Minimizing Broadcast Latency in Ad Hoc Wireless Networks
Shankar M Banik
The Citadel
171 Moultrie Street
Charleston, SC 29409
1-843-953-5039
shankar.banik@citadel.edu

Sridhar Radhakrishnan
University of Oklahoma
200 Felgar Street
Norman, OK 73019
1-405-325-1867
sridhar@ou.edu

ABSTRACT
Network wide broadcasting in ad-hoc wireless networks provides important control and route establishment functionality for a number of unicast and multicast protocols. In broadcasting, a source node sends the same message to all the nodes in the network. Broadcasting based on Minimum Connected Dominating Set (MCDS) is a frequently used approach to reduce the communication overhead. A set of connected nodes in a graph is a connected dominating set, if all the nodes in the graph are either in the set or neighbors of nodes in the set. MCDS is such a set with minimum cardinality. Several broadcasting algorithms in ad-hoc wireless networks based on neighborhood information use MCDSs. This approach minimizes the total number of transmissions in broadcasting process but it does not guarantee minimum broadcast latency. Minimum latency broadcast is NP-hard for ad-hoc wireless network [1]. A proper collision-free broadcast scheduling among nodes in MCDS will minimize the broadcast latency. In this paper, we propose a collision-free broadcast scheduling algorithm to minimize the broadcast latency for a given network with the number of transmissions equal to cardinality of MCDS.

1. INTRODUCTION
Ad-hoc wireless networks are a set of wireless nodes which communicate with each other without any fixed infrastructure. In such a network, each node communicates with other nodes either directly or with the help of other intermediate nodes. This kind of network is used in battlefields and disaster recovery situations.

The main idea of broadcasting is to send the same message from a source to all the nodes in the network. In wireless ad-hoc network, when two or more nodes transmit a message at the same time to a common node which is neighbor to both of them, a collision occurs at the common node. As a result some nodes may need to retransmit their messages. In this type of network, the source node may need the assistance of intermediate nodes to reach the remote nodes. So the retransmission of messages is more expensive in this situation, as it consumes power of all the intermediate nodes. Proper selection of intermediate nodes is necessary in order to assure a collision-free broadcast. Another important parameter in broadcasting is broadcast latency which is defined by the time taken by a message to reach all the nodes in the network.

We use a graph $G = (V, E)$ to model an ad-hoc wireless network where $V$ represents the set of wireless nodes and $E$ represents the set of edges between the wireless nodes. We define an edge $e \in E$ between two nodes $a, b \in V$, if both $a$ and $b$ are within their wireless transmission ranges. To simplify our model, we can assume that all the nodes in $V$ have the same transmission range. This implies that if there is any edge $e \in E$ connecting two nodes $a, b \in V$, then $a$ is within $b$‘s range and $b$ is within $a$‘s range.

A minimum connected dominating set for a graph is a set of connected nodes where a node in the graph is either in the MCDS or is adjacent to at least one node in MCDS. Finding an MCDS for an arbitrary graph is known to be NP-hard [2] and approximation algorithms are proposed in [3]. Several papers on broadcasting [4], [5] in ad hoc networks propose heuristic algorithms to find MCDS. They try to minimize the number of transmissions in broadcasts using the MCDSs. But minimizing broadcast latency has not been considered as performance criteria in the above mentioned papers. In [1], Gandhi et al. showed that minimum latency broadcast is NP-hard for wireless ad-hoc networks. In this paper, we show that proper scheduling among nodes in MCDS for broadcasting is necessary to minimize the broadcast latency. We also provide a collision-free broadcast scheduling algorithm that minimizes broadcast latency.

2. BROADCAST SCHEDULING
Broadcasting using MCDS reduces communication overhead since only the nodes in MCDS are allowed to retransmit. But proper collision-free scheduling among the nodes in MCDS is necessary to ensure minimum latency broadcast. We illustrate the
importance of scheduling using an example network in Figure 1. Time required to broadcast a message by a node to all its 1-hop neighbors is same for every node in the network. We consider this time as a cycle. In broadcasting process, all 1-hop neighbors of the source are covered at the end of first cycle. If the diameter of network from the source node is \( n \), then the lower bound on total time for broadcasting would be \( n \) cycles.

![Graph G = (V, E) representing an ad-hoc network](image)

Figure 1: A graph \( G = (V, E) \) representing an ad-hoc network

Assume that node-3 in Figure 1 has some data to broadcast. From node-3 one of the MCDS is \( \{3, 4, 6, 9, 12\} \). So that total number of transmissions is 5. The diameter of the network from the source node-3 is 4. So the lower bound for total time to complete broadcasting process is 4 cycles. With the above MCDS, we get the following two broadcasting schedules \( S_1 \) and \( S_2 \) among MCDS nodes.

<table>
<thead>
<tr>
<th>Schedule ( S_1 )</th>
<th>Schedule ( S_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) 3 ( \rightarrow ) 1, 2, 4, 5, 6</td>
<td>(1) 3 ( \rightarrow ) 1, 2, 4, 5, 6</td>
</tr>
<tr>
<td>(2) 6 ( \rightarrow ) 8, 7</td>
<td>(2) 4 ( \rightarrow ) 8, 9, 10</td>
</tr>
<tr>
<td>(3) 4 ( \rightarrow ) 9, 10</td>
<td>(3) 6 ( \rightarrow ) 7, 9 ( \rightarrow ) 11, 12</td>
</tr>
<tr>
<td>(4) 9 ( \rightarrow ) 11, 12</td>
<td>(4) 12 ( \rightarrow ) 13, 14</td>
</tr>
</tbody>
</table>

Considering schedule \( S_1 \), the total number of transmissions is 5 and total time for broadcasting is 5 cycles. During the first cycle, source node-3 transmits the broadcast message. After cycle 1, all 1-hop neighbors of node-3 receive the message. Now node-4 and node-6 are ready to rebroadcast in the next cycle. As node-4 and node-6 are neighbors, only one node is allowed to transmit due to CSMA/CA nature of underlying MAC protocol. Node-6 is chosen to transmit the message. After cycle 2, only node-4 is contending for channel and transmits the message in cycle 3. In the same way node-9 and node-12 rebroadcasts in cycle 4 and cycle 5 respectively. In schedule \( S_2 \), after cycle 1, node-4 is selected to rebroadcast instead of node-6. At this stage, node-6 and node-9 are contending for channel. As they are not neighbors both of them can transmit in the same cycle. In cycle 3, both node-6 and node-9 rebroadcast the message. Transmission by node-12 in cycle 4 completes the broadcasting process. We see that in both the schedules total number of transmissions is 5, but total time in schedule \( S_1 \) is 5 cycles whereas the total time in schedule \( S_2 \) is 4 cycles. Proper scheduling among MCDS nodes for rebroadcasting is important for minimizing the total broadcast latency.

The lower bound for the number of transmissions in broadcasting is the cardinality of MCDS. By increasing the number of simultaneous transmissions, we can reduce the broadcasting time. In wireless networks, several nodes can transmit simultaneously if they are not neighbors or if they do not have any common neighbors. Set of contending nodes is a set of nodes in a network, which are trying to acquire channel at a particular instance of time. This set of contending nodes may contain several sets of conflicting nodes. Set of conflicting nodes is a subset of nodes to the set of contending nodes, which are either neighbors or have uncovered common neighbors from previous broadcasts.

CSMA/CA nature of underlying MAC protocol allows only one node from a set of conflicting nodes to transmit at any particular time. Selection of nodes from contending nodes should be such that it increases the number of simultaneous transmissions in next cycles of broadcasting process. In this paper, we propose an algorithm based on greedy strategy which always tries to minimize the broadcasting time by increasing simultaneous transmissions in the network.

**Broadcast Scheduling Algorithm (BSA)**

- **Input:** \( U \): [Set of nodes which can transmit simultaneously]
- **Output:** \( V \): [Set of nodes which can transmit simultaneously]

**Step 1:** Select a node \( p \) \( \in U \) such that \( |E_p| \) is maximum.

- If there are nodes \( m_1, m_2, \ldots, m_k \) \( \in U \) such that
  
  \[
  |E_{m_i}| = |E_{m_2}| = \ldots = |E_{m_k}|
  \]

  Let \( t = \text{Max} \{\text{Depth}(j) | j = m_1, m_2, \ldots, m_k\} \).

  Select the node \( p \) \( \in U \) corresponding to \( t \).

**Step 2:** \( V = V \cup \{p\}, U = U - \{p\} \), \( UC = UC - V \).

**Step 3:** If the set \( U \) is not empty go to step 1.

At the end, this algorithm returns a set of nodes \( V \) which can transmit simultaneously. During the selection process, BSA first tries to reach maximum number of MCDS nodes such that the number of nodes contending for the channel in next cycle will be high which in turn increases the probability of simultaneous transmissions. If contending nodes reach same number of uncovered nodes in MCDS, BSA selects a node \( p \) such that the depth of the sub-tree of BFST from the source node \( s \) through node \( p \) is maximum, which in turn ensures that it increases the number of simultaneous transmission in the system.

### 3. REFERENCE