Localized Broadcasting with Guaranteed Delivery and Bounded Transmission Redundancy

Hosna Jabbari, Majid Khabbazian and Vijay K. Bhargava

***Abstract*— The common belief is that localized broadcast algorithms are not able to guarantee both full delivery and a good bound on the number of transmissions. In this paper, we propose the first localized broadcast algorithm that guarantees full delivery and a constant approximation ratio to the minimum number of required transmissions in the worst case. The proposed broadcast algorithm is a self-pruning algorithm based on 1-hop neighbor information. Using the proposed algorithm, each node determines its status (forwarding/non-forwarding) in $O(\Delta_G \log \Delta_G)$, where Δ_G is the maximum node degree of the network. By extending the proposed algorithm, we show that localized broadcast algorithms can achieve both full delivery and a constant approximation ratio to the optimum solution with message complexity $O(N)$, where N is the total number of nodes in the network and each message contains a constant number of bits. We also show how to save bandwidth by reducing the size of piggybacked information. Finally, we relax several system-model assumptions, or replace them with practical ones, in order to improve the practicality of the proposed broadcast algorithm.**

I. Introduction

Broadcasting is a fundamental communication primitive in which a message is sent from a source node to all other nodes in the network. Broadcasting has many applications, including route discovery in wireless ad hoc routing protocols, and is frequently used to adapt network changes caused by the dynamic nature of ad hoc networks. The simplest broadcast mechanism is flooding, in which every node retransmits the first copy of the received message to all of its 1-hop neighbors. Despite its simplicity, flooding can cause a large number of redundant transmissions, wasting valuable network resources such as bandwidth and power. It can also lead to significant performance degradation and network congestion. Therefore, it is essential to design efficient broadcast algorithms in order to reduce the number of redundant transmissions. Unfortunately, it is not practical to design a delivery-guaranteed broadcast algorithm that eliminates all redundant transmissions. This is because the problem of finding the minimum Connected Dominating Set (CDS) in a unit disk graph can be reduced to the problem of eliminating all of the redundant transmissions. A set of nodes is called a Dominating Set (DS) if every node in the network either belongs to the set or is a 1-hop neighbor of a node in the set. It is well known that finding the Minimum CDS (MCDS) of a unit disk graph is NP-hard in general [1], [2]. Note that every CDS can be used as a backbone of the network to broadcast the message. On the other hand, the forwarding nodes in the delivery-guaranteed broadcast algorithms form a CDS. Therefore, broadcast algorithms can be used to find a CDS if a source node is selected to initiate the broadcast. Consequently, the problems of finding a minimum CDS and of designing an optimum broadcast algorithm can be reduced to each other.

To reduce the number of redundant transmissions broadcast algorithms typically impose some bandwidth overhead to, for example, collect neighbor information via message exchanges. It is desirable to reduce this bandwidth overhead to improve the practicality of the broadcast algorithm for ad hoc networks with frequent topology changes. In [3], the authors prove that every distributed algorithm for constructing a non-trivial CDS has the lower message complexity bound of $\Omega(N \log N)$, where N is the number of nodes and the message size is a constant multiple of the number of bits representing the node IDs (a CDS is said to be trivial if it consists of all nodes). Let us define a distributed broadcast algorithm as a non-trivial broadcast algorithm if the forwarding nodes always form a non-trivial CDS. In this paper, we show that the same lower bound does not hold for non-trivial broadcast algorithms, although they are closely related to the problem of finding a non-trivial CDS. In fact, we show that an extension of our proposed non-trivial broadcast algorithm requires $O(N)$ messages, where the message size is a constant. Moreover, we show that the number of forwarding nodes using the extended algorithm is within a constant factor of the optimum solution (i.e., minimum CDS).

Computational overhead also plays an important role in designing efficient broadcast algorithms. Flooding is an ideal broadcast algorithm in terms of computational overhead, since each node requires almost no computation (it simply broadcasts the first copy of the received message). The existing broadcast algorithms typically require some computation for selecting forwarding nodes or for self-pruning. In this paper, we show how to efficiently implement the proposed broadcast algorithm using some computational geometry techniques to reduce the computational overhead.

A. Related Work

Different approaches can be used to classify the existing broadcast algorithms. One approach is based on whether or not they use a previously constructed backbone. One solution for reducing the number of redundant transmissions is to form a connected dominating set of nodes and use them as a backbone to broadcast the message. As mentioned earlier, finding the Minimum CDS (MCDS) is an NP-hard problem. However, there are distributed algorithms to find a small-sized CDS with constant approximation ratio to the MCDS [3], [4]. The main drawback of this solution is that maintaining a CDS is often costly in networks with frequent topology changes [5].

The second approach to classifying the existing broadcast algorithms is based on whether global or local information is employed by the algorithm. An algorithm is global if it uses whole or partial global state information. On the other hand, a distributed algorithm is localized if it is based on solely local information. It is clear that global broadcast algorithms are not appropriate for mobile ad hoc networks due to the dynamic nature of the network. In addition, they may not scale well when the number of nodes in the network increases.

The localized broadcast algorithms typically use k-hop neighbor information, where $k \geqslant 0$. A node can collect k-hop neighbor information using k rounds of neighbor set information exchange. Some broadcast algorithms do not need any neighbor information (i.e., $k = 0$). Flooding and probabilistic broadcast algorithms such as [6] and [7] fit in this category. These algorithms cannot typically guarantee full delivery or effectively reduce the number of transmissions. Moreover, to decide whether or not to broadcast, they may use a threshold (such as probability of broadcast) which may not be easy to find for different network situations.

Most existing localized broadcast algorithms use k-hop neighbor information, where $k \geqslant 1$ is a small number. These algorithms can be further divided into neighbor-designating (sender-based) and self-pruning (receiver-based). In neighbor-designating algorithms [5], [8], the forwarding status of each node is determined by other nodes. In other words, each broadcasting node selects a subset of its k-hop neighbors to forward the message. The main design challenge for neighbor-designating algorithms is to choose a small subset of nodes to forward the message. For example, in [5], the authors propose a broadcast algorithm based on 1-hop neighbor information that selects the smallest subset of its 1-hop neighbors with the maximum coverage area, where the coverage area of a set of nodes is the union of their transmission coverage. When 2-hop neighbor information is available, it can be extended to select the minimum number of 1-hop neighbors that cover all 2-hop neighbors. This procedure is known as the minimum forwarding set problem. There are heuristic algorithms in the literature that give a constant approximation ratio to the minimum forwarding set problem [9]. However, as shown in [8], even an optimal solution to this problem may not result in a small-sized CDS in the network. In [8], the authors extend the minimum forwarding set problem to the case where the complete 2-hop topology information is available. Note that two rounds of information exchange provide partial topology information for the 2-hop neighbor set. To get complete 2-hop topology information, each node must either get the position information of the 2-hop neighbors or perform three rounds of information exchange [8]. The authors show that their proposed algorithm can provide a probabilistic constant approximation ratio to MCDS, but point out that it performs poorly when the network becomes extremely dense.

In the self-pruning algorithms, each node determines its status (forwarding/non-forwarding) using local information [10], [11], [12]. Clearly, the main design challenge with regard to self-pruning algorithms is to find an effective self-pruning condition. Different self-pruning conditions have been proposed in the literature. For example, one simple self-pruning condition is whether all the neighbors have received the message before the defer timer expiration. Similar to the neighbor-designating algorithms, the existing self-pruning algorithms do not provide a constant approximation ratio to the optimal solution (MCDS).

B. *Our Contribution*

In the first part of the paper, we consider two general structures commonly used in neighbor-designating and self-pruning algorithms based on 1-hop neighbor information. We show that using these structures we are not able to guarantee both full delivery and a good bound on the number of forwarding nodes in the network. An open question is whether localized broadcast algorithms can provide both full delivery and constant approximation to the optimal solution (MCDS). As mentioned earlier, finding MCDS is an NP-hard problem. In the absence of global network information, this problem becomes even more challenging. It is a common belief that localized broadcast algorithms (e.g. self-pruning algorithms) cannot guarantee a constant approximation ratio to the optimal solution [10], [13]. One contribution of this paper is to show that localized broadcast algorithms are able to guarantee a reasonable bound on the number of forwarding nodes. In particular, we propose a self-pruning algorithm and prove that it provides full delivery as well as guaranteeing a constant approximation ratio to the optimal solution (MCDS). We design efficient algorithms with nearly optimum computational complexity to compute the proposed self-pruning condition.

It is proved in [3] that every distributed algorithm for constructing a non-trivial CDS has a lower message complexity bound of $\Omega(N \log N)$, where N is the number of nodes and the message size is a constant multiple of the number of bits representing the node IDs. We show that the same message complexity lower bound does not apply to the non-trivial broadcast algorithms. In fact, we show that the message complexity of an extension of our proposed algorithm is $O(N)$, where N is the number of nodes and the message size is a constant. We prove that the extended broadcast algorithm can also guarantee full delivery and a constant approximation ratio to MCDS.

We show how to reduce the bandwidth requirements by reducing the amount of piggybacked information in each broadcast packet. We relax several system-model assumptions, or replace them with practical ones, to improve the practicality of the proposed broadcast algorithm. Finally, we verify the analytical results using simulation and provide a comparison with one of the best broadcast algorithms based on 1-hop information. The simulation results show that the proposed broadcast algorithm performs better than what was analytically guaranteed for the worst case.

II. System Model

In this section, we define our system model. We discuss how to relax some of its underlying in Section V.

We consider a wireless network as a collection of N nodes represented by points located at their positions in a 2-D plane. Each node is equipped with an omni-directional antenna that has radio transmission range of R. Two nodes are called neighbors if they are in transmission range of each other (i.e., the Euclidean distance between them is less than or equal to R). We assume that each node is able to obtain its position using an existing positioning technique, such as the Global Positioning System (GPS). Each node periodically broadcasts a short HELLO message containing its unique ID and its position. Therefore, each node knows the position of its 1-hop neighbors as well.

To prove that a broadcast algorithm guarantees full delivery, we assume that nodes are static during the broadcast and that the Medium Access Control (MAC) layer is ideal, i.e., there are no transmission errors such as collision and contention. In addition, we assume that the network is connected. A broadcast algorithm can be considered a delivery-guaranteed broadcast algorithm if it guarantees full delivery under these assumptions. Note that without these assumptions, even flooding may not achieve full delivery.

III. Broadcast Algorithms Based on 1-hop Neighbor Information

In this section, we consider two general structures commonly used in neighbor-designating and self-pruning algorithms based on 1-hop neighbor information. We show that using these structures we are not able to guarantee both full delivery and a good bound on the number of forwarding nodes in the network.

A. Neighbor-Designating Algorithms

Algorithm 1 shows a general structure of neighbor-designating broadcast algorithms based on 1-hop neighbor information. As shown in Algorithm 1, each node schedules a broadcast for a received message if the node is selected by the sender and if it has not received the same message before. Algorithm 1 can be extended by allowing each node to schedule a broadcast for a message as long as the node is selected by the sender and the message has not been previously broadcast. Clearly, each message is broadcast once at most by a node in both Algorithm 1 and its extension. However, using Algorithm 1, each node may only schedule a broadcast when it receives a message for the first time. By contrast, in the algorithm extension, a broadcast schedule can be set at any time. For example, a message can be dropped after the first reception but scheduled for broadcast the second time.

Corollary 1 shows that almost all the nodes on the boundary of the network (including all the nodes on the convex hull [14] of the network) will broadcast the message if Algorithm 1 or its extension is used as the basic structure in a delivery-guaranteed broadcast algorithm based on 1-hop neighbor information. Consequently, neighbor-designating algorithms based on 1-hop neighbor information may not be efficient if a large number of nodes (compared to the total number of nodes) are located on the boundary of the network. It also follows from Theorem 1 that a broadcast algorithm based on 1-hop neighbor information cannot guarantee both full delivery and a good bound on the number of forwarding nodes if it uses Algorithm 1 or its extension as its basic structure.

Theorem 1. *Let us represent a node N_A with a point A located at its position and use $\overline{PQ}$ to denote the Euclidean distance between two points P and Q. Suppose Algorithm 1 or its extension is used as the basic structure in a delivery-guaranteed broadcast algorithm based on 1-hop neighbor information. Node N_A will be selected to forward the message if there is a point P such that $\overline{AP} \leqslant R$ and for every node $N_B \neq N_A$ in the network $\overline{BP} > R$.*

Proof: Each node has 1-hop neighbor information, thus only N_A knows whether or not there is a node at the position of point P. Consequently, N_A must be selected to forward the message in order to guarantee full delivery. ∎

Corollary 1. *Almost all of the nodes on the boundary of the network will broadcast the message if Algorithm 1 or its extension is used as the basic structure in a delivery-guaranteed broadcast algorithm based on 1-hop neighbor information.*

Proof: Suppose that N_A is a node on the boundary of the network. With high probability, there is a point P outside the network boundary such that $\overline{AP} \leqslant R$ and $\overline{BP} > R$ for every node $N_B \neq N_A$. For example, suppose that there is a line L containing A such that all the nodes are either on the line or one side of it. This situation occurs, for instance, when A is a vertex of the network convex hull [14]. Consider a point P on the other side of the line L such that $\overline{AP} = R$ and $AP \perp L$. It is clear that $\overline{BP} > R$ for every node $N_B \neq N_A$. Therefore, based on Theorem 1, N_A has to be selected to forward the message. ∎

Corollary 2. *Suppose Algorithm 1 or its extension is used as the basic structure in a delivery-guaranteed broadcast algorithm based on 1-hop neighbor information. The ratio of the number of broadcasting nodes over the minimum number of required broadcasts can be as large as N, where N is the number of nodes in the network.*

Proof: As shown in Figure 1, suppose that all the nodes are located on a line segment AB of length R or on a circle $C_{O,\frac{R}{2}}$ with a radius $\frac{R}{2}$, where R is the radio transmission range. Clearly, one broadcast is enough to transmit a message to all the nodes in the network for each scenario. However, based on Theorem 1, all of the nodes will broadcast the message, because for every node N_A, we can find a point P outside the network boundary such that $\overline{AP} = R$ and AP is orthogonal to the line segment AB/Circle $C_{O,\frac{R}{2}}$. It is easy to show that $\overline{BP} > R$ for every node $N_B \neq N_A$. Consequently, the ratio of broadcasting nodes over the minimum number of required broadcasts is N. ∎

Algorithm 1: A general structure of neighbor-designating algorithms

Extract the required information from the received message M;
if M has been received before or does not contain node's *ID*
  drop the message;
else
  set a defer timer;
When defer timer expires
  Select a subset of neighbors to forward the message;
  Attach the list of forwarding nodes to the message;
  Broadcast the message;

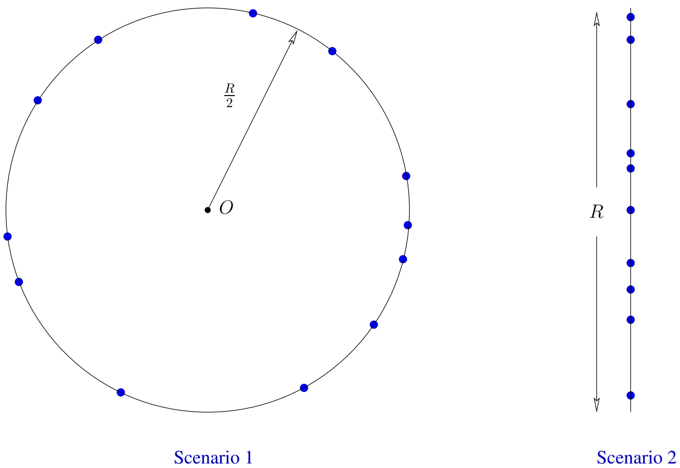

Fig. 1. Two worst-case examples for Corollary 2.

B. Self-Pruning Algorithms

Algorithm 2 shows a general structure of self-pruning broadcast algorithms based on 1-hop neighbor information. Using this structure, a node schedules a broadcast for the first received copy of the message. When the defer timer expires, the node may refrain from broadcasting the message if a certain self-pruning condition is satisfied. Theorem 2 shows that this structure cannot guarantee both full delivery and a good bound on the number of forwarding nodes. Note that in Algorithm 2, no information is piggybacked with the broadcast packet. An extension of Algorithm 2 is to allow nodes to piggyback some information in the broadcast packet. Clearly, this extension of Algorithm 2 is stronger than both Algorithm 1 and Algorithm 2. In the next section, we show that a broadcast algorithm based on 1-hop neighbor information can guarantee both full delivery and a good bound on the number of forwarding nodes (a constant approximation ratio to MCDS) if it employs this extension of Algorithm 2.

Theorem 2. *Suppose Algorithm 2 is used as the basic structure in a delivery-guaranteed broadcast algorithm based on 1-hop neighbor information. The ratio of the number of broadcasting nodes over the minimum number of required broadcasts is $O(N)$ in the worst case.*

Proof: Consider the case where all the nodes are on two circles $C_{O,\frac{R}{2}}$ and $C_{O,\frac{3R}{2}}$ centered at O with radii $\frac{R}{2}$ and $\frac{3R}{2}$, respectively. As shown in Figure 2, suppose that for every node N_{A_i} on $C_{O,\frac{R}{2}}$ there is a corresponding node N_{B_i} on $C_{\frac{3R}{2}}$ (and vice versa) such that $\overline{A_iB_i} = R$. By contradiction, suppose that neither N_{A_i} nor N_{B_i} broadcast the message. Note that N_{A_i} and N_{B_i} do not share any neighbor. Since they have only 1-hop neighbor information and there is no information piggybacked in the broadcast packet, N_{A_i} (or N_{B_i}) cannot be sure whether its corresponding node has received the message. Therefore, in order to guarantee full delivery, either N_{A_i} or N_{B_i} has to broadcast the message. However, only a constant number of transmissions is enough to send the message to all the nodes (at most 11 transmissions for the considered case). Therefore, the ratio of broadcasting nodes over the minimum number of required broadcasts is $O(N)$. ■

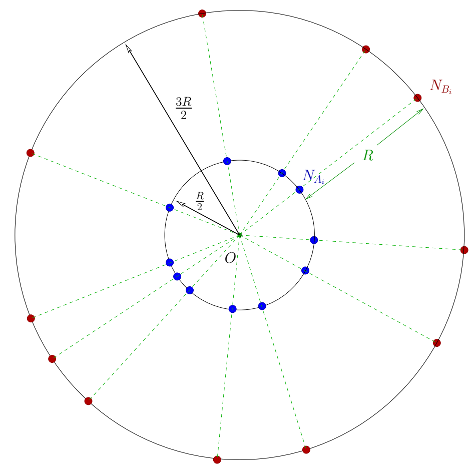

Fig. 2. A worst-case example for Theorem 2.

Algorithm 2: A general structure of self-pruning algorithms

if the message M has been received before
  Drop the message;
else
  Set a defer timer;
when defer timer expires
  Decide whether or not to broadcast M;

IV. An Efficient Self-Pruning Broadcast Algorithm

In the previous section, we considered two structures typically used in the broadcast algorithms based on 1-hop neighbor information. We showed that these structures cannot guarantee the following properties simultaneously:

- full network delivery
- good bound on the number of broadcasting nodes

An open question is whether a localized broadcast algorithm (e.g., a neighbor-knowledge-based broadcast algorithm) is able to guarantee these properties. In this section, we propose a novel self-pruning broadcast algorithm based on 1-hop neighbor information that not only guarantees full delivery but also a constant approximation ratio to the optimal solution (MCDS). As its basic structure, the proposed broadcast algorithm (Algorithm 3) uses a simple extension of Algorithm 2 in which each node is allowed to piggyback some information within the broadcast packet. In Algorithm 3, each node adds a list of its 1-hop neighbor IDs and positions into the broadcast packet. Upon receiving a packet, this information is extracted and used to determine the node's status (forwarding/non-forwarding) based on the following self-pruning condition.

Definition 1 (Responsibility Condition). *Node N_A is pruned (has non-forwarding status) if it is not responsible for any of its neighbors. Node N_A is not responsible for its neighbor N_B if N_B has received the message or if there is another node N_C such that N_C has received the message and N_B is closer to N_C that N_A.*

Algorithm 3: The proposed broadcast algorithm

Extract the required information from the received packet;
Add the broadcasting node and its 1-hop neighbor information (except the node's own information) to the list $List_{ID}^{Rec}$.
if the message has been received before
    drop the message;
else
    set a defer timer;
When defer timer expires
    if the responsibility condition is satisfied
        Remove the previous information attached to the message;
        Attach the list of 1-hop neighbors to the message;
        Broadcast the packet;

Suppose N_A computes a list of nodes that have received the message ($List_{N_A}^{Rec}$) by collecting the information piggybacked in the broadcast packets. To check the responsibility condition, N_A can first determine which neighbors have not received the message. The responsibility condition is then satisfied if and only if there is a neighbor N_B that has not received the message and

$$\forall N_C \in List_{N_A}^{Rec}: \quad \overline{AB} \leqslant \overline{CB}.$$

In Section IV-B, we describe efficient algorithms for computing the responsibility condition.

Example 1. *As shown in Figure 3, N_A has seven neighbors. Suppose that N_A has received the message from N_H. Recall that N_A extracts the list of N_H's neighbors from the received packet. Therefore, it knows that N_E, N_F and N_G have received the message and N_B, N_C and N_D have not. According to the responsibility condition, N_A is not required to broadcast because*

$$\overline{BE} < \overline{BA}, \quad \overline{CF} < \overline{CA} \quad \textit{and} \quad \overline{DG} < \overline{DA}\,(\textit{or}\ \overline{DF} < \overline{DA}).$$

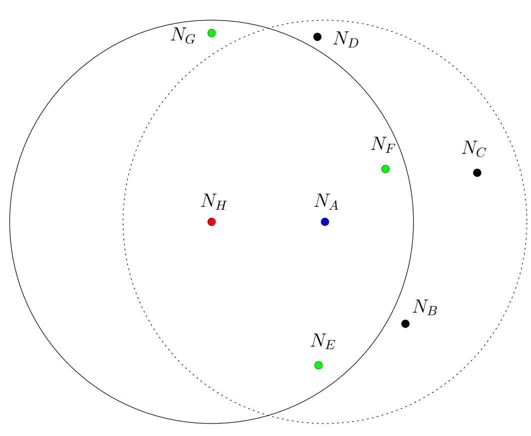

Fig. 3. An example of self-pruning based on the responsibility condition.

A. Analysis of the Proposed Broadcast Algorithm

In this section, we prove that Algorithm 3 can achieve full delivery and a constant approximation ratio to MCDS. In order to prove these properties, we assume that nodes are static during the broadcast, the networks is connected and the MAC layer is ideal, i.e., there is no communication error. As mentioned earlier in Section II, even flooding cannot guarantee full delivery without these assumptions.

Theorem 3. *Algorithm 3 guarantees that all the nodes in the network will receive the message.*

Proof: Using Algorithm 3, each node broadcasts the message at most once. Therefore, broadcasting will eventually terminate. By contradiction, suppose there is at least one node that has not received the message after the broadcast termination. Let us consider the set

$$\begin{aligned}\Lambda = \{(N_X, N_Y) | N_X \text{ and } N_Y \text{ are neighbors,} \\ N_X \text{ has received the message and} \\ N_Y \text{ has not received the message}\}\end{aligned}$$

Suppose N_S is the source node (the node that initiated the broadcast) and N_T is a node that has not received the message. The network is connected, thus there is a path between N_S and N_T. Clearly, we can find two neighbor nodes N_B and N_A along the path from N_T to N_S such that N_B has not received the message, while N_A has. Consequently, $(N_A, N_B) \in \Lambda$, thus $\Lambda \neq \emptyset$. As a result,

$$\exists (N_{A'}, N_{B'}) \in \Lambda \text{ s.t. } \forall (N_X, N_Y) \in \Lambda: \quad \overline{A'B'} \leqslant \overline{XY}. \quad (1)$$

Clearly, $N_{A'}$ has not broadcast since $N_{B'}$ has not received the message. Therefore, there must be a node $N_{C'}$ such that $N_{C'}$ has received the message and $\overline{C'B'} < \overline{A'B'} \leqslant R$. This result contradicts to (1), since $(N_{C'}, N_{B'}) \in \Lambda$. ∎

Lemma 1. *Using Algorithm 3, the number of broadcasting nodes inside a disk $D_{O,\frac{R}{4}}$ with a radius $\frac{R}{4}$ is bounded by a constant.*

Proof: As shown in Figure 4, consider three disks $D_{O,\frac{R}{4}}$, $D_{O,\frac{3R}{4}}$ and $D_{O,\frac{5R}{4}}$, centered at O with radii $\frac{R}{4}$, $\frac{3R}{4}$ and $\frac{5R}{4}$, respectively. Let us define

$$\mathcal{R}_1 - \mathcal{R}_2 = \{P | P \in \mathcal{R}_1 \text{ and } P \notin \mathcal{R}_2\},$$

where $\mathcal{R}_1$ and $\mathcal{R}_2$ are two regions and P is a point. Suppose k is the minimum number such that for every set of k points $P_i \in D_{O,\frac{5R}{4}} - D_{O,\frac{3R}{4}}$, $1 \leqslant i \leqslant k$, we have

$$\exists P_i, P_j: \quad i \neq j \quad \text{and} \quad \overline{P_i P_j} \leqslant \frac{R}{2}.$$

Note that the area $D_{O,\frac{5R}{4}} - D_{O,\frac{3R}{4}}$ can be covered with a constant number of disks with radius $\frac{R}{4}$. If the number of points inside $D_{O,\frac{5R}{4}} - D_{O,\frac{3R}{4}}$ is greater than the number of covering disks, at least one covering disk will contain more than one point. For two points P_1 and P_2 inside a disk with a radius $\frac{R}{4}$, we have $\overline{P_1 P_2} \leqslant \frac{R}{2}$. Therefore, k is bounded by a constant (the number of covering disks plus one).

We prove that the number of broadcasting nodes inside $D_{O,\frac{R}{4}}$ is less than or equal to k. By contradiction, suppose that there are more than k broadcasting nodes inside $D_{O,\frac{R}{4}}$. Let $N_{A_i} \in D_{O,\frac{R}{4}}$, $0 \leqslant i \leqslant k$ be the first $k+1$ broadcasting nodes ordered chronologically based on their broadcast time. Based on the responsibility condition, for every node N_{A_i} there is a corresponding neighbor N_{B_i} such that $N_{B_i} \notin List^{Rec}_{N_{A_i}}$ and $\overline{A_iB_i} \leqslant \overline{CB}_i$ for every node $N_C \in List^{Rec}_{N_{A_i}}$, where $List^{Rec}_{N_{A_i}}$ is the list of nodes that have received the message based on N_{A_i}'s collected information. Note that every node inside $D_{O,\frac{3R}{4}}$ will receive the message after N_{A_0}'s broadcast. Therefore, $N_{B_i} \notin D_{O,\frac{3R}{4}}$ for $1 \leqslant i \leqslant k$. On the other hand, N_{B_i} is a neighbor of N_{A_i}, thus $N_{B_i} \in D_{O,\frac{5R}{4}}$. Consequently,

$$\forall 1 \leqslant i \leqslant k : \quad N_{B_i} \in D_{O,\frac{5R}{4}} - D_{O,\frac{3R}{4}}.$$

It follows that

$$\forall 1 \leqslant i \leqslant k : \quad \overline{A_iB_i} > \frac{R}{2}. \tag{2}$$

Suppose $1 \leqslant i < j \leqslant k$. The nodes N_{A_i} and N_{A_j} are neighbors, because $\overline{A_iA_j} \leqslant \frac{R}{2}$. Recall that N_{A_i} piggybacks the list of its neighbors within the broadcast packet. Therefore, $N_{B_i} \in List^{Rec}_{N_{A_j}}$, thus $N_{B_i} \neq N_{B_j}$. It follows that $N_{B_1} \ldots N_{B_k}$ are k different points inside $D_{O,\frac{5R}{4}} - D_{O,\frac{3R}{4}}$. Therefore

$$\exists N_{B_i}, N_{B_j} : \quad i < j \quad \text{and} \quad \overline{B_iB_j} \leqslant \frac{R}{2}. \tag{3}$$

This is a contradiction, because $N_{B_i} \in List^{Rec}_{N_{A_j}}$ and based on (2) and (3)

$$\overline{A_jB_j} > \frac{R}{2} \Longrightarrow \overline{A_jB_j} > \overline{B_iB_j}.$$

Thus, according to the responsibility condition, N_{A_j} is not responsible for N_{B_j}. ■

Theorem 4. *Algorithm 3 gives a constant approximation ratio to the optimal solution (MCDS).*

Proof: Clearly, a disk with radius R can be covered with a constant number of disks with radius $\frac{R}{4}$. Therefore, based on Lemma 1, the number of broadcasting nodes inside a disk with radius R is bounded by a constant C_{min}. Let $|MCDS|$ be the number of nodes in the MCDS. Each broadcasting node is inside the transmission range of at least one node in th MCDS. On the other hand, the number of broadcasting nodes inside the transmission range of each node in the MCDS is bounded by C_{min}. Therefore, the total number of broadcasting nodes is not more than $C_{min} \times |MCDS|$. ■

The following theorem shows that the message complexity of the proposed broadcast algorithm is $O(N)$, where N is the total number of nodes in the network and each message contains a node's ID and its position. In Section V, we will show that the message size can be reduced to a constant number of bits by removing the node ID from the message and using a constant number of bits to represent each node's position.

Theorem 5. *The message complexity of Algorithm 3 is $O(N)$, where N is the total number of nodes in the network and each message contains a node's ID and its position.*

Proof: There are two types of messages used in the proposed broadcast algorithm. First, each node broadcasts its ID and position in a short HELLO message. Second, each broadcasting node piggybacks the list of its 1-hop neighbor IDs and positions into the broadcast packet. We will consider this packet as n separate messages, where n is the number of 1-hop neighbors of the broadcasting node. Therefore, each message contains exactly one node's ID together with its position. As shown in the proof of Theorem 4, the number of broadcasting neighbors of each node is bounded by a constant C_{min}. Therefore, each node's ID and position appears in at most $C_{min}+1$ messages, including the HELLO message. Consequently, the total number of messages is not more than $(C_{min}+1) \times N$. ■

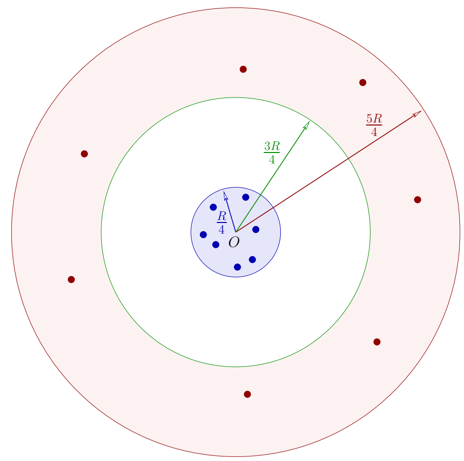

Fig. 4. Three co-centered disks; $N_{A_i} \in D_{O,\frac{R}{4}}$ and $N_{B_i} \in D_{O,\frac{5R}{4}} - D_{O,\frac{3R}{4}}$.

B. Efficient Algorithms for Computing the Responsibility Condition

The problem of computing the responsibility condition can be expressed by the following geometry problem:

Definition 2 (Closest-Point Problem (CPP)). *Let $\mathcal{P} = \{P_0, P_1, \ldots, P_m\}$ and $\mathcal{Q} = \{Q_1, Q_2, \ldots, Q_n\}$ be two disjoint sets of points. Is there a point $Q \in \mathcal{Q}$ such that*

$$\forall P \in \mathcal{P} : \quad \overline{QP_0} \leqslant \overline{QP}?$$

The closest-point problem attempts to establish whether or not P_0 is the closest point in $\mathcal{P}$ to at least one point in $\mathcal{Q}$. One approach to solving CPP is to find the closest point in $\mathcal{P}$ to every point in $\mathcal{Q}$. The trivial algorithm for finding the closest point in $\mathcal{P}$ to a query point $Q \in \mathcal{Q}$ is to compute the distance from Q to all the points in $\mathcal{P}$, keeping track of the "closest point so far". Clearly, using this algorithm, the computational complexity of solving the CPP would be $O(m \times n)$. Finding the closest point in $\mathcal{P}$ to a query point $Q \in \mathcal{Q}$ is a well-known problem called the Nearest Neighbor Search (NNS) problem. To solve the NNS problem, several space-partitioning methods, including kd-tree and the Voronoi diagram [14], can be used. A kd-tree is a data structure used for organizing points in k-dimensional space. A kd-tree for the set of 2-D points $\mathcal{P}$ can be constructed in $O(m \log m)$. Moreover, the computational complexity of a query is $O(\log m)$ [14]. Therefore, using kd-tree, the CPP can be solved in $O(m \log m + n \log m)$. Similarly, we can use the Voronoi diagram to solve CPP. The Voronoi diagram is a partitioning of a plane with n generating points into convex polygons called Voronoi cells, such that each cell contains exactly one generating point and every node in the cell is closer to the generating point of the cell than to other generating points. It is known that the Voronoi diagram can be constructed in $O(n \log n)$ and that each cell has $O(n)$ edges in the worst case. Therefore, using the set of points $\mathcal{P}$ as the generating points, the Voronoi diagram can be generated in $O(m \log m)$. The node P_0 is the closest node in $\mathcal{P}$ to a query point Q if and only if Q is in the Voronoi cell of P_0. We can check whether a query point Q is in the Voronoi cell of P_0 in $O(\log m)$, since the Voronoi cell of P_0 is a convex polygon with at most $O(m)$ edges. Therefore, the CPP can be solved in $O(m \log m + n \log m)$ using the Voronoi diagram. Note that we only need to compute the Voronoi cell of P_0. The Voronoi cell of P_0 can be computed in $O(m \log h)$, where h is the number of edges of the cell. The average number of vertices of a Voronoi cell is less than six [14], thus h is a constant on average. Therefore, the average case computational complexity of this approach is $O(m + n)$.

Suppose P_0 is not the closest point in $\mathcal{P}$ to any node in $\mathcal{Q}$. In this case, $\Omega(n)$ is a lower bound for every algorithm to solve the CPP for sets $\mathcal{P}$ and $\mathcal{Q}$, because the algorithm cannot ignore any node in the set $\mathcal{Q}$. On the other hand, every algorithm to solve the CPP has to consider all the points in $\mathcal{P}$ if P_0 is the closest point in $\mathcal{P}$ to a node in $\mathcal{Q}$. Thus, $\Omega(m)$ is also a lower bound for such algorithms. Considering both cases, it follows that $\Omega(m + n)$ is a lower bound for every algorithm to solve the CPP. We described algorithms that compute the CPP in $O((n + m) \log m)$, in the worst case. These algorithms are nearly optimal in terms of computational complexity, because $\log m$ is a small factor in practice. Moreover, by constructing the Voronoi cell of P_0, we showed that the CPP can be solved in $O(n + m)$ in the average case.

Theorem 6. *The worst-case complexity of computing the responsibility condition is $O(\Delta_G \log \Delta_G)$, where Δ_G is the maximum node degree of the network.*

Proof: Based on Algorithm 3, a node N_A stores the broadcasting neighbors' information and their 1-hop neighbors' information (except N_A's information) in $List_{N_A}^{Rec}$. The node N_A can compute $List_{N_A}^{NotRec}$, the list of its neighbors that have not received the message, in $O(l \times n)$, where $1 \leqslant l \leqslant n$ is the number of broadcasting neighbors and n is the total number of neighbors. Consider the set of points

$$\mathcal{P} = \{P_0, P_1, \ldots, P_m\}$$

where P_0 is located at the position of N_A and $P_1, P_2, \ldots, P_m$ are located at the positions of the nodes in $List_{N_A}^{Rec}$. Suppose

$$\mathcal{Q} = \{Q_1, Q_2, \ldots, Q_k\}$$

is the set of points located at the position of the nodes in $List_{N_A}^{NotRec}$. As mentioned earlier, by constructing the Voronoi cell of P_0, we can compute the CPP in $O((m+k) \log m)$ for sets $\mathcal{P}$ and $\mathcal{Q}$. Clearly, the responsibility condition is satisfied if and only if there exists a point $Q \in \mathcal{Q}$ such that

$$\forall P \in \mathcal{P} : \quad \overline{QP_0} \leqslant \overline{QP}.$$

As shown in the proof of Theorem 4, l (the number of broadcasting neighbors of N_A) is bounded by a constant. Moreover, we have $k \leqslant n$, $m \leqslant l \times \Delta_G$ and $n \leqslant \Delta_G$, where Δ_G is the maximum node degree of the network. Therefore, the complexity of computing the responsibility condition is

$$O(l \times n) + O((m + k) \log m) = O(\Delta_G \log \Delta_G).$$

Note that the CPP can be solved in $O(k + m)$ in the average case. Therefore, the average complexity of computing the responsibility condition is

$$O(l \times n) + O(m + k) = O(\Delta_G).$$

■

C. Reducing Bandwidth Requirements

In Algorithm 3, each forwarding node piggybacks the list of its 1-hop neighbors in the broadcast packet. In this section, we show how a forwarding node can reduce bandwidth overhead by piggybacking a list of a subset of its 1-hop neighbors.

Definition 3 (Representative Set). *Let $\mathcal{P}$ be a set of points $\{P_1, P_2, \ldots, P_n\}$ inside a circle $C_{O,R}$. The set $\mathcal{S} \subseteq \mathcal{P}$ is a representative set for $\mathcal{P}$ if for every point Q outside $C_{O,R}$ (i.e., $\overline{OQ} > R$)*

$$\exists P' \in \mathcal{S} \quad s.t. \quad \forall P \in \mathcal{P} : \quad \overline{P'Q} \leqslant \overline{PQ}.$$

A representative set $\mathcal{S}_{min}$ of $\mathcal{P}$ is called the minimum representative set of $\mathcal{P}$ if $\mathcal{S}_{min}$ has the minimum cardinality among all the representative sets of $\mathcal{P}$.

Definition 4 (Representative Neighbor Set). *We say P is the corresponding point of the node N_X if P is located at the position of N_X. Let $\mathcal{P}$ be the set of corresponding points of N_A's neighbors. A subset of N_A's neighbors is called a representative neighbor set if their corresponding point set is a representative set of $\mathcal{P}$. The set of N_A's representative neighbors with the minimum number of nodes is called the minimum representative neighbor set.*

Lemma 2. *In Algorithm 3, a forwarding node can piggyback a list of its representative neighbors instead of the list of all its neighbors, without affecting the forwarding status of any other node in the network.*

Proof: Let $List_{N_A}^{Rep}$ be a list of representative nodes of N_A. Suppose N_A piggybacks $List_{N_A}^{Rep}$ instead of the list of all its neighbors and $N_B \notin List_{N_A}^{Rep}$ is a neighbor of N_A. The node N_B is required in computing the responsibility condition of N_A's neighbor N_C if N_C has a neighbor N_D that has not received the message and $\overline{BD} < \overline{CD}$. Note that N_D is not a neighbor of N_A since all of N_A's neighbors have received the message. Therefore, N_D is outside $C_{A,R}$, where $C_{A,R}$ is the transmission range of N_A. Based on the definition of representative neighbor set, there exists a node $N_E \in List_{N_A}^{Rep}$ such that

$$\overline{ED} \leqslant \overline{BD} \quad \Longrightarrow \quad \overline{ED} < \overline{CD}.$$

Therefore, for each neighbor of N_A required in computing the responsibility condition of N_C, we can find a node in $List_{N_A}^{Rep}$ that leads to the same result. ∎

Lemma 2 shows that forwarding nodes can save bandwidth by piggybacking the list of their representative neighbors instead of the list of all their neighbors. Clearly, finding the minimum representative neighbor set is equivalent to computing the minimum representative set of $\mathcal{P}$, where $\mathcal{P}$ is the set of points corresponding to the node's neighbors. Let us denote the Voronoi diagram of $\mathcal{P}$ by Vor($\mathcal{P}$). Assume that $Cell(P)$ is the set of vertices of the cell of Vor($\mathcal{P}$) that corresponds to a point $P \in \mathcal{P}$. Note that $Cell(P)$ includes a vertex v_∞ "at infinity" if the Voronoi cell of P has an infinite edge. Theorem 7 shows that the minimum representative set of $\mathcal{P}$ is unique and can be computed in $O(|\mathcal{P}| \log |\mathcal{P}|)$.

Lemma 3. *Suppose $\mathcal{P}$ is a set of points inside the circle $C_{O,R}$ and $\mathcal{S}_{min}$ is the minimum representative set of $\mathcal{P}$. We have*

$$P \in \mathcal{S}_{min} \quad \Longleftrightarrow \quad \exists V \in Cell(P): \quad \overline{OV} > R.$$

Proof: Suppose that

$$\forall V \in Cell(P): \quad \overline{OV} \leqslant R.$$

In this case, the entire Voronoi cell of P will be inside the circle $C_{O,R}$. Therefore, for every point Q outside the circle we have

$$\exists P' \in \mathcal{P}: \quad \overline{P'Q} \leqslant \overline{PQ},$$

because Q is outside the Voronoi cell of P. Consequently, P is not required to be in a representative set of $\mathcal{P}$, thus it is not in the minimum representative set of $\mathcal{P}$. Now, suppose that

$$\exists V \in Cell(P): \quad \overline{OV} > R.$$

Since the point P is inside the circle, the line segment PV crosses the circle in a point U. Consider a point Q on the line segment UV such that $Q \neq U$ and $Q \neq V$. Clearly, Q is outside the circle $C_{O,R}$, i.e. $\overline{OQ} > R$. Since the Voronoi cell of P is a convex polygon, the point Q is inside the cell and is not located on any edge of it. Therefore, we have

$$\forall P' \in \mathcal{P} \backslash \{P\}: \quad \overline{PQ} < \overline{P'Q}.$$

Consequently, P must be in every representative set of $\mathcal{P}$ including its minimum representative set. ∎

Theorem 7. *Suppose $\mathcal{P}$ is a set of points inside a circle $C_{O,R}$. The minimum representative set of $\mathcal{P}$ is unique and can be computed in $O(|\mathcal{P}| \log |\mathcal{P}|)$.*

Proof: Based on Lemma 3, a point P is in the minimum representative set of $\mathcal{P}$ if and only if

$$\exists V \in Cell(P): \quad \overline{OV} > R.$$

There is a single Voronoi diagram associated with each set of generating points. Therefore, the minimum representative set is unique. From Lemma 3, it also follows that the minimum representative set of $\mathcal{P}$ is a subset of every representative set of $\mathcal{P}$. To compute the minimum representative set of $\mathcal{P}$, we first compute the Voronoi diagram Vor($\mathcal{P}$). For each Voronoi cell, we then check whether it has a vertex outside the circle $C_{O,R}$. The Voronoi diagram of $\mathcal{P}$ can be computed in $O(|\mathcal{P}| \log |\mathcal{P}|)$. Moreover, the average number of vertices of a cell is less than six [14]. In other words, the total number of vertex-checking operations is less than $6 \times |\mathcal{P}|$. Therefore, the complexity of computing the representative set is

$$O(|\mathcal{P}| \log |\mathcal{P}|) + O(|\mathcal{P}|) = O(|\mathcal{P}| \log |\mathcal{P}|).$$

∎

In the worst case, we have $|\mathcal{S}_{min}| = |\mathcal{P}|$, where $\mathcal{S}_{min}$ is the minimum representative set of $\mathcal{P}$. This situation occurs, for example, when all the points in $\mathcal{P}$ are located in a line segment or on a circle. However, it can be shown that

$$\lim_{|\mathcal{P}| \to \infty} \frac{E(|\mathcal{S}_{min}|)}{|\mathcal{P}|} = 0,$$

where $E(|\mathcal{S}_{min}|)$ is the expected cardinality of the minimum representative set of $\mathcal{P}$. In order to avoid complex analytical analysis, we used simulation to analyze $E(|\mathcal{S}_{min}|)$. The simulation results (given in Section VI) show that

$$E(|\mathcal{S}_{min}|) < \pi \sqrt{|\mathcal{P}|}, \quad 1 \leqslant |\mathcal{P}| \leqslant 500.$$

Suppose that the forwarding nodes can piggyback a list of their representative neighbors instead of the list of all their neighbors. In this case, Theorem 8 shows that a node can simply avoid broadcasting if it is not in the list of the nodes piggybacked in at least one received packet.

Theorem 8. *Based on the responsibility condition, a node N_A can avoid broadcasting if for at least one received packet, N_A is not in the list of representative neighbors of the broadcasting node.*

Proof: Suppose N_A receives the message from node N_B and that N_A is not in the list of N_B's representative nodes piggybacked in the packet. We prove that N_A's status is non-forwarding based on the responsibility condition. N_A's status is non-forwarding if all of its neighbors have received the message, so we assume that this is not the case. Suppose N_C is a neighbor of N_A that has not received the message. Therefore, N_C is not a neighbor of N_B, hence it is outside $C_{B,R}$, where $C_{B,R}$ is the transmission range of N_B. Since N_A is not in the representative neighbor set of N_B, there exists a node $N_D \neq N_A$ such that N_D is in the representative neighbors set of N_B and

$$\overline{DC} \leqslant \overline{AC}.$$

Note that $\overline{DC} \neq \overline{AC}$, because based on Lemma 3, the Voronoi cell of the point A is inside $C_{B,R}$. Therefore,

$$\overline{DC} < \overline{AC}.$$

It follows that N_A is not responsible for any of its neighbors and therefore has a non-forwarding status. ∎

V. Relaxing Some of the System-Model Assumptions

The assumptions made in Section II are often used in the existing broadcast algorithms in order to model the network. Some of these assumptions are not practical. For example, a node may obtain its position using the Global Positioning System (GPS). However, the position is not accurate due to several sources of error including positioning error and the limited number of bits used to represent the position. In this section, we discuss how to relax some of the system-model assumptions or replace them with more practical ones in order to improve the practicality of our proposed broadcast algorithm. We show that all of the extensions of the proposed algorithm can provide both full delivery and a constant approximation ratio to the optimal solution under the new assumptions.

A. Broadcasting Under Uncertain Position Information

As mentioned earlier, the assumption of having the precise position is not practical. In practice, there are several sources of error. For example, the position information obtained by GPS typically includes some errors, which vary between different GPS devices. Another source of error is roundoff error or representation error, which is associated with the fact that a finite number of bits is used to represent the position information. Typically, each node in the network is able to reduce the representation error by assigning a fairly large number of memory bits to represent a number. However, to reduce the bandwidth and power overhead, it is desirable to use as few bits as possible to represent the position information added in the HELLO messages or the broadcast packets. In this section, we show that a constant number of bits suffices to maintain the main properties of the proposed broadcast algorithm.

Let δ be an upper bound on the maximum position error. Therefore, the maximum error in computing the distance between two points is $2\delta+\epsilon$, where ϵ is the maximum roundoff error. Suppose $\mathcal{E}$ is an upper bound for $2\delta+\epsilon$ and is known by all the nodes in the network. Thus, we have

$$\overline{AB} - \mathcal{E} \leqslant \Gamma(\overline{AB}) \leqslant \overline{AB} + \mathcal{E}, \tag{4}$$

where $\Gamma(\overline{AB})$ is the approximated distance between the points A and B. Algorithm 4 shows how to compute the responsibility condition under uncertain position information. Let $List_{N_A}^{Neigh}$ be the list of N_A's neighbors, $List_{N_A}^{Brd}$ be the list of N_A's neighbor that have broadcast the message and $List_{N_A}^{Rec}$ be the list of nodes that have received the message, i.e., all of the nodes in $List_{N_A}^{Brd}$ together with their 1-hop neighbors. Algorithm 4 first computes $List_{N_A}^{NotRec}$, the list of its neighbors that have not received the message (Lines 1-8). Node N_A assumes that its neighbor N_B has received the message if and only if

$$\exists N_C \in List_{N_A}^{Brd}: \quad \Gamma(\overline{BC}) \leqslant R - \mathcal{E}. \tag{5}$$

Note that Algorithm 4 is not provided the precise position information, so it acts conservatively in order to make sure N_B has received the message. Algorithm 4 returns true if and only if N_A has a neighbor $N_B \in List_{N_A}^{NotRec}$ such that

$$\forall N_C \in List_{N_A}^{Rec}: \quad \Gamma(\overline{AB}) - \mathcal{E} \leqslant \Gamma(\overline{BC}) + \mathcal{E}. \tag{6}$$

The output of Algorithm 4 determines whether the responsibility condition is satisfied.

Algorithm 4: Computing the responsibility condition under uncertain position information

Input: $List_{ID}^{Neigh}$, $List_{ID}^{Brd}$, $List_{ID}^{Rec}$ and the max error $\mathcal{E}$.
output: true or false.

1. empty($List_{ID}^{NotRec}$); //make the list empty
2. for ($i = 1$; $i \leqslant$ length($List_{ID}^{Neigh}$); ++i)
3. $\quad chk \leftarrow true$;
4. $\quad$ for ($j = 1$; $j \leqslant$ length($List_{ID}^{Brd}$); ++j)
5. $\quad\quad$ if (dist($List_{ID}^{Neigh}[i], List_{ID}^{Brd}[j]$) $\leqslant R - \mathcal{E}$)
6. $\quad\quad\quad chk \leftarrow false$; break;
7. $\quad$ if (chk)
8. $\quad\quad$ Add($List_{ID}^{Neigh}[i], List_{ID}^{NotRec}$)

9. for ($i = 1$; $i \leqslant$ length($List_{ID}^{NotRec}$); ++i)
10. $\quad d \leftarrow dist(List_{ID}^{NotRec}[i], ID)$;
11. $\quad chk \leftarrow true$;
12. $\quad$ for ($j = 1$; $j \leqslant$ length($List_{ID}^{Rec}$); ++j)
13. $\quad\quad$ if (dist($List_{ID}^{NotRec}[i], List_{ID}^{Rec}[j]$) $< d - 2\mathcal{E}$)
14. $\quad\quad\quad chk \leftarrow false$; break;
15. $\quad$ if (chk)
16. $\quad\quad$ return true;
17. return false;

Theorem 9. *Algorithm 3 guarantees full delivery under uncertain position information if it uses Algorithm 4 to compute the responsibility condition.*

Proof: The proof is similar to that of Theorem 3. The broadcasting will eventually terminate, since each node transmits the message only once. By contradiction, suppose there is a node that has not received the message after termination of broadcasting. Let us consider the set

$$\begin{aligned}\Lambda = \{(N_X, N_Y) | N_X &\text{ and } N_Y \text{ are neighbors,}\\ &N_X \text{ has received the message and}\\ &N_Y \text{ has not received the message}\}\end{aligned}$$

Suppose N_S is the source node and N_T is a node that has not received the message. The network is connected, thus there is a path between N_S and N_T. Clearly, we can find two neighbor nodes N_B and N_A along the path from N_T to N_S such that N_B has not received the message, while N_A has. Consequently, $(N_A, N_B) \in \Lambda$, thus $\Lambda \neq \emptyset$. As a result,

$$\exists (N_{A'}, N_{B'}) \in \Lambda \ \text{ s.t. } \ \forall (N_X, N_Y) \in \Lambda: \quad \overline{A'B'} \leqslant \overline{XY}. \tag{7}$$

Node $N_{B'}$ has not received the message. Therefore, for every broadcasting node N_F we have $\overline{FB'} > R$, thus

$$\Gamma(\overline{FB'}) > R - \mathcal{E}.$$

Consequently, based on (5), $N_{A'}$ does not consider $N_{B'}$ as a node that has received the message. Clearly, $N_{A'}$ has not broadcast since $N_{B'}$ has not received the message. Considering the responsibility condition, it follows that there is a node $N_{C'}$ such that $N_{C'}$ has received the message and

$$\Gamma(\overline{B'C'}) + \mathcal{E} < \Gamma(\overline{A'B'}) - \mathcal{E}.$$

Using (4) we get

$$\overline{B'C'} < \overline{A'B'}.$$

This result contradicts (7), since $(N_{C'}, N_{B'}) \in \Lambda$. ∎

Theorem 10. *Using Algorithm 4 to compute the responsibility condition, the proposed broadcast algorithm (Algorithm 3) guarantees a constant approximation ratio to the optimum solution under uncertain position information if $\mathcal{E} = \frac{R}{8} - \lambda R$, where $0 < \lambda \leqslant \frac{1}{8}$ and $\frac{1}{\lambda}$ is bounded by a constant number.*

Proof: We show that the number of broadcasting nodes inside a disk $D_{O,\frac{R}{4}}$ with a radius $\frac{R}{4}$ is bounded by a constant. Using the same approach used in the proof of Theorem 4, it then follows that the total number of broadcasting nodes is bounded by a constant factor of that of the optimum solution. Let $D_{O,\frac{R}{4}}$, $D_{O,\frac{3R}{4}}$ and $D_{O,\frac{5R}{4}}$ be three disks centered at O with radii $\frac{R}{4}$, $\frac{3R}{4}$ and $\frac{5R}{4}$, respectively. Suppose k is the minimum number, such that for every set of k points $P_i \in D_{O,\frac{5R}{4}} - D_{O,\frac{3R}{4}}$, $1 \leqslant i \leqslant k$, we have

$$\exists P_i, P_j : \quad i \neq j \quad \text{and} \quad \overline{P_i P_j} \leqslant \lambda R.$$

Note that $\frac{1}{\lambda}$ is bounded by a constant, so the area $D_{O,\frac{5R}{4}} - D_{O,\frac{3R}{4}}$ can be covered with a constant number of disks with radius $\frac{\lambda}{2}R$. If the number of points inside $D_{O,\frac{5R}{4}} - D_{O,\frac{3R}{4}}$ is greater than the number of covering disks, at least one covering disk will contain more than one point. For two points P_1 and P_2 inside a disk with radius $\frac{\lambda}{2}R$, we have $\overline{P_1 P_2} \leqslant \lambda R$. Therefore, k is bounded by a constant (the number of covering disks plus one). We prove that the number of broadcasting nodes inside $D_{O,\frac{R}{4}}$ is less than or equal to k. By contradiction, suppose that there are more than k broadcasting nodes inside $D_{O,\frac{R}{4}}$. Let $N_{A_i} \in D_{O,\frac{R}{4}}$, $0 \leqslant i \leqslant k$ be the first $k+1$ broadcasting nodes ordered chronologically based on their broadcast time. Based on the responsibility condition, for each node N_{A_i} there is a corresponding neighbor N_{B_i} such that $N_{B_i} \notin List^{Rec}_{N_{A_i}}$ and

$$\forall N_C \in List^{Rec}_{N_{A_i}} : \quad \Gamma(\overline{A_i B_i}) - \mathcal{E} \leqslant \Gamma(\overline{B_i C}) + \mathcal{E},$$

where $List^{Rec}_{N_{A_i}}$ is the list of nodes that have received the message based on N_{A_i}'s collected information. Suppose $1 \leqslant i < j \leqslant k$. Nodes N_{A_i} and N_{A_j} are neighbors, because $\overline{A_i A_j} \leqslant \frac{R}{2}$. Recall that N_{A_i} piggybacks the list of all its neighbors within the broadcast packet. Therefore, $N_{B_i} \in List^{Rec}_{N_{A_j}}$, thus $N_{B_i} \neq N_{B_j}$. Note that every node $N_C \in D_{O,\frac{3R}{4}}$ will receive the message after N_{A_0}'s broadcast, because $\overline{A_0 C} \leqslant R$. Therefore, N_C is in the list of nodes piggybacked by N_{A_0} in the broadcast packet. Thus

$$\forall 1 \leqslant i \leqslant k : \quad N_C \in List^{Rec}_{N_{A_i}}.$$

It follows that

$$\forall 1 \leqslant i \leqslant k : \quad N_{B_i} \notin D_{O,\frac{3R}{4}}.$$

On the other hand, N_{B_i} is a neighbor of N_{A_i}, thus $N_{B_i} \in D_{O,\frac{5R}{4}}$. Consequently,

$$\forall 1 \leqslant i \leqslant k : \quad N_{B_i} \in D_{O,\frac{5R}{4}} - D_{O,\frac{3R}{4}}.$$

Thus,

$$\forall 1 \leqslant i \leqslant k : \quad \overline{A_i B_i} > \frac{R}{2}. \tag{8}$$

There are k different nodes $N_{B_1} \dots N_{B_k}$ inside $D_{O,\frac{5R}{4}} - D_{O,\frac{3R}{4}}$. Therefore,

$$\exists N_{B_i}, N_{B_j} : \quad i < j \quad \text{and} \quad \overline{B_i B_j} \leqslant \lambda R. \tag{9}$$

From (4) we have

$$\Gamma(\overline{B_i B_j}) + \mathcal{E} \leqslant \overline{B_i B_j} + 2\mathcal{E}.$$

Thus, using (9) we get

$$\Gamma(\overline{B_i B_j}) + \mathcal{E} \leqslant \frac{R}{8} + \mathcal{E}. \tag{10}$$

On the other hand, we have $\overline{A_j B_j} > \frac{R}{2}$. Hence,

$$\Gamma(\overline{A_j B_j}) - \mathcal{E} > \frac{R}{2} - 2\mathcal{E}. \tag{11}$$

Since $\mathcal{E} \leqslant \frac{R}{8}$, we have

$$\frac{R}{8} + \mathcal{E} \leqslant \frac{R}{2} - 2\mathcal{E}.$$

Thus, based on (10) and (11) we get

$$\Gamma(\overline{B_i B_j}) + \mathcal{E} < \Gamma(\overline{A_j B_j}) - \mathcal{E}$$

This result is a contradiction, because based on (6), N_{A_j} is not responsible for N_{B_j}. ■

Example 2. *Suppose that $\mathcal{E} \leqslant \frac{R}{9}$. Therefore, we have*

$$\mathcal{E} = \frac{R}{8} - \lambda R,$$

where $\lambda \geqslant \frac{1}{72}$. Note that $\frac{1}{\lambda}$ $(\leqslant 72)$ is bounded by a constant. Thus, based on Theorem 10, the proposed broadcast algorithm can guarantee a constant approximation ratio to the optimal solution if the maximum error $\mathcal{E}$ is less than or equal to $\frac{R}{9}$.

As shown in Theorem 10, Algorithm 3 can guarantee a constant approximation ratio to MCDS if the maximum error is bounded by a factor of the transmission range. As a result, employing the same approach used in the proof of Theorem 5, we can show that the message complexity of Algorithm 3 under uncertain position information is still $O(N)$, where N is the total number of nodes in the network and each message contains a node ID and its position. Note that node IDs can be simply removed from the message without affecting the functionality of the proposed broadcast algorithm. It is because the responsibility condition is based solely on the approximate/precise position information of the nodes.

Suppose that all of the nodes in the network are in an $L \times L$ square area. Clearly, each node requires an infinite number of bits in general to precisely represent its position. However, allowing errors in position information, node position can be represented by a limited number of bits, provided that the size of square area is finite. For example, assume that we allow the maximum position error of αR. As shown in Figure 5, the $L \times L$ square area can be partitioned into $\lceil \frac{L}{\sqrt{2}\alpha R} \rceil \times \lceil \frac{L}{\sqrt{2}\alpha R} \rceil$ grid cells of size $\sqrt{2}\alpha R \times \sqrt{2}\alpha R$. Let the tuple (I, J) denote the coordinate of the cell in the grid. For instance, (1,1) indicates the bottom left cell of the gird. Suppose each node uses the coordinate of the cell in which it is located to indicate its position. In this case, the position of the node can be estimated by the position of the center of the cell. Therefore, the number of bits required to represent the approximate position is at most

$$\lceil \lg_2 \left(\lceil \frac{L}{\sqrt{2}\alpha R} \rceil \right) \rceil + \lceil \lg_2 \left(\lceil \frac{L}{\sqrt{2}\alpha R} \rceil \right) \rceil \approx 2 \lg_2 \left(\frac{L}{\alpha R} \right)$$

and the maximum position error is one-half of the cell diagonal, i.e.,

$$\frac{\sqrt{2}(\sqrt{2}\alpha R)}{2} = \alpha R.$$

Note that the number of bits required to represent the approximate position increases as $\frac{L}{R}$ increases.

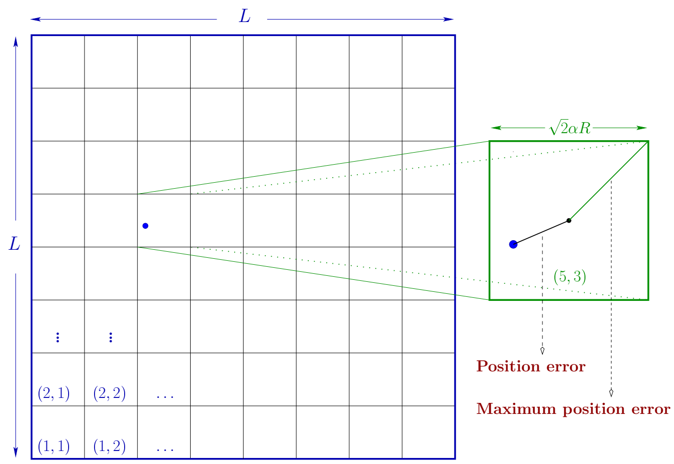

Fig. 5. Partitioning the network into square cells.

Lemma 4. *Suppose that the node N_A is required to send its position with the maximum error αR to its neighbor N_B. In this case, N_A requires only a constant number of bits to represent its position if $\frac{1}{\alpha}$ is bounded by a constant.*

Proof: Suppose (i, j) is the coordinate of the cell in which N_A is located and that the side of each grid cell is $\sqrt{2}\alpha R$. Node N_A can transmit the tuple

$$(i \mod (\lfloor \frac{\sqrt{2}}{\alpha} \rfloor + 2), j \mod (\lfloor \frac{\sqrt{2}}{\alpha} \rfloor + 2))$$

to node N_B. Suppose (i, j) and (i', j') are the coordinates of two different cells such that

$$i \equiv i' \mod (\lfloor \frac{\sqrt{2}}{\alpha} \rfloor + 2),$$

and

$$j \equiv j' \mod (\lfloor \frac{\sqrt{2}}{\alpha} \rfloor + 2).$$

Let P_1 and P_2 be two points inside the cells (i, j) and (i', j'), respectively. It is easy to show that

$$\overline{P_1P_2} \geqslant (\lfloor \frac{\sqrt{2}}{\alpha} \rfloor + 1) \times (\sqrt{2}\alpha R) \quad \Longrightarrow \quad \overline{P_1P_2} > 2R.$$

It follows that a circle with radius R cannot intersect both cells. Therefore, N_B can uniquely identify the cell using the tuple

$$(i \mod (\lfloor \frac{\sqrt{2}}{\alpha} \rfloor + 2), j \mod (\lfloor \frac{\sqrt{2}}{\alpha} \rfloor + 2)).$$

Consequently, N_A is required to transmit only

$$2\lceil \lg_2 \left(\lfloor \frac{\sqrt{2}}{\alpha} \rfloor + 2 \right) \rceil \approx \lg_2(\frac{1}{\alpha}) + 1$$

bits, which is constant if $\frac{1}{\alpha}$ is bounded by a constant. Node N_B can approximate the position of N_A by taking the position of the center of the cell. ■

As mentioned earlier, the message complexity of Algorithm 3 is $O(N)$ under uncertain position information if the maximum position error is bounded. Moreover, using Lemma 4 , it follows that the size of each message can be reduced to a constant number of bits by removing the node ID and using a constant number of bits to represent the approximate position. In [3], the authors proved that every distributed algorithm for constructing a non-trivial CDS has the lower message complexity bound of $\Omega(N \log N)$, where N is the number of nodes and the message size is a constant multiple of the number of bits representing the node IDs. Although finding a non-trivial CDS and designing a non-trivial broadcast algorithm are closely related problems, our results show that the same message complexity lower bound does not apply for the non-trivial broadcast algorithms.

B. Relaxing the Homogeneous Network Assumption

In practice, devices may have different transmission ranges. In this case, the network topology can be represented by a directed graph, for which there is a link from N_A to N_B if N_B is in transmission range of N_A. Suppose G is an undirected graph obtained by removing uni-directional links of the topology graph. We assume that G is connected and define two nodes as neighbors if there is a link between them in G (i.e., they are in transmission range of each other). Note that many wireless medium access control protocols, such as IEEE 802.11, require bi-directional links.

Let us use R_{N_A} to denote a lower bound of N_A's transmission range. Suppose each node includes the lower bound of its transmission range in the HELLO message. Moreover, assume that the forwarding nodes include the lower bounds of their neighbors' transmission ranges in the broadcast packets. The responsibility condition of node N_A is satisfied if N_A has a neighbor N_B such that N_B has not received the message, and for every node N_C that has received the message

$$\overline{AB} \leqslant \overline{CB} \quad \text{or} \quad \overline{CB} > R_{N_C}.$$

Using a similar approach as that used in the proof of Theorem 3, we can show that Algorithm 3 guarantees full delivery in a heterogeneous network if it employs the new responsibility condition.

Suppose R_{min} is a lower bound for the transmission ranges of all the nodes in the network and R_{max} is an upper bound. The following theorem shows that the proposed broadcast algorithm can still guarantee a good bound on the number of forwarding nodes in a heterogeneous network if $\frac{R_{max}}{R_{min}}$ is bounded by a constant.

Theorem 11. *The proposed broadcast algorithm can guarantee a constant approximation ratio to the optimal solution in a heterogeneous network if $\frac{R_{max}}{R_{min}}$ is bounded by a constant.*

Proof: Let $D_{O,\frac{R_{min}}{4}}$, $D_{O,\frac{3R_{min}}{4}}$ and $D_{O,\frac{R_{min}}{4}+R_{max}}$ be three disks centered at O with radii $\frac{R_{min}}{4}$, $\frac{3R_{min}}{4}$ and $\frac{R_{min}}{4}+R_{max}$, respectively. Suppose k is the minimum number such that for every set of k points $P_i \in D_{O,\frac{R_{min}}{4}+R_{max}} - D_{O,\frac{3R_{min}}{4}}$, $1 \leqslant i \leqslant k$, we have

$$\exists P_i, P_j : \quad i \neq j \quad \text{and} \quad \overline{P_iP_j} \leqslant \frac{R_{min}}{2}.$$

Note that $\frac{R_{max}}{R_{min}}$ is bounded by a constant, so the area $D_{O,\frac{R_{min}}{4}+R_{max}} - D_{O,\frac{3R_{min}}{4}}$ can be covered with a constant number of disks with radius $\frac{R_{min}}{4}$. If the number of points inside $D_{O,\frac{R_{min}}{4}+R_{max}} - D_{O,\frac{3R_{min}}{4}}$ is greater than the number of covering disks, at least one covering disk will contain more than one point. For two points P_1 and P_2 inside a disk with radius $\frac{R_{min}}{4}$, we have $\overline{P_1P_2} \leqslant \frac{R_{min}}{2}$. Therefore, k is bounded by a constant (the number of covering disks plus one). Using a technique similar to that used in the proof of Lemma 1, we can show that the number of broadcasting nodes inside $D_{O,\frac{R_{min}}{4}}$ is less than or equal to k. Since $\frac{R_{max}}{R_{min}}$ is bounded by a constant, a disk with radius R_{max} can be covered with a constant number of disks with radii R_{min}. It follows that the total number of broadcasting nodes is bounded by a constant factor of that of the optimum solution (refer to the proof of Theorem 4).

■

C. Broadcasting in Three-Dimensions

We assumed in Section II that the nodes are placed in a 2-D plane. In general, the nodes can be located in 3-D space, as the network area may not be perfectly flat. In this case we assume that the nodes are provided with their position in 3-D. Interestingly, all the properties of the proposed broadcast algorithm are preserved in 3-D. Assuming $\overline{AB}$ as the Euclidean distance of two points in 3-D, we can use the proof of Theorem 3 to show that Algorithm 3 guarantees full delivery in 3-D space. Note that the transmission range of a node in 3-D is a sphere. Replacing circles by spheres in the proofs of Lemma 1 and Theorem 4, we can similarly argue that Algorithm 3 guarantees a constant approximation ratio to the optimal solution in 3-D. Finally, using a technique similar to that used for 2-D, we can show that, in 3-D, the proposed algorithm not only preserves its main properties under uncertain position information (provided that the maximum error is bounded) but also can benefit from the proposed bandwidth-overhead reduction technique introduced in Section IV-C.

VI. Simulation

A. Reducing Bandwidth Overhead

As shown in Section IV-C, a forwarding node can piggyback the list of a subset of its neighbors (its representative neighbor set) instead of all of them without violating the status (forwarding/non-forwarding) of any other node in the network. To avoid the complexity of mathematical analysis, we used simulation to find the average cardinality of the minimum representative set. For a given number of neighbors $1 \leqslant n \leqslant 500$, we randomly put n points inside a unit circle and computed the minimum representative set $\mathcal{S}_{min}$. To get the average cardinality $E(|\mathcal{S}_{min}|)$, we ran the simulation 10^4 times for each given n. Figure 6 shows the ratio $\frac{E(|\mathcal{S}_{min}|)}{n} \times 100$. As shown in Figure 6, a greater bandwidth saving can be achieved on average when the number of neighbors increases. For instance, we can save on average more than 50% of the bandwidth overhead when $n \geqslant 33$. As illustrated in Figure 6, we can approximate the expected minimum number of representative nodes by $\pi\sqrt{n}$ for $1 \leqslant n \leqslant 500$.

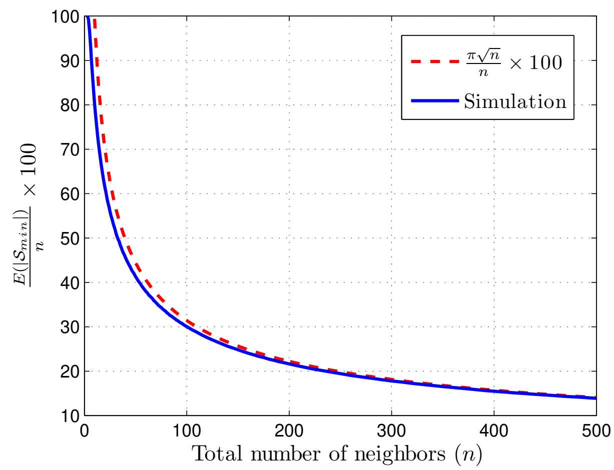

Fig. 6. The ratio $\frac{E(|\mathcal{S}_{min}|)}{n} \times 100$ against the total number of neighbors.

B. Performance of the Proposed Broadcast Algorithm

To verify the theoretical results, we used simulation to compute the average number of forwarding nodes. We implemented a C++ simulator that uses a simplified version of IEEE 802.11 to simulate its broadcasting mode. In the broadcasting mode of IEEE 802.11, scheduling is done according to the p-persistent CSMA/CA protocol. In the p-persistent CSMA/CA protocol, when a node has a message to transmit, it initiates a defer timer by a random number and starts listening to the channel. If the channel is busy, it continues to listen until the channel becomes idle. When the channel is idle, it starts decrementing the defer timer at the end of each time unit. The message is broadcast when the timer expires. Table I summarizes some of the parameters used in the simulator. The total size of the data packet was 256 bytes and the bandwidth of the wireless channel was set to 2 Mb/s. The signal propagation speed was assumed to be $3 \times 10^8\, m/s$ and the maximum defer time was set to 0.01 sec. In each simulation run, we randomly placed N nodes in a $1000 \times 1000\, m^2$ square area. Only one broadcast was initiated in each simulation run by a randomly selected node.

Figures 8 and 9 show the ratio of the number of broadcasting nodes over the total number of nodes. To get the results shown in Figure 8, we varied the transmission range from 100 to 300 and fixed the total number of nodes to 1000. Figure 9 shows the result of the experiment for which the transmission range was fixed to 250 and the total number of nodes vary from 200 to 1000. We compared the performance of our proposed algorithm with that of the broadcast algorithm proposed in [5]. In [5], Liu et al. showed that the number of broadcasting nodes using their proposed flooding algorithm is significantly lower than that of previous notable broadcast algorithms [15], [16]. We also considered the ratio-8 approximation of MCDS [3] as a benchmark. As shown in Figures 8 and 9, our proposed broadcast algorithm can significantly reduce the number of forwarding nodes. Moreover, it slightly outperforms the ratio-8 approximation. Figure 7 illustrates an instance of using the proposed algorithm for the case where $R = 300$m, $N = 400$ and nodes are placed in a square area of $1000 \times 1000\text{m}^2$. As shown in this figure, only 10 nodes (represented by stars) among 400 nodes broadcast the message.

Another important metric in efficient broadcast algorithms is the end-to-end broadcast latency which is the time it takes for the message to reach all the nodes in the network. Figure 10 shows the average latency of our proposed algorithm for a single message broadcast and compares it with that of Liu's broadcast algorithm. As shown in this figure, the average latency of our broadcast algorithm is 56% to 36% of that of Liu's algorithm when the total number of nodes varies from 200 to 1000. The low average latency of the proposed broadcast algorithm is due to the fact that the algorithm does not increase the back-off delay of any node in the network and the total number of forwarding nodes is low on average.

Parameter	Value
Packet Size	256 bytes
Bandwidth	2 Mb/sec
Size of Square Area	$1000 \times 1000\text{m}^2$
Transmission Range	100–300m
Interference Range	$2 \times$ Transmission Range
Signal Propagation Speed	10^8 meter/sec
Number of Nodes	200–1000
Maximum Defer Time	0.01 sec

TABLE I
SIMULATION PARAMETERS

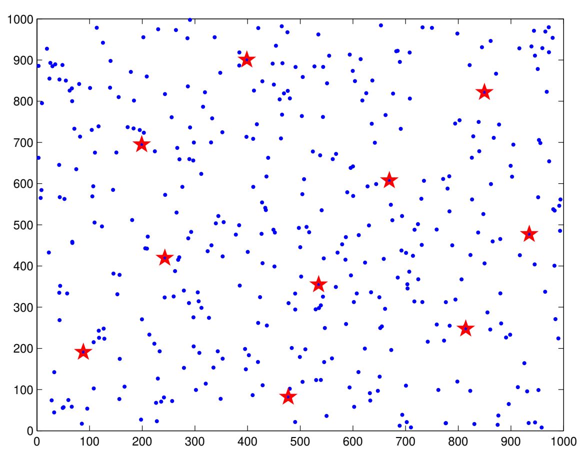

Fig. 7. Broadcasting nodes in a $10^3 \times 10^3 m^2$ square area with 400 nodes.

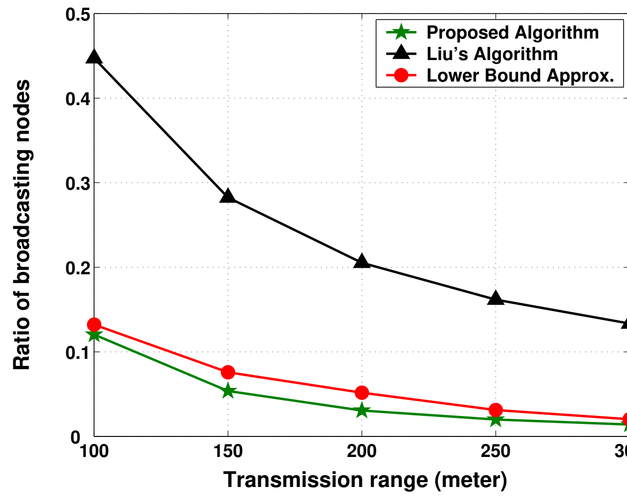

Fig. 8. Ratio of broadcasting nodes vs. transmission range.

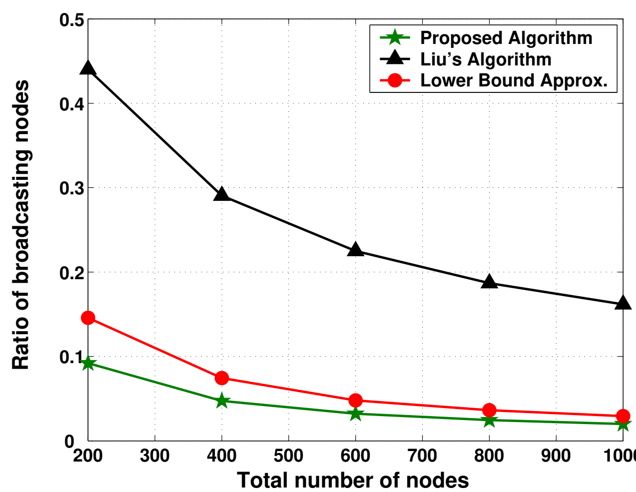

Fig. 9. Ratio of broadcasting nodes vs. total number of nodes.

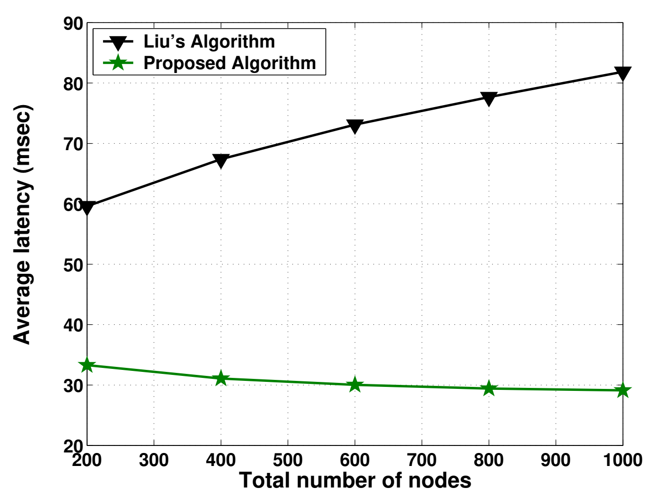

Fig. 10. Average latency vs. total number nodes.

VII. Conclusion

We considered two general structures typically used in broadcast algorithms based on 1-hop neighbor information. We showed that a broadcast algorithm cannot guarantee both full delivery and a good bound on the number of transmissions if it uses any of these structures. It is commonly believed that a localized broadcast algorithm is not able to guarantee a good bound on the number of transmissions. We presented a localized broadcast algorithm based on 1-hop information and proved that it guarantees both full delivery and a constant approximation ratio to the optimal solution. The proposed broadcast algorithm has low message and computational complexities. In fact, we proved that the message complexity of the proposed algorithm is less than the lower message complexity bound of finding a non-trivial CDS (a closely related problem) and that its computational complexity is nearly optimal. Moreover, we proposed a technique to further reduce the bandwidth overhead. We relaxed several system-model assumptions, or replaced them with practical ones, to improve the practicality of the broadcast algorithm. Finally, we verified the analytical results using simulation.

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] 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.

[4] S. Funke, A. Kesselman, U. Meyer, and M. Segal, "A simple improved distributed algorithm for minimum CDS in unit disk graphs," *ACM Sensor Networks (TOSN)*, vol. 2, no. 3, pp. 444–453, 2006.

[5] 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.

[6] Y. Tseng, S. Ni, and E. Shih, "Adaptive approaches to relieving broadcast storms in a wireless multihop mobile ad hoc networks," *In Proc. International Conference on Distributed Computing Systems (ICDCS)*, pp. 481–488, 2001.

[7] Y. Sasson, D. Cavin, and A. Schiper, "Probabilistic broadcast for flooding in wireless mobile ad hoc networks," *In Proc. IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1124–1130, 2003.

[8] 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.

[9] G. Calinescu, I. Mandoiu, P. Wan, and A. Zelikovsky, "Selecting forwarding neighbors in wireless ad hoc networks," *ACM Mobile Networks and Applications*, vol. 9, no. 2, pp. 101–111, 2004.

[10] J. Wu and F. Dai, "Broadcasting in ad hoc networks based on self-pruning," *In Proc. IEEE INFOCOM*, pp. 2240–2250, 2003.

[11] 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.

[12] 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.

[13] 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.

[14] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, *Computational Geometry: Algorithms and Applications, 2^{nd} ed.* New York: Springer-Verlag, 2000.

[15] Y. Cai, K. Hua, and A. Phillips, "Leveraging 1-hop neighborhood knowledge for efficient flooding in wireless ad hoc networks," *In Proc. IEEE International Performance, Computing, and Communications Conference (IPCCC)*, pp. 347–354, 2005.

[16] J. Wu and H. Li, "On calculating connected dominating set for efficient routing in ad hoc wireless networks," *In Proc. International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DiaLM)*, pp. 7–14, 1999.