Efficient Broadcasting in Mobile Ad Hoc Networks

Majid Khabbazian, *Student Member*, *IEEE*, and Vijay K. Bhargava, *Fellow*, *IEEE*

Abstract—This paper presents two efficient broadcasting algorithms based on 1-hop neighbor information. In the first part of the paper, we consider sender-based broadcasting algorithms, specifically the algorithm proposed by Liu et al. In their paper, Liu et al. proposed a sender-based broadcasting algorithm that can achieve local optimality by selecting the minimum number of forwarding nodes in the lowest computational time complexity $O(n \log n)$, where n is the number of neighbors. We show that this optimality only holds for a subclass of sender-based algorithms. We propose an efficient sender-based broadcasting algorithm based on 1-hop neighbor information that reduces the time complexity of computing forwarding nodes to $O(n)$. In Liu et al.'s algorithm, n nodes are selected to forward the message in the worst case, whereas in our proposed algorithm, the number of forwarding nodes in the worst case is 11. In the second part of the paper, we propose a simple and highly efficient receiver-based broadcasting algorithm. When nodes are uniformly distributed, we prove that the probability of two neighbor nodes broadcasting the same message exponentially decreases when the distance between them decreases or when the node density increases. Using simulation, we confirm these results and show that the number of broadcasts in our proposed receiver-based broadcasting algorithm can be even less than one of the best known approximations for the minimum number of required broadcasts.

Index Terms—Wireless ad hoc networks, flooding, broadcasting, localized algorithms.

1 INTRODUCTION

Broadcasting is a fundamental communication operation in which one node sends a message to all other nodes in the network. Broadcasting is widely used as a basic mechanism in many ad hoc network protocols. For example, ad hoc on-demand routing protocols such as AODV [1] and DSR [2] typically use broadcasting in their route discovery phase. Broadcasting is also used for topology updates, for network maintenance, or simply for sending a control or warning message. The simplest broadcasting algorithm is flooding, in which every node broadcasts the message when it receives it for the first time. Using flooding, each node receives the message from all its neighbors in a collision-free network. Therefore, the broadcast redundancy significantly increases as the average number of neighbors increases. High broadcast redundancy can result in high power and bandwidth consumption in the network. Moreover, it increases packet collisions, which can lead to additional transmissions. This can cause severe network congestion or significant performance degradation, a phenomenon called the broadcast storm problem [3]. Consequently, it is crucial to design efficient broadcasting algorithms to reduce the number of required transmissions in the network.

A set of nodes is called a Dominating Set (DS) if any node in the network either belongs to the set or is a 1-hop neighbor of a node in the set. The set of broadcasting nodes forms a Connected DS (CDS). Therefore, the minimum number of required broadcasts is not less than the size of the minimum CDS. Unfortunately, finding the minimum CDS is NP-hard, even for the unit disk graphs [4], [5]. However, there are some distributed algorithms that can find a CDS whose size is smaller than a constant factor of the size of the minimum CDS [6], [7]. These algorithms can be employed to find a small-sized CDS that can be used as a virtual backbone for broadcasting in ad hoc networks. However, this approach is not efficient in networks with frequent topology changes, as maintaining a CDS is often costly [8].

The main objective of efficient broadcasting algorithms is to reduce the number of broadcasts while keeping the bandwidth and computational overhead as low as possible. One approach to classify broadcasting algorithms is based on the neighbor information they use. Some broadcasting algorithms such as flooding and probabilistic broadcasting algorithms [9], [10] do not rely on neighborhood knowledge. These algorithms cannot typically guarantee full delivery and/or effectively reduce the number of broadcasts. 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. In the second category, broadcasting algorithms require having 2-hop or more neighbor information. The broadcasting algorithms in this category can reduce the number of broadcasts in the network and guarantee full delivery [11], [12], [13]. However, they may induce high overhead in highly dynamic networks as they need to maintain 2-hop network connectivity.

In this paper, we propose two broadcasting algorithms based on 1-hop neighbor information. The first proposed

- M. Khabbazian is with the Department of Electrical and Computer Engineering, University of British Columbia, 2523 Pearkes Lane, Vancouver, BC V6T 2C3, Canada. E-mail: majidk@ece.ubc.ca.
- V.K. Bhargava is with the Department of Electrical and Computer Engineering, University of British Columbia, 2356 Main Mall, Vancouver, BC V6T 1Z4, Canada. E-mail: vijayb@ece.ubc.ca.

Manuscript received 3 Jan. 2007; revised 25 Aug. 2007; accepted 9 Apr. 2008; published online 19 June 2008.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-0002-0107.
Digital Object Identifier no. 10.1109/TMC.2008.94.

algorithm is a sender-based algorithm. In sender-based algorithms, the broadcasting nodes select a subset of their neighbors to forward the message. We compare our proposed broadcasting algorithm to one of the best sender-based broadcasting algorithms that use 1-hop information [8]. In [8], Liu et al. propose a broadcasting algorithm that reduces the number of broadcasts and achieves local optimality by selecting the minimum number of forwarding nodes with minimum time complexity $O(n \log n)$, where n is the number of neighbors. We show that this optimality only holds for a subclass of sender-based broadcasting algorithms employing 1-hop information and prove that our proposed sender-based algorithm can achieve full delivery with time complexity $O(n)$. Moreover, Liu et al.'s algorithm selects n forwarding nodes in the worst case, while our proposed algorithm selects 11 nodes in the worst case. Based on our simulation results, our sender-based algorithm results in fewer broadcasts than does Liu et al.'s algorithm. All these interesting properties are achieved at the cost of a slight increase in end-to-end delay. Thus, our first proposed algorithm is preferred to Liu et al.'s algorithm when the value of n is typically large, and it is important to bound the packet size.

We also propose a receiver-based broadcasting algorithm in this paper. In receiver-based algorithms, the receiver decides whether or not to broadcast the message. The proposed receiver-based algorithm is a novel broadcasting algorithm that can significantly reduce the number of broadcasts in the network. We show that using our proposed receiver-based algorithm, two close neighbors are not likely to broadcast the same message. In other words, we prove that the probability of broadcast for a node N_A exponentially decreases when the distance between N_A and its broadcasting neighbor decreases or when the density of nodes increases. Based on our experimental results, the number of broadcasts using our receiver-based algorithm is less than one of the best known approximations for the minimum number of required broadcasts.

The rest of this paper is organized as follows: In Section 2, we describe the system model and network assumptions. In Section 3, we discuss our proposed sender-based broadcasting algorithm and its characteristics. We propose a simple and highly efficient receiver-based broadcasting algorithm in Section 4 and prove an interesting property of the algorithm. We also relax some system model assumption in this section. In Section 5, we verify the theoretical results using simulation and compare the number of forwarding nodes of our proposed broadcasting algorithms with that of one of the best existing broadcasting algorithms and an approximated lower bound of the optimal solution. Finally, we provide conclusions in Section 6.

2 System Model

Our system model is very similar to that used by Liu et al. [8]. We assume that all nodes are located in a 2D plane and have a transmission range of R. Therefore, the topology of the network can be represented by a unit disk graph. We assume that the network is connected. Two nodes are considered neighbors if they are in the transmission range of each other. We suppose that each node knows its location via a localization technique such as Global Positioning System (GPS) or the lightweight techniques summarized in [14]. Each node periodically broadcasts a very short *Hello* message, which includes its *ID* and position. Thus, each node gets the position of its neighbors as well. In the medium access control (MAC) layer, we assume that scheduling is done according to the p-persistent CSMA/CA protocol, which is based on IEEE 802.11 in the broadcast mode. 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.

3 An Efficient Sender-Based Broadcasting Algorithm

3.1 Algorithm Structure

Our first proposed broadcasting algorithm is a sender-based algorithm, i.e., each sender selects a subset of nodes to forward the message. Each message can be identified by its source *ID* and a sequence number incremented for each message at the source node. Algorithm 1 is a general sender-based broadcasting algorithm and indicates the structure of our proposed sender-based broadcasting algorithm. Upon expiration of the timer, the algorithm requests the MAC layer to schedule a broadcast. The message scheduled in the MAC layer is buffered and then broadcast with a probability p. This adds another delay (i.e., the MAC-layer delay) in broadcasting the message. The MAC-layer delay in IEEE 802.11 is a function of several factors including the network traffic. Note that there is a chance that a node changes its decision (regarding the selected nodes or regarding whether to broadcast) during the MAC-layer delay due to receiving other copies of the message. This chance is not negligible when the delay in the MAC layer is comparable to the average value of the timer set in the broadcasting algorithm. As stated in [15], one solution to this problem is a cross-layer design in which the network layer is given the ability to modify or remove packets that are present in the MAC-layer queue. This solution allows the broadcasting algorithms to perform close to their ideal performance even for very small average timer values [15]. In the entire paper, we assume that either the MAC-layer delay is negligible compared to the average delay set by the algorithm or the network layer (hence, the algorithm) is able to modify or remove packets buffered in the MAC-layer queue (in this case, the algorithm does not require to set a defer timer).

The sender-based broadcasting algorithms can be divided into two subclasses. In the first subclass, each node decides whether or not to broadcast solely based on the first received message and drops the rest of the same messages that it receives later. Liu et al.'s algorithm falls in this subclass and achieves local optimality by selecting the minimum number of forwarding nodes in the lowest computational time complexity.

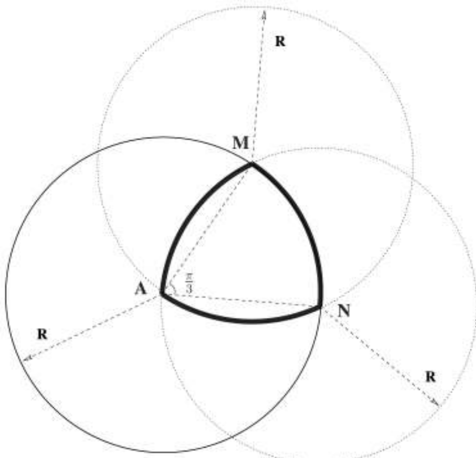


Fig. 1. A bulged slice around A.

In the second subclass of sender-based broadcasting algorithms, each node can decide whether or not to broadcast after each message reception. However, if a node broadcasts a message, it will drop the rest of the same messages that it receives in the future. Therefore, each message is broadcast once at most by a node using the broadcasting algorithms in both subclasses. Our first proposed broadcasting algorithm falls in this subclass of sender-based broadcasting algorithms. We show that the proposed algorithm can reduce both the computational complexity of selecting the forwarding nodes and the maximum number of selected nodes in the worst case.

Algorithm 1 shows the basic structure of our proposed sender-based broadcasting algorithm. 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 scheduled the same message before. Clearly, each message is broadcast once at most by a node, which is similar to Liu et al.'s algorithm. However, in Liu et al.'s algorithm, each node may only schedule a broadcast when it receives a message for the first time. In contrast, in Algorithm 1, 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. Clearly, the main design issue in Algorithm 1 is how to select the forwarding nodes.

Algorithm 1. A general sender-based algorithm

```
Extract information from the received message M
if M has been scheduled for broadcast or does not
  contain node's ID then
    drop the message
else
    set a defer timer
end if
When defer timer expires
    Select a subset of neighbors to forward the message
    Attach the list of forwarding node to the message
    Schedule a broadcast
```

3.2 Forwarding-Node Selection Algorithm

Let us consider point A as the node N_A and a circle $\mathcal{C}_{A,R}$ centered at A with a radius R as the transmission range of N_A. We use $\overline{AB}$ to denote the distance between two points A and B. Before delving into the algorithm description and proofs, we need to define the following terms:

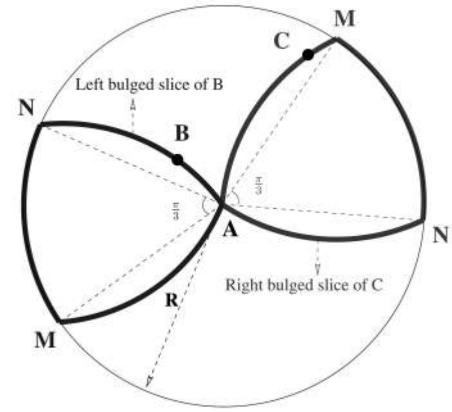


Fig. 2. Left bulged slice of B and right bulged slice of C around A.

Definition 1 (bulged slice). *As illustrated in Fig. 1, we define a bulged slice around A as the intersection area of three circles with radius R and centers A, M, and N, where $\overline{AM} = R$, $\overline{AN} = R$, and $\overline{MN} = R$. Note that in any bulged slice AMN, we have $\angle MAN = \frac{\pi}{3}$.*

Definition 2 (right/left bulged slice). *As shown in Fig. 2, let A and B be two points such that $0 < \overline{AB} \leq R$ and AMN be a bulged slice around A. Suppose that the point B is on one of the arcs $\widehat{AM}$ or $\widehat{AN}$ of the bulged slice AMN. In this case, AMN is called the right bulged slice of B around A if it contains the $\frac{\pi}{3}$ clockwise rotation of point B around A and is called its left bulged slice around A otherwise.*

Definition 3 (bulged angle). *Let $\mathcal{B}_1$ and $\mathcal{B}_2$ be two bulged slices around A. The bulged angle $\angle_A(\mathcal{B}_1, \mathcal{B}_2)$ is defined to be equal to $0 \leq \alpha < 2\pi$ if $\mathcal{B}_2$ is an α counterclockwise rotation of $\mathcal{B}_1$ around A.*

Definition 4 (B-coverage set). *A subset of neighbors of N_A is called a B-coverage set of N_A if any nonempty bulged slice around A contains at least one node from the set. A bulged slice is empty if there is no node inside it.*

Definition 5 (slice-based selection algorithm). *A forwarding-node selection algorithm is called a slice-based selection algorithm (or slice-based algorithm) if for any node N_A, it selects a B-coverage set of it.*

A node can have several different B-coverage sets. Therefore, there is more than one slice-based selection algorithm. For example, a trivial slice-based selection algorithm would be one that selects all of the neighbors as the B-coverage set. Clearly, this algorithm will result in flooding if it is used as the forwarding-node selection scheme in Algorithm 1. In this section, we first show that Algorithm 1 can achieve full delivery if it uses any slice-based algorithm to select the forwarding nodes. We then present an efficient slice-based

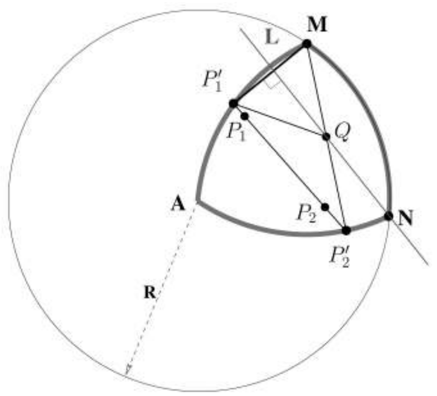


Fig. 3. Upper bound on the distance between nodes inside a bulged slice.

algorithm that selects 11 nodes in the worst case and has computational complexity $O(n)$, where n is the number of neighbors.

Lemma 1. *For any two points P_1 and P_2 inside a bulged slice, we have*

$$\overline{P_1P_2} \leq R.$$

Proof. As shown in Fig. 3, the line passing through P_1 and P_2 intersects the bulged slice AMN at P_1' and P_2'. Clearly, $\overline{P_1P_2} \leq \overline{P_1'P_2'}$. Therefore, to prove the lemma, it is sufficient to show that $\overline{P_1'P_2'} \leq R$. This is easy to show if both P_1' and P_2' are on the same arc of the bulged slice. Thus, without loss of generality, we can assume that P_1' and P_2' are on the arcs $\widehat{AM}$ and $\widehat{AN}$, respectively. Let us consider the perpendicular bisector of the line segment $P_1'M$ (line L). Line L passes through N because $\overline{NM} = \overline{NP_1'} = R$. Since the point P_2' is on the arc $\widehat{AN}$, the line segment MP_2' will cross the line L at a point Q. Using triangle inequality, we have

$$\overline{P_1'P_2'} \leq \left(\overline{QP_2'} + \overline{QP_1'}\right) = \left(\overline{QP_2'} + \overline{QM}\right) = P_2'M = R.$$

Note that Q is on the line L; hence, $\overline{QP_1'} = \overline{QM}$. □

Consider two points A and B such that $R < \overline{AB} \leq 2R$. As shown in Fig. 4, the line segment AB intersects the circle $\mathcal{C}_{A,R}$ at point Q. Let AQM and AQN be the left and right bulged slices of Q around A, respectively. The following lemmas hold:

Lemma 2. *A point P is inside the bulged slice AQM or AQN if $\overline{AP} \leq R$ and $\overline{BP} \leq R$.*

Proof. It is easy to show that for any triangle $\triangle ABC$, $\overline{AM} \leq \overline{AB}$ or $\overline{AM} \leq \overline{AC}$, where M is a point on the line segment BC. Consequently, in the triangle $\triangle PAB$ (shown in Fig. 4), we have

$$\overline{PQ} \leq \overline{AP} \leq R \quad \text{or} \quad \overline{PQ} \leq \overline{BP} \leq R.$$

Therefore, $\overline{PQ} \leq R$. Thus, based on the bulged slice definition, the point P is inside the bulged slice AQM or AQN. □

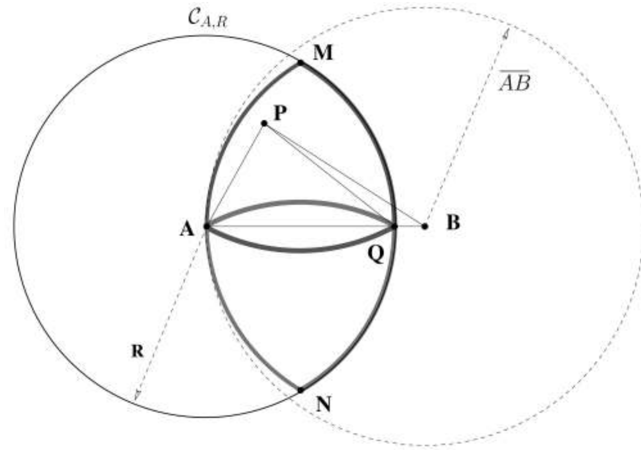


Fig. 4. Other properties of bulge slice from Lemmas 2 and 3.

Lemma 3. *For any point $P \neq A$ inside the bulged slice AQM or AQN, we have*

$$\overline{BP} < \overline{BA}.$$

Proof. Using triangle inequality, we get

$$\overline{BP} \leq \overline{BQ} + \overline{QP} \leq \overline{BQ} + R = \overline{BA}.$$

Note that this equality holds only when $\overline{BP} = \overline{BQ} + \overline{QP}$ and $\overline{QP} = R$ or simply when $P = A$. □

Theorem 1. *In a collision-free network, Algorithm 1 can achieve full delivery if it uses a slice-based selection algorithm to select the forwarding nodes.*

Proof. Using Algorithm 1, 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 broadcasting termination. Let us consider the following set:

$$\begin{aligned}\Lambda = \{(N_X, N_Y, N_Z) | & N_X \text{ has broadcast the message,} \\ & N_Z \text{ has not received the message, and} \\ & N_Y \text{ is the neighbor of both } N_X \text{ and } N_Z\}.\end{aligned}$$

Suppose N_S is the node that initiated broadcasting, 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_C and N_B along the path from N_T to N_S such that N_C has not received the message, while N_B has received it. Suppose that N_B has received the message from N_A. Consequently, $(N_A, N_B, N_C) \in \Lambda$; thus, $\Lambda \neq \emptyset$. As a result, we have

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

Obviously, $N_{A'}$ and $N_{C'}$ are not neighbors, because $N_{C'}$ has not received the message. Thus, $\overline{A'C'} > R$. Using Lemma 2, B' is inside the bulged slice $A'P_1P_2$ or $A'P_1P_3$, where P_1 is the intersection of line segment $A'C'$ and the circle $\mathcal{C}_{A',R}$, and $A'P_1P_2$ and $A'P_1P_3$ are the left and the right bulged slices of P_1 around A', respectively. Without loss of generality, assume that B' is inside the bulged slice $A'P_1P_2$. Since $N_{A'}$ has at least one neighbor (i.e., $N_{B'}$) in this slice, there must be a selected node N_D in the slice that has forwarded the message. Using Lemma 1, we get $\overline{B'D} \leq R$; hence, nodes N_D and $N_{B'}$ are neighbors. Therefore, $(N_D, N_{B'}, N_{C'}) \in \Lambda$. However, this contradicts (1) because using Lemma 3, we have

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

□

Algorithm 2 shows our proposed slice-based selection algorithm. Suppose that node N_A uses the proposed algorithm to select the forwarding nodes from its neighbors. Let us assume that N_A stores all of its neighbors' *ID*s and locations in an array of length n, where n is the number of neighbors. The algorithm selects the first node N_{S_1} randomly from the array. The first node can also be selected deterministically by, for example, selecting the node that is the farthest away from N_A. Let $\mathcal{LB}_A(P)$ and $\mathcal{RB}_A(P)$ denote the left bulged slice and right bulged slice of P around A, respectively. Suppose that N_{S_i} is the last node selected by the algorithm. To select the next node, the algorithm iterates through the array and selects the node $N_{S_{i+1}}$ such that it is inside the slice $\mathcal{LB}_A(S_i)$, $\angle_A(\mathcal{LB}_A(S_i), \mathcal{LB}_A(S_{i+1})) \neq 0$, and

$$\begin{aligned} &\forall N_B \text{ inside } \mathcal{LB}_A(S_i): \\ &\angle_A(\mathcal{LB}_A(S_i), \mathcal{LB}_A(B)) \leq \angle_A(\mathcal{LB}_A(S_i), \mathcal{LB}_A(S_{i+1})). \end{aligned} \tag{2}$$

If there is no such node, the algorithm selects $N_{S_{i+1}}$ such that

$$\begin{aligned} &\forall N_B \text{ inside } \mathcal{C}_{A,R}: \\ &\angle_A(\mathcal{LB}_A(S_i), \mathcal{RB}_A(S_{i+1})) \leq \angle_A(\mathcal{LB}_A(S_i), \mathcal{RB}_A(B)). \end{aligned} \tag{3}$$

The algorithm terminates by selecting the last node N_{S_m} if N_{S_m} is inside $\mathcal{LB}_A(S_1)$ or N_{S_1} is inside $\mathcal{LB}_A(S_m)$ or $S_{m+1} = S_1$.

Algorithm 2 A slice-based selection algorithm
Input: $List_A[1 \dots n]$: List of all neighbors of N_A
Output: A B-coverage set of N_A : $\{N_{S_i}\}$

$ind \leftarrow 1$; $i \leftarrow 0$
repeat
  $ang\_max \leftarrow 0$; $ang\_min \leftarrow 2\pi$
  $i \leftarrow i + 1$
  $N_{S_i} \leftarrow List_A[ind]$
  $chk \leftarrow false$
  for $j = 1$; $j \leq \text{length}(List_A)$; j++ **do**
    if $List_A[j]$ is in $\mathcal{LB}_A(S_i)$ **then**
      if $\angle_A(\mathcal{LB}_A(S_i), \mathcal{LB}_A(List_A[j])) > ang\_max$ **then**
        $chk \leftarrow true$
        $ind\_max \leftarrow j$
        $ang\_max \leftarrow \angle_A(\mathcal{LB}_A(S_i), \mathcal{LB}_A(List_A[j]))$
      end if
    else
      if $\angle_A(\mathcal{LB}_A(S_i), \mathcal{RB}_A(List_A[j])) < ang\_min$ **then**
        $ind\_min \leftarrow j$
        $ang\_min \leftarrow \angle_A(\mathcal{LB}_A(S_i), \mathcal{RB}_A(List_A[j]))$
      end if
    end if
  end for
  if chk **then**
    $ind \leftarrow ind\_max$
  else
    $ind \leftarrow ind\_min$
  end if
until S_1 is in $\mathcal{LB}_A(List_A[ind])$ OR $List_A[ind]$ is in $\mathcal{LB}_A(S_1)$
if $ind \neq 1$ **then**
  $N_{S_{i+1}} \leftarrow List_A[ind]$
end if

Lemma 4. *Suppose the proposed algorithm selects m nodes $\{N_{S_1}, N_{S_2}, \dots, N_{S_m}\}$. For any $1 \leq i < m - 2$, we have*

$$\angle_A(\mathcal{LB}_A(S_i), \mathcal{LB}_A(S_{i+2})) > \frac{\pi}{3}.$$

Proof. Based on (2), (3), and the algorithm termination condition, we can show that

$$\angle_A(\mathcal{LB}_A(S_i), \mathcal{LB}_A(S_{i+2})) > \angle_A(\mathcal{LB}_A(S_i), \mathcal{LB}_A(S_{i+1}))$$

for any $1 \leq i < m - 2$. By contradiction, assume that

$$\angle_A(\mathcal{LB}_A(S_i), \mathcal{LB}_A(S_{i+2})) \leq \frac{\pi}{3}.$$

Therefore, S_{i+2} is inside $\mathcal{LB}_A(S_i)$. Thus, using (2), we have

$$\angle_A(\mathcal{LB}_A(S_i), \mathcal{LB}_A(S_{i+2})) \leq \angle_A(\mathcal{LB}_A(S_i), \mathcal{LB}_A(S_{i+1})),$$

which is a contradiction. □

Theorem 2. *The proposed slice-based selection algorithm will select at most 11 nodes.*

Proof. By contradiction, assume that the algorithm selects more than 11 nodes. Therefore, S_{11} is not in $\mathcal{LB}_A(S_1)$. Using Lemma 4, we get

$$\begin{aligned} &\angle_A(\mathcal{LB}_A(S_1), \mathcal{LB}_A(S_{11})) \\ &\quad = \sum_{i=1}^{5} (\angle_A(\mathcal{LB}_A(S_{2i-1}), \mathcal{LB}_A(S_{2i+1}))) > 5 \times \frac{\pi}{3}. \end{aligned}$$

Therefore, $\angle_A(\mathcal{LB}_A(S_{11}), \mathcal{LB}_A(S_1)) < (2\pi - 5 \times \frac{\pi}{3}) = \frac{\pi}{3}$. Consequently, S_1 is inside $\mathcal{LB}_A(S_{11})$; thus, the proposed slice-based algorithm will terminate after selecting S_{11}. □

The above theorem gives an upper bound on the number of nodes selected by the proposed selection algorithm. In Section 5, using simulation, we show that the average number of selected nodes (when the nodes are distributed uniformly) is less than six.

Theorem 3. *Time complexity of the proposed slice-based selection algorithm is $O(n)$, where n is the number of neighbors.*

Proof. The algorithm selects the first node in $O(1)$. To select each of the other nodes, the algorithm performs $O(n)$ operations by checking all the neighbors in the array. Therefore, the complexity of the algorithm is $O(m \times n)$, where m is the number of selected nodes. Using Theorem 2, we have $m \leq 11$; thus, the time complexity of algorithm is $O(n)$. □

3.3 Reducing the Number of Forwarding Nodes

In the sender-based broadcasting algorithms, each broadcasting node attaches a list of its selected forwarding nodes to the message before broadcasting it. This procedure will increase the bandwidth and power required to broadcast the message. As shown earlier, our proposed slice-based selection algorithm reduces the number of selected forwarding nodes to 11 in the worst case. In this section, we show how to further reduce the number of selected nodes.

Recall that the proposed slice-based algorithm selects a subset of N_A's neighbors such that there is at least one selected node in any nonempty bulged slice around A. Suppose N_A extracts the list of the forwarding nodes from each message it receives. Let $\mathcal{L}_A$ be a subset of N_A's neighbors that has broadcast the message or been selected by other nodes to forward it. Since all of the selected forwarding nodes are required to broadcast the message, it is sufficient for N_A to find a subset of its neighbors $\mathcal{S}_A$ such that any nonempty bulged slice around A contains at least one node from $\mathcal{S}_A \cup \mathcal{L}_A$. Algorithm 2 can be simply extended to achieve this in $O(n)$. Note that the extended algorithm can start with a node from $\mathcal{L}_A$ and select any node in $\mathcal{L}_A$ as soon as it appears in the left bulged slice of the previously selected node. Finally, the extended algorithm removes all of the nodes in $\mathcal{L}_A$ from the set of selected nodes.

3.4 Maximizing the Minimum Node Weight of B-Coverage Set

Suppose node N_A assigns a weight to each of its neighbors. The weight can represent the neighbor's battery lifetime, its distance to N_A, the average delay of the node, the level of trust, or a combination of them. In some scenarios, we may desire to find a B-coverage set such that its minimum node weight is the maximum or its maximum node weight is the minimum among that of all B-coverage sets. For example, assume that the weight of each node represents its battery lifetime in a wireless network. It may be desirable to select the nodes with a higher battery lifetime to forward the message in order to keep the nodes with a lower battery lifetime alive. Algorithm 3 shows how to find a B-coverage set such that its minimum node weight is the maximum among that of all B-coverage sets. A similar approach can be used to find a B-coverage set such that its maximum node weight is the minimum.

Algorithm 3. Maximizing the minimum node weight
Input: $List_A[1 \ldots n]$: List of all neighbors of N_A
Output: A B-coverage set of N_A with highest minimum node weight

1: $SList_A \leftarrow sort(List_A)$ {Sort the neighbor nodes by their weights} $\{SList[i] \geq SList[j] \Leftrightarrow i \leq j\}$
2: $H \leftarrow n;\ T \leftarrow 1; m \leftarrow \lfloor \frac{n}{2} \rceil$
3: $St \leftarrow Algorithm\_2(SList[1])$
4: **if** St is a B-coverage set for N_A **then**
5:   **return** SList[1]
6: **end if**
7: **while** $H > T + 1$ **do**
8:   $St \leftarrow Algorithm\_2(SList[1 \ldots m])$ {Pass m nodes with the highest weights to Algorithm 2 as the input}
9:   **if** St is a B-coverage set for N_A **then**
10:     $H \leftarrow m$
11:     $m \leftarrow \lceil \frac{T+m}{2} \rceil$
12:   **else**
13:     $T \leftarrow m$
14:     $m \leftarrow \lceil \frac{H+m}{2} \rceil$
15:   **end if**
16: **end while**
17: **return** $(\text{Algorithm\_2}(SList[1 \ldots H]))$

Algorithm 3 first sorts the nodes by their weights in decreasing order. Then, in each step, it passes m nodes with the highest weights to Algorithm 2 as input and gets a set of (at most 11) nodes as output, where $1 \leq m \leq n$ is an integer initially set to $\lceil \frac{n}{2} \rceil$. If the output set is a B-coverage set, Algorithm 3 sets H to m and decreases m to $\lceil \frac{T+m}{2} \rceil$, where T and H are variables initially set to 1 and n, respectively. Otherwise, it sets T to m and increases m to $\lceil \frac{H+m}{2} \rceil$. After a finite number of steps, we get $H = T + 1$. Algorithm 3 then returns the output of $\text{Algorithm\_2}(SList[1 \ldots H])$.

Corollary 1. *Algorithm 3 will select at most 11 nodes.*

Proof. The proof is clear, as Algorithm 3 returns an output of Algorithm 2 (Line 17). □

Theorem 4. *The time complexity of Algorithm 3 is $O(n \log n)$.*

Proof. Algorithm 3 requires $O(n \log n)$ operations to sort the list of neighbors $List_A[1 \ldots n]$. The computational complexity of Algorithm 2 is $O(n)$. Therefore, Algorithm 3 performs $O(n)$ operations in each iteration of the while loop. The while loop terminates after $O(\log n)$ iterations because it uses a binary search approach to find the minimum value of H. Consequently, the order of Algorithm 3 is $O(n \log n + \log n \times n)$. □

Theorem 5. *The minimum weight of nodes of the B-coverage set selected by Algorithm 3 is the maximum among that of all B-coverage sets.*

Proof. Suppose that St_{min} is a B-coverage set such that the minimum weight of nodes in St_{min} is greater than or equal to that of other B-coverage sets. Let $N_X \in St_{min}$ be the node with the minimum weight in St_{min}. Assume that N_A has K neighbors with weights greater than or equal to the weight of N_X. Therefore, the output of $\text{Algorithm\_2}(SList[1 \ldots K])$ is a B-coverage set. Note that Algorithm 3 finds the minimum H such that the output of $\text{Algorithm\_2}(SList[1 \ldots H])$ is a B-coverage set for N_A. Therefore, $H \leq K$, and thus, the minimum weight of nodes of the B-coverage set selected by Algorithm 3 is greater than or equal to the weight of N_X. □

3.5 Similarity with a Topology Control Algorithm

In [16] and [17], the authors proposed a cone-based topology control algorithm, where each node makes local decisions about its transmission power. The objective of the algorithm is to minimize the transmission power of each node without violating the network connectivity. In order to do that, each node N_A transmits with the minimum power P_α such that in every nonempty cone of degree α around N_A, there is some node that N_A can reach with power P_α. A cone is nonempty if there is at least a node in the cone that N_A can reach using its maximum power. For $\alpha = \frac{2\pi}{3}$, they

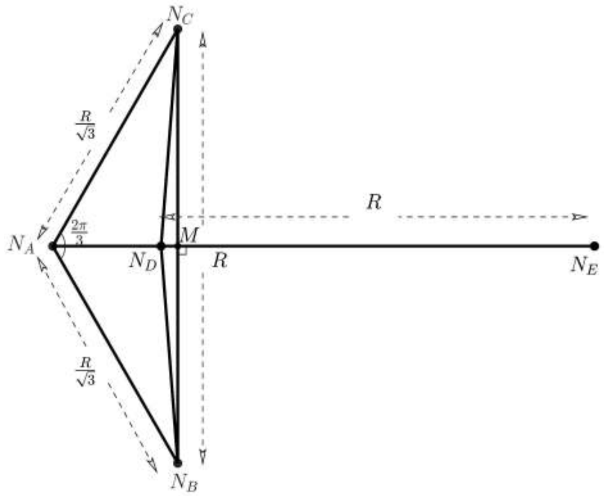


Fig. 5. A counterexample for $\alpha = \frac{2\pi}{3}$.

proved that the network remains connected if the cone-based algorithm is employed.

Suppose that we use cones instead of bulged slices in the proposed forwarding-node selection algorithm. Therefore, the algorithm will select the forwarding node set such that any nonempty cone of degree α around N_A contains at least one node from the forwarding node set. Surprisingly, this algorithm will not guarantee full delivery. Fig. 5 shows a counterexample for the case where $\alpha = \frac{2\pi}{3}$. Fig. 6 shows that even for $\alpha = \frac{\pi}{3}$, full delivery cannot be guaranteed. In both Figs. 5 and 6, the node N_A initiates broadcasting and selects only N_B and N_C to forward the message. Suppose that N_D is close enough to the point M such that it is the only node that can reach N_E. In this case, N_E will not receive the message because N_D is not selected by neither N_B nor N_C to forward the message. Note that the cone-based and the forwarding-node selection algorithms use different approaches. In the cone-based algorithm, a node N_A *increases its power* from zero until there is a node in each nonempty cone around N_A. However, in the forwarding-node selection algorithm, a node N_A *selects* some nodes (the forwarding nodes) until there is a selected node in each nonempty bulged slice around N_A.

4 A Highly Efficient Receiver-Based Broadcasting Algorithm

In this section, we propose a novel receiver-based broadcasting algorithm that can significantly reduce redundant broadcasts in the network. As mentioned earlier, in receiver-based broadcasting algorithms, the receiver of the message decides whether or not to broadcast the message. Therefore, a potential advantage of receiver-based broadcasting algorithms over sender-based ones is that they do not increase the size of the message by adding a list of forwarding nodes.

4.1 Algorithm Structure

Algorithm 4 shows a general approach used in several receiver-based broadcasting algorithms [13], [18]. Our proposed receiver-based broadcasting algorithm employs

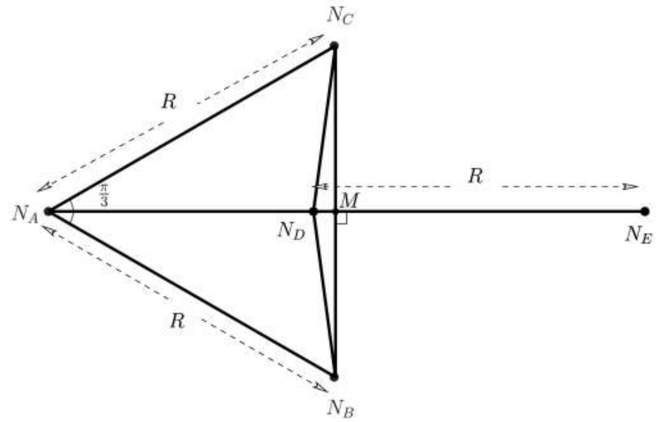


Fig. 6. A counterexample for $\alpha = \frac{\pi}{3}$.

this approach. Clearly, the main design challenge of Algorithm 4 is to determine whether or not to broadcast a received message. A trivial algorithm is to refrain broadcasting if and only if all the neighbors have received the message during the defer period. Although this algorithm is simple to implement, it has limited effect in reducing the number of redundant broadcasts. Suppose N_A's defer time expires at t_0. Using the above strategy, node N_A will broadcast if some of its neighbors (at least one) have not received the message by t_0. However, this broadcast is redundant if all such neighbors receive the message from other nodes after time t_0. This scenario typically occurs when t_0 is small compared to the maximum defer time. In the next section, we introduce a responsibility-based scheme (RBS) that further reduces the redundant broadcasts without any changes in the MAC-layer defer-time design.

Algorithm 4. A general receiver-based algorithm

```
Extract information from the received message M
if M has been received before then
  drop the message
else
  set a defer timer
end if
When defer timer expires
  decide whether or not to schedule a broadcast
```

4.2 Responsibility-Based Scheme

Algorithm 5 shows the proposed RBS. The main idea of Algorithm 5 is that a node avoids broadcasting if it is not responsible for any of its neighbors. A node N_A is not responsible for a neighbor N_B if N_B has received the message or if there is another neighbor N_C such that N_C has received the message and N_B is closer to N_C than it is to N_A. Suppose N_A stores *ID*s of all its neighbors that have broadcast the message during the defer period. When executed by a node N_A, Algorithm 5 first uses this information to determine which neighbors have not received the message (Lines 1-9 of Algorithm 5). It then returns *false* if and only if it finds a neighbor N_B that has not received the message and

$$\overline{AB} \leq \overline{BC}$$

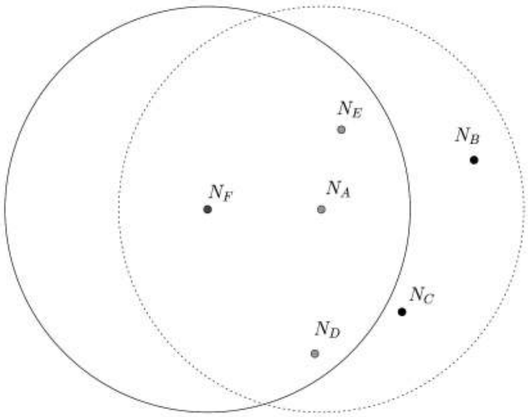


Fig. 7. An example of an RBS decision.

for any N_A's neighbor N_C that has received the message. The output of RBS determines whether or not the broadcast is redundant.

Algorithm 5. RBS
Input: $List_A$: List of all neighbors of N_A, and $List_B$: List of broadcasting neighbors
Output: true or false

1: $List_C \leftarrow List_A$
2: **for** $i = 1$; $i \leq \text{length}(List_C)$; i++ **do**
3: **for** $j = 1$; $j \leq \text{length}(List_B)$; j++ **do**
4: **if** $\text{dist}(List_C[i], List_B[j]) \leq R$ **then**
5: $\text{removeElement}(List_C[i], List_C)$
6: break
7: **end if**
8: **end for**
9: **end for**
10: $List_D \leftarrow List_A - List_C$
11: **for** $i = 1$; $i \leq \text{length}(List_C)$; i++ **do**
12: $check \leftarrow true$
13: **for** $j = 1$; $j \leq \text{length}(List_D)$; j++ **do**
14: **if** $\text{dist}(List_C[i], List_D[j]) < \text{dist}(List_C[i], N_A)$ **then**
15: $check \leftarrow false$
16: break
17: **end if**
18: **end for**
19: **if** check **then**
20: **return** (false)
21: **end if**
22: **end for**
23: **return** (true)

Example 1. As shown in Fig. 7, N_A has five neighbors. Suppose that N_A has received a message from N_F. Note that N_A has the position of all its neighbors. Therefore, it can find that N_E and N_D have received the message but N_B and N_C have not. As shown in Fig. 7, N_A is not required to broadcast because

$$\overline{BE} < \overline{BA} \quad \text{and} \quad \overline{CD} < \overline{CA}.$$

Theorem 6. *In a collision-free network, Algorithm 4 can achieve full delivery if it uses the proposed RBS to determine whether or not to broadcast.*

Proof. Using Algorithm 5, each node broadcasts a 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 broadcasting 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 node that initiated broadcasting, 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, we have

$$\exists (N_{A'}, N_{B'}) \in \Lambda \quad \text{s.t.} \quad \forall (N_X, N_Y) \in \Lambda : \quad \overline{A'B'} \leq \overline{XY}. \quad (4)$$

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 $\overline{C'B'} < \overline{A'B'} \leq R$. This result contradicts (4), since $(N_{C'}, N_{B'}) \in \Lambda$. □

Theorem 7. *The time complexity of the proposed RBS is $O(n^2)$, where n is the number of neighbors.*

Proof. Algorithm 5 consists of two parts. In the first part (Lines 1-9), the algorithm generates a list of neighbors that have not received the message ($List_C$). Clearly, the time complexity of this part is $O(kn)$, where $1 < k \leq n$ is the number of broadcasting neighbors. In the second part, the algorithm checks whether there is a node N_B such that N_B has not received the message and $\overline{BA} \leq \overline{BC}$ for any neighbor $N_C \in List_D$. The time complexity of this part is $O(lm)$, where $0 \leq l \leq n$ is the number of neighbors that have not received the message, and $1 \leq m \leq n$ is the number of neighbors that have received it. Therefore, the complexity of the algorithm is $O(lm + kn)$. □

4.3 A Property of the Proposed RBS

In the simulation section (Section 5), we show that the proposed RBS can significantly reduce the number of broadcasts in the network. In particular, our simulation shows that using RBS, the average number of broadcasts is less than one of the best known approximations for the minimum number of required broadcasts. To justify this, we prove a property of the proposed RBS.

Assume that nodes are placed randomly inside a square area of size $L \times L$ using a homogeneous planar Poisson distribution. Therefore, nodes are independently and uniformly distributed in the area. Moreover, we have

$$Prob(\text{number of nodes in area } \tau = k) = \frac{(\delta\tau)^k e^{-\delta\tau}}{k!},$$

where δ is the density of nodes [19], [20]. Suppose node N_B receives the message from N_A for the first time. For simplicity, assume that circle $\mathcal{C}_{A,2R}$ is completely inside the square area. Corollary 2 shows that the probability that N_B broadcasts the message exponentially decreases when the distance $\overline{AB}$ decreases or when the node density δ increases. This result is further confirmed by simulation in Section 5.

Example 2. Suppose $R=250$ m, $L=1{,}000$ m, and $\delta L^2=300$ (i.e., there are about 300 nodes in the network). Let $Prb(Brd_B)$ be the probability that N_B broadcasts the message after receiving it from N_A. Using Theorem 8, we get $Prb(Brd_B) \leq 1.26 \times 10^{-2}$, $Prb(Brd_B) \leq 1.4 \times 10^{-3}$, and $Prb(Brd_B) \leq 10^{-4}$ when $\overline{AB} = 100$ m, $\overline{AB} = 80$ m, and $\overline{AB} = 60$ m, respectively.

Let $\mathcal{R}_1, \mathcal{R}_2, \ldots \mathcal{R}_k$ be k nonoverlapping regions inside the network. Suppose that $\zeta_{\mathcal{R}}$ is the event

$$\zeta_{\mathcal{R}} = \{\text{The region } \mathcal{R} \text{ contains at least one node}\}.$$

Since the nodes are placed by homogeneous planar Poisson distribution, the events $\zeta_{\mathcal{R}_i}$ are independent [20]. Consequently, we have

$$Prb(\zeta_{\mathcal{R}_1}, \zeta_{\mathcal{R}_2}, \ldots, \zeta_{\mathcal{R}_k}) = Prb(\zeta_{\mathcal{R}_1}) Prb(\zeta_{\mathcal{R}_2}) \ldots Prb(\zeta_{\mathcal{R}_k}). \quad (5)$$

Lemma 5 generalizes (5) to the case where the regions may overlap each other.

Lemma 5. *Let $\mathcal{R}_1, \mathcal{R}_2, \ldots \mathcal{R}_k$ be k regions inside the network. We have*

$$Prb(\zeta_{\mathcal{R}_1}, \zeta_{\mathcal{R}_2}, \ldots, \zeta_{\mathcal{R}_k}) \geq Prb(\zeta_{\mathcal{R}_1}) Prb(\zeta_{\mathcal{R}_2}) \ldots Prb(\zeta_{\mathcal{R}_k}).$$

Proof. The proof is by induction on the number of regions. The lemma is true if the number of regions is one (i.e., $k=1$). Suppose that the inequality holds for $k=d$ regions. We have

$$\begin{aligned} &Prb\big(\zeta_{\mathcal{R}_1}, \zeta_{\mathcal{R}_2}, \ldots, \zeta_{\mathcal{R}_d} | \bar{\zeta}_{\mathcal{R}_{d+1}}\big) \\ &\quad = Prb\big(\zeta_{\mathcal{R}_1 - \mathcal{R}_{d+1}}, \zeta_{\mathcal{R}_2 - \mathcal{R}_{d+1}}, \ldots, \zeta_{\mathcal{R}_d - \mathcal{R}_{d+1}}\big), \end{aligned} \quad (6)$$

where $\bar{\zeta}_{\mathcal{R}_i}$ is the complement of $\zeta_{\mathcal{R}_i}$, and $R_i - R_j$ is the collection of all points inside R_i and outside R_j. Note that

$$Prb\big(\zeta_{\mathcal{R}_1}, \zeta_{\mathcal{R}_2}, \ldots, \zeta_{\mathcal{R}_d} | \zeta_{\mathcal{R}_1 - \mathcal{R}_{d+1}}, \zeta_{\mathcal{R}_2 - \mathcal{R}_{d+1}}, \ldots, \zeta_{\mathcal{R}_d - \mathcal{R}_{d+1}}\big) = 1.$$

Thus,

$$\begin{aligned} &Prb(\zeta_{\mathcal{R}_1}, \zeta_{\mathcal{R}_2}, \ldots, \zeta_{\mathcal{R}_d}) \\ &\quad \geq Prb\big(\zeta_{\mathcal{R}_1 - \mathcal{R}_{d+1}}, \zeta_{\mathcal{R}_2 - \mathcal{R}_{d+1}}, \ldots, \zeta_{\mathcal{R}_d - \mathcal{R}_{d+1}}\big). \end{aligned} \quad (7)$$

It follows from (6) and (7) that

$$Prb(\zeta_{\mathcal{R}_1}, \zeta_{\mathcal{R}_2}, \ldots, \zeta_{\mathcal{R}_d}) \geq Prb\big(\zeta_{\mathcal{R}_1}, \zeta_{\mathcal{R}_2}, \ldots, \zeta_{\mathcal{R}_d} | \bar{\zeta}_{\mathcal{R}_{d+1}}\big). \quad (8)$$

We have

$$\begin{aligned} &Prb(\zeta_{\mathcal{R}_1}, \zeta_{\mathcal{R}_2}, \ldots, \zeta_{\mathcal{R}_d}) \\ &\quad = Prb\big(\zeta_{\mathcal{R}_{d+1}}\big) Prb\big(\zeta_{\mathcal{R}_1}, \zeta_{\mathcal{R}_2}, \ldots, \zeta_{\mathcal{R}_d} | \zeta_{\mathcal{R}_{d+1}}\big) \\ &\qquad + Prb\big(\bar{\zeta}_{\mathcal{R}_{d+1}}\big) Prb\big(\zeta_{\mathcal{R}_1}, \zeta_{\mathcal{R}_2}, \ldots, \zeta_{\mathcal{R}_d} | \bar{\zeta}_{\mathcal{R}_{d+1}}\big). \end{aligned}$$

Therefore, using (8), we get

$$\begin{aligned} &Prb\big(\zeta_{\mathcal{R}_{d+1}}\big) Prb\big(\zeta_{\mathcal{R}_1}, \zeta_{\mathcal{R}_2}, \ldots, \zeta_{\mathcal{R}_d} | \zeta_{\mathcal{R}_{d+1}}\big) \\ &\quad \geq \big(1 - Prb\big(\bar{\zeta}_{\mathcal{R}_{d+1}}\big)\big) Prb(\zeta_{\mathcal{R}_1}, \zeta_{\mathcal{R}_2}, \ldots, \zeta_{\mathcal{R}_d}). \end{aligned}$$

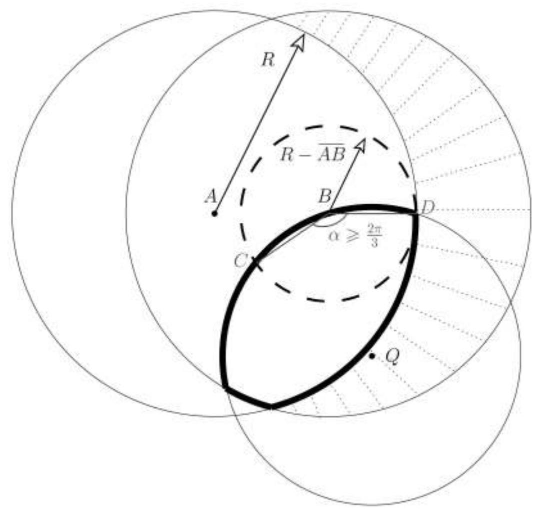


Fig. 8. Finding a lower bound for $\Delta(I(\mathcal{D}_{A,R}, \mathcal{D}_{B,R}, \mathcal{D}_{Q,\overline{QB}}))$.

Thus, using an induction hypothesis, we get

$$\begin{aligned} &Prb\big(\zeta_{\mathcal{R}_1}, \zeta_{\mathcal{R}_2}, \ldots, \zeta_{\mathcal{R}_d}, \zeta_{\mathcal{R}_{d+1}}\big) \\ &\quad \geq Prb\big(\zeta_{\mathcal{R}_{d+1}}\big) Prb(\zeta_{\mathcal{R}_1}, \zeta_{\mathcal{R}_2}, \ldots, \zeta_{\mathcal{R}_d}) \\ &\quad \geq Prb(\zeta_{\mathcal{R}_1}) Prb(\zeta_{\mathcal{R}_2}) \ldots Prb(\zeta_{\mathcal{R}_d}) Prb\big(\zeta_{\mathcal{R}_{d+1}}\big). \end{aligned}$$

□

Lemma 6. *Let $\mathcal{D}_{A,R}$ and $\mathcal{D}_{B,R}$ be two disks with radius R and centers A and B, respectively. Suppose $\overline{AB} \leq R$. As shown in Fig. 8, consider a point Q such that $R < \overline{QA}$ and $\overline{QB} \leq R$. Let $\mathcal{D}_{Q,\overline{QB}}$ be a disk with radius $\overline{QB}$. We have*

$$\Delta\Big(\mathcal{I}(\mathcal{D}_{A,R}, \mathcal{D}_{B,R}, \mathcal{D}_{Q,\overline{QB}})\Big) \geq \frac{\pi(R - \overline{AB})^2}{3},$$

where $\mathcal{I}(\mathcal{D}_{A,R}, \mathcal{D}_{B,R}, \mathcal{D}_{Q,\overline{QB}})$ is the intersection of disks $\mathcal{D}_{A,R}$, $\mathcal{D}_{B,R}$, and $\mathcal{D}_{Q,\overline{QB}}$, and $\Delta(R)$ is the area of region R.

Proof. For any point P on the circle $\mathcal{C}_{A,R}$, we have

$$\overline{BP} \leq \overline{AP} - \overline{AB} = R - \overline{AB}.$$

Therefore, as shown in Fig. 8, the disk $\mathcal{D}_{B,(R-\overline{AB})}$ is inside $\mathcal{I}(\mathcal{D}_{A,R}, \mathcal{D}_{B,R})$. Consequently, we have

$$\Delta\Big(I(\mathcal{D}_{A,R}, \mathcal{D}_{B,R}, \mathcal{D}_{Q,\overline{QB}})\Big) \geq \Delta\Big(I(\mathcal{D}_{B,(R-\overline{AB})}, \mathcal{D}_{Q,\overline{QB}})\Big).$$

Since $\overline{QA} \geq R$, using triangle inequality, we get

$$\overline{QB} \geq \overline{QA} - \overline{AB} \geq R - \overline{AB}.$$

Therefore, we have

$$\angle QBC \geq \frac{\pi}{3} \quad \text{and} \quad \angle QBD \geq \frac{\pi}{3},$$

and hence, $\angle CBD \geq \frac{2\pi}{3}$. Therefore,

$$\begin{aligned} \Delta\Big(I(\mathcal{D}_{B,(R-\overline{AB})}, \mathcal{D}_{Q,\overline{QB}})\Big) &\geq \frac{\Delta\Big(\mathcal{D}_{B,(R-\overline{AB})}\Big)}{3} \\ &= \frac{\pi(R - \overline{AB})^2}{3}. \end{aligned}$$

□

Theorem 8. *Suppose* $d \leq R$ *is the distance between two nodes* N_A *and* N_B. *We have*

$$Prb(Brd) \leq 1 - e^{-\delta\gamma e^{-\delta\frac{\pi(R-d)^2}{3}}},$$

where $Prb(Brd)$ *is the probability that* N_B *broadcasts the message after receiving it from* N_A, *and*

$$\gamma = \Delta(\mathcal{D}_{B,R}) - \Delta\big(\mathcal{I}(\mathcal{D}_{A,R}, \mathcal{D}_{B,R})\big)$$

is the area of the hatched crescent shown in Fig. 8.

Proof. Node N_B is not required to broadcast if and only if

$$\zeta^* : \forall N_Q \in \Lambda : \quad \exists N_P \text{ inside } I(\mathcal{D}_{A,R}, \mathcal{D}_{B,R}, \mathcal{D}_{Q,\overline{QB}}),$$

where

$$\Lambda = \{N_X | \overline{AX} > R \quad \text{and} \quad \overline{BX} \leq R\}.$$

Note that the nodes' positions have a Poisson distribution. Therefore, using Lemma 6, we get

$$\begin{aligned} &Prb\Big(\exists N_P \text{ inside } I(\mathcal{D}_{A,R}, \mathcal{D}_{B,R}, \mathcal{D}_{Q,\overline{QB}})\Big) \\ &= 1 - e^{-\delta\Delta\left(I(\mathcal{D}_{A,R}, \mathcal{D}_{B,R}, \mathcal{D}_{Q,\overline{QB}})\right)} \geq 1 - e^{-\delta\frac{\pi(R-d)^2}{3}}. \end{aligned} \tag{9}$$

Thus,

$$\begin{aligned} Prb(Brd) &= 1 - Prb(\zeta^*) \\ &= 1 - \sum_{k=0}^{\infty} Prob(|\Lambda| = k) Prb(\zeta^* | (|\Lambda| = k)), \end{aligned}$$

where $|\Lambda|$ is the cardinality of the set Λ. Therefore, using (9) and Lemma 5, we get

$$\begin{aligned} Prb(Brd) &\leq 1 - \sum_{k=0}^{\infty} \frac{(\delta\gamma)^k e^{-\delta\gamma}}{k!} \left(1 - e^{-\delta\frac{\pi(R-d)^2}{3}}\right)^k \\ &= 1 - e^{-\delta\gamma e^{-\delta\frac{\pi(R-d)^2}{3}}}, \end{aligned}$$

where γ is the area of the hatched crescent in Fig. 8 (collection of all points Q, $QA > R$, and $QB \leq R$):

$$\begin{aligned} \gamma &= \Delta(\mathcal{D}_{B,R}) - \Delta\big(\mathcal{I}(\mathcal{D}_{A,R}, \mathcal{D}_{B,R})\big) \\ &= R^2\left(\pi - 2\arccos\left(\frac{d}{2R}\right)\right) + d\sqrt{R^2 - \left(\frac{d}{2}\right)^2}. \end{aligned}$$

□

Corollary 2. *Using Theorem 8, we get*

$$Prb(Brd) \leq \delta\gamma e^{-\delta\frac{\pi(R-\overline{AB})^2}{3}}.$$

Proof. Consider the function

$$f(x) = x + e^{-x} - 1.$$

It is easy to show that $f(x)$ has a global minimum at $x = 0$. Therefore, we have

$$1 - e^{-x} \leq x$$

for any real number x. As a result, we get

$$Prb(Brd) \leq 1 - e^{-\delta\gamma e^{-\delta\frac{\pi(R-d)^2}{3}}} \leq \delta\gamma e^{-\delta\frac{\pi(R-d)^2}{3}}.$$

□

It is also possible that node N_B receives the message from more than one neighbor in its defer period. In this case, the number of N_B's neighbors that have received the message increases, and the number of that have not received the message decreases. Consequently, the probability that N_B is required to broadcast the message further decreases compared to the case where N_B receives the message from only one neighbor. It is worth mentioning that RBS can guarantee that the number of forwarding nodes is within a constant factor of the optimal solution (minimum CDS) if it is provided with 2-hop neighbor information [21].

4.4 Relaxing Some System Model Assumptions

We assumed in Section 2 that the nodes are placed in a 2D plane. However, this assumption is not used in the proof of Theorem 6. Therefore, the proposed receiver-based algorithm can also achieve full delivery when the nodes are distributed in a 3D space. Note that in this case, RBS uses 3D node positions.

We can also relax the assumption that all the nodes have the same transmission range R. When the nodes' transmission ranges are different, the topology graph should be defined as a directed graph for which there is a link from N_A to N_B if N_B is in the transmission range of N_A. Suppose G is an undirected graph obtained by removing unidirectional 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 the transmission range of each other). Note that many wireless MAC protocols such as IEEE 802.11 require bidirectional links. Let us assume that nodes put not only their *ID* and position but also their transmission range into the hello messages that they periodically broadcast. Therefore, the neighbors of a node know both its position and transmission range. In this case, nodes can use Algorithm 6 to decide whether or not to broadcast.

Algorithm 6 is a modified version of RBS. When executed by a node N_A, Algorithm 6 uses the position and transmission range of the broadcasting nodes to determine which neighbors have not received the message. It then returns *false* if and only if it finds a neighbor N_B that has not received the message and

$$\overline{AB} \leq \overline{BC} \quad \text{or} \quad \overline{BC} > \text{transmission range of } N_C$$

for any N_A's neighbor N_C that has received the message.

Algorithm 6. Modified RBS
Input: $List_A$: List of all neighbors of N_A, and $List_B$: List of broadcasting neighbors
Output: true or false

$List_C \leftarrow List_A$
for $i = 1$; $i \leq \text{length}(List_C)$; i++ **do**
  for $j = 1$; $j \leq \text{length}(List_B)$; j++ **do**
    if $\text{dist}(List_C[i], List_B[j]) \leq R_{List_B[j]}$ **then**
      $\{R_{List_B[j]}$: Transmission range of node $List_B[j]\}$
      $\text{removeElement}(List_C[i], List_C)$
      break
    end if

```
    end for
end for
List_D ← List_A − List_C
for i = 1; i ≤ length(List_C); i++ do
    check ← true
    for j = 1; j ≤ length(List_D); j++ do
        if dist(List_C[i], List_D[j]) < dist(List_C[i], N_A)
        then
            if dist(List_C[i], List_D[j]) ≤ R_{List_D[j]} then
                check ← false
                break
            end if
        end if
    end for
    if check then
        return (false)
    end if
end for
return (true)
```

Corollary 3. *In a collision-free network, Algorithm 4 can achieve full delivery if it uses the modified RBS to determine whether or not to broadcast.*

Proof. The proof is similar to the proof of Theorem 6. By contradiction, suppose there is at least one node that has not received the message after the broadcasting termination. Let us consider the set

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

Suppose N_S is the node that initiated broadcasting, and N_T is a node that has not received the message. There is a path between N_S and N_T in G; thus, we can find two neighbors N_A and N_B along the path from N_S to N_T such that N_A has received the message and N_B has not received it. Consequently, $(N_A, N_B) \in \Lambda$; thus, $\Lambda \neq \emptyset$. As a result, we have

$$\exists (N_{A'}, N_{B'}) \in \Lambda \quad \text{s.t.} \quad \forall (N_X, N_Y) \in \Lambda : \quad \overline{A'B'} \leq \overline{XY}. \quad (10)$$

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

$$\overline{C'B'} < \overline{A'B'} \quad \text{and} \quad \overline{C'B'} \leq \text{Transmission range of } N_{C'}.$$

This result contradicts (10) because $(N_{C'}, N_{B'}) \in \Lambda$. □

We can relax the assumption of having precise position information as well. The reader is referred to [21] for more details about broadcasting under uncertain position information.

5 Simulation

5.1 Average Number of Nodes Selected by the Proposed Sliced-Based Algorithm

In Section 3, we proved that the proposed forwarding-node selection algorithm selects 11 nodes in the worst case. In practice, the number of selected nodes is typically less than

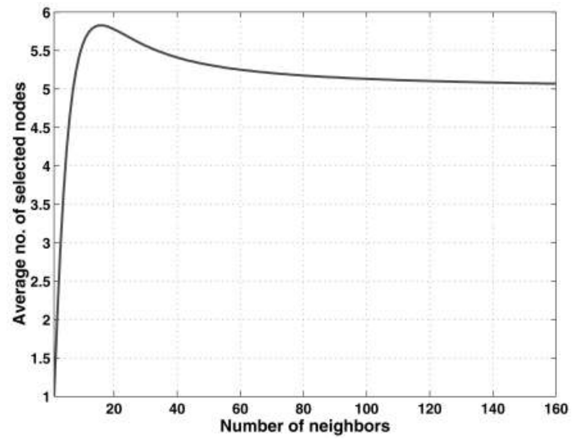


Fig. 9. Average number of nodes selected by the proposed slice-based algorithm.

11. To avoid the complexity of mathematical analysis, we used a simulation to find the average number of selected nodes. For a given number of neighbors $1 \leq n \leq 160$, we randomly put n points inside a circle with radius R. We then ran the proposed selection algorithm and obtained the number of selected nodes. To get the average number of selected nodes, we ran simulation 10^6 times for each given n. As shown in Fig. 9, the average number of selected nodes is less than six and approaches five when n increases. Note that the proposed sliced-based selection algorithm does not necessarily select a B-coverage with a minimum number of nodes. However, there is a sliced-based selection algorithm that can find a B-coverage with a minimum number of nodes in $O(n \log n)$ and can consequently reduce the average number of selected nodes. It is worth mentioning that Fig. 9 shows the average number of selected nodes by the source node (the node that initiates the broadcasting). For the rest of broadcasting nodes, the average number of selected nodes is at least one less than that for the source node because of the optimization technique introduced in Section 3.

5.2 Probability of Broadcast Using the Proposed RBS

Suppose that the proposed receiver-based algorithm is used for broadcasting in the network. Assume that node N_B receives a message from N_A for the first time. It has been proven that the probability of N_B broadcasting the message ($Prb(Brd_B)$) exponentially decreases when the distance $\overline{AB}$ decreases or when the node density δ increases. We used simulation to confirm this theoretical result. For the simulation, we considered two nodes N_A and N_B with distance $0 < d \leq R$ from each other. We uniformly placed nodes with density δ inside the network and checked whether or not N_B was required to broadcast the message. We ran simulation 10^6 times for a given δ and R. We then estimated $Prb(Brd_B)$ by the ratio of the number of times N_B was required to broadcast over the total number of runs.

Figs. 10, 11, 12, and 13 show the simulation results for several values of δ, d, and R. As shown in these figures, the probability of broadcast exponentially decreases when d decreases or when δ increases. For example, when $R = 300$ m

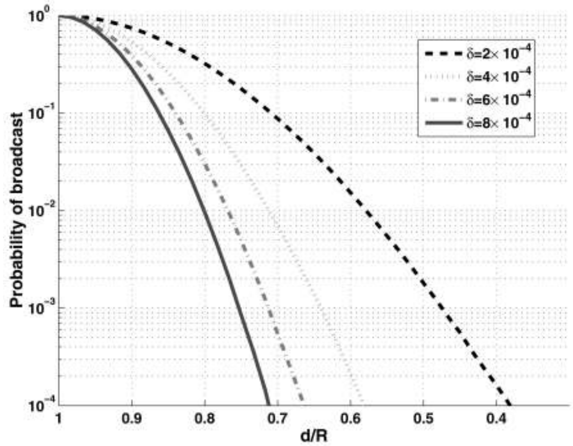


Fig. 10. Probability of broadcast for $R = 300$ m.

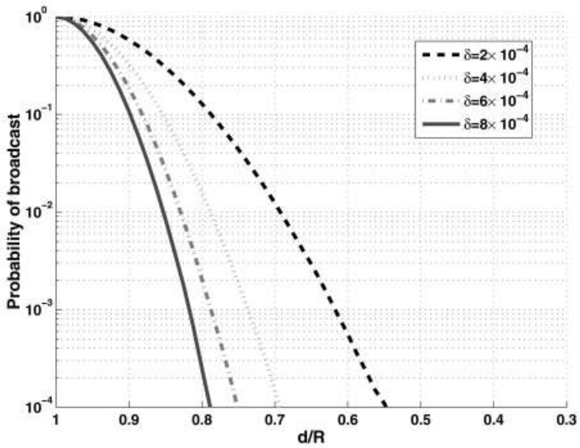


Fig. 11. Probability of broadcast for $R = 400$ m.

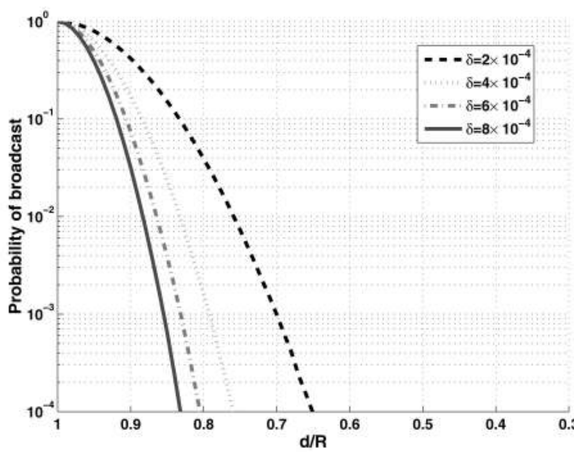


Fig. 12. Probability of broadcast for $R = 500$ m.

and $\delta = 4 \times 10^{-4}$, the probability of broadcast is 0.1 for $d = 250$ m and reduces to 10^{-4} for $d = 200$ m. This property can justify why the proposed receiver-based algorithm can significantly reduce the number of broadcasts in the network. Fig. 14 illustrates an instance of using RBS for the case where $R = 300$ m, $\delta = 4 \times 10^{-4}$, and nodes are placed in a square area of $1{,}000 \times 1{,}000$ m^2. As shown in Fig. 14, only nine nodes (represented by stars) among 400 nodes broadcast the message.

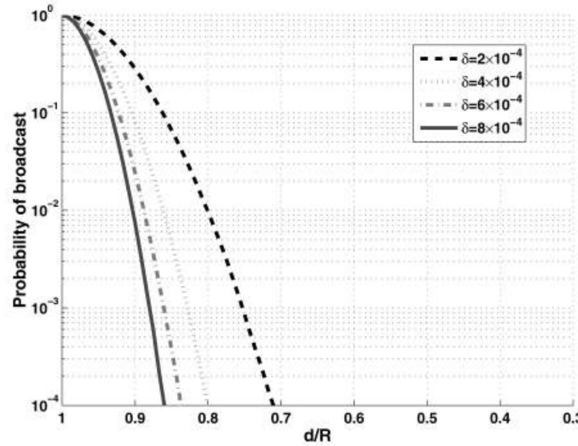


Fig. 13. Probability of broadcast for $R = 600$ m.

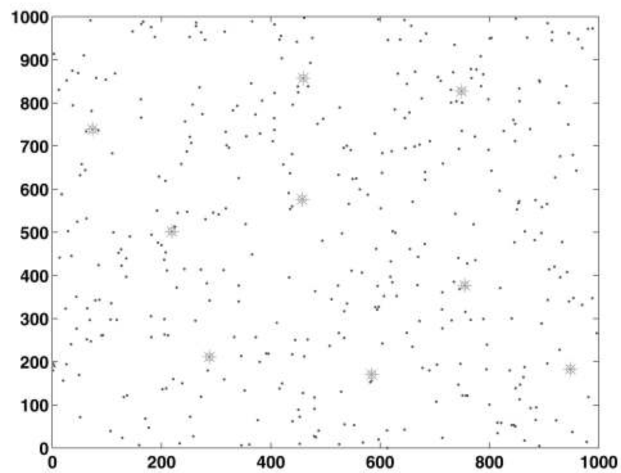


Fig. 14. Broadcasting nodes in a $1{,}000 \times 1{,}000$ m^2 square area with 400 nodes.

TABLE 1
Simulation Parameters

Parameter	Value
Simulator	ns-2 (version 2.27)
MAC Layer	IEEE 802.11
Propagation Model	two-ray ground
Packet Size	256 bytes
Bandwidth	2 Mb/sec
Size of Square Area	$1000 \times 1000\text{m}^2$
Transmission Range	50–300m
Number of Nodes	25–1000

5.3 Performance of Proposed Sender-Based and Receiver-Based Algorithms

The main objective of efficient broadcasting algorithms is to reduce the number of broadcasts. Therefore, we considered the ratio of broadcasting nodes over the total number of nodes as the metric to evaluate the performance of the proposed broadcasting algorithms. Using the ns-2 simulator, we evaluated this metric against two parameters: transmission range and node density. In each simulation run, we uniformly distributed N nodes in a $1{,}000 \times 1{,}000$ m^2 square area, where $N = \delta \times (\text{network area})$. A randomly generated topology was discarded if it led to a disconnected network. Only one broadcasting occurred in each simulation run by a randomly selected node. Table 1

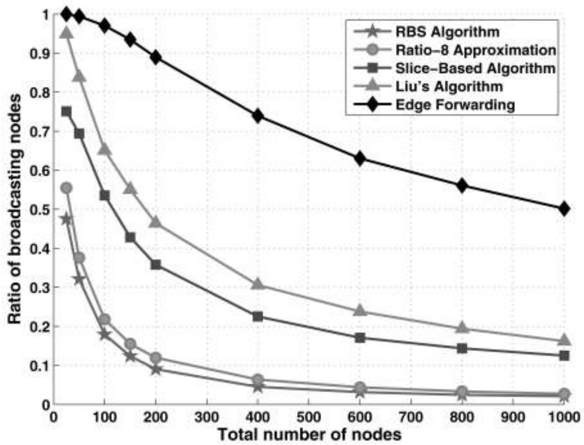


Fig. 15. Ratio of broadcasting nodes versus total number of nodes (uniform distribution).

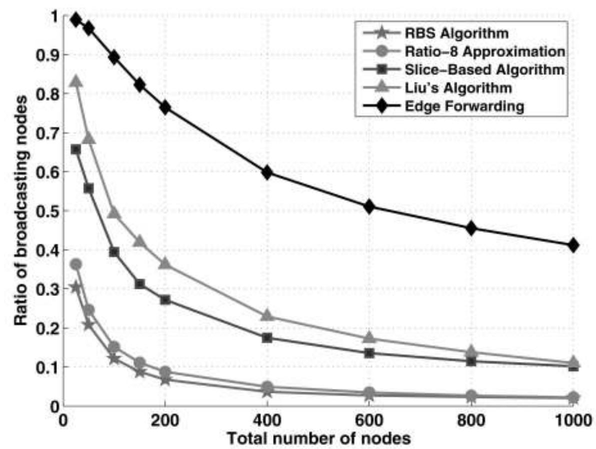


Fig. 17. Ratio of broadcasting nodes versus total number of nodes (Gaussian distribution).

summarizes some of the parameters used in ns-2. As shown in the table, the total number of nodes N varies within 25-1,000, and the transmission range varies within 50-300 m. As a result, the simulation covers very sparse and very dense networks as well as the networks with large diameters.

Figs. 15 and 16 show the average ratio of broadcasting nodes for 100 separate runs. The performance of our proposed algorithm is compared with the performance of Liu et al.'s algorithm [8] and the Edge Forwarding algorithm [22]. Using the Edge Forwarding algorithm, each node divides its transmission coverage into six equal-size sectors and decides whether or not to broadcast based on the existence of forwarding nodes in some overlapped areas. In [8], Liu et al. show that the number of redundant broadcasts using their broadcasting algorithm is significantly lower than that of previous notable broadcasting algorithms [22], [23]. As proved earlier, our proposed sender-based algorithm has lower computational complexity and selects fewer forwarding nodes than Liu et al.'s algorithm. The simulation results, shown in Figs. 15 and 16, indicate two interesting facts. First, our proposed sender-based algorithm does not require more broadcasts than Liu et al.'s proposed broadcasting algorithm. Second, the number of broadcasts using RBS is significantly lower than the number associated with the other implemented algorithms. In fact, as shown in Figs. 15 and 16, this number is even less than one of the best-known approximations for the minimum number of required broadcasts [6]. Note that in RBS, the probability that two close nodes broadcast the same message is very low. As a result, the number of broadcasting nodes is statistically bounded in a finite region (e.g., the transmission range of a node). However, using the slice-based and Liu et al.'s algorithms, the chance that two close nodes broadcast the same message is not negligible. For example, using Liu et al.'s algorithm, a node N_A 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. As expected, most of the nodes around the transmission boundary of N_A will be selected by N_A since they often have contributions in the maximum coverage area of N_A's 1-hop neighbors. Therefore, it is likely for Liu et al.'s algorithm to select two close nodes around the transmission boundary of a node.

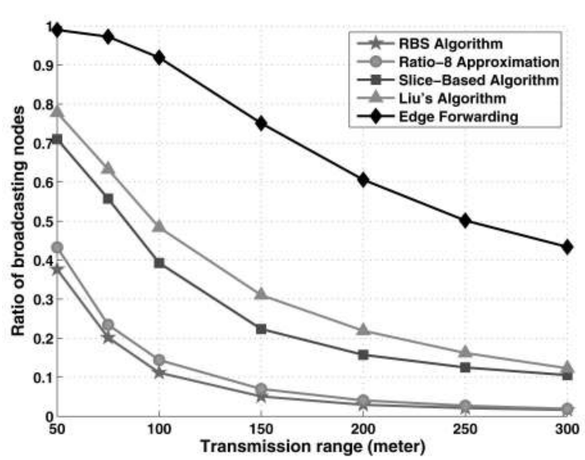


Fig. 16. Ratio of broadcasting nodes versus transmission range (uniform distribution).

We repeated the simulation to consider a few more scenarios. In the first scenario, we changed the node distribution from a uniform to a 2D Gaussian distribution. Both the center and the variance of the Gaussian distribution were randomly selected for each run. The variance was selected from the range 200-400 to avoid a very dense population of nodes around the center of the distribution. As shown in Fig. 17, the number of broadcasting nodes decreases for all the broadcasting algorithms when a Gaussian distribution is used to distribute the nodes in the region. The simulation results indicate that the RBS algorithm still performs significantly better than other broadcasting algorithms considered in this work.

In the second scenario, we used a uniform distribution to distribute the nodes and evaluated the impact of message-reception failure on the performance of broadcasting algorithms. We considered two networks of 100 and 400 nodes. The probability of message-reception failure was assumed to be equal and independent for each node in the network. For both networks, the maximum transmission

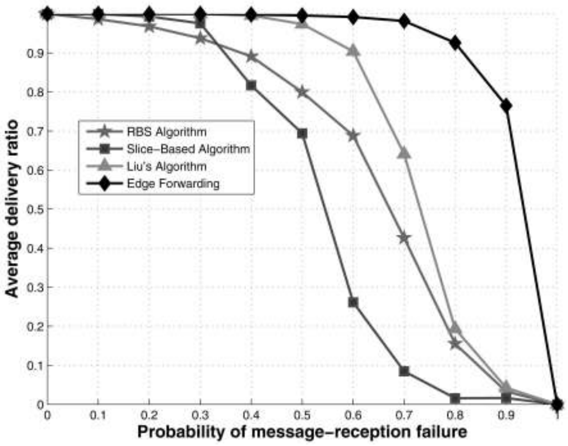


Fig. 18. Average delivery ratio versus probability of message-reception failure $(N = 100)$.

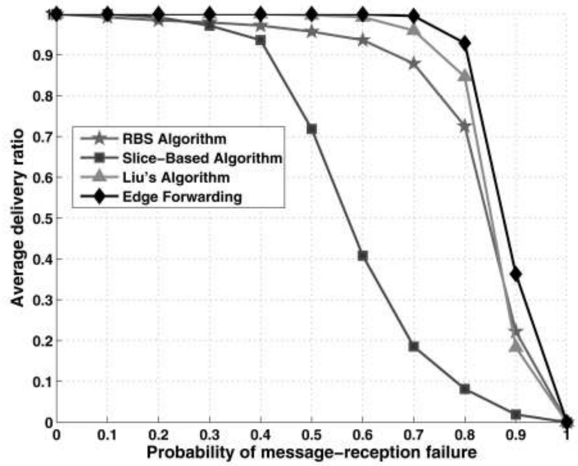


Fig. 19. Average delivery ratio versus probability of message-reception failure $(N = 400)$.

range was set to 250 m. Figs. 18 and 19 compare the average delivery ratio of the broadcasting algorithms for different probabilities of message-reception failure. As shown in these figures, the Edge Forwarding algorithm is the most robust broadcasting algorithm against message-reception failure. This is because the impact of message loss is less when the broadcast redundancy is high. Interestingly, the robustness of the RBS algorithm significantly improves as the node density increases. The simulation results indicate that the slice-based algorithm is the least robust broadcasting algorithm against message-reception failure. This is mainly due to the fact that the slice-based algorithm selects a small number of nodes (less than six on the average) to forward the message. Therefore, when the probability of message-reception failure is high, it is very likely that most of the selected nodes fail to receive and thus forward the message.

Finally, we simulated the broadcasting algorithms in a mobile wireless setting. The nodes were initially distributed using a uniform distribution. In the simulation, we used a random walk mobility model and set the maximum velocity to 10 m/s. We fixed the transmission range to 250 m and varied the total number of nodes within 25-1,000. The simulation results indicate that all the broadcasting algorithms considered in this paper can achieve a high delivery ratio (above 95 percent on the average) for $N \geq 50$, where N is the total number of nodes in the network. This is mainly because the implemented broadcasting algorithms make broadcasting decisions "on the fly." For $N = 25$, the implemented algorithms failed to achieve a high delivery ratio because the network could easily get disconnected due to the nodes' mobility. Clearly, broadcasting algorithms cannot achieve a high delivery ratio in such scenarios. It is worth mentioning that for $N \geq 50$, the ratio of broadcasting nodes is almost the same as the case where there is no mobility (see Fig. 15).

6 Conclusion and Future Work

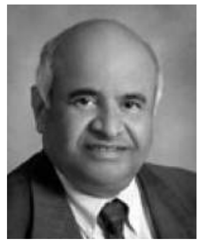
In the first part of this paper, we proposed a forwarding-node selection algorithm that selects at most 11 nodes in $O(n)$, where n is the number of neighbors. This limited number of nodes is an improvement over Liu et al.'s algorithm, which selects n nodes in the worst case and has time complexity $O(n \log n)$. Moreover, we showed that our proposed forwarding-node selection algorithm results in fewer broadcasts in the network. In the second part of the paper, we proposed an efficient receiver-based algorithm and showed why it significantly reduces the number of forwarding nodes in the network. We also relaxed some system model assumptions that are typically used in the broadcasting algorithms. Interestingly, the 2-hop-based version of our proposed receiver-based algorithm can guarantee constant approximation to the optimal solution (minimum CDS). As far as the authors know, this is the first broadcasting algorithm that constructs a CDS "on the fly" and can guarantee both full delivery and a constant approximation ratio to the optimal solution. As part of our future work, we will investigate the necessary conditions to guarantee both full delivery and constant approximation ratio to the minimum CDS.

References

[1] C. Perkins, *Ad Hoc on Demand Distance Vector (AODV) Routing*, IETF Internet draft, work in progress, 1997.
[2] D. Johnson and D. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," *Mobile Computing*, T. Imielinski and H.F. Korth, eds., Kluwer Academic Publishers, 1996.
[3] S. Ni, Y. Tseng, Y. Chen, and J. Sheu, "The Broadcast Storm Problem in a Mobile Ad Hoc Network," *Proc. ACM MobiCom '99*, pp. 151-162, 1999.
[4] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W.H. Freeman, 1990.
[5] B. Clark, C. Colbourn, and D. Johnson, "Unit Disk Graphs," *Discrete Math.*, vol. 86, pp. 165-177, 1990.
[6] P. Wan, K. Alzoubi, and O. Frieder, "Distributed Construction of Connected Dominating Set in Wireless Ad Hoc Networks," *Proc. IEEE INFOCOM '02*, vol. 3, pp. 1597-1604, 2002.
[7] S. Funke, A. Kesselman, U. Meyer, and M. Segal, "A Simple Improved Distributed Algorithm for Minimum CDS in Unit Disk Graphs," *ACM Trans. Sensor Networks*, vol. 2, no. 3, pp. 444-453, 2006.
[8] H. Liu, P. Wan, X. Jia, X. Liu, and F. Yao, "Efficient Flooding Scheme Based on 1-Hop Information in Mobile Ad Hoc Networks," *Proc. IEEE INFOCOM*, 2006.

[9] Y. Tseng, S. Ni, and E. Shih, "Adaptive Approaches to Relieving Broadcast Storms in a Wireless Multihop Mobile Ad Hoc Networks," *Proc. 21st Int'l Conf. Distributed Computing Systems (ICDCS '01),* pp. 481-488, 2001.
[10] Y. Sasson, D. Cavin, and A. Schiper, "Probabilistic Broadcast for Flooding in Wireless Mobile Ad Hoc Networks," *Proc. IEEE Wireless Comm. and Networking Conf. (WCNC '03),* pp. 1124-1130, 2003.
[11] W. Lou and J. Wu, "Double-Covered Broadcast (DCB): A Simple Reliable Broadcast Algorithm in Manets," *Proc. IEEE INFOCOM '04,* pp. 2084-2095, 2004.
[12] J. Wu and F. Dai, "Broadcasting in Ad Hoc Networks Based on Self-Pruning," *Proc. IEEE INFOCOM '03,* pp. 2240-2250, 2003.
[13] W. Peng and X. Lu, "On the Reduction of Broadcast Redundancy in Mobile Ad Hoc Networks," *Proc. ACM MobiHoc '00,* pp. 129-130, 2000.
[14] T. He, C. Huang, B. Blum, J. Stankovic, and T. Abdelzaher, "Range-Free Localization Schemes in Large Scale Sensor Networks," *Proc. ACM MobiCom '03,* pp. 81-95, 2003.
[15] A. Keshavarz-Haddad, V. Ribeiro, and R. Riedi, "DRB and DCCB: Efficient and Robust Dynamic Broadcast for Ad Hoc and Sensor Networks," *Proc. Fourth Ann. IEEE Conf. Sensor, Mesh and Ad Hoc Comm. and Networks (SECON '07),* June 2007.
[16] R. Wattenhofer, L. Li, P. Bahl, and Y. Wang, "Distributed Topology Control for Power Efficient Operation in Multihop Wireless Ad Hoc Networks," *Proc. IEEE INFOCOM '01,* pp. 1388-1397, 2001.
[17] L. Li, J. Halpern, P. Bahl, Y. Wang, and R. Wattenhofer, "A Cone-Based Distributed Topology-Control Algorithm for Wireless Multi-Hop Networks," *IEEE/ACM Trans. Networking,* vol. 13, pp. 147-159, 2005.
[18] M. Sun, W. Feng, and T. Lai, "Broadcasting in Ad Hoc Networks Based on Self-Pruning," *Proc. IEEE Global Telecomm. Conf. (GLOBECOM '01),* pp. 2842-2846, 2001.
[19] A. Papoulis, *Probability and Statistics.* Prentice Hall, 1990.
[20] R. Chang and R. Lee, "On the Average Length of Delaunay Triangulations," *BIT Numerical Math.,* vol. 24, pp. 269-273, 1984.
[21] M. Khabbazian and V.K. Bhargava, "Localized Broadcasting with Guaranteed Delivery and Bounded Transmission Redundancy," *IEEE Trans. Computers,* vol. 57, no. 8, pp. 1072-1086, Mar. 2008.
[22] Y. Cai, K. Hua, and A. Phillips, "Leveraging 1-Hop Neighborhood Knowledge for Efficient Flooding in Wireless Ad Hoc Networks," *Proc. 24th IEEE Int'l Performance, Computing, and Comm. Conf. (IPCCC '05),* pp. 347-354, 2005.
[23] J. Wu and H. Li, "On Calculating Connected Dominating Set for Efficient Routing in Ad Hoc Wireless Networks," *Proc. Third Int'l Workshop Discrete Algorithms and Methods for Mobile Computing and Comm. (DIAL-M '99),* pp. 7-14, 1999.

Majid Khabbazian received the BSc degree in computer engineering from the Sharif University of Technology, Tehran, Iran, in 2002 and the MASc degree in electrical and computer engineering from the University of Victoria, British Columbia, Canada, in 2004. He is currently a PhD candidate in the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, British Columbia, Canada, where he is a university graduate fellowship holder. He was a university graduate fellowship recipient at the University of Victoria. His current research interests include cryptography and wireless network security. He is a student member of the IEEE.



Vijay K. Bhargava received the BSc, MSc, and PhD degrees from Queen's University, Kingston, Ontario, Canada in 1970, 1972, and 1974, respectively. Currently, he is a professor and the head of the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, British Columbia, Canada. He was with the University of Victoria from 1984 to 2003 and with Concordia University, Montreal, from 1976 to 1984. He is a coauthor of the book *Digital Communications by Satellite* (John Wiley & Sons, 1981), a coeditor of *Reed-Solomon Codes and Their Applications* (IEEE, 1994), and a coeditor of *Communications, Information and Network Security* (Kluwer Academic Publishers, 2003). His research interests are in wireless communications. He is very active in the IEEE and was nominated by the IEEE Board of Directors for the office of IEEE President-Elect. He has served on the Board of the IEEE Information Theory Society and the IEEE Communications Society. He is a past president of the IEEE Information Theory Society. He is the editor-in-chief for the *IEEE Transactions on Wireless Communications*. He received the IEEE Centennial Medal in 1984, the IEEE Canada's McNaughton Gold Medal in 1995, the IEEE Haraden Pratt Award in 1999, the IEEE Third Millennium Medal in 2000, the IEEE Graduate Teaching Award in 2002, and the Eadie Medal of the Royal Society of Canada in 2004. He is a fellow of the IEEE, the Engineering Institute of Canada (EIC), the Canadian Academy of Engineering, and the Royal Society of Canada.