On the Power of Local Broadcast Algorithms

Hosna Jabbari, Majid Khabbazian, Ian F. Blake and Vijay K. Bhargava
University of British Columbia
2356 Main Mall, Vancouver, BC, Canada V6T 1Z4

***Abstract*— There are two main approaches, static and dynamic, to broadcasting in wireless ad hoc networks. In the static approach, local algorithms determine the status (forwarding/non-forwarding) of each node proactively based on local topology information and a globally known priority function. In this paper, we first show that local broadcast algorithms based on the static approach cannot achieve a good approximation factor to the optimum solution (an NP-hard problem). However, we show that a constant approximation factor is achievable if (relative) position information is available. In the dynamic approach, local algorithms determine the status of each node "on-the-fly" based on local topology information and broadcast state information. Using the dynamic approach, it was recently shown that local broadcast algorithms can achieve a constant approximation factor to the optimum solution when (approximate) position information is available. However, using position information can simplify the problem. Also, in some applications it may not be practical to have position information. Therefore, we wish to know whether local broadcast algorithms based on the dynamic approach can achieve a constant approximation factor without using position information. We answer this question in the positive - we design a local broadcast algorithm in which the status of each node is decided "on-the-fly" and prove that the algorithm can achieve both full delivery and a constant approximation to the optimum solution.**

I. Introduction

Wireless ad hoc networks have emerged to support applications, in which it is required/desired to have wireless communications among a variety of devices without relying on any infrastructure or central management. In ad hoc networks, wireless devices, simply called nodes, have limited transmission range. Therefore, each node can directly communicate with only those within its transmission range (i.e., its neighbors) and requires other nodes to act as routers in order to communicate with out-of-range destinations.

One of the fundamental operations in wireless ad hoc networks is broadcasting, where a node disseminates a message to all other nodes in the network. This can be achieved through flooding, in which every node transmits the first copy of the received message. However, flooding can impose a large number of redundant transmissions, which can result in significant waste of constrained resources such as bandwidth and power. In general, not every node is required to forward/transmit the message in order to deliver it to all nodes in the network. A set of nodes form a Dominating Set (DS) if every node in the network is either in the set or has a neighbor in the set. A DS is called a Connected Dominating Set (CDS) if the subgraph induced by its nodes is connected. Clearly, the forwarding nodes, together with the source node, form a CDS. On the other hand, any CDS can be used for broadcasting a message (only nodes in the set are required to forward). Therefore, the problems of finding the minimum number of required transmissions (or forwarding nodes) and finding the Minimum Connected Dominating Set (MCDS) can be reduced to each other. Unfortunately, finding the MCDS (and hence minimum number of forwarding nodes) was proven to be NP hard even when the whole network topology is known [1], [2]. A desired objective of many efficient broadcast algorithms is to reduce the total number of transmissions to preferably within a constant factor of its optimum. For local algorithms and in the absence of global network topology information, this is commonly believed to be very difficult or impossible [3], [4].

The existing local broadcast algorithms can be classified based on whether the forwarding nodes are determined statically (based on only local topology information) or dynamically (based on both local topology and broadcast state information) [5]. In the static approach, the distinguishing feature of local algorithms over other broadcast algorithms is that using local algorithms any local topology changes can only affect the status of those nodes in the vicinity. Therefore, local algorithms can provide scalability as the constructed CDS can be updated, efficiently. The existing local algorithms in this category typically use a priority function known by all nodes in order to determine the status of each node [5]. In this paper we show that, using only local topology information and a globally known priority function, the local broadcast algorithms based on the static approach are not able to guarantee a good approximation factor to the optimum solution (i.e., MCDS). On the other hand, we show that local algorithms based on the static approach can achieve interesting results such as a constant approximation factor and shortest path preservation if the nodes are provided with position information.

In the dynamic approach, the status of each node (hence the CDS) is determined "on-the-fly" during the broadcast progress. Using this approach, the constructed CDS may vary from one broadcast instance to another even when the whole network topology and the source node remain unchanged. Consequently, the broadcast algorithms based on the dynamic approach typically have small maintenance cost and are expected to be robust against node failures and network topology changes. Many local broadcast algorithms in this category use local neighbor information to reduce the total number of transmissions and to guarantee full delivery (assuming no loss at the MAC/PHY layer). Others, such as probability-based and counter-based algorithms [6]–[8], do not rely on neighbor information. These algorithms typically cannot guarantee full delivery but eliminate the overhead imposed by broadcasting "Hello" messages or exchanging neighbor information.

Many of the existing neighbor-information-based broadcast algorithms in this category can be further classified as neighbor-designating and self-pruning algorithms. In neighbor-designating algorithms [9]–[11], each forwarding node selects some of its local neighbors to forward the message. Only the selected nodes are then required to forward the message in the next step. For example, a forwarding node u may select a subset of its 1-hop neighbors such that any 2-hop neighbor of u is a neighbor of at least one of the selected nodes [9]. In self-pruning algorithms [3], [12], [13], on the other hand, each node decides by itself whether or not to forward a message. The decision is made based on a self-pruning condition. For example, a simple self-pruning condition employed in [12] is whether all neighbors have been covered by previous transmissions. In other words, a node can avoid forwarding/rebroadcasting a message if all of its neighbors have received the message from other nodes in the network.

In [14], it was shown that neither neighbor-designating nor self-pruning algorithms can guarantee both full delivery and a constant approximation if they use only 1-hop neighbor information and do not piggyback information into the broadcast packets. The authors then proposed a self-pruning algorithm based on partial 2-hop neighbor information and proved that the algorithm achieves a constant approximation to the optimum solution and guarantees full delivery. However, in their proposed algorithm, each node was assumed to have its (approximate) position information, which is not practical in some applications/scenarios. Also, having position information can provide non-trivial information in wireless ad hoc networks and can greatly simplify the problem. As such, we wish to know whether similar results can be obtained without using position information. In this paper, we answer this remaining question in the positive - we propose a local broadcast algorithm based on 2-hop neighbor information and prove that it guarantees a constant approximation to the optimum solution. The proposed algorithm is both neighbor-designating and self-pruning, i.e. the status of each node is determined by itself and/or other nodes. A common drawback of many neighbor-designating algorithms is that each forwarding node may select $\Omega(n)$ of its neighbors in the worst case, where n is the total number of neighbors. Consequently, these algorithms cannot bound the packet size since a list of all selected node has to be piggybacked in the packet. As shown in Section IV, using our proposed algorithm, each forwarding node selects at most one of its neighbors, thus it does not have this drawback.

The paper is organized as follows. In Section II, we describe our system model and assumptions. In Sections III and IV we analyze the power of local broadcast algorithms based on the static and dynamic approach, respectively. In Section V, we use simulation to confirm the analytical results presented in Section IV. Finally, we conclude the paper in Section VI. Most of the proofs are placed in the appendix.

II. System Model and Assumptions

We assume that the network consists of N nodes equipped with omnidirectional antennas. Every node u has a unique *id*, denoted $id(u)$, and every packet is stamped by the *id* of its source node and a nonce, a randomly generated number by the source node. For simplicity, we assume that all nodes are located in two-dimensional space. However, all the results presented in this paper can be readily extended to three-dimensional ad hoc networks.

To model the network, we assume two nodes u and v are connected by an edge if and only if $|uv| \leqslant R$, where $|uv|$ denotes the Euclidean distance between nodes u and v and R is the transmission range of the nodes. This model is, up to scaling, identical to the unit disk graph model, which is a typical model for two-dimensional ad hoc networks. In reality, however, the transmission range can be of arbitrary shape as the wireless signal propagation can be affected by many unpredictable factors. Finally, we assume that the network is connected and static during the broadcast and that there is no loss at the MAC/PHY layer. These assumptions are necessary in order to prove whether or not a broadcast algorithm can guarantee full delivery. Note that without these assumptions even flooding cannot guarantee full delivery.

III. Broadcasting Using the Static Approach

Let the k-*neighborhood* of a node u, denoted $G_k(u)$, be the subgraph induced by nodes within k-hops of u. Suppose each node is given a globally fixed priority function $Pr(id(w), G_{h'}(w))$ which gets a node's *id*, $id(w)$, and its local topology information, $G_{h'}(w)$, as inputs and returns a real number that determines the priority of w. For example, a node priority can be determined based on its *id*, its degree (i.e. the number of its 1-hop neighbors) or its neighbor connectivity ratio (i.e. ratio of pairs of neighbors that are not directly connected to all possible pairs of neighbors).

In local broadcast algorithms based on the static approach, status (forwarding/non-forwarding) of each node u, $Stat(u)$, is a function of $id(u)$, $G_h(u)$ and $Pr(id(v), G_{h'}(v))$, where $v \in G_h(u)$ and the parameters h and h' are fixed constant numbers. Note that status of each node does not depend on that of other nodes. Therefore, any local topology change can only affect the status of the nodes in the vicinity. In designing local broadcast algorithms, we are looking for status functions that not only guarantee constructing a CDS (hence full delivery) but also ensure that the constructed CDS has small size, preferably within a constant factor of the optimum. In the following, we show that no such status function exists. The idea is to find a graph in which any status function fails either in constructing a CDS or finding a CDS whose size is smaller than $\Omega(N)$, where N is the total number of nodes in the network. Our approach is to find a large number of nodes for which both the local topology, $G_h(.)$, and the relative priority of the nodes in $G_h(.)$ are the same.

Without loss of generality, we can assume that $R = 1$. As shown in Figure 1, let us distribute N nodes on the x-axis between the coordinates 0 and $2(h + h') + 1$ such that the coordinate of i'th node is $\frac{(i-1)}{(N-1)} \times (2(h + h') + 1)$, where

$1 \leqslant i \leqslant N$ and $N \gg 2(h+h')+1$. It is easy to see that $G_{h'}(u)$ and $G_{h'}(v)$ are isomorphic if $u, v \in [h', 2h+h'+1]$. Based on the definition of priority function, the relative priority of two nodes u and v only depends on their *ids* if $G_{h'}(u)$ and $G_{h'}(v)$ are isomorphic. Therefore, we can distribute all nodes in the interval $[h', 2h+h'+1]$ such that their priorities increase as their coordinates (their distance to the origin) increase. Using this distribution, all nodes in the unity interval $[h'+h, h'+h+1]$ will have similar view of local topology $G_h(.)$ and priority relationship between the nodes in $G_h(.)$. Therefore, the output of the status function is the same for all nodes in this unity interval, i.e., either all or none of the nodes in the unity interval will be selected. Clearly, the latter case is not possible since at least one node in every unity interval has to be selected (otherwise the graph induced by the selected nodes will be disconnected). On the other hand, selecting all nodes in the interval $[h'+h, h'+h+1]$ will result in a large set of selected nodes compared to the MCDS. It is because the size of MCDS for the simple graph shown in Figure 1 is no more than $2 \times (2(h+h')+1)$ [1]. However, if we select all nodes in a unity interval, the size of the obtained CDS would be at least

$$\lfloor \frac{N-1}{2(h+h')+1} \rfloor + 2(h+h'-1).$$

Therefore, the approximation factor would be at least

$$\frac{\lfloor \frac{N-1}{2(h+h')+1} \rfloor + 2(h+h'-1)}{2 \times (2(h+h')+1)} \in \Omega(N).$$

Consequently, local broadcast algorithms based on the static approach are not able to guarantee a good approximation factor in the worst case. Note that this result does not imply that local broadcast algorithms cannot achieve a good bound on the average.

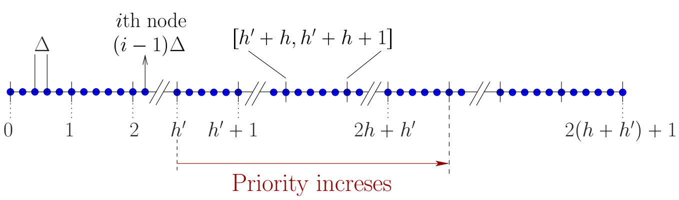

Fig. 1. Distributing nodes on a line segment with length $2(h+h')+1$.

A. Using Position Information

In the context of broadcast algorithms based on the static approach, we may wish to know whether using position information can help us to get a better result in the worst case. Following, we show that a constant approximation factor can be achieved in the worst case if position information is available.

Theorem 1: Suppose all nodes are located in a square with size $L \times L$ such that $(\frac{L}{R})$ is bounded by a constant. Let $\alpha \geqslant \frac{1}{c_0}$, where $c_0 \in O(1)$. The size of a CDS is guaranteed to be within a factor of MCDS if and only if it contains a constant number of nodes in each square cell of size $\alpha R \times \alpha R$.

As shown in Figure 2, let us assume that all nodes in the network are located in a square area of size $L \times L$. Considering Theorem 1 , it is natural to divide the network area into small square cells of size $\alpha R \times \alpha R$ and search for algorithms that guarantee a constant number of selected nodes in each cell.

Lemma 1: Let $u \in \mathcal{C}_i$ and $v \in \mathcal{C}_j$ be two nodes located in two different square cells $\mathcal{C}_i$ and $\mathcal{C}_j$, where the size of each cell is $\alpha R \times \alpha R$. For any pair of nodes $u' \in \mathcal{C}_i$ and $v' \in \mathcal{C}_j$ we have

$$|u'v'| \leqslant |uv| + 2\sqrt{2}\alpha R.$$

Let $S_{sq}(\alpha)$ denote a set constructed by dividing the network area into small square size of $\alpha R \times \alpha R$ (as shown in Figure 2) and selecting one node in each non-empty cell. Clearly, $S_{sq}(\alpha)$ is a dominating set (DS) if $\alpha \leqslant \frac{\sqrt{2}}{2}$. Therefore, we use the notation $DS_{sq}(\alpha)$ instead of $S_{sq}(\alpha)$ wherever $\alpha \leqslant \frac{\sqrt{2}}{2}$. In general, the graph induced by $DS_{sq}(\alpha)$ is not connected. Let us use $G(V, R)$ to denote a graph constructed by connecting two nodes in V if and only if their Euclidean distance is at most R. The following theorem states that $DS_{sq}(\alpha)$ will be a CDS if the network remains connected when all nodes reduce their transmission range R to $(1-2\sqrt{2}\alpha)R$, where $2\sqrt{2} < \frac{1}{\alpha}$ is a constant number. In this case, we say the network satisfies the *high-connectivity* condition.

Theorem 2: Suppose $\frac{1}{\alpha} > 2\sqrt{2}$ is a constant number (i.e., $\frac{1}{\alpha} \in O(1)$). For any network $G(V, R)$, $DS_{sq}(\alpha)$ will be a CDS whose size is within a constant factor of the MCDS if $G(V, (1-2\sqrt{2}\alpha)R)$ is connected

Networks with high density typically satisfy the high-connectivity condition. For example, when the density is high, it is expected that the network remains connected if all nodes reduce their transmission range to, say, 90% of the original. The high-connectivity condition, however, may not be always guaranteed. An alternative approach is to require the selected nodes to increase their transmission power (hence their transmission range) by a constant. We say a network satisfies the *high-transmission* condition if (upon the need) the nodes can increase their transmission range by a constant.

Theorem 3: Suppose $\frac{1}{\alpha} \geqslant \sqrt{2}$ is a constant number. $DS_{sq}(\alpha)$ will be a CDS whose size is within a constant factor of the MCDS if all nodes in $DS_{sq}(\alpha)$ increase their transmission range R to $(1+2\sqrt{2}\alpha)R$.

As stated in Theorem 3, when the high-transmission condition is satisfied, we can construct a small sized CDS by selecting one node in each none-empty cell. The selection algorithm can be carried out locally. Having the position of all 1-hop neighbors (hence all nodes in the cell), each node can decide whether it is the selected node in the cell it is located. The selection criteria can be based on the node's *id* or other parameters such as node's coordinates and battery life time. For example, a node may select itself if it has the smallest id in the cell. When the network satisfies the high-transmission condition, however, a more effective criteria may be to select a node with higher battery life time (to increase the network life time) or to select the closest node to the origin of the cell or the one with less mobility (to get a more stable CDS in mobile ad hoc networks).

[1]The exact size of MCDS is $\left\lceil \frac{l-1}{\lfloor \frac{N-1}{l} \rfloor \times \frac{l}{N-1}} \right\rceil$, where $l = 2(h+h')+1$.

There are many optimization techniques to further reduce the total number of selected nodes in $DS_{sq}(\alpha)$ or to relax the requirement of transmitting at higher power for many selected nodes. For example, suppose that node u_s is the selected node in cell $\mathcal{C}_i$. The node u_s does not require to increase its transmission power if every node v within transmission range of a node $u \in \mathcal{C}_i$ is within the transmission range of either u_s or a selected neighbor of u_s. This can be formally expressed as

$$\begin{aligned} &\forall u, v \text{ s.t. } u \in \mathcal{C}_i \wedge |uv| \leqslant R: \\ &\exists w_s \in DS_{sq}(\alpha) \text{ s.t. } |u_s w_s| \leqslant R \wedge |w_s v| \leqslant R. \end{aligned} \tag{1}$$

Both the high-connectivity and high-transmission conditions can be relaxed if we allow selecting more than one yet a constant number of nodes in each non-empty cell. Following, we describe a simple algorithm that can achieve a constant approximation to the MCDS without using any condition such as high-connectivity.

Suppose we divide the network into square cells with size $\frac{\sqrt{2}}{2}R \times \frac{\sqrt{2}}{2}R$. Clearly, all nodes in a cell are 1-hop neighbors of each other. Two cells $\mathcal{C}_i$ and $\mathcal{C}_j$ are called neighbors iff

$$\exists u \in \mathcal{C}_i, v \in \mathcal{C}_j \text{ s.t. } |uv| \leqslant R.$$

In this case, nodes u and v are called the connectors of the neighbor cells $\mathcal{C}_i$ and $\mathcal{C}_j$ and u is referred to as the connector to the cell $\mathcal{C}_j$ through the node v. As shown in Figure 2, each cell is a neighbor of at most 20 other cells if the side of each square cell is set to $\frac{\sqrt{2}}{2}R$. Assume that each node has a list of its 2-hop neighbors together with their positions. Suppose u is a node located in the cell $\mathcal{C}_i$. The node u selects itself as a member of CDS if, based on a criteria, it is the selected connector to a neighboring cell $\mathcal{C}_j$ through the node $v \in \mathcal{C}_j$. The designed criteria must be symmetric in the sense that u selects itself as a connector to the cell $\mathcal{C}_j$ through the node $v \in \mathcal{C}_j$ if and only if v selects itself as a connector to the cell cell $\mathcal{C}_i$ through the node $u \in \mathcal{C}_j$. As an example of a symmetric criteria, a node $u \in \mathcal{C}_i$ can select itself iff there exists a node v in a neighboring cell $\mathcal{C}_j$ such that $|uv|$ is minimum among all the possible connectors of the cells $\mathcal{C}_i$ and $\mathcal{C}_j$. Any tie can be broken using, for example, nodes' *id*s. Note that node u has a list of its 2-hop neighbors (as well as their positions) therefore it can compute the set of all possible connectors between its own cell and any neighboring cell. Clearly, the constructed set is a CDS whose size is within a constant factor of its optimum because the total number of nodes in every cell is bounded by a constant and the side of the square cell is αR, where $\sqrt{2} \leqslant \frac{1}{\alpha} \in O(1)$ (see Theorem 1). Note that, in practice, many of the selected nodes can be pruned using similar conditions as (1) in order to get a smaller CDS.

An important application of constructing a CDS is to employ it as a backbone for routing. When a CDS is constructed, only the nodes in the set are required to forward packets towards the destination. Therefore, each path between the source node s and destination node d can be represented as

$$s, w_1, w_2, \ldots, w_k, d,$$

where w_i are in CDS. Let $l(s,d)$ and $l_{CDS}(s,d)$ denote the length of the shortest path between s and d in the original graph and the graph induced by CDS $\cup \{s, d\}$, respectively. In general, the ratio $\frac{l_{CDS}(s,d)}{l(s,d)}$ can be very large, in the worst case. For example, suppose all nodes in the network (N nodes in total) are located on a circle with radius $\frac{R}{2\sin(\frac{\pi}{N})}$ such that the distance between any two neighbors is R. In this case, it is easy to show that there are two nodes s and d such that

$$\frac{l_{MCDS}(s,d)}{l(s,d)} = \frac{N}{2} - 1,$$

where $l_{MCDS}(.)$ is the length of the shortest path in the minimum connected dominating set. Consequently, even MCDS cannot provide a good approximation of the shortest path in the original graph. Theorem 4, on the other hand, shows that the length of the shortest path in $CDS_{sq}(\alpha)$ constructed based on the high-transmission condition is at most one more than that of the original shortest path.

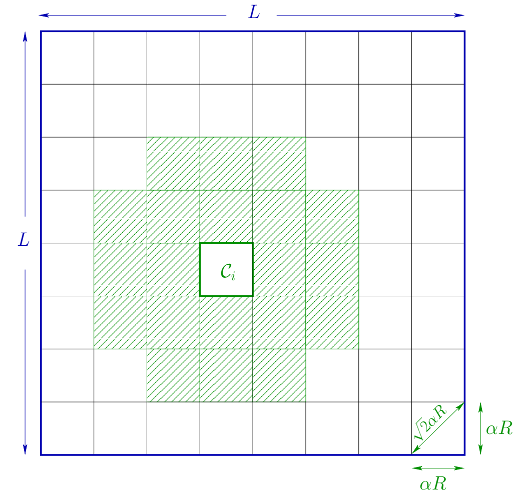

Fig. 2. Network partitioning and possible neighbors of a cell $\mathcal{C}_i$ for the case $\alpha = \frac{\sqrt{2}}{2}$.

Theorem 4: For the CDS constructed based on the high-transmission condition we have

$$l_{CDS}(s,d) \leqslant l(s,d) + 1.$$

Employing a symmetric criteria to construct the CDS or using Condition (1) will change the shortest path approximation to

$$l_{CDS}(s,d) \leqslant 2 \times l(s,d) + 1.$$

IV. Broadcasting Using The Dynamic Approach

Using the dynamic approach, the status (forwarding/non-forwarding) of each node is determined "on-the-fly" as the broadcasting message propagates in the network. In particular, in neighbor-designating broadcast algorithms, each forwarding node selects a subset of its neighbors to forward the packet and in self-pruning algorithms each node determines its own status based on a self-pruning condition after receiving the first or several copies of the message. It was recently proved that self-pruning broadcast algorithms (hence broadcast algorithms based on the dynamic approach) are able to guarantee both full delivery and a constant approximation factor to the optimum solution (MCDS) [14]. However, the proposed algorithm in

[14] employes position information in order to design a strong self-pruning condition. In the previous section, we observed that position information can simplify the problem of reducing total number of nodes. Moreover, having position information may not be practical in some applications. Therefore, it is interesting to know if both full delivery and a constant approximation factor to MCDS can be achieved when position information is not available. In this section, we design a hybrid (i.e., both neighbor-designating and self-pruning) broadcast algorithm and show that the algorithm can achieve both full delivery and constant approximation without using position information.

A. The Proposed Local Broadcast Algorithm

Suppose each node has a list of its 2-hop neighbors. This can be achieved in two rounds of information exchange. In the first round, each node broadcasts its *id* to its 1-hop neighbors. Therefore, at the end of the first round, each node has a list of its 1-hop neighbors. In the second round, each node can transmit its *id* together with the list of its 1-hop neighbors.

When the broadcast process terminates, a node has a forwarding status (i.e., is a member of CDS) if it has broadcast the packet and non-forwarding status (i.e., is not a member of CDS), otherwise. However, during the broadcast process, each node may take several statuses represented by different colors as follows

- white: The node has not received the packet;
- green: The node has received the packet;
- red: The node was selected to forward/broadcast the packet;
- black: The node has broadcast the packet.

At the beginning of the broadcast process, all nodes are whites except the source node (the node that initiates the broadcast), which is green. When the status of a node is changed from white to green, it schedules a broadcast by placing a copy of the message in its MAC layer queue. There are at least two sources of delay in the MAC layer. First, a message may not be at the head of the queue so it has to wait for other packets to be transmitted. Second, in contention based channel access mechanisms such as CSMA/CS, to avoid collision, a packet at the head of the queue has to wait for a random amount of time before getting transmitted. In this paper, we assume that a packet can be removed from the MAC layer queue if it is no longer required to be transmitted. Therefore, the broadcast algorithm has access to two functions to manipulate the MAC layer queue. The first function is the scheduling/placing function, which is responsible for putting a message in the MAC layer queue. We assume that the scheduling function handles duplicate packets, i.e., it does not place the packet in the queue if a copy of it is already in the queue. The second function is called to remove a packet form the queue. We assume that removing function does not do anything if there is no copy of the packet in the queue.

Algorithm 1 shows our proposed local broadcast scheme. When a node receives a packet, it first extracts some information from the packet and updates a self-pruning condition called the *black condition*. Suppose that a white or a green node u receives a packet from a broadcasting node v. Assume that u is not selected by v to forward the message. If the self-pruning condition is satisfied, node u removes the packet from the MAC layer queue (if there is any). Otherwise, u places the packet in its queue (if there is no copy of it there) and sets its color to green. A non-black node, which is selected to forward, has to change its color to red and place the packet in the queue. Finally, a broadcasting node has to select one of its neighbors (to forward the message) and include its *id* in the packet if the self-pruning condition is not satisfied. Note that a selected node (a red node) has to broadcast even if the self-puring condition is satisfied.

Algorithm 1 The proposed hybrid algorithm

```
Extract information from the received packet
if color== black then
    Return;
end if
if color==white then
    color ← green;
end if
Update the black condition and the list of sole neighbors;
if the black condition is not satisfied then
    Schedule the packet; {(*only update the embedded sole
    neighbor if the packet is already in the queue*)}
    if the node is selected then
        color ← red;
    end if
else {(*there is no sole neighbor in this case*)}
    if the node is selected then
        color ← red;
        Schedule the packet; {(*only remove the sole neigh-
        bor if the packet is already in the queue*)}
    else
        Remove the packet form the queue;
    end if
end if
Set the color to black when the packet is transmitted;
```

The self-pruning condition is at the core of the proposed broadcast algorithm. In our broadcast algorithm, we use the following self-pruning condition

*Definition 1 (**black condition**):* We say the *black condition* is satisfied for a node u if, based on u's collected information, for any of its neighbors, v, there is a node $w \neq u$ whose color is either red or black. Note that the color of all the red nodes will be changed to black, eventually.

*Definition 2 (**sole neighbor**):* A node v is called a *sole neighbor* of u if, based on u's collected information, there is no node $w \neq u$ such that w is a neighbor of v and the color of w is either red or black. Clearly, a node has a sole neighbor if and only if its black condition is not satisfied.

As mentioned earlier, a broadcasting node whose black condition is not satisfied has to select one of its neighbors to forward the message. In our proposed algorithm, the selected node is a sole neighbor of the broadcasting node. If there are more than one sole neighbors, the algorithm can select one of them randomly or based on a criteria. For example, it can select the node with the maximum battery life time or the one with the most number of white neighbors.

B. Analysis of the Proposed Broadcast Algorithm

In this section, we prove that the proposed broadcast algorithm guarantees full delivery as well as a constant approximation to the optimum solution irrespective of the (sole neighbor's) selection criteria and the random delay in the MAC layer (hence, the random sequence of the broadcasting nodes). In order to prove these properties, we assume that nodes are static during the broadcast, the networks is connected and there is no loss at the MAC/PHY layer. Note that even flooding cannot guarantee full delivery without these assumptions.

Theorem 5: Algorithm 1 guarantees full delivery.

Proof: Every node broadcasts a message at most once. Therefore, the broadcast process eventually terminates. By contradiction, assume that node d does not receive the message after broadcast termination. Since the network is connected, there is a path from the source node s (the node that initiates the broadcast) to node d. Clearly, we can find two nodes u and v on the path such that u and v are neighbors, u has received the packet and v has not received it. The node u has not broadcast the packet since v has not received it. Therefore, the black condition must have been satisfied for u. Thus, v must have a neighbor w, whose color is either red or black. Note that all red nodes will eventually broadcast the message and change their color to black. This is a contradiction as, based on the assumption, v cannot have a broadcasting neighbor. ∎

Lemma 2: Using Algorithm 1, the number of broadcasting nodes inside a disk $D_{O,\frac{R}{2}}$ centered at O with a radius $\frac{R}{2}$ is bounded by a constant.

Proof: Clearly, all nodes inside $D_{O,\frac{R}{2}}$ are neighbors of each other, thus they receive each others broadcast packets. The broadcasting nodes can be divided into two types based on whether or not the black condition was satisfied for them just before they broadcast the packet. Note that the black condition may be satisfied for a broadcasting node only if the node has been selected to forward the message. It is because a selected node has to broadcast the packet irrespective of the black condition. Consider two disks centered at O with radii $\frac{R}{2}$ and $\frac{3R}{2}$, respectively. Suppose k is the minimum number such that for every set of k nodes $w_i \in D_{O,\frac{3R}{2}} - D_{O,\frac{R}{2}}$, $1 \leqslant i \leqslant k$, we have

$$\exists w_i, w_j : \quad i \neq j \quad \text{and} \quad |w_i w_j| \leqslant R.$$

The area $D_{O,\frac{3R}{2}} - D_{O,\frac{R}{2}}$ can be covered with a constant number of disks with radius $\frac{R}{2}$. Clearly, we will have at least two nodes w_i and w_j in a covering disk (hence $|w_i w_j| \leqslant R$) if the number of nodes inside $D_{O,\frac{3R}{2}} - D_{O,\frac{R}{2}}$ is more than the number of covering disks. Thus, k is bounded by a constant. We prove that, for each type, the number of broadcasting nodes inside $\mathcal{D}_{O,\frac{R}{2}}$ is bounded by a constant. By contradiction, suppose that there are more than k broadcasting nodes for which the black condition is not satisfied. Let $u_0, u_1, \ldots u_k$ be the first $k+1$ broadcasting nodes ordered chronologically based on their broadcast time and $a_0, a_1, \ldots a_k$ their corresponding sole neighbor. Note that all the nodes in the disk $\mathcal{D}_{O,\frac{R}{2}}$ will receive the packet after u_0 broadcast it. Therefore,

$$\forall a_i, 1 \leqslant i \leqslant k : \quad a_i \in D_{O,\frac{3R}{2}} - D_{O,\frac{R}{2}}.$$

Thus, there are two nodes a_i, a_j, $i < j$ such that $|a_i a_j| \leqslant R$. The node u_i has broadcast before u_j and is a neighbor of it. Therefore, u_j is aware of u_i's sole neighbor a_i. This is a contradiction because based on the definition of sole neighbor, a_j cannot be a sole neighbor of u_j.

Similarly, we can show that there is a constant number k' such that every set of k' nodes inside $D_{O,\frac{3R}{2}}$ contains at least two neighboring nodes. Let $v_1, \ldots v_{k'} \in D_{O,\frac{3R}{2}}$ be the first k' broadcasting nodes (inside $D_{O,\frac{R}{2}}$), ordered chronologically based on their broadcast time, for which the black condition is satisfied. Note that a broadcasting node must have been selected (by another node) to forward the packet if its black condition is satisfied. Let $v_1, \ldots v_{k'}$ be the sole neighbors of $b_1, b_2, \ldots b_{k'}$. Clearly, $b_i \in D_{O,\frac{3R}{2}}$. Thus, there are two nodes b_i and b_j, $i < j$ such that $|b_i b_j| \leqslant R$. This is a contradiction because b_i and b_j are neighbors and b_j receives the b_i broadcast packet thus it cannot have a sole neighbor in $D_{O,\frac{R}{2}}$ as $v_i \in D_{O,\frac{R}{2}}$. ∎

Corollary 1: Using Algorithm 1, every node has at most a constant number of broadcasting neighbors.

Proof: A disk with a radius R can be covered with a constant number of disks with radii $\frac{R}{2}$. [2] The proof is, then, straightforward using Lemma 2. ∎

Theorem 6: Algorithm 1 has a constant approximation factor to the optimal solution (MCDS).

Proof: The proof is straightforward using Lemma 2 and Theorem 1. ∎

C. Computing the Black Condition

To compute the black condition, each node u maintains a list of its sole neighbors $List_u^{sole}$. Initially, all 1-hop neighbors are placed in the sole neighbor list. The list gets updated every time the node receives a copy of the message. Suppose node u receives a copy of the message from its neighbor v. To update $List_u^{sole}$, it retrieves the list of neighbors of v and subtracts it from $List_u^{sole}$. The black conditions is satisfied if and only if $List_u^{sole}$ becomes empty after one or several updates. Based on Corollary 1, each node has a constant number of broadcasting neighbors, hence the total number of updates is bounded by a constant. Therefore, to compute the black condition, only a constant number of subtractions has to be performed, hence the complexity of computing the black condition is the same as the complexity of computing a single list subtraction.

Let Δ denote the maximum node degree in the network. An update consists of subtracting two lists of size at most Δ. When the network topology changes are not as frequent as broadcasting in the network, in a pre-computation stage, each node can sort the list of its neighbors and share the sorted list (instead of an unsorted list) with its neighbors in

[2] The constant is 7 [15].

the second round of information exchange. When the lists are sorted, the subtraction can be carried out in $O(\Delta)$, which is optimum since each element of the two lists has to be accessed at least once. However, if topology changes occur at higher rates than broadcasting, the nodes may sort the list of their own neighbors and the broadcasting neighbors reactively upon receiving a copy of the broadcasting message. Clearly, in this case the complexity of computing the black condition will be $O(\Delta \log \Delta)$. Note that the the naive method for subtracting two lists (without using sorting) has the computational complexity $O(\Delta^2)$.

D. The Strong Black Condition

A broadcasting node for which the black condition is satisfied does not select any forwarding node and is selected by a broadcasting node whose black condition is not satisfied. Also, a broadcasting node selects at most one node to forward the packet. Therefore, the black condition is satisfied for at most half of the broadcasting nodes. Consequently, to prove the algorithm guarantees a constant approximation ratio it is sufficient to show that the number of broadcasting nodes for which the black condition is not satisfied is within a constant approximation factor of the minimum number of required broadcasting nodes. Recall that a selected node has to broadcast even if its black condition is satisfied. To further reduce the number of broadcasting nodes we can relax this assumption by allowing the selected nodes to avoid broadcasting under the following self-pruning condition.

*Definition 3 (**strong black condition**):* We say the *strong black condition* is satisfied for a node u if, based on u's collected information, any of its neighbors has either a black neighbor or a red neighbor whose priority (e.g., its *id*) is higher than u.

Note that the strong black condition is only used for selected nodes to check whether they need to broadcast. Other nodes can determine their status based on the black condition (a weaker condition). Clearly, using the strong black condition for the selected nodes will not result in more broadcasting nodes compared to the case where it is not used. Also, the following theorem states that the full delivery is guaranteed if the selected nodes get pruned under the strong black condition.

Theorem 7: Algorithm 1 guarantees full delivery (assuming no loss at the MAC/PHY layer) if selected nodes avoid broadcasting under the strong black condition.

(Proof in Appendix).

E. Extending the Network Model

The results presented in the paper can be extended to the case where the nodes are distributed in three-dimensional space. In other words, when the nodes are distributed in three-dimensions it can be shown that local broadcast algorithms based on the static approach can provide a constant approximation if nodes have their position information. By replacing circles with balls, it can be similarly shown that Algorithm 1 can provide both full delivery and a constant approximation to the optimum solution.

Algorithm 1 (the proposed algorithm based on the dynamic approach) can be extended to the case where the nodes have different transmission ranges. In this case, it can be proved that the algorithm guarantees a constant approximation ratio if $\frac{R_{Max}}{R_{Min}}$ is constant, where R_{Max} is the maximum transmission range and R_{Min} is the minimum transmission range of the nodes in the network and two nodes have a link iff both of them are in transmission range of each other. Similar to the proof of Theorem 6, this can be proved by showing that the number of broadcasting nodes inside any disk $D_{O,R_{Min}}$ is constant. Also, we can use the quasi unit disk graph to model the network [16]. In this model there is a link between two nodes if their Euclidean distance is less than γR, $0 \leqslant \gamma \leqslant 1$, and there is no link if the Euclidean distance is more than R. This model is closer to reality than the unit disk graph model. Using quasi unit disk graphs model we can show that Algorithm 1 guarantees a constant approximation ratio if $\frac{1}{\gamma}$ is constant. Similarly, the proof is by showing that the number of broadcasting nodes in any disk $D_{O,\gamma R}$ is constant.

V. Experimental Results

One of the major contributions of this work is the design of a local broadcast algorithm based on the dynamic approach (Algorithm 1) that can achieve both full delivery and a constant approximation ratio to the optimum solution without using position information. To confirm the analytical results, we implemented Algorithm 1 and used it in a simulation to compute the ratio of broadcasting nodes (i.e., number of broadcasting nodes/total number of nodes). We also implemented Wan-Alzoubi-Frieder algorithm [17] and used it as an approximation of the minimum number of broadcasting nodes. Note that Wan-Alzoubi-Frieder algorithm is not a local algorithm and is only used as a benchmark as it has an approximation factor of at most 8 [3]. Both Wan-Alzoubi-Frieder algorithm (referred to as ratio-8 algorithm) and Algorithm 1 were implemented in C++. To compute the number of broadcasting nodes, we uniformly distributed the nodes in a square of size $1000 \times 1000 m^2$. We assumed there is no collision in the MAC layer and allowed only one broadcast at each simulation run. Also, we used the strong black condition in Algorithm 1 to reduce the total number of broadcasting nodes. Figures 3 and 4 show the ratio of broadcasting nodes for over 1000 runs. To get the results shown in Figure 3, we set the transmission range to $250m$ and varied the total number of nodes from 25 to 1000. In Figure 4, the number of nodes was fixed to 1000 and the transmission range was varied from $50m$ to $300m$. The transmission range and the total number of nodes were selected from a large interval so that the simulation covers very sparse and very dense networks as well as the networks with large diameters. Interestingly, both Figures 3 and 4 show that the ratio of broadcasting nodes using Algorithm 1 is very close to that using Wan-Alzoubi-Frieder algorithm.

VI. Conclusions

In this paper, we investigated capabilities of local broadcast algorithms in reducing the total number of required transmis-

[3]The approximation factor is at most 7.8 as proved in [18]

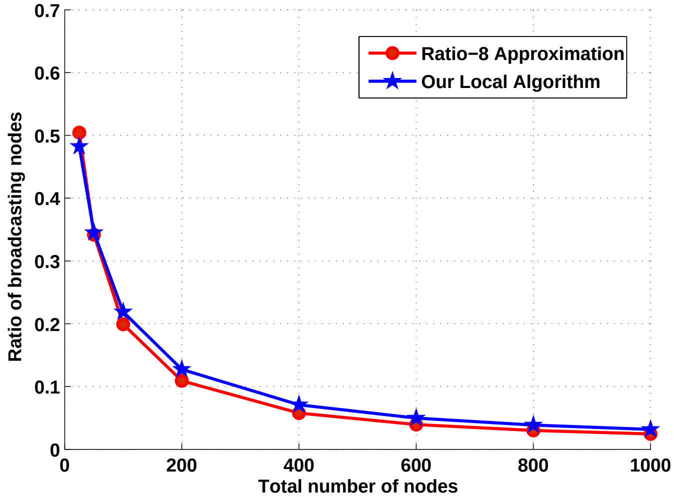

Fig. 3. Ratio of broadcasting nodes vs. total number of nodes.

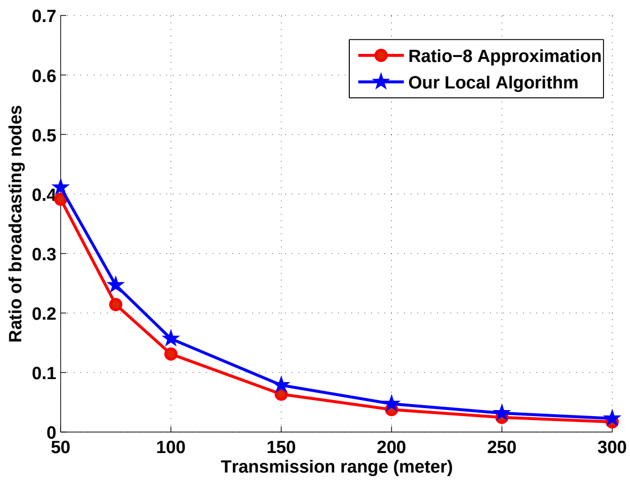

Fig. 4. Ratio of broadcasting nodes vs. transmission range.

sions. As proved, local broadcast algorithms based on the static approach may not be able to guarantee a small sized CDS if the position information is not available. It was shown that using relative position information can greatly simplify the problem of reducing the total number of selected nodes using the static approach. In fact, we showed that a constant approximation factor is achievable using position information. Using the dynamic approach, it was recently shown that a constant approximation is possible using (approximate) position information [14]. In this paper, we showed that local broadcast algorithms based on the dynamic approach do not require position information to guarantee a constant approximation factor to MCDS. The results presented in the paper can be extended to the case where the nodes are distributed in three-dimensional space. Also, the proposed algorithm based on the dynamic approach can be extended to the case where the nodes have different transmission ranges or when the network is modeled using the quasi unit disk graph model.

References

[1] M. Garey and D. Johnson, *Computers and Intractability; A Guide to the Theory of NP-Completeness*. New York: W. H. Freeman & Co., 1990.
[2] B. Clark, C. Colbourn, and D. Johnson, "Unit disk graphs," *Discrete Mathematics*, vol. 86, pp. 165–177, 1990.
[3] J. Wu and F. Dai, "Broadcasting in ad hoc networks based on self-pruning," *In Proc. IEEE INFOCOM*, pp. 2240–2250, 2003.
[4] J. Wu and W. Lou, "Forward-node-set-based broadcast in clustered mobile ad hoc networks," *Wireless Communications and Mobile Computing*, vol. 3, no. 2, pp. 155–173, 2003.
[5] J. Wu and F. Dai, "A generic distributed broadcast scheme in ad hoc wireless networks," *IEEE Transactions on Computers*, vol. 53, no. 10, pp. 1343–1354, 2004.
[6] S. Ni, Y. Tseng, Y. Chen, and J. Sheu, "The broadcast storm problem in a mobile ad hoc network," *In Proc. ACM International Conference on Mobile Computing and Networking (MOBICOM)*, pp. 151–162, 1999.
[7] Z. Haas, J. Halpern, and L. Li, "Gossip-based ad hoc routing," *In Proc. IEEE INFOCOM*, pp. 1707–1716, 2002.
[8] D. Y. Sasson and A. Schiper, "Probabilistic broadcast for flooding in wireless mobile ad hoc networks," *In Proc. IEEE Wireless Communications and Networking Conference*, pp. 1124–1130, 2003.
[9] H. Liu, P. Wan, X. Jia, X. Liu, and F. Yao, "Efficient flooding scheme based on 1-hop information in mobile ad hoc networks," *In Proc. IEEE INFOCOM*, 2006.
[10] J. Wu, W. Lou, and F. Dai, "Extended multipoint relays to determine connected dominating sets in manets," *IEEE Trans. on Computers*, vol. 55, no. 3, pp. 334–347, 2006.
[11] M. Khabbazian and V. K. Bhargava, "Efficient broadcasting in mobile ad hoc networks," *IEEE Transactions on Mobile Computing: accepted for publication*, 2008.
[12] W. Peng and X. Lu, "On the reduction of broadcast redundancy in mobile ad hoc networks," *In Proc. ACM Interational Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pp. 129–130, 2000.
[13] I. Stojmenovic, M. Seddigh, and J. Zunic, "Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks," *IEEE Trans. on Parallel and Distributed Systems*, vol. 13, pp. 14–25, 2002.
[14] M. Khabbazian and V. K. Bhargava, "Localized broadcasting with guaranteed delivery and bounded transmission redundancy," *IEEE Transactions on Computers*, vol. 57, no. 8, pp. 1072–1086, 2008.
[15] C. T. Zahn, "Black box maximization of circular coverage," *Journal of Research of the National Bureau of Standards B.*, vol. 66, pp. 181–216, 1962.
[16] L. Barriére, P. Fraigniaud, and L. Narayanan, "Robust position-based routing in wireless ad hoc networks with unstable transmission ranges," pp. 19–27, 2001.
[17] P. Wan, K. Alzoubi, and O. Frieder, "Distributed construction of connected dominating set in wireless ad hoc networks," *In Proc. IEEE INFOCOM*, vol. 3, pp. 1597–1604, 2002.
[18] W. Wu, H. Du, X. Jia, Y. Li, and S. C. Huang, "Minimum connected dominating sets and maximal independent sets in unit disk graphs," *Theoretical Computer Science*, vol. 352, no. 1, pp. 1–7, 2006.

Appendix

Proof of Theorem 1: Suppose every cell of size $\alpha R \times \alpha R$ contains at most a constant number of nodes. The transmission range of each node can be covered with a constant number of cells as $\alpha \geqslant \frac{1}{c_0}$ and $c_0 \in O(1)$. Therefore, the number of 1-hop neighbors of each node that belong to CDS is bounded by a constant number C. Since every node in the CDS is a 1-hop neighbor of at least one node in the MCDS we get

$$|CDS| \leqslant C \times |MCDS|,$$

where $|CDS|$ and $|MCDS|$ denote the size of the CDS and size of the optimum solution, respectively.
Now, suppose that the size of the constructed CDS is guaranteed to be withing a constant factor of its optimal. Based on Lemma 2, Algorithm 1 guarantees that the the number of CDS nodes in every disk with radius $\frac{R}{2}$ is a constant. Consequently, the size of MCDS is $O((\frac{L}{R})^2)$ which is a constant when $(\frac{L}{R})$

is bounded by a constant number. By contradiction, assume that for every number m there is a distribution of nodes inside the network such that there exists a square cell in which the number of CDS nodes is larger than m. Therefore, the approximation factor increases as m increase, thus the algorithm cannot guarantee a constant approximation. ■

Proof of Theorem 2: The graph $G(R, (1-2\sqrt{2}\alpha)R)$ is connected. Thus, for every pair of nodes s and d there exists a sequence of nodes

$$s, w_1, w_2, \dots, w_{k-1}, w_k, d,$$

such that the Euclidean distance between any two consecutive nodes is no more than $(1-2\sqrt{2}\alpha)R$. Let $\mathcal{N}(u)$ denote the selected node in the cell in which the node u is located. The following sequence is a path between s and d in $G(V, R)$

$$s, \mathcal{N}(w_1), \mathcal{N}(w_2), \dots \mathcal{N}(w_{k-1}), \mathcal{N}(w_k), d.$$

It is because, based on Lemma 1, the distance between two consecutive nodes in the path is at most

$$(1-2\sqrt{2}\alpha)R + 2\sqrt{2}\alpha R = R.$$

When nodes s and d belong to the CDS we have $s = \mathcal{N}(s)$ and $d = \mathcal{N}(d)$. Based on Theorem 1, the size of constructed CDS is within a constant factor of its optimum. ■

Proof of Theorem 3: The proof is very similar to the proof of Theorem 2. ■

Proof of Theorem 4: Let

$$s, w_1, w_2, \dots, w_{k-1}, w_k, d,$$

be the shortest path between nodes s and d. Suppose $CDS_{sq}(\alpha)$ is a CDS constructed based on the high-transmission condition and $\mathcal{N}(u)$ is the selected node in the cell in which the node u is located. The sequence

$$s, \mathcal{N}(s), \mathcal{N}(w_1), \mathcal{N}(w_2), \dots \mathcal{N}(w_{k-1}), \mathcal{N}(w_k), d$$

is a path in the graph induced by $CDS_{sq}(\alpha) \cup \{s, d\}$. Consequently, we have $l_{CDS} \leqslant k+2$, thus

$$l_{CDS}(s, d) \leqslant l(s, d) + 1.$$

Note that node $\mathcal{N}(w_1)$ may not be in the transmission range of s. In this case node s needs to forward the packet to $\mathcal{N}(s)$ (i.e., the selected node in its cell). On the other hand, all nodes in d's cell are in the transmission range of $\mathcal{N}(w_k)$, thus $\mathcal{N}(w_k)$ can directly forward the packet to d.

If the CDS is constructed using a symmetric criteria, nodes s and d can be connected in the graph induced by the union of the CDS and the nodes s and d through

$$s, \mathcal{N}(s, w_1), \mathcal{N}(w_1, s), \mathcal{N}(w_1, w_2), \dots, \mathcal{N}(w_k, d), \mathcal{N}(d, w_k), d$$

where $\mathcal{N}(u, v)$ denotes the connector of u's cell to v's cell. Consequently, we have $l_{CDS}(s, d) \leqslant 2(k+1)+1$, thus

$$l_{CDS}(s, d) \leqslant 2 \times l(s, d) + 1.$$

■

Proof of Theorem 7: By contradiction, assume that node d does not receive the message after broadcast termination. Using the same argument in the proof of Theorem 5, we can show that there are two neighboring nodes u and w such that u has not received the message and w's color is red. Suppose w' has the highest priority among the red-neighbors of u. Clearly, w' cannot avoid broadcasting since its strong black condition cannot be satisfied. This is a contradiction as, based on the assumption, v cannot have a broadcasting neighbor.