi -110016, INDIA[§]

Technical Report TR-2006-09
Department of Statistics, Rice University

[1] § Vinay Ribeiro was postdoc at the department of statistics when this work was done.

Abstract

This work proposes two novel deterministic, timer-based broadcast schemes, *Dynamic Reflector Broadcast* (DRB) and *Dynamic Connector Connector Broadcast* (DCCB), which perform ideally in terms of the following three crucial measures of performance. First, deterministic broadcast schemes guarantee full reachability over an idealistic lossless MAC layer. Second, timer-based schemes stand out for their robustness against node failure as well as more general changes in the network topology. Third, DRB and DCCB operate provably within a factor of the optimal efficiency (number of rebroadcasts). To the best of our knowledge no other deterministic timer-based scheme possesses this property. NS-2 simulations employing the 802.11b MAC protocol confirm our analysis. The factor can be estimated to be quite small. DCCB requires only a given Dominating Set (DS) while the DRB operates on a Weakly Connected Dominating Set (WCDS) which improves its efficiency. Algorithms that construct DS and WCDS are well known. In addition, we propose a novel timer-based algorithm for building efficient WCDS, that will be analyzed and compared with existing algorithms. Novel to the proposed schemes is their *hybrid backbone* consisting of a given, static Dominating Set (DS) and a dynamically computed set of connecting nodes. As an additional contribution, this work studies the trade-off of timer settings (and thus latency) against number of rebroadcasts, as well as robustness of the proposed algorithms. To the best of our knowledge, no prior study of these issues exists.

Contents

List of Figures

Chapter 1

Introduction

Wireless Ad Hoc networks consist of nodes that communicate with each other through wireless channels without any fixed infrastructure, relaying information through other participating nodes via a multiple transmissions, called *hops*. Such networks include mobile Ad Hoc and sensor networks which have a host of applications, such as disaster relieve, communication in remote areas, and monitoring in hazardous environment, to name but a few.

In Ad Hoc networks, broadcast plays a particularly important role, relaying a message generated by one node to all other nodes. Broadcast is an integral part of a variety of protocols that provide basic functionality and efficiency to higher-layer services. Clearly, in the absence of fixed infrastructure, the Ad Hoc network must self-organize in order for nodes to locate and route messages to each other. In the presence of mobility, this location information needs to be constantly updated. Examples include coordinated and distributed computing, a prime task in sensor networks, multi-casting, and several unicast routing protocols such as Dynamic Source Routing (DSR), Ad Hoc On demand Distance Vector (AODV), Zone Routing Protocol (ZRP), and Location Aided Routing (LAR) [16].

One simple approach for broadcasting is *blind flooding*, where each node rebroadcasts a packet as soon as it receives it for the first time. While simple and effective, blind flooding also produces redundant broadcasts and wastes precious bandwidth and power. These detrimental effects become particularly dramatic in dense networks, where blind flooding leads to the so-called *broadcast storm* which manifests in heavy contention and collisions (see the pioneering work of [28]).

Broadcast algorithms aim at avoiding the broadcast storm by forwarding a broadcast only over a subset of nodes called *backbone*. The performance of such a broadcast algorithm is usually measured by three metrics: *efficiency* measured in *number of rebroadcasts* reflects directly on the bandwidth and power consumption, *reachability* reports the fraction of nodes that actually receive the broadcast packet, and *latency* indicates the time between first transmission and the first time the last node in the network received the broadcast.

The most common classification [38, 25] labels a broadcast scheme *probabilistic methods* and *deterministic methods*. As their defining difference, the probabilistic methods do not guarantee full-reachability, that is they reach only a random number of nodes [28, 14, 44, 5, 43, 36, 18]. Deterministic methods guarantee full-reachability assuming an ideal MAC layer. Examples include multipoint-relaying [31], marking [41, 11, 39, 6, 40], neighbor elimination [30, 34, 33], clustering [29, 20, 23, 42] and node-forwarding [4, 22, 24].

Also, broadcast schemes are divided into *static* ones, that always use the same backbone and *dynamic* ones that recompute a backbone for each broadcast in order to adapt to changing network topology and broadcast state [40]. In some dynamic schemes the nodes decide whether to join the backbone after hearing the packet once or twice [35, 22].

An important class of dynamic schemes is formed by the *timer-based schemes* where each network node starts a random timer upon hearing the broadcast and deciding upon timer expiration whether to rebroadcast based on the information collected from all overheard broadcast packets. Timer-based schemes include most probabilistic schemes (see [28, 36, 18]) as well as a few deterministic schemes (see [30, 34, 33]). A recent scheme uses non-random timers which depend on neighbor locations [15].

In this report, we concentrate on deterministic timer-based schemes, a class with attractive properties. By definition, their reachability is ideal assuming no loss at the MAC layer. In addition, being dynamic they are adaptive to network conditions and should be expected to be more robust to node failure and mobility than static schemes. As a result of their random timers, however, timer-based schemes incur a larger latency when compared to other schemes.

The contribution of this work is two-fold. First, we introduce two novel deterministic timer-based schemes: Dynamic Reflector Broadcast (DRB) and Dynamic Connector-Connector Broadcast (DCCB); both possess an efficiency within a factor of the optimum, a property which other deterministic timer-based schemes do not share and which we establish both analytically and in simulation. Moreover, we propose a novel distributed timer-based algorithm for building a WCDS which is used for DRB. Second, we study by simulation the effect of the settings of the random timer for existing and proposed deterministic timer-based schemes. To the best of our knowledge, this issue has not been addressed in the literature before. We summarize the technical report as follows.

The principal distinguishing feature of DRB and DCCB from other existing timer-based schemes is their *hybrid* backbone consisting of a fixed and a variable part. The fixed set of backbone nodes forms a *Dominating Set* (DS), i.e., it covers the entire network provided no loss occurs on the MAC/PHY layer. Hence, both algorithms are deterministic. The fixed DS is assumed to be given and efficient. The algorithms select the variable portion of the backbone nodes which have the task to connect the fixed given DS; they are selected randomly based on timers and overheard information; hence, both algorithms are timer-based.

The efficiency in terms of rebroadcasts of both schemes lies within a small factor of the optimum as our analysis shows. To allow for flexibility, DCCB assumes nothing about the DS. To gain yet another factor of efficiency, on the other hand, DRB assumes that the DS actually forms a *Weakly Connected Dominating Set* (WCDS), meaning that connecting all of its members which are within two hops of each other produces a *Connected Dominating Set* (CDS). Several algorithms for building an efficient WCDS or DS in a distributed fashion are known to exist [1, 37, 21, 3, 32]. In addition to these algorithms, we propose a novel timer-based algorithm to construct a WCDS called *crystal-DS algorithm.* This algorithm uses just one message transmission per node. The crystal-DS algorithm builds an WCDS in a manner similar to the growth of crystals. The DS originates at a single node (seed) and then expands with time to form an intricate structure that eventually covers the entire network. The resulting DS, which we call a *crystal-DS* , has the strengths of efficient WCDS algorithms [1, 37].

Our simulations with NS-2.29 [27] confirm our analytical findings regarding efficiency. Furthermore, they reveal the superior robustness to node failure of DCCB and DRB in terms of reachability. We suggest that robustness is a direct consequence of the hybrid nature of the backbone which allows for adaptively replacing nodes of the variable part that are no longer available. Efficiency, on the other hand, results from a clever scheme for connecting the given DS inherent to DCCB and DRB. Thereby, simulation indicates that DRB is particularly efficient, and DCCB particularly robust.

The effect of timer settings on performance has found only little attention in the literature so far. By simulation we find that the duration of the random timers controls a trade-off between the efficiency (number of rebroadcasts) and the latency for all timer-based schemes. Large timer averages increase the latency but reduce the number of rebroadcasts, while small timer averages show the opposite effect. Notably, NS-2 simulations show that the number of rebroadcasts quickly drops close to its minimum value as the average timer duration is increased beyond a critical threshold. This phenomenon can be explained by noting that with high average timer duration *redundant* rebroadcasts caused by the delay of MAC layer do not occur with probability close to one. Thus, beyond a critical threshold for the random duration, the performance of the scheme is sufficiently close to the idealistic one with high probability. We call these timer settings the *efficient regime* of a timer-based scheme. We compute the performance in the efficient regime assuming idealistic MAC/PHY layer, thus leveraging the power of simplistic assumptions all while establishing the relevance of our findings via the notion of the efficient regime. Moreover, we explain some potential extensions to DRB and DCCB algorithm for making them even more robust and adaptive to the node failure or topology changes in Ad Hoc or sensor networks.

The structure of this report is as follows. We describe existing timer-based schemes in Chapter 2. The WCDS and DS building algorithms and our novel timer-based algorithm are described in Chapter 3. We propose the DRB and DCCB schemes in Chapter 4. In Chapter 5, we study and compare the performance and robustness of the timer-based schemes through simulations. We analyze the backbone size of DRB and DCCB schemes over efficient DS and WCDS in Section 6. As extensions to our work, we propose two local repair algorithms on DRB and DCCB schemes for topology changing wireless networks in Chapter 7. Finally we conclude the paper in Chapter 8.

Chapter 2

Related Broadcast Schemes

In this Chapter we argue for our selection of broadcast schemes to which we compare to DRB and DCCB in Section 5. There exist only two deterministic timer-based broadcast schemes in the literature: *Scalable Broadcast Algorithm* (SBA) and the *Stojemnovic* scheme. Indeed, a timer-based approach brings about inherent elements of randomness which are easier to handle in a probabilistic scheme where full reachability does not have to be guaranteed. A comprehensive comparison of the entire host of existing probabilistic schemes is beyond the scope of the work and would distract from the actual objective of demonstrating the benefits of a hybrid backbone. For a list of probabilistic schemes [28, 36, 18] and existing comparative work see [38, 25] and references therein. Area- and distance-based schemes [28, 36], e.g., use actual geographic information about the nodes which naturally changes the tradeoff game. Notably, counter- and color-based [18] schemes are probabilistic and timer-based but use only the explicit or implicit information provided by overheard broadcast packets. The more recent color-based schemes exhibit performance similar to the counter-based ones [18]. At last, a new scheme was proposed [15] most recently, where non-random timers are set depending on neighbor locations such as to optimize efficiency or latency. We do not include this scheme in our comparison since their timers are non-random.

In summary, we find it appropriate and useful to focus in this work on the class of deterministic timer-based broadcast schemes and to add the counter-based broadcast as a representative probabilistic timer-based scheme for which a comparison is meaningful.

2.1 SBA scheme

The SBA scheme of Peng and Lu proposed the Scalable Broadcast Algorithm (SBA) to reduces the number of broadcast nodes as follows [30]. In this algorithm, the status of a forward node is computed on-the-fly. When a node v receives a broadcast packet, instead of forwarding the packet immediately, it waits for a random time. Denote the set of neighbors of node v by $N(v)$. For each of its neighbors w that has forwarded the broadcast packet, node v removes w and $N(w)$ from $N(v)$. If the resulting set of nodes does not become empty after the random time, node v forwards the broadcast node; otherwise node v does not forward the packet.

2.2 Stojmenovic scheme

Stojmenovic proposed a timer-based scheme for broadcasting [34, 33] based on Wu and Li's marking process [41]. This scheme improves Wu and Li's scheme in two ways: (1) Instead of using information of 2-hop neighbors for marking the nodes, it applies the geographic information, and uses only 1-hop information to implement the marking process. Therefore, each node only maintains a list of its neighbors and their geographic positions. Then it calls the marked nodes "gateways", the broadcast scheme only uses the gateways to forward the packet to all node in the network. (2) In addition, the set of forwarding nodes is further reduced by running a neighbor elimination algorithm over the gateways, similar to the one used in SBA.

2.3 Counter-based scheme

The counter-based scheme is a simple probabilistic scheme [28]. Recall that it does not guarantee that every node will receive the broadcast packet even when there are no collisions and other losses. In the counter-based scheme, when a node receives the broadcast packet for the first time it starts a random timer and counts how many times it overhears the same broadcast until expiration. If this number is less than some threshold η then it rebroadcasts the packet. Clearly, a larger η increases the probability of nodes rebroadcasting the packet and with it the reachability. However, large values of η also result in inefficiencies in term of number of rebroadcasts.

Chapter 3

Building Efficient DS and WCDS Structures

In this chapter, first we review some exiting works on building DS and WCDS. Moreover, we propose a distributed timer-based algorithm for building a WCDS that we call it crystal-DS .

3.1 Background on Dominating Sets

Many algorithms to build a WCDS have been proposed in the literature [1, 37, 7, 8, 12]. In some algorithms the generated WCDS based on an independent set. Consequently these algorithms keep the size of backbone at most a constant factor of the *Minimum Connected Dominating Set* (MCDS) [1, 37]. For example, the algorithm proposed by Wan, Alzoubi and Frieder build a WCDS using the minimum ID or the weights of the nodes on a rooted spanning tree, which is an independent set [37]. However, building and maintaining a near optimal WCDS is costly. Minor changes like failure or movement of a single WCDS node are not repairable locally, and require global reconstruction [9]. Note that finding an MCDS of a wireless network is an NP-complete problem [26], so such algorithms are used to estimate the MCDS.

The most common algorithms for constructing an efficient DS are clustering algorithms. These select a set of nodes as clusterheads. The set of clusterheads build an Independent Dominating Set (IDS) which is smaller than a constant factor of the MCDS size [21, 3]. There are also some algorithms that build a Maximal Independent Set (MIS) [19, 37] (MIS is also a DS) and can be classified as clustering algorithm. However, some clustering algorithms does not build necessary an IDS [32, 7, 8].

In some algorithms, the clusterheads (dominators) are elected based on the lowest node identifier (ID), or higher degree of connectivity [21]. More generally the clusterheads can be selected based on unique wights of the nodes [3]. The weight is a function of ID, connectivity degree, power, mobility and security of the node. The selection algorithm process as the following: an unmarked node which learns of a neighboring node marked as a dominator, marks itself a dominatee. An unmarked node which has the highest weight among its unmarked neighbors marks itself a dominator and marks its neighbors as its dominatees. Another way of electing dominators is to use random timers at the nodes. In this approach, an unmarked node backs off for a random time duration; if it hears a dominator during this time it becomes a dominatee, otherwise it marks itself as a dominator and its neighbors as dominatees upon timer expiration [2]. However, this algorithm needs synchronized transmissions in the network in order to build an IDS by the clusterheads.

3.2 The Crystal-DS Algorithm

We call the DS-build algorithm the *crystal-DS algorithm* because it builds a WCDS in a manner similar to the growth of crystals. The DS originates at a single node called a *seed* and then expands with time to eventually cover the entire network. The choice of the seed can be arbitrary, such as the first node that needs to broadcast a message. The crystal-DS algorithm requires no a priori information about the system and is extremely efficient; it performs only one message transmission per node.

During the building phase of the backbone all nodes are marked either as *dominators* or *dominatees*. All messages transmitted during the building phase contain the identifier of the node generating the message and announce how the node is marked. For short, we refer to the former as a *dominator message* and to the latter as a *dominatee message*. Initially all nodes are unmarked. Details of the construction follow now.

Crystal-DS Algorithm

1. Initially the seed transmits a dominator message and marks itself a

dominator.

2. Any unmarked node that receives a dominator message waits for a time interval of random duration and then transmits a dominatee message. It then marks itself a dominatee.

3. Any unmarked node that receives a dominatee message sets a timer of random duration. If the node hears a dominator message before expiration of the timer then it proceeds as in step 2; otherwise it sends out a dominator message and marks itself a dominator.

4. Each dominatee maintains a cache of dominators within its range. A dominatee with a pair of dominators A and B in its cache is called an *$A \leftrightarrow B$ reflector*. A dominatee can be a reflector for multiple pairs of dominators.

3.2.1 Properties of Crystal-DS Structure

The set of dominators created by the crystal-DS algorithm has the property that connecting 2-hop away dominators results in a CDS (i.e. the set is a WCDS). For the proof, note that by the end of crystal-DS algorithm, all nodes will be marked either as dominators or dominatees. This is a consequence of our assumption that the network is connected. The WCDS property of the set of dominators results from the following lemma.

Lemma 1 *For any pair of dominators in the crystal-DS there exists a path connecting them on which dominators and reflectors strictly alternate.*

The lemma implies that one does not have to pass through two reflectors in a row to forward a message from one dominator to another.

For a formal proof of the lemma, consider any dominator D other than the seed and denote by R the first node from which D heard a dominatee message. In addition, label as $D \uparrow$ the first dominator from which R heard a dominator message. Clearly D and $D \uparrow$ have a two-hop path between them through R. Label the operation of traveling from D to $D \uparrow$ through R as $\uparrow$. Starting from any dominator D, repeating the operation $\uparrow$ a finite number of times will trace a path to the seed. By combining two such paths to the seed, one from dominator D and one from D', we obtain a path connecting D and D' that consists solely of two-hop paths between dominators.

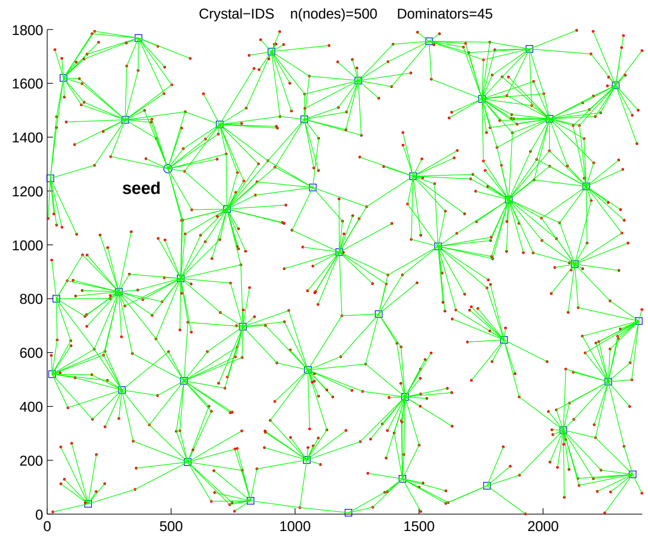


Figure 3.1: A simulation of a network with uniformly distributed nodes. Note that the Independent Dominating Set (IDS) generated by the crystal-DS algorithm is well-spread within the network. Several dominatee nodes can act as reflectors between pairs of dominators.

In addition, we can show that the set of dominators build an IDS, under the assumption of ideal MAC. Because all neighbors of a dominator must necessarily be dominatees, no two dominators are within hearing range of each other. Thus the dominators form an IDS. Also, note that by finding paths from each dominator to the seed as described in the proof, we build a CDS with a sparse tree structure.

Figure 3.1 depicts the Seed node (circle) and Dominator nodes (squares) in a homogeneous network created by the crystal-DS algorithm (see later for a definition of homogeneous). Observe that the dominators are well-spread throughout the network. Note also the presence of several reflectors between any pair of neighboring dominators, a fact we will exploit while performing broadcast.

3.2.2 Crystal-DS Algorithm vs. Clustering Algorithms

Unlike the crystal-DS algorithm, the clustering algorithms described in Section 3.1 cannot build a WCDS. There are two reasons for this. First, the crystal-DS algorithm exploits more global information than the clustering algorithm. In the crystal-DS algorithm we start at a single node from which the DS grows outward. This growth process implicitly collects global information about the network that the helps build a timer-based WCDS. However, note that each node decides to become a dominator or dominatee based only on local information. The clustering algorithm, in contrast, starts in a distribution fashion and its growth process does not contain any global network information.

Second, the crystal-DS algorithm uses both positive and negative information whereas the clustering uses only negative information to build an DS. In the crystal-DS algorithm a node joins the DS if it knows that there exists another dominator two hops away *and* that there does not exist a dominator 1-hop away. In the clustering scheme, nodes only use information that there is no dominator 1-hop away. This extra information that the crystal-DS algorithm uses helps it build an intricate structure such as that depicted in Figure 3.1.

Chapter 4

DRB and DCCB Schemes

The major difference of our proposed schemes to existing ones lies in their hybrid broadcast backbone. In a nutshell, the proposed algorithms assume a given static Dominating Set (DS), respectively a Weakly Connected Dominating Set (WCDS) for which they build a set of connecting nodes for each broadcast on the fly. To this end, the algorithms employ a distributed marking algorithm. The idea of a hybrid backbone is novel. The first, called *dynamic reflector broadcast* (DRB) relays packets on a WCDS set for broadcasting; the choice of WCDS is arbitrary. The only requirement is that nodes need to be informed about their WCDS neighbors. The second, called *dynamic connector-connector broadcast* DCCB, relays packets on any DS for broadcasting. In DCCB nodes need to be informed about 1 and 2-hop away DS nodes. The same idea have been used in [10] to distribute the work load among nodes systematically for increasing the life time of sensor nodes.

4.1 Dynamic Reflector Broadcast (DRB)

The DRB algorithm assumes that a WCDS has been computed and that all *network* nodes know their WCDS neighbors. The nodes of the WCDS are called *dominators*; further, the network nodes which have more than one dominator as neighbor are called *reflectors*. The scheme works as follows.

Dynamic Reflector Broadcast Algorithm

1. The broadcast may be initiated by any node by simply broadcasting a

packet to all its neighbors.

2. Any dominator that hears a particular broadcast packet for the first time rebroadcast the packet after a short delay. It also appends to the packet the list of all neighboring dominators (including itself) and removes any other appended list. A typical setting is a delay uniformly distributed in the interval $[0, T_D]$.

3. Any reflector that hears a particular broadcast packet for the first time starts a timer of random duration. A typical setting is a uniform distribution in the interval $[0, T_R]$ where T_R is much larger than T_D. Until its timer expires the reflector listens to all rebroadcasts of the same packet and logs as "done" the sending dominators corresponding the target dominators listed in the packet. Upon expiration of the timer it checks whether any of its neighboring dominators is not logged as "done". If not, it takes no further action. If so, it rebroadcasts the packet to all its neighbors. It also appends to the packet the list of all its neighboring dominators and removes any other appended list, in order to inform all neighboring reflectors.

Some notes on parameter settings are in order. The timer for dominators T_D helps avoiding collisions between dominators which may receive the broadcast within short time of each other. For a network with collision free MAC layer, T_D can be set to zero. The timer for reflectors T_R will be set depending on density of the reflectors, packet size and background traffic (see the in-depth discussion in Section 5).

For an illustration we refer to Fig 4.1 where the dominators are indicated by squares, the reflectors by red circles and the originating node by S. The packet may take different paths, depending on the realized timer values. For example, if $R4$ happens to have the shortest timer among $R2 - R5$, then the broadcast begins its path as $S \to D3 \to R4 \to D1$. If even the timer of $R1$ expires before the one of $R2$ then the packet's path is $S \to D3 \to R4 \to D1 \to R1 \to D2$ with the obvious bifurcation at $D3$: $\ldots \to R5 \to D4 \to R6 \to D5$. Ideally, no other rebroadcasts occur; in particular, $R2$ and $R3$ will take no action, providing they are being informed of the broadcast state by the relay of $R4 \to D1$ and the broadcast of $D2$, respectively, before their timers expire.

In other words, under the assumption of an ideal MAC layer, the broadcast will happen along a tree which is, however, random. In reality, MAC and

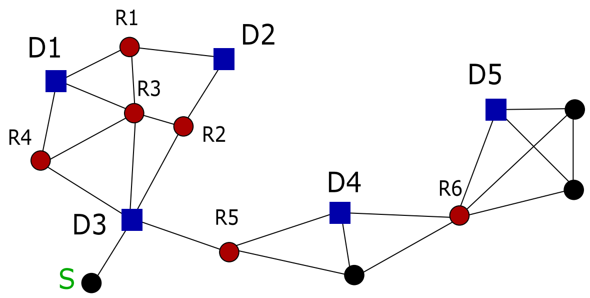


Figure 4.1: Set of dominators and reflectors for DRB scheme. The broadcast packet can reach $D2$ through the paths $D3 \rightarrow R2$, $D3 \rightarrow R3 \rightarrow R1$ and $D3 \rightarrow R4 \rightarrow D1 \rightarrow R1$.

physical layer consume time to transmit the packet due to background traffic and reflectors may broadcast redundant packets. To continue the above example (see Fig 4.1) if the timers of reflectors $R3$ and $R4$ end shortly after each other, the networking layers of both reflectors may hand the packet to their MAC layer for broadcast (to serve $D1$) before hearing the other's rebroadcast. In Section 5 we will study timer-based schemes with realistic MAC layer, and propose some techniques to avoid this cross-layer problem.

Another important property of DRB is:

Lemma 2 *In a network with lossless MAC layer, DRB reaches all nodes.*

Proof: By design of DRB, the dominator of the originating node must eventually rebroadcast the packet. Also by design, the broadcast reaches the set of dominators which are within 2-hops of each other and which contains the originator's dominator. By definition of a WCDS, this set is the entire DS. Since, all dominators rebroadcast the packet once, all nodes are covered.

4.2 Dynamic Connector-Connector Broadcast (DCCB)

The DCCB algorithm assumes that a DS has been computed, whose members are again called *dominators* and that all *network* nodes know the dominators within 1 or 2 hops distance. Note that it is necessary to connect all dominators within 3 hops of each other in order to guarantee a CDS over a lossless

MAC layer. The network nodes which have more than one dominator within at most 2 hops distance are called *connector*. The scheme works as follows.

Dynamic Connector-Connector Broadcast Algorithm

1. Every network node A compiles the lists of dominators that are exactly 1-hop and 2-hops away, denoted by $L_1(A)$ and $L_2(A)$ respectively. If A is itself a dominator it adds itself to the list $L_1(A)$. Every node A appends these lists $L_1(A)$ and $L_2(A)$ to the broadcast message in the packet when rebroadcasting it, removing any other appended lists.

2. When a dominator hears a broadcast packet for the first time it rebroadcasts the packet after a short random delay. In a typical setting, the delay is uniform in the interval $[0, T_D]$.

3. When a connector node A first hears a packet m, it starts a random timer. In a typical setting the duration is uniform in the interval $[0, T_C]$ where T_C is much larger than T_D. The connector generates new lists $L_1(A, m)$ and $L_2(A, m)$ specific to the packet m which it initializes to $L_1(A)$ and $L_2(A)$ respectively.

4. Whenever a connector A hears a broadcast of the packet m, say by node B, including the first time it hears m, it deletes all elements of $L_1(A, m)$ and $L_2(A, m)$ that belong to $L_1(B)$. Being in range of B, these dominators are dealt with. It also deletes from $L_2(A, m)$ any elements that belong to $L_2(B)$.

5. If at timer expiration, either $L_1(A, m)$ or $L_2(A, m)$ are non-empty, node A rebroadcasts the packet.

Similarly as with DRB, the parameter T_D can be set to zero for a network with collision free MAC layer. We consider random timers for dominator to avoid collisions between dominators which receive the broadcast packet shortly after each other. The parameter T_C should be set depending on node density in the network, packet size and background traffic. We refer to the discussion in Section 5.

For a demonstration of the algorithm we refer to Figure 4.2 which shows a set of dominators (squares) and connectors (pink circles), as well as the node S which originates the broadcast. As with DRB, the broadcast packet may take several different paths according to which timers expire first, resulting

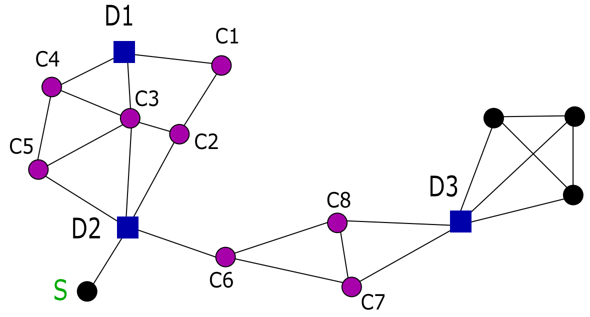


Figure 4.2: Set of dominators and connectors in DCCB scheme. The broadcast packet can reach $D1$ through the paths $D2 \rightarrow C3$, $D2 \rightarrow C5 \rightarrow C4$ and $D2 \rightarrow C2 \rightarrow C1$.

in backbones that vary randomly according to the timer values as well as others random effects including transmission delay caused by the MAC layer and background traffic.

The need to connect dominators within 3 hops of each other (see, e.g., $D2$ and $D3$ in Figure 4.2) leads to a source of inefficiencies not present with a WCDS. If $D1$ receives the packet via $D2 \rightarrow C5 \rightarrow C4$, e.g., then $C2$ will still attempt to forward the packet to $D1$, which is an example for redundant connector between 3-hop away dominators.

Again, an important property is easily established.

Lemma 3 *In a network with lossless MAC layer, DCCB reaches all nodes.*

Proof: In DCCB algorithm, when a dominator transmit the packet, the connectors forward the packet to dominators in 2-hop and 3-hop away unless the dominators already has received the packet. So, when the algorithm finishes, there will be no dominator which has not received the packet and a dominator distance in 2-hop or 3-hop away has the packet. Since, at the beginning there is a dominator which transmit the packet, the DCCB leaves no dominator unreached. Since, all dominators rebroadcast the packet once, all nodes are covered.

To **summarize** or design choices, the hybrid nature of both DRB and DCCB are motivated combining two benefits, robustness from the dynamic part of the backbone, and efficiency from the static part. DRB is expected to perform particularly efficient due to its tree shaped backbone under idealistic MAC layer assumptions. DCCB, on the other hand, can relay on any DS and is expected to be particularly robust due to the larger set of connectors

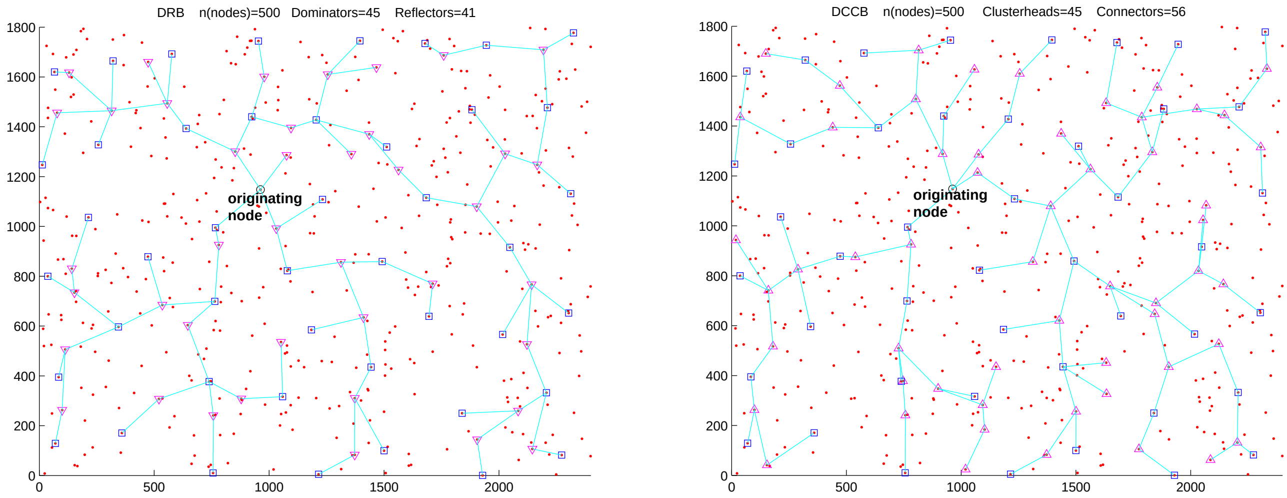


Figure 4.3: Generated random spanning tree for a broadcast packet by DRB and DCCB schemes in a homogeneous network.

between dominators. In addition, generated DS by clustering algorithms are locally repairable [3]. Therefore, DCCB could show desirable performance in Ad Hoc networks with fast topology changes. Given near optimal DS, both algorithms should provide backbones within a factor of the optimum. All these conjectures are established by analysis and simulation in the remaining Chapters.

Chapter 5

Timer-based Schemes in Simulation

In this section we study the performance of different timer-based schemes for a homogeneous dense network by simulation. The metrics used are efficiency, i.e., number of rebroadcasts, latency, and reachability. We also address robustness, characteristics of timer-based schemes to be noted in simulation as well as a the potential for a cross-layer design to mitigate the effects of using timers on latency.

In all simulations we employ the NS-2.29 simulator [27]. Nodes are uniformly distributed in a 2400m by 1800m area and each node has the default radio range of 250m. We employ the IEEE 802.11b MAC protocol with basic transmission rate 1Mbps and broadcast packet size of 125 Bytes. All results have been computed by averaging the performance over 200 realization for each scheme, by varying the originating node of broadcast randomly among the nodes. Packet size is kept constant and no background traffic is present in any simulation. Clearly, both factors increase the time from reception to eventual relaying of a broadcast packet. Though an in-depth study is beyond the scope of this paper, some first order valuable insight into the effect of these factors on performance could potentially be gained for timer-based schemes by conceptually absorbing the additional random delays into the timers.

5.1 Using Timers: Trading off Latency for Efficiency

Using timers constitutes a clever mechanism to detect network topology on the fly with the potential for superior performance at the price of latency, as we will demonstrate. Less obviously, timer-based schemes perform less efficiently over realistic MAC and physical layers with non-zero packet transmission time, contention and collision than if these layers were idealistic.

Indeed, in a timer-based scheme, the network layer of the node may decide at timer expiration to rebroadcast the packet based on information collected from all overheard broadcasts of the same packet. It forwards the packet to the MAC layer where it is temporarily buffered. Before the MAC layer transmits the packet the node may obtain new information from further rebroadcasts which would reverse its decision to rebroadcast. However, in the absence of any cross-layer design, the broadcast scheme (network layer) cannot stop the MAC layer which results in a redundant transmission. The longer the waiting time in the MAC queue as compare to the network timer, the higher the chance for such redundancy. Clearly, the chance increases by adding to the volume of background traffic, using large broadcast packets, reducing the capacity and specially increasing density of the nodes.

Essentially, keeping all other factors constant, increasing the average of the network layer timer results in fewer redundant transmissions. Simulation results depicted in Figure 5.1(a) confirm this intuition. We used 500 nodes with uniformly distributed timer, and varied the average timer length. The trade-off between the number of broadcast nodes and latency for all timer-based schemes is only too obvious, with very small timers essentially resulting in a *blind flood*. Note that each point on the plots shows the average number of rebroadcasts, reachability and latency of a timer-based scheme over 200 realizations.

5.2 Efficient Regime

While small time settings result practically in a blind flood according to Subsection 5.1, the number of rebroadcasts quickly drops close to its minimum value as the average timer duration is increased beyond a critical threshold. This is explained by the observation that redundancy caused by the lower layers occurs with probability practically zero, thus resulting in a performance close to that of an idealistic MAC/PHY layer. Note that the critical threshold depends on density, MAC/PHY layer and background traffic in the network. We call timer settings that result number of rebroadcasts within an acceptable predefined tolerance of the one over idealistic MAC/PHY layer the *efficient regime* of a timer-based scheme. Typically, also the reachability is close to idealistic in the efficient regime.

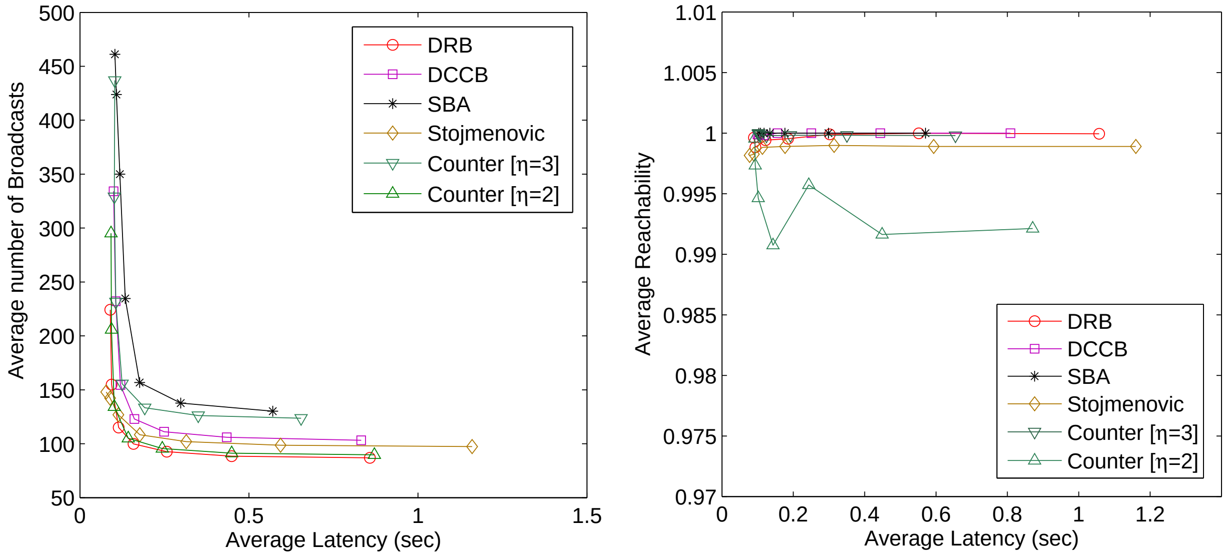
The relevance of the efficient regime for theoretical work lies in establishing a regime of timer settings where the performance can be assessed within an acceptable tolerance by assuming idealistic MAC/PYS layer. In practice, an efficient regime with acceptable tolerance is found by simply letting the timer values be large enough such that the scheme performs beyond the distinct "knee-point" in the performance plots (see Figure 5.1(a)). The efficient regime provides also a reasonable setting of timer values for a meaningful empirical comparison of broadcast schemes.

A prominent finding of our simulations is that DRB uses fewer number of rebroadcasts than other timer-based schemes for any given average latency in the network. Note that this result does not change when we vary the packet size though we do not show these results here due to space limitations. Assuming settings in the efficient regime for the moment, the counter-based scheme with threshold $\eta = 2$ is competitive, but has inferior reachability (see Figure 5.1(b)). With threshold set to $\eta = 3$, however, it is less efficient than DCCB and the Stojmenovic schemes. Finally, as we will show later in this section, as the density of nodes increases, DCCB becomes more efficient than the Stojmenovic scheme.

Outside their efficient regimes, i.e., for the small timer values with performance left of the "knee-points", the schemes use a large number of broadcasts (see Figure 5.1(a), left part). The Stojmenovic scheme has less number of rebroadcasts than other schemes for very short timers, because it only uses subset of nodes (the "gateways") for broadcasting. In practice, the schemes must employ sufficiently large average timer values to ensure that they are efficient an issue which we address via a suggestion of a cross-layer at the end of the section.

5.3 Impact of Node Density on Performance

We next analyze how the performance of timer-based schemes in their efficient regimes depends on node density. To vary the density we change



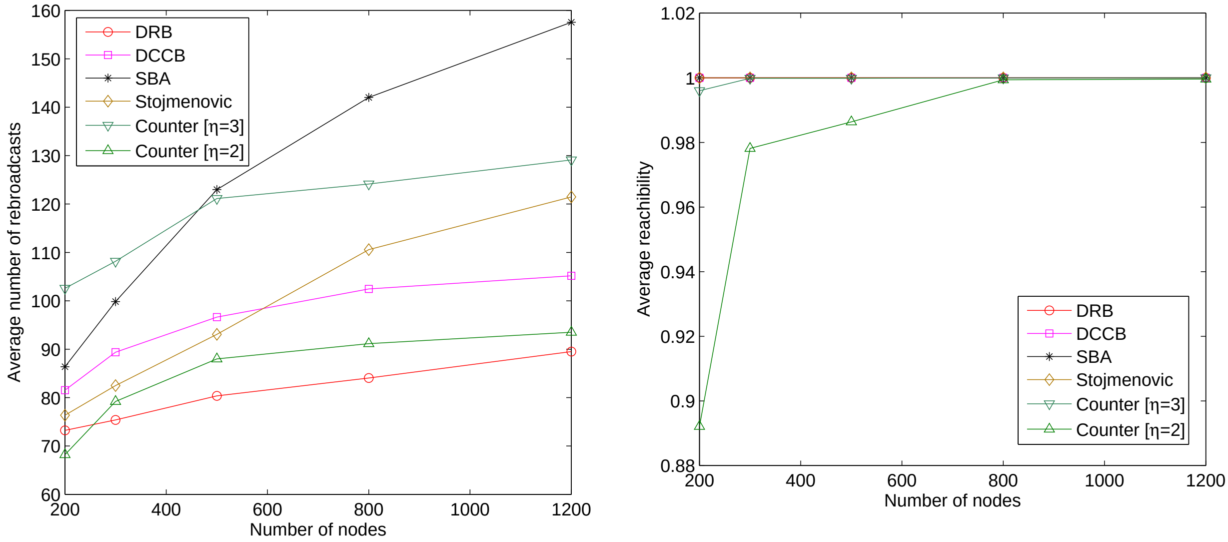
(a) (b)

Figure 5.1: Efficiency and reachability of timer-based schemes against latency.

number of nodes in the area from 200 to 1200. Figure 5.2(a) indicates that the number of rebroadcasts for DRB and DCCB remains roughly constant as the density increases; in fact, it depends only on the area of homogeneous dense network as we will show in Section 6. In contrast, as our results indicate, the number of broadcast nodes increases with density for the SBA and Stojmenovic schemes. Our results also verify that the number of broadcast nodes in the counter-based schemes is bounded by a function of the area of the network as has been established in previous work [18]. For a comparison of performance see also Figure 5.2(b).

5.4 Robustness to Node Failure

Though hardly addressed in the literature, node failure can be detrimental to broadcast schemes. Nodes can fail due to loss of power, being switched off, or malfunction. Here, we study the robustness of broadcasting schemes in terms of the average efficiency and the average reachability for different



(a) (b)

Figure 5.2: Efficiency and reachability in efficient regimes against node density.

probabilities of node failure.

Static schemes with sparse backbones are particularly sensitive to the node failure which may cause a disconnection of part of the backbone and consequently a low reachability. Hence one way to improve robustness of static schemes is to trade it off against efficiency and to increase the redundancy of the backbone.

Being dynamic, timer-based schemes are quite robust to node failure. In a way, timer-based schemes enjoy the robustness of a static scheme with a very redundant backbone, but eventually only a small subset of these nodes get used.

We simulated a network of 500 nodes. Node failure is assumed to occur with equal probability p for all nodes and independently of each other. Figs 5.3(a) and 5.3(b) compare the average efficiency and reachability of the schemes for different p. Notably, SBA achieves full reachability because it potentially includes all network nodes in the broadcast backbone, it will ensure that the packet reaches all nodes the network is connected.

DRB and DCCB both achieve excellent reachability. However, they can-

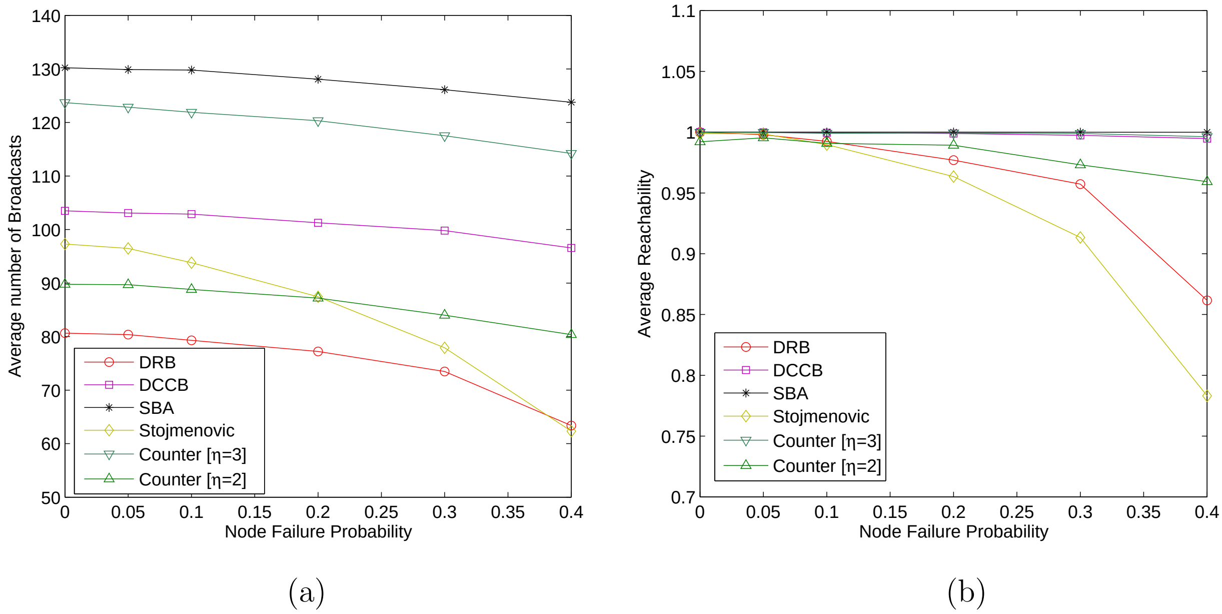


Figure 5.3: Efficiency and reachability in efficient regime against node failure probability.

not guarantee full-reachability like SBA because part of their backbones (the corresponding WCDS and DS) are predetermined before the broadcast and are not replaceable during a broadcast.

Counter-based schemes also enjoy good reachability. The Stojmenovic scheme, however, is less robust when compared to other schemes, because its potential backbone is smaller the potential backbone of other schemes.

5.5 Cross-Layer Design to Guarantee Efficiency

We finally propose a cross-layer design that allows timer-based schemes to perform in their efficient regimes even for very small average timer values. As we discussed in section 5.1, when the delay in the MAC queue is comparable to the average timer values used by the timer-based scheme at the network layer, a node may learn that no action is required of it only after handing the broadcast packet to the MAC layer, unable to stop the redundant transmission.

One solution is to give the network layer, and hence the broadcast scheme, the ability to annihilate packets that are present in the MAC queue. Alternatively, the MAC layer could be given the ability to check with the network layer whether to drop a broadcast packet just before transmitting the packet. The potential gains in terms of efficiency can be read off directly from Figure 5.1(a): the knee-points moves to the left and the efficient regime of the schemes is reached for a very small latency.

Chapter 6

Analysis: Efficiency and Complexity

Among the performance metrics for broadcasting in wireless networks, its efficiency and overhead stand out for their relevance. Indeed, the number of transmissions directly impacts power consumption, bandwidth utilization, and latency. Using fewer nodes for broadcasting reduces the chance for contention and collision which waste resources and add to the latency. This fact has been proved analytically in [17]. The same paper shows also that a broadcast scheme has throughput within constant factor of broadcast capacity if and only if the backbone size is within a constant factor of MCDS. Here, we prove that DRB and DCCB backbones have this property in their efficient regimes.

Our analysis here builds on the following two idealistic assumptions[1] which are at the basis of most existing analytical work in this field. We assume, first, that the MAC layer is collision free and, second, that the transmission time of packets is zero. While these assumptions are idealistic in general, we point out that in their efficient regimes and in the absence of background traffic timer-based schemes perform roughly as they would under the above stated first assumption (cpre. section 5). Indeed, large timer values or an appropriate cross-layer design would practically annihilate the chances for redundant rebroadcasts caused by transmission delays below the network layer.

[1] We stress the fact, that our simulations did *not* impose these assumptions and employed the realistic representation of the MAC layer as implemented in ns-2.

In order to address the optimal performance of the schemes we restrict our attention to static (pre-determined) portions of the backbones which are near optimal. More precisely, we impose that the DCCB scheme operates with a near optimal DS computed by a clustering algorithm [21, 3], and that the DRB scheme operates with a near optimal WCDS generated by a [37].

6.1 Number of Rebroadcasts

Denote the size of near optimal WCDS and DS as N_{WCDS} and N_{DS} respectively. Here, we compute the number of rebroadcasts of DRB and DCCB under the assumptions explained above in terms of N_{WCDS} and N_{DS}. We also compute the *approximation factor* which is the ratio of the number of rebroadcasts to that of the MCDS size.

6.1.1 Number of rebroadcasts in DRB

We recall that for idealistic MAC/PHY layer the DRB broadcast algorithm guarantees that every reflector transmits only if the packet reaches a *new* dominator, so it uses at most $N_{\mathrm{WCDS}} - 1$ reflectors. For a more generally valid worst case bound on the approximation factor consider a dominator that fails or rebroadcasts the packet after a long delay. Then, several independent reflectors may try to forward the packet to it. However, still assuming the ideal MAC layer imposed on this section these reflectors can not be within range of each other, and their number is at most 5 for each dominator by simple geometry. Thus, the maximum number of rebroadcasts is $(5 + 1)N_{\mathrm{WCDS}} = 6N_{\mathrm{WCDS}}$. Combining with the bound for the independent sets of [1] in terms of MCDS size the approximation factor is at most $4 * 6 = 24$.

6.1.2 Number of rebroadcasts in DCCB

For a generally valid worst case bound on the approximation factor consider all connectors which directly and indirectly forward the packet to the same dominator. However, still assuming the ideal MAC layer implies that direct connectors and indirect connectors build two independent sets. We can show by geometric arguments that there are at most 5 direct and 18 indirect connectors for each dominator. Thus the number of nodes used for broadcasting is at most $(1 + 5 + 18)N_{\text{DS}} = 24N_{\text{DS}}$. Again, from [1] the approximation factor is at most $4 * 24 = 96$.

6.2 Complexity and Overhead

First, we show that the length of the list which is appended to the broadcast packets is bounded. Since, dominators of near optimal WCDS or DS build an independent set, number of dominators in 1-hop away are 2- hop away from a fixed node is at most 5 and 18 respectively. So, the length of the appended list is at most $5S$ in DRB and it is at most $(5 + 18)S$ in DCCB, where S is the length of array which represent the ID of a dominator. Moreover, it shows that the amount of computation that is imposed to a node after hearing a broadcast packet is bounded (at most 5 comparisons in DRB and 23 comparisons in DCCB).

Second, we show that the number of overhearings of the same broadcast packet is bounded for every node. A node only hears the transmissions that target some dominators which are at most within 2 hops from the node in the DRB scheme, and within 3 hops in DCCB. The number of such dominators is less than 25 in DRB and 49 in DCCB [1]. Moreover, the schemes imply that no two neighbor nodes target the same dominator, which implies that the number of transmissions heard by a node and which target the same dominator is less than 5. Therefore, the number of overhearings is less than $5 * 25$ in DRB and $5 * 49$ in DCCB.

From the two arguments above, we conclude that a broadcast packet imposes at most $O(1)$ computation to every node and therefore the total computation of the nodes is $O(n)$, where n is the number of nodes. Note that here we do not consider the complexity of building a near optimal WCDS or DS. It has been shown that the complexity of the WCDS built by the algorithm proposed in [37] is $O(n \log(n))$ and the DS build by the algorithm proposed in [21] is $O(n)$.

6.3 Analysis in Homogeneous Dense Networks

The analytical results can be computed more accurately in homogeneous dense networks using some additional geometric properties of these networks. Here, we use the following notation: the network consists of n uniformly distributed nodes in a planar area of size A; extensions to higher dimensions are straightforward.

6.3.1 A general benchmark: Minimum CDS

A Minimum Connected Dominating Set (MCDS) uses the minimum number of nodes required to broadcast a packet to all nodes in the network. Finding an MCDS of a wireless network is an NP-complete problem [26]. However, we are able to provide a simple lower bound for the number of nodes of an MCDS, and thus of *any* CDS.

Theorem 1 *Let $\mathcal{M}$ denote the MCDS of a homogeneous dense network. Then*

$$\mathbb{E}[\#\mathcal{M}] \overset{(a)}{\geq} 1 + \frac{A - \pi R^2}{(\pi/3 + \sqrt{3}/2)R^2} \overset{(b)}{\simeq} 1.6420 \frac{A}{\pi R^2} \tag{6.1}$$

The bound (a) becomes tight in the limit of an infinitely dense network. The approximation (b) holds as $(A/R^2 \to \infty)$.

Proof of theorem 1 First, we find a lower bound for a connected random homogeneous network. Then we explain how bound will become tighter for a dense homogeneous network.

Denote, $S = \pi R^2$ and $S' = (\pi/3 + \sqrt{3}/2)R^2$ which are the area covered by first broadcast and the maximum area covered by new rebroadcasting [28]. Also, denote $M = \lceil \frac{A-S+S'}{S'} \rceil$ which is the minimum number of connected nodes in a network which can cover the area A.

$$\begin{aligned}
\mathbb{E}[\#\mathcal{M}] &= \sum_{k=1}^{\infty} P[\#\mathcal{M} \geq k] \\
&= \sum_{k=1}^{\infty} P[\text{nodes not covered by } k \text{ broadcasting}] \\
&\geq \sum_{k=1}^{M-1} P[\text{one node is uncovered}] \\
&\geq \sum_{k=1}^{M-1} \frac{A - S + (k-1)S'}{A} \\
&= (M-1)\frac{A-S}{A} - \frac{(M-1)M}{2}\frac{S'}{A}
\end{aligned}$$

$$\simeq \frac{A}{2S'} = .82\frac{A}{\pi R^2}$$

The computed lower bound becomes larger in the homogeneous and very dense networks. Since, $P[\text{nodes not covered by } k \text{ broadcasting}] = 1$ for $0 < k < M$, because the area is not fully covered, so $\#\mathcal{M} \geq M$

$$\mathbb{E}[\#\mathcal{M}] \geq M \simeq 1.64\frac{A}{\pi R^2}$$

6.3.2 Size of Cluster-DS and Crystal-DS structures

We start with a useful result that exploits *independence* of a set of nodes, meaning that no two nodes of the set are within range of each other.

Theorem 2 *Let $\mathcal{I}$ denote an independent dominating set of a homogeneous dense network. Then*

$$\frac{A}{\pi R^2} \overset{(a)}{\leq} \mathbb{E}[\#\mathcal{I}] \overset{(b)}{\leq} 3.6276\frac{A}{\pi R^2} \quad (6.2)$$

The bound (a) holds in the limit of an infinitely dense and (b) holds in the limit ($A/R^2 \to \infty$).

Proof of theorem 2: The bounds of Theorem 2 are resulted from two Lemmas 4 and 5.

Lemma 4 *Assume that $\mathcal{I}$ is a dominating set in a homogeneous dense network. Then*

$$\mathbb{E}[\#\mathcal{I}] \geq \frac{A}{\pi R^2} \quad (6.3)$$

Proof of lemma 4: Here, we use the same method for the proof of theorem 1. Denote, $M = \lceil \frac{A}{S} \rceil$, which is the minimum number of nodes needed to cover the area A.

$$\begin{aligned}
\mathbb{E}[\#\mathcal{I}] &= \sum_{k=1}^{\infty} P[\#\mathcal{I} \geq k] \\
&= \sum_{k=1}^{\infty} P[\text{nodes not covered by } \ k \text{ dominators}] \\
&\geq \sum_{k=1}^{M-1} P[\text{one node is uncovered}] \\
&\geq \sum_{k=1}^{M-1} \frac{A-kS}{A} = M - 1 - \frac{(M-1)M}{2}\frac{S}{A} \\
&\simeq \frac{A}{2S} = .5\frac{A}{\pi R^2}
\end{aligned}$$

Similar to the proof of last theorem, when network is homogeneous and very dense, $P[\text{nodes not covered by } k \text{ broadcasting}] = 1$ for $0 < k < M$. Thus, $\#\mathcal{I} \geq M$ and

$$\mathbb{E}[\#\mathcal{I}] \geq M \simeq \frac{A}{\pi R^2}$$

Lemma 5 *Assume that $\mathcal{I}$ is an Independent Set in a homogeneous dense network. Then*

$$\mathbb{E}[\#\mathcal{I}] \leq \frac{\pi\sqrt{3}}{6}\frac{4A}{\pi R^2} = 3.6276\frac{A}{\pi R^2} \tag{6.4}$$

Proof of lemma 5: The nodes of IS are not in the radio range of each other, so half radio range circles around these nodes are disjoint. When A is large in compare to circles, by the circle packing theorem at most $A\pi\sqrt{3}/6$ can be covered by the circles. So, it gives us an upper bound for number of circles (independent nodes) in the area A

$$\mathbb{E}[\#\mathcal{I}] \ \leq \frac{\pi\sqrt{3}}{6}\frac{A}{\pi(R/2)^2} = 3.6276\frac{A}{\pi R^2}$$

Theorem 2 provides useful bounds for the number dominators for independent cluster- and crystal-DS structures which lies at the basis for deriving efficiency bounds for DCCB and DRB. Next, we add considerably sharper results which hold, however, only asymptotically (for DCCB) or are obtained from simulation (for DRB).

Cluster-DS: Assuming in addition to a homogeneous and dense network that the clustering scheme chooses the dominators locally, we may apply the classical theory for the so-called *disc parking problem* [13]: Here, discs are "arriving" iteratively in the plane at independent random locations. They "park", or remain, if they don't intersect with any of the already parked discs; otherwise, they "leave". The process continues until no other disc can possibly be parked; the system is said to be in the "jamming limit". In the cluster setting, parking means becoming a dominator, while leaving translates into becoming a dominatee. A famous result from statistical physics states that at the jamming limit the fraction of the covered area will on average and in the limit of an infinite area be roughly 0.547 [13].

Now, consider the nodes of a cluster-DS as being the center of discs of radius $R/2$. Note that this system satisfies the jamming limit condition in the large homogeneous dense networks. Therefore the number of nodes in the cluster-DS can be estimated by the following equation, asymptotically as $A/R^2 >> 1$:

$$N_{\mathrm{DS}} \simeq 0.547 \frac{4A}{\pi R^2} = 2.188 \frac{A}{\pi R^2} \tag{6.5}$$

Note that only the fraction of network area by radio range enters here. Also, comparing (6.5) and (6.2) we see that an independent cluster-DS differs in size from any other independent DS, such as the crystal-DS , at most by the small constant factor of $3.62/2.18 = 1.65$.

Crystal-DS : To the best of our knowledge there is no suitable "jamming limit" to deal with crystal-DS and we have to content ourselves with (6.2) and the asymptotic worst case difference factor from cluster-DS of 1.65 (see (6.5)). In fact, simulations show that they are extremely close indicating that for a wide variety of network scenarios we have consistently the following empirical average

$$N_{\mathrm{WCDS}} \simeq 2.2 \frac{A}{\pi R^2}. \tag{6.6}$$

6.3.3 Number of rebroadcasts in DRB scheme

We recall that for idealistic MAC/PHY layer the DRB broadcast algorithm guarantees that every reflector transmits only if the packet reaches a *new* dominator. Thus, there are at most $N_{\mathrm{WCDS}} - 1$ used reflectors and from equation (6.6) we find in homogeneous dense networks

$$< \#\text{broadcasts in DRB} > \;\simeq\; 2 * 2.2\frac{A}{\pi R^2} = 4.4\frac{A}{\pi R^2}. \qquad (6.7)$$

Combining the equations (6.1) and (6.7) give us $4.4/1.64 = 2.6$ as the *approximation factor* of DRB, meaning the ratio of the number of broadcasts of a DRB to that of the MCDS.

6.3.4 Number of rebroadcasts in DCCB scheme

We divide the connectors used in a realization of DCCB into two groups: the first group set of connectors directly (in 1-hop) forwards the packet to at least one dominator; the second group of connectors requires always two hops. In other words, from the algorithm, for the direct connectors the list $L_1(., m)$ is not empty and for the indirect connectors only the list $L_2(., m)$ is not empty when the random timer expires. The number of direct connectors is at most $N_{\mathrm{DS}}-1$, since each one reaches a new dominator over an ideal MAC layer. In order to compute the number of indirect connectors, we bound the number of dominators at 2-3 hops. Simple geometric considerations of non-overlapping discs show: at most 23 dominators for a 2-hop and 47 dominators for a 2- or 3-hop distance [1]. However, these pessimistic bounds poorly reflect the average behavior in a typical network. Figure 6.1 indicates from simulations in a homogeneous network that on average we find 7 dominators at 2-hop and 10 at 3-hop distance from an arbitrary dominator in the dense limit. Thus, there are almost $10N_{\mathrm{DS}}/2$ couples of 3-hop away dominators in the network. We can easily show that at most one indirect connector forwards the packet between two dominators which are at 3-hops from each other. So, the number of indirect dominators is at most $5N_{\mathrm{DS}}$ and the number of rebroadcasts in DCCB is at most equal to $N_{\mathrm{DS}}-1+N_{\mathrm{DS}}+5N_{\mathrm{DS}}$. Combining this with (6.5), we have

$$< \#\text{broadcasts in DCCB} > \;\leq\; 15.2\frac{A}{\pi R^2} \qquad (6.8)$$

By combining (6.1) and (6.8) we bound the approximation factor of DCCB to be less than $15.2/1.64 = 9.3$.

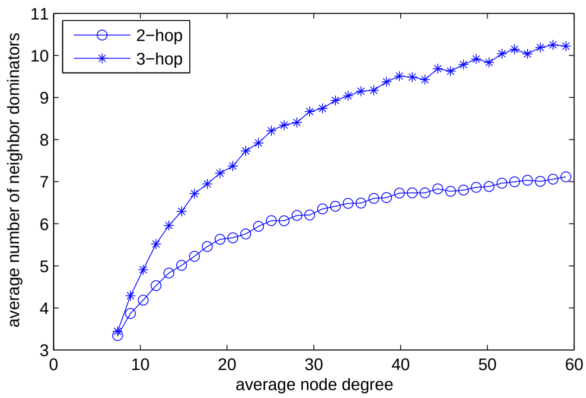


Figure 6.1: Average number of neighbor dominators in a cluster-DS.

Chapter 7

Extensions to DRB and Crystal-DS Algorithms

The ability of dynamic nodes to join the broadcast backbone voluntarily give DRB and DCCB the capability to adapt to the scarceness of resources and to repair node failure. Also, the crystal-DS algorithm has this capability for choosing an appropriate set of dominators. In addition, the crystal-DS algorithm can become more scalable to large networks by mixing it with clustering algorithms. We develop these ideas in the following not with an eye on quantifying savings but to demonstrate the potential of these extensions.

7.1 Adaptivity to Scarcity of Resources

We assign a weight to each node that reflects its degree of connectivity, power consumption, or mobility and allow reflectors (connectors in DCCB) with higher priority to join the DRB backbone during broadcast. This is accomplished by shortening the random back-off timing for those nodes. Similarly preference can be given to nodes for the crystal-DS algorithm.

7.2 Repair Mechanism to Counter Node Failure

Though hardly addressed in the literature, it is important for broadcast schemes to be robust to node failure. Node failure can occur in practice when a node runs out of power, is deliberately switched off, or malfunctions due to some reason, all scenarios which should not be ruled out in Ad Hoc networks and sensor networks in particular.

As we explained in Section 4.2, DCCB scheme guarantees full-reachability on any DS, and some clustering algorithms are robust against topology changes in the network [21, 3]. So DCCB is applicable for the networks with node failure and even mobility.

The DRB broadcast is robust to node failure, providing very high reachability even with non-negligible node failure, thanks to its use of a crystal-DS and dynamic reflectors (see Section 5.4). In order to provide full-reachability in the event of node failure, however, any broadcast scheme must have a repair mechanism. We next propose a repair algorithm for DRB which makes it adapt to failures that occur in the crystal-DS .

The DRB-repair algorithm needs periodic messages from all nodes. Every nodes sends a "Hello" message periodically and transmit its ID and a list of its dominators. As a consequence, all nodes will be aware about the dominators which are in 1-hop or 2-hops away and have not failed. We explain the repair-algorithm for different failure scenarios.

Failure of non-reflector dominatees: Dominatees that are not reflectors do not participate in a broadcast. Thus the failure of such a node does not affect DRB, or any scheme, for that matter.

Failure of reflectors: We consider two separate cases of reflector failure.

Case (i): The failed reflector is not the only reflector between any pair of dominators. The inherent robustness of the mesh of reflectors and its operation in DRB makes node repair trivial in this case. Such a reflector is redundant and other reflectors automatically perform its broadcasting tasks.

Case (ii): The failed reflector is the only reflector for a pair of dominators. In this case, when the backbone is used periodically, the two dominators will infer that they have no reflector between them. The dominators then initiate a 3-hop route discovery to connect to each other through a 2-hop path. We found this simple scheme quite effective as we demonstrate in Figure 7.1.

Failure of dominator: Again, disconnection needs to be discovered as before; all nodes in the radio range of a failed dominator change their scheme to *DCCB* with cluster-DS algorithm as described before. This repairing can fix the dominator failure which has a strong local effect in the backbone.

We demonstrate the repair using the constellation of Figure 7.1. There, the nodes Di, Ri and ni are dominators, reflectors and ordinary dominatees respectively. If the ordinary dominatee $n3$ fails, the broadcast message can

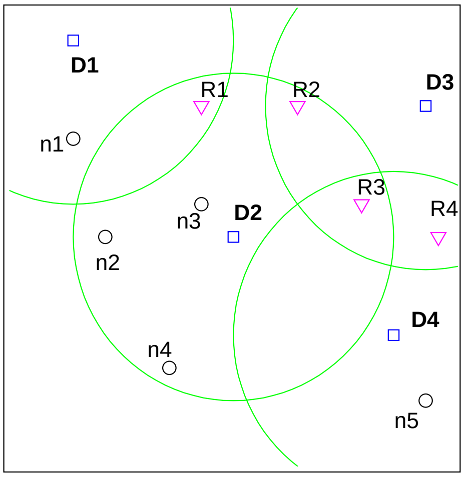


Figure 7.1: Different cases of node failure in DRB obtained by failing nodes $n3$, $R2$, $R1$ and $D2$.

still move through the dominators and reflectors.

If the reflector $R2$ fails, $R3$ will relay the message between $D2$ and $D3$. But, if the reflector $R1$ fails then no other reflector can replace it. Thus when $D1$ broadcasts the message and it does not hear any reflector relaying it to $D2$, it infers disconnection between itself and $D2$; its route discovery will find node $n1$ which is two hops away from $D2$ via $n2$.

If dominator $D2$ fails, the nodes in the radio range of $D2$ switch to DCCB scheme.

The DRB-repair algorithm tries to repair the crystal-DS backbone locally. However, the due to its quasi-global nature (see [25] for the definition of quasi-global), the crystal-DS backbone might not always be locally repairable and only global repairing guarantees the connectivity over the backbone. However, in some cases local repair algorithm can not fix a WCDS [9]. So, the algorithm does not grantee full-reachability.

Figure 7.2 shows two subgraph $G1$ and $G2$ that are connected to each other over crystal-DS backbone only by dominator node $D1$. In other words, $D1$ is a bridge node on the backbone between $G1$ and $G2$. When $D1$ fails, two subgraphs $G1$ and $G2$ will get disconnected and no local repairing algorithm can connect to subgraphs by working on the nodes around $D1$. However, by

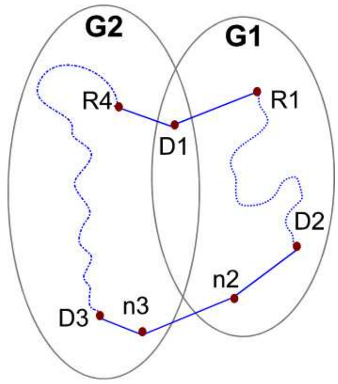


Figure 7.2: Failure of $D1$ causes a global disconnectivity over the backbone ($D1$ is a bridge for the backbone subgraph).

a global repairing two subgraph can be connected by a path like $D2$, $n2$, $n3$, $D3$ that was not used by DRB in crystal-DS backbone.

7.3 Scalability to Large or Merging Networks

In large networks the building of a crystal-DS may start at various locations, be it for lack of coordination, or for the need for time-efficiency or some local broadcasting needs. In this case, the crystal-DS algorithm keeps the seed identifier as a distinguishing mark. If a node learns of a neighbor with a different seed ID, it infers that it lies at the border of two crystal-DS backbones. Using the clustering algorithms at the border node allows us to connect the two backbones with different seeds. The same procedure will also enable the merging of networks that may come into each other's range due to mobility. The border nodes simply mark themselves as connectors between the two merged networks and the broadcast message disseminates from one network to another. Periodic rebuilds will replace such temporary fixes with a more efficient backbone built from a single seed.

Chapter 8

Conclusions

The contribution of this paper is two-fold. First, we introduce two novel deterministic timer-based schemes: DRB and DCCB. Both possess an efficiency within a factor of the optimum, a property which other deterministic timer-based schemes do not share and which we establish both analytically and in simulation. The principal distinguishing feature of DRB and DCCB from other existing timer-based schemes is their *hybrid* backbone consisting of a fixed and a variable part. The idea for using a hybrid backbone is novel. We also proposed a novel $O(n)$ timer-based algorithm for building an WCDS called the crystal-DS algorithm. The WCDS built by this algorithm is an IDS under the assumption of ideal MAC. So, it keeps number of WCDS within small constant factor of MCDS for such cases.

Second, we study by simulation the effect of the settings of the random timer for existing and proposed deterministic timer-based schemes. To the best of our knowledge, this issue has not been addressed in the literature before. We showed how the performance of these scheme depends on the random timer. Simulations not shown due to space constraints indicate that performance also depends on the broadcast packet size, the capacity of the wireless channel, the density of the nodes as well as the volume of background traffic.

The simulations presented demonstrate that DRB and DCCB have better reachability than counter-based probabilistic scheme. Also, they use a smaller number of nodes when compared to deterministic timer-based schemes. We proposed a simple cross-layer design which helps the timer-based schemes to work close to their maximal efficiency without sacrificing latency. Notably, combining analysis and empirical estimates of geometric efficiency constants we establish that close to their maximal efficiency DRB and DCCB operate within a small constant factor of the optimum, provided that the given WCDS and DS have sizes within small constant factor of the optimum.

Acknowledgment

Financial support comes in part from NSF, grant number ANI-0338856, and from Texas ATP, project number 003604-0036-2003.

Bibliography

[1] K. Alzoubi, P.-J. Wan, and O. Frieder. Weakly-connected dominating sets and sparse spanners in wireless ad hoc networks. In *ICDCS*, pages 96–104. IEEE Computer Society, 2003.

[2] D. J. Baker and A. Ephremides. The architectural organization of a mobile radio network via a distributed algorithm. *IEEE Trans. Comm.*, COM-29(11):1694–1701, 1981.

[3] S. Basagni. Distributed clustering for ad hoc networks. In *ISPAN*, pages 310–315. IEEE Computer Society, 1999.

[4] G. Calinescu, I. I. Mandoiu, P.-J. Wan, and A. Z. Zelikovsky. Selecting forwarding neighbors in wireless ad hoc networks. *Mob. Netw. Appl.*, 9(2):101–111, 2004.

[5] J. Cartigny and D. Simplot. Border node retransmission based probabilistic broadcast protocols in ad-hoc networks. In *HICSS*, page 303. IEEE Computer Society, 2003.

[6] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. *ACM Wireless Networks Journal*, 8(5):481–494, 2002.

[7] Y. P. Chen and A. L. Liestman. Approximating minimum size weakly-connected dominating sets for clustering mobile ad hoc networks. In *MobiHoc*, pages 165–172. ACM Press, 2002.

[8] Y. P. Chen and A. L. Liestman. A zonal algorithm for clustering ad hoc networks. *International Journal of Foundations of Computer Science*, 14(2):305–322, 2003.

[9] Y. P. Chen and A. L. Liestman. Maintaining weakly-connected dominating sets for clustering ad hoc networks. *Ad Hoc Networks*, 3(5):629–642, 2005.

[10] W. S. Conner, J. Chhabra, M. Yarvis, and L. Krishnamurthy. Experimental evaluation of synchronization and topology control for in-building sensor network applications. In *WSNA*, pages 38–49, 2003.

[11] F. Dai and J. Wu. Distributed dominant pruning in ad hoc networks. In *ICC*, 2003.

[12] D. Dubhashi, A. Mei, A. Panconesi, J. Radhakrishnan, and A. Srinivasan. Fast distributed algorithms for (weakly) connected dominating sets and linear-size skeletons. *J. Comput. Syst. Sci.*, 71(4):467–479, 2005.

[13] J. Feder, E. L. Hinrichsen, and T. Jossang. Geometry of random sequential adsorption. *Journal of Statistical Physics*, 44(5/6):448476, 1986.

[14] Z. J. Haas, J. Y. Halpern, and E. L. Li. Gossip based ad-hoc routing. *CoRR*, 2002.

[15] M. Heissenbüttel, T. Braun, M. Wälchli, and T. Bernoulli. Optimized stateless broadcasting in wireless multi-hop networks. In *INFOCOM*, 2006.

[16] X. Hong, K. Xu, and M. Gerla. Scalable routing protocols for mobile ad hoc networks. *IEEE Network*, 16:11–21, 2002.

[17] A. Keshavarz-Haddad, V. Ribeiro, and R. Riedi. Broadcast capacity in multihop wireless networks. In *MobiCom*, 2006.

[18] A. Keshavarz-Haddad, V. Ribeiro, and R. Riedi. Color-based broadcasting for ad hoc networks. In *WiOpt*, 2006.

[19] F. Kuhn, T. Moscibroda, T. Nieberg, and R. Wattenhofer. Fast deterministic distributed maximal independent set computation on growth-bounded graphs. In *DISC*, pages 273–287, 2005.

[20] T. J. Kwon and M. Gerla. Efficient flooding with passive clustering (pc) in ad hoc networks. *SIGCOMM Comput. Commun. Rev.*, 32(1):44–56, 2002.

[21] C. R. Lin and M. Gerla. Adaptive clustering for mobile wireless networks. *IEEE Journal on Selected Areas in Communications*, 15(7):1265–1275, 1997.

[22] W. Lou and J. Wu. On reducing broadcast redundancy in ad hoc wireless networks. *IEEE Transactions on Mobile Computing*, 1(2):111–123, 2002.

[23] W. Lou and J. Wu. A cluster-based backbone infrastructure for broadcasting in manets. In *IPDPS*, page 221.1. IEEE Computer Society, 2003.

[24] W. Lou and J. Wu. Double-covered broadcast (dcb): A simple reliable broadcast algorithm in manets. In *INFOCOM*, 2004.

[25] W. Lou and J. Wu. *Localized Broadcasting in Mobile Ad Hoc Networks Using Neighbor Designation*, chapter 28. in Mobile Computing Handbook, CRC Press, 2005.

[26] M. V. Marathe, H. Breu, H. B. Hunt III, S. S. Ravi, and D. J. Rosenkrantz. Simple heuristics for unit disk graphs. *Networks*, 25:59–68, 1995.

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

[28] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu. The broadcast storm problem in a mobile ad hoc network. In *MobiCom*, pages 151–162. ACM Press, 1999.

[29] E. Pagani and G. P. Rossi. Reliable broadcast in mobile multihop packet networks. In *MobiCom*, pages 34–42. ACM Press, 1997.

[30] W. Peng and X.-C. Lu. On the reduction of broadcast redundancy in mobile ad hoc networks. In *MobiHoc*, 2000.

[31] A. Qayyum, L. Viennot, and A. Laouiti. Multipoint relaying for flooding broadcast messages in mobile wireless networks. In *HICSS*, volume 9, page 298. IEEE Computer Society, 2002.

[32] P. Sinha, R. Sivakumar, and V. Bharghavan. Enhancing ad hoc routing with dynamic virtual infrastructures. In *INFOCOM*, 2001.

[33] I. Stojmenovic. Comments and corrections to 'dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks'. *IEEE Trans. Parallel Distrib. Syst.*, 15(11):1054–1055, 2004.

[34] I. Stojmenovic, M. Seddigh, and J. Zunic. Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks. *IEEE Trans. Parallel Distrib. Syst.*, 13(1):14–25, 2002.

[35] J. Sucec and I. Marsic. An efficient distributed network-wide broadcast algorithm for mobile ad hoc networks. Caip technical report 248, Rutgers University, 2000.

[36] Y.-C. Tseng, S.-Y. Ni, and E.-Y. Shih. Adaptive approaches to relieving broadcast storms in a wireless multihop mobile ad hoc network. In *IEEE Transactions on Computers*, volume 52, pages 545–557, 2003.

[37] P.-J. Wan, K. Alzoubi, and O. Frieder. Distributed construction of connected dominating set in wireless ad hoc networks. *Mob. Netw. Appl.*, 9(2):141–149, 2004.

[38] B. Williams and T. Camp. Comparison of broadcasting techniques for mobile ad hoc networks. In *MobiHoc*, pages 194–205. ACM Press, 2002.

[39] J. Wu and F. Dai. Broadcasting in ad hoc networks based on self-pruning. In *INFOCOM*, volume 3, pages 2240–2250, 2003.

[40] J. Wu and F. Dai. A generic distributed broadcast scheme in ad hoc wireless networks. *IEEE Trans. Comput.*, 53(10):1343–1354, 2004.

[41] J. Wu and H. Li. On calculating connected dominating set for efficient routing in ad hoc wireless networks. In *DIALM*, pages 7–14. ACM Press, 1999.

[42] J. Wu and W. Lou. Forward-node-set-based broadcast in clustered mobile ad hoc networks. In *Wireless Communications and Mobile Computing*, volume 3, pages 155–173, 2003.

[43] C.-C. Yang and C.-Y. Chen. A reachability-guaranteed approach for reducing broadcast storms in mobile ad hoc networks. In *Proceedings of IEEE Semiannual VTC*, 2002.

[44] D. C. Yoav Sasson and A. Schiper. Probabilistic broadcast for flooding in wireless mobile ad hoc networks. In *Proceedings of IEEE Wireless Communications and Networking Conference*, 2003.