Minimizing Broadcast Latency and Redundancy in Ad Hoc Networks

Rajiv Gandhi*
Dept. of Computer Science
University of Maryland
College Park, MD 20742
gandhi@cs.umd.edu

Srinivasan Parthasarathy†
Dept. of Computer Science
University of Maryland
College Park, MD 20742
sri@cs.umd.edu

Arunesh Mishra‡
Dept. of Computer Science
University of Maryland
College Park, MD 20742
arunesh@cs.umd.edu

ABSTRACT

Network wide broadcasting is a fundamental operation in ad hoc networks. In broadcasting, a source node sends a message to all the other nodes in the network. In this paper, we consider the problem of collision-free broadcasting in ad hoc wireless networks. Our objective is to minimize the latency and the number of retransmissions in the broadcast. We show that minimum latency broadcasting is NP-hard for ad hoc wireless networks. We also present a simple and distributed collision-free broadcasting algorithm for broadcasting a message. For networks with bounded node transmission ranges, our algorithm *simultaneously* guarantees that the latency and the number of retransmissions are within $O(1)$ times their respective optimal values. Our algorithm and analysis extends to the case when multiple messages are broadcast from multiple sources. Experimental studies indicate that our algorithms perform much better in practice than the analytical guarantees provided for the worst case.

Keywords

Ad Hoc Networks, Broadcasting, Approximation Algorithms

1. INTRODUCTION

Ad hoc networks are a set of wireless mobile nodes which communicate with one another. Nodes in these networks do not rely on any pre-existing routing infrastructure for communication, but instead communicate either directly or with the help of other intermediate nodes in the network. The distributed, wireless and self-configuring nature of ad hoc networks make them suitable for a wide variety of applications. At the same time, these characteristics also introduce several challenging and interesting research issues in the design of communication protocols for these networks.

*Research supported by NSF Award CCR-9820965.

†Research supported by NSF Award CCR-0208005.

‡Research supported by a NIST grant.

Network wide broadcasting is a fundamental operation in ad hoc networks. Its goal is to transmit a message from a source to all the other nodes in the network. Several ad hoc network protocols assume the availability of an underlying broadcast service. Applications which make use of broadcasting include LAN Emulation, host paging, sending an alarm, and establishing unicast routes [22]. Broadcasting is also a common operation in many distributed computing applications and can be used for service or resource discovery in unstructured environments. In the absence of other mechanisms, broadcast also serves as a last resort for other group communication operations such as multicast.

Any communication protocol for ad hoc networks should contend with the issue of interference in the wireless medium. When two or more nodes transmit a message to a common neighbor at the same time, the common node will not receive any of these messages. In such a case, we say that a collision has occurred at the common node. In multi-hop ad hoc networks where all the nodes may not be within the transmission range of the source, intermediate nodes may need to assist in the broadcast operation by retransmitting the message to other remote nodes in the network. Retransmissions use up valuable resources in the network such as power and bandwidth. Hence, it is important to choose the intermediate nodes carefully so as to avoid redundancy in retransmissions. Another metric of interest in broadcast protocols is the end-to-end broadcast latency which is the time taken by the message to reach all the nodes in the network.

One of the earliest broadcast mechanisms proposed in the literature is flooding [15, 17], where every node in the network retransmits a message to its neighbors after receiving it. Although flooding is extremely simple and easy to implement, it was shown in [22] that flooding can be very costly and can lead to serious redundancy, bandwidth contention and collision: a situation known as *broadcast storm*. Since then, a lot of research has been directed towards designing broadcast protocols which are collision-free and which reduce redundancy by reducing the number of retransmissions.

Recently, there has been a tremendous interest for supporting real-time multimedia traffic over ad hoc networks [31]. Ad hoc sensor networks are also expected to play a big role in military communications, disaster relief and rescue operations [21]. These applications impose stringent end-to-end latency requirements on the underlying protocols. In lieu of

these real-time applications, it is necessary to have broadcast protocols which can meet the combined requirements of collision-free delivery, low end-to-end latency and low redundancy. To the best of our knowledge, none of the existing broadcast protocols address all these issues together.

In this paper, we study collision-free broadcasting in ad hoc networks. We provide algorithms which guarantee that all nodes in the network receive the broadcast messages without collision. Our algorithms simultaneously guarantee low broadcast latency and low number of retransmissions. The following is a summary of our contributions in this paper.

1.1 Our Contributions

- We show that the problem of minimum latency broadcasting in ad hoc networks is NP-hard. This implies that there are no efficient broadcast algorithms which can minimize the latency of broadcast in ad hoc networks, unless P=NP.
- For networks with bounded node transmission ranges, we present a *collision-free* broadcast algorithm which *simultaneously* produces *provably* good solutions in terms of latency and the number of retransmissions. In particular, both the latency and the number of retransmissions of our algorithm are within $O(1)$ times their respective optimal values.
- We extend our algorithm and analysis to the case when multiple messages are broadcast from multiple sources, in networks with uniform node transmission ranges.
- We study the performance of our algorithms under various network conditions through simulations.

Our algorithms and analysis make use of certain simple but powerful geometric properties of wireless networks. All our approximation factors are independent of network characteristics. In particular, they are independent of the number of network nodes and the maximum degree of nodes in the network. Our algorithms are simple and have easy distributed implementations. Experimental studies indicate that their performance is much better in practice on typical inputs than the analytical guarantees we provide for the worst case.

The remainder of this paper is organized as follows: We review related work in the next section. In Section 3, we present the details of our model and the formal problem statement. In Section 4, we show that minimum latency broadcasting in ad hoc networks is NP-hard. In Section 5, we present and analyze our broadcast algorithms and extend it to the case when multiple messages are broadcast from multiple sources. In Section 6, we discuss the distributed implementation of our algorithm. In Section 7, we evaluate the performance of our algorithms experimentally through simulations. Section 8 contains conclusions and directions for future work.

2. RELATED WORK

As noted earlier, flooding is one of the earliest protocols for multicasting and broadcasting in ad hoc networks [15, 17]. In flooding, every node in the network retransmits the message to its neighbors after receiving it. Ni *et al.*[22] study the flooding protocol analytically and experimentally and show that it can lead to severe contention, collision and redundant retransmissions: a situation referred to as broadcast storm. They also propose neighbor-knowledge and coverage based heuristics to determine which nodes should retransmit the message. Several other heuristics have been proposed where nodes make use of their neighborhood information to determine if they need to retransmit a message [19, 25, 26, 24, 28, 23]. Although the goal of all the above protocols is to minimize the number of retransmissions, none of them guarantee any bounds for their solution with respect to the optimal number of retransmissions.

In a series of papers [27, 12, 11], it was proposed that a connected dominating set (CDS) can be used as a virtual backbone for routing in ad hoc networks. A dominating set in a graph $G = (V, E)$ is defined as a set of vertices $V' \subseteq V$, such that every node in $V - V'$ is adjacent to some node in V'. A connected dominating set (CDS) is a dominating set whose induced subgraph is connected. An $MCDS$ is a CDS of minimum size. The notion of CDS occurs naturally in broadcast protocols since the set of retransmitting nodes in any broadcast protocol forms a CDS. Minimizing the number of retransmissions directly translates to finding an $MCDS$ in a graph.

Guha and Khuller [14] studied $MCDS$ in general graphs and showed that computing $MCDS$ is NP-hard. They also gave two greedy approximation algorithms for this problem. In [10], it was shown that computing $MCDS$ is NP-hard even for unit disk graphs ($UDGs$). $UDGs$ model ad hoc networks where all the node transmission ranges are uniform. Several researchers have addressed the issue of efficient approximation algorithms for $MCDS$ for $UDGs$ [29, 5, 20]. Although these results lead to efficient broadcast algorithms in terms of the number of retransmissions, it does not directly lead to low broadcast latency.

There has been relatively less research directed towards low latency broadcasting in ad hoc networks. However, some results exist in radio network literature whose models are essentially the same as ours. In particular, Chlamtac and Kutten [7] study the complexity of minimum latency broadcast scheduling with interference and show that the problem is NP-hard for general graphs. Recently, Elkin and Kortsarz [13] showed a logarithmic hardness of approximation for the same problem. However, these results do not directly extend to ad hoc networks since ad hoc networks are modeled by a very restricted class of geometric graphs called disk graphs. In [8], Chlamtac and Kutten gave an algorithm for constructing a broadcast tree and for collision-free scheduling of a message along this broadcast tree. They proved that for arbitrary graphs, the broadcast latency of their schedule is within $O(\ln(N/r)^2)$ times the optimal, where N is the number of network nodes and r is the graph theoretic radius of the network. For ad hoc networks with bounded node transmission ranges, we improve upon these results in this paper, by showing the NP-hardness for disk graphs and by presenting algorithms with broadcast latency and number of retransmissions which are within $O(1)$ times their optimal values.

Basagni *et al.*[3] present a mobility transparent broadcast

scheme for mobile multi-hop radio networks. In their scheme, nodes compute their transmit times once and for all in the beginning. They provide two schemes with bounded latency. These schemes have approximation factors which are linear and polylogarithmic in the number of network nodes. In effect, they assume that the topology of the network is completely unknown. Although, their schemes are extremely attractive for highly mobile environments, their approximation factors are far from what is achievable in static and relatively less mobile environments where the broadcast tree and broadcast schedule can be computed efficiently. In addition, they do not bound the number of retransmissions in their schemes. Other results which deal with broadcasting in unknown topologies include [6, 9].

Hung *et al.*[16] provide centralized and distributed deterministic algorithms for broadcasting and experimentally study the efficiency of their algorithms with respect to collision-free delivery, number of retransmissions and broadcast latency. While their centralized algorithm is guaranteed to be collision-free, their distributed algorithm is not. They also do not provide any guarantees with respect to the number of retransmissions and latency of the broadcast schedule. For an excellent survey of many of the above mentioned protocols, the reader is referred to the survey by Williams and Camp [30]. This survey provides a neat characterization and experimental evaluation of many of these protocols under a wide range of network conditions.

Finally, we mention another area of research which is related to that of broadcast scheduling: broadcast time in radio networks. This area investigates the actual amount of time required by a broadcast message to reach all the network nodes under the same interference model as that of ours. However, the computation of a broadcast schedule before a message originates is disallowed. The reader is referred to [2, 1, 4, 18] for more details. To the best of our knowledge, these results model the network as an arbitrary graph which is not an accurate model for ad hoc networks. These results may not be applicable for scenarios where it is possible to schedule the broadcasting of a message in advance and where the cost of broadcast schedule computation can be amortized over multiple messages.

3. PRELIMINARIES

3.1 Network Model

We model an ad hoc network using a directed graph $G = (V, E)$. The nodes in V are embedded in the plane. Each node $u \in V$ has a transmission range, $range(u) \in [r_{min}, r_{max}]$. Let $d(u, v)$ denote the Euclidean distance between u and v. An arc $(u, v) \in E$ iff v is in the transmission range of u, i.e., $d(u, v) \leq range(u)$. Such graphs are called Disk Graphs. When the node transmission ranges are uniform, G is called a Unit Disk Graph (UDG). We say that an edge (u, v) is uni-directional if $(u, v) \in E$ and $(v, u) \notin E$. We say that an edge (u, v) is bi-directional if both $(u, v) \in E$ and $(v, u) \in E$. A UDG can be treated as an undirected graph since all its edges are bi-directional.

We assume that time is discrete. Since the medium of transmission is wireless; whenever a node transmits a message, all its out-neighbors hear the message. We say that there is a collision at node w, if w hears a message from two transmitters at the same time. In such a case, we also say that the two transmissions interfere. A node w receives a message collision-free iff w hears the message without any collision.

3.2 Problem Statement

We are given a disk graph $G = (V, E)$ and a set of messages $M = \{1, 2, \ldots, m\}$. We are also given a set of sources for these messages: $sources = \{s_j | s_j \text{ is the source of message } j\}$. A node can transmit message j only after it receives message j collision-free. A broadcast *schedule* specifies, for each message j and each node i, the time at which node i receives message j collision-free and the time at which it transmits message j. If a node does not transmit a message then its transmit time for that message is 0. The latency of the broadcast schedule is the first time at which every node receives all messages. The number of retransmissions is the total number of times every node transmits any message. Our goal is to compute a broadcast schedule in which the latency and the number of retransmissions are minimized.

3.3 Background: Approximation Algorithms

Many interesting optimization problems are NP-hard. Computing optimal solutions to such problems in polynomial time is not possible unless $P = NP$. A common approach to NP-hard problems is to develop *approximation algorithms*, which have polynomial running times and which produce provably good solutions with costs close to the optimal.

Definition. An α-*approximation* algorithm for a minimization problem Π is an algorithm that runs in polynomial time and for every instance I of Π it produces a solution of cost at most $\alpha \mathrm{OPT}(I)$, $\alpha > 1$, where $\mathrm{OPT}(I)$ refers to cost of an optimal solution for instance I. In this case, we also say that the approximation algorithm has an approximation factor of α.

Notice that in the definition, we compare the cost of our solution with the cost of an optimal solution. But finding the cost of an optimal solution is NP-hard. In order to establish the approximation guarantee, we must find in polynomial time, a *lower bound* on the cost of an optimal solution. Lower bounds closer to the optimal cost yield better approximation guarantee.

4. NP-HARDNESS

We prove that minimum latency broadcast in ad hoc networks is NP-hard via a reduction from 3-SAT. This reduction is similar to the reduction from 3-SAT for general graphs. However, the ad hoc network reduction is complicated by the fact that its nodes are to be embedded on the plane and that its arcs be consistent with the transmission ranges of the nodes. Due to lack of space, we do not give the reduction here but state the following theorem without proof.

THEOREM 4.1. *Minimum latency broadcast problem in ad hoc networks is NP-hard.*

```
BroadcastTree(G = (V, E), s)
1   P ← {s} // P is the set of primary nodes.
2   T_BFS ← BFS tree in G with root s
3   l ← maximum number of levels in T_BFS
4   for i ← 2 to l do
5       L_i ← set of all nodes at level i in T_BFS
6       for each w ∈ L_i do
7           if (P ∩ N_i(w) = ∅) then
8               P ← P ∪ {w}
9               parent(w) ← L_{i-1} ∩ N_i(w)
10          else // w is a secondary node
11              parent(w) ← any node in P that dominates w
12  for each u ∈ V do
13      Children(u) ← {w|parent(w) = u}
14  V_broadcast ← V
15  E_broadcast ← {(u, w)|u = parent(w)}
16  return T_broadcast = (V_broadcast, E_broadcast)
```

Figure 1: Algorithm for constructing the broadcast tree

```
ScheduleBroadcasts(T_broadcast, s)
1   for each node u ∈ T_broadcast do
2       transmitTime(u) ← 0
3   Transmitters ← {s}
4   while (Transmitters ≠ ∅) do
5       u ← any node in Transmitters
6       I_1(u) ← {t|∃w ∈ Children(u) that hears a message at time t}
7       I_2(u) ← {t|∃w ∈ N_o(u) that receives a message collision-free at time t}
8       I(u) ← I_1(u) ∪ I_2(u)
9       transmitTime(u) ← min{t|t > receiveTime(u) and t ∉ I(u)}
10      for each w ∈ Children(u) do
11          receiveTime(w) ← transmitTime(u)
12      Transmitters ← Transmitters \ {u}
13      Transmitters ← Transmitters ∪ {w|w ∈ Children(u) and Children(w) ≠ ∅}
14  return transmitTime
```

Figure 2: Algorithm for scheduling the broadcasts

```
BroadcastMessage(G = (V, E), s)
1   T_broadcast ← BroadcastTree(G, s)
2   ScheduleBroadcasts(T_broadcast, s)
```

Figure 3: Algorithm for broadcasting a single message from source s

5. BROADCAST ALGORITHMS

We first present an algorithm for computing the broadcast schedule for a single message. We then extend the algorithm for scheduling multiple messages from the same source. Finally we present the algorithms for scheduling multiple messages from multiple sources.

5.1 Scheduling Broadcasts for a Single Message

The algorithm in Figure 3 takes as input a directed graph $G = (V, E)$ and a source node s. Let $N_i(u)$ and $N_o(u)$ denote the set of in-neighbors and out-neighbors of a node $u \in V$. The algorithm consists of two parts – (i) constructing a *broadcast tree*, $T_{broadcast}$, rooted at s in which if a node $u \in V$ is a parent of a node $w \in V$ then u is responsible for transmitting the message to w without any collision at w (pseudo-code in Figure 1), (ii) scheduling the transmit times for all nodes in V such that every node receives the message collision-free (pseudo-code in Figure 2).

The broadcast tree $T_{broadcast}$, is constructed as follows. The set of nodes V, is partitioned into a set of *primary nodes*, P, and a set of *secondary nodes* S. The set of primary nodes P forms a dominating set in G, i.e., each node in S is in the transmission range of some node in P. P is constructed by considering each level of the BFS tree in order. Initially, P is empty. Let $L_j, j = 1, 2, \ldots, l$, be the set of nodes at level j in the BFS tree rooted at source s. Let w be any node in L_i. Node w is added to P if and only if P constructed so far does not dominate w, otherwise $w \in S$. For each node $u \in V$, its parent, $parent(u)$, in $T_{broadcast}$, is determined as follows. Let $P_i = P \cap L_i$ and $S_i = S \cap L_i$ be the set of primary nodes and secondary nodes respectively at level i in the BFS tree. If $u \in P_i$ then $parent(u)$ is any one of its in-neighbors in L_{i-1}. If $u \in S_i$ then $parent(u)$ is any one of it's in-neighbors in $P_{i-1} \cup P_i$. All nodes whose parent is u constitute the set $Children(u)$.

The transmissions are scheduled following a *greedy* strategy. Note that the nodes that retransmit the message are the internal (non-leaf) nodes in $T_{broadcast}$. Any retransmitting node, u, retransmits at the minimum time t that satisfies the following three constraints – (i) u has received the message collision-free before time t, (ii) no node in $Children(u)$ is hearing any transmissions at time t, (iii) no node in $N_o(u) \setminus Children(u)$ is receiving the message collision-free at time t, where $N_o(u)$ is the set of out-neighbors of u.

5.2 Analysis

Lemmas 5.1, 5.2, and 5.3 represent some useful properties of our algorithm.

LEMMA 5.1. *For any two primary nodes u and w in P, $d(u, w) > r_{min}$.*

PROOF. Without loss of generality, assume that u was chosen in P before w. If $d(u, w) \leq r_{min}$ then w would be dominated by u and would not have been chosen in P by our algorithm. □

LEMMA 5.2. *If $(u, w) \in E_{broadcast}$ then both u and w can not be secondary nodes, i.e. $\{u, w\} \cap P \geq 1$.*

PROOF. Without loss of generality, let $u = parent(w)$. If $w \in P$ then the claim is true. If w is a secondary node then w chooses a primary node as it's parent in line 16 of the pseudo-code BROADCASTTREE in Figure 1. Hence, u must be a primary node. □

LEMMA 5.3. *In our algorithm every node retransmits at most once.*

PROOF. Consider any node $v \in V$. If v is a leaf in $T_{broadcast}$ then $Children(v) = \emptyset$ and v is never included in the list $Transmitters$ (line 13 in Figure 2). Hence, v never retransmits. If v is an internal node in $T_{broadcast}$ then it retransmits exactly once at $transmitTime(v)$. Node v is then removed from $Transmitters$ (line 12 in Figure 2). □

LEMMA 5.4. *Consider a disk D, of radius $r \geq r_{min}$. If $Primaries(r)$ represent the set of primary nodes within D then $|Primaries(r)| \leq k_1 r^2 / r_{min}^2$, where k_1 is a constant.*

PROOF. Let c be the center of the disk D. For each primary node $u \in Primaries(r)$, consider a disk of radius $r_{min}/2$ centered at u. Let H denote the set of all such disks. Thus, $|Primaries(r)| = |H|$. By Lemma 5.1, no two primary nodes are within distance r_{min} from each other. Thus, H is a set of non-intersecting disks. Each primary node in D is at most a distance of r from c. Any point on a disk in H is at a distance of at most $r + (r_{min}/2)$ from c. The size of H is upper-bounded by the number of non-intersecting disks of radius $r_{min}/2$ whose areas are completely contained within a disk of radius $r + (r_{min}/2)$. Hence we have

$$\begin{aligned} |H| &\leq \frac{\pi(r + \frac{r_{min}}{2})^2}{\pi(\frac{r_{min}}{2})^2} \\ &= \left(\frac{2r}{r_{min}} + 1\right)^2 \\ &\leq \left(\frac{3r}{r_{min}}\right)^2 \end{aligned}$$

□

LEMMA 5.5. *Consider an internal node u in $T_{broadcast}$ that has received the message collision-free. Recall that $I(u)$ (line 8 in SCHEDULEBROADCASTS) is the set of interference times that u must avoid while choosing $transmitTime(u)$. $|I(u)| \leq 2\,|Primaries(3r_{max})| \leq k_2 r_{max}^2 / r_{min}^2$, where k_2 is a constant.*

PROOF. Let us represent the transmission range of any node u by a disk of radius $range(u) \leq r_{max}$. Let w be a transmitter such that u and w share a common out-neighbor and $transmitTime(w) \in I(u)$. Thus, $d(u, w) \leq 2r_{max}$. Let $B(u)$ be the set of all such nodes w. Since each node retransmits at most once, $|I(u)| \leq |B(u)|$. We will now upper-bound $|B(u)|$. Nodes in $B(u)$ are either primary nodes, $B^p(u)$, or secondary nodes, $B^s(u)$. Let us first bound $|B^p(u)|$.

Note that $|B^p(u)| \leq |Primaries(2r_{max})|$, where we can obtain $|Primaries(2r_{max})|$ from Lemma 5.4. We will now bound $|B^s(u)|$. Let $v \in B^s(u)$. By Lemma 5.2, every child of v in $T_{broadcast}$ is a primary node. These primary nodes are at a distance of at most $3r_{max}$ from u. Thus, $|B^s(u)| \leq |Primaries(3r_{max})|$. Using Lemma 5.4, we get $|B^p(u)| + |B^s(u)| \leq 2\,|Primaries(3r_{max})| \leq k_2 r_{max}^2/r_{min}^2$, for some constant k_2. □

5.2.1 Latency Bound

Recall that l is the maximum number of levels in the BFS tree. Let OPT_{lat} be the latency in an optimal solution. Note that $l \leq OPT_{lat}$.

THEOREM 5.6. *Algorithm* BROADCASTMESSAGE *gives an approximation guarantee of* $O(r_{max}^2/r_{min}^2)$.

PROOF. Recall that for any node u, $receiveTime(u)$ is the time when u receives the message collision-free from its parent in $T_{broadcast}$. By induction on the number of levels in the BFS tree, we will show that for any $i = 0, 1, \ldots, l$, and for some constant k,

$$\forall v \in L_i, receiveTime(v) \leq t_i = k(r_{max}^2/r_{min}^2)i \qquad (1)$$

We then substitute $i = l$ and use $l \leq OPT_{lat}$ to prove the theorem.
Base Case: $i = 0$. In this case, expression (1) is trivially true since $L_i = \{s\}$ and by definition, s receives the message at time 0.
Induction Hypothesis: Assume that expression (1) is true for $i = 0, 1, \ldots, q$.
Induction Step: We will show that equation (1) is true for $i = q + 1$. Let $z \in L_{q+1}$. Let $y = parent(z)$ and $x = parent(y)$. A node u may retransmit the message only after it receives the message collision-free. It also avoids all times in $I(u)$ to schedule the transmission. After satisfying these constraints u chooses $transmitTime(u)$ greedily. Hence, we get

$$\begin{aligned} receiveTime(z) &\leq receiveTime(y) + 1 + |I(y)| \\ receiveTime(y) &\leq receiveTime(x) + 1 + |I(x)| \\ receiveTime(z) &\leq receiveTime(x) + 2 + |I(y)| + |I(x)| \end{aligned}$$

Note that either $x \in L_{q-1}$ or $x \in L_q$. Using Lemma 5.5, induction hypothesis, and choosing $k \geq 4k_2$, we get

$$\begin{aligned} receiveTime(z) &\leq t_q + 2 + 2k_2(r_{max}^2/r_{min}^2) \\ receiveTime(z) &\leq k(r_{max}^2/r_{min}^2)q + 2 + 2k_2(r_{max}^2/r_{min}^2) \\ &\leq k(r_{max}^2/r_{min}^2)(q+1) \end{aligned}$$

□

As a corollary, our algorithm gives a constant factor approximation guarantee for graphs where maximum and minimum node transmission ranges are bounded.

COROLLARY 5.7. *If the maximum and minimum node transmission ranges in* G *are bounded then our algorithm has an approximation factor of* $O(1)$.

5.2.2 Bound on the number of retransmissions

Let OPT_{ret} be the set of nodes that retransmit in an optimal algorithm. Without loss of generality, we assume that each node in OPT_{ret} retransmits exactly once (otherwise, the number of retransmissions in the optimal solution would only increase and give us a better lower bound).

LEMMA 5.8. *The total number of retransmissions in our algorithm is at most* $2\,|P|$.

PROOF. In our algorithm, each node retransmits at most once (Lemma 5.3). Hence, the total number of retransmissions is at most $|P| + |S_{ret}|$, where P is the set of primary nodes and S_{ret} is the set of retransmitting secondary nodes. By Lemma 5.2, in $T_{broadcast}$, every child of a secondary node is a primary node with a unique parent. Hence, $|S_{ret}| \leq |P|$. □

LEMMA 5.9. *Recall that* $Primaries(r_{max})$ *is the set of primaries in a disk of radius* r_{max}. *Then we have* $|OPT_{ret}| \geq \frac{|P|}{|Primaries(r_{max})|}$.

PROOF. Any node has a maximum transmission range of r_{max}. The maximum number of primary nodes in a disk of radius r_{max} is $|Primaries(r_{max})|$. Since each node in OPT_{ret} can reach at most $|Primaries(r_{max})|$ primary nodes, we get the required bound. □

THEOREM 5.10. *The number of retransmissions in our algorithm is* $O(r_{max}^2/r_{min}^2)$ *times the number of retransmissions in an optimal algorithm.*

PROOF. Combining Lemmas 5.8, 5.9, and 5.4, we get the result. □

COROLLARY 5.11. *If the maximum and minimum node transmission ranges are bounded in* G, *then the number of retransmissions in our algorithm is at most* $O(1)$ *times the optimal number of retransmissions.*

5.3 Algorithms for Scheduling Multiple Messages

We present three algorithms for scheduling broadcasts of multiple messages. The first is a "one-to-all" broadcast algorithm which schedules the broadcast of multiple messages from a single source. The second and third are "all-to-all" broadcast algorithms which schedule the broadcast of multiple messages from many sources. For the remainder of the section, we assume that the network has a uniform transmission range. Recall that $M = \{1, \ldots, m\}$ is the set of messages. We now present our algorithm for scheduling the broadcast of M from a single source s.

5.3.1 Single Source Multiple Messages (SSMM)

Figure 4 presents a sketch of our single source multiple message scheduling algorithm. The algorithm consists of three stages. The first stage computes the broadcast tree from s

ScheduleBroadcasts_SSMM$(G=(V,E),M,s)$
Stage 1.
$T \leftarrow$BroadcastTree(G,s)
Stage 2.
/* Identical to ScheduleBroadcasts(T,s) except line 9 which is replaced by the next three lines */
9.1 Let $u \in L_q$
9.2 $transmitTime(u) \leftarrow \min\{t | t > t_q \text{ and } t \notin I(u)\}$
9.3 $delay(u) = transmitTime(u) - receiveTime(u)$
Stage 3.
for $j \leftarrow 1$ to m **do**
  for each node $u \in T$ **do**
    if $u = s$ **then**
      transmit message j at time $3(j-1)k+1$
    else
      after receiving message j, delay
      transmission of j by $delay(u)$ time units
return all transmit times

Figure 4: Single Source Multiple Message (SSMM)

using the algorithm given in Figure 1. The second stage computes a schedule for a single message using a modified version of the scheduling algorithm presented in Figure 2. The third stage reuses this schedule efficiently for every message in M. We now present the details of the last two stages.

Stage 2 is identical to the algorithm presented in Figure 2 except Line 9 which is replaced by lines 9.1, 9.2 and 9.3 in Figure 4. Recall that L_q is the set of nodes at level q in the BFS tree rooted at s. Also recall that t_q is the time by which all nodes in L_q are guaranteed to get the message (see expression (1)). Let $u \in L_q$. Line 9.2 ensures that for any level q, all nodes in L_q receive the message before any node in L_q retransmits it. Line 9.3 specifies $delay(u)$, which is the amount of time by which u delays the transmission of a message after receiving it.

In Stage 3, s transmits messages 1 through m sequentially with a uniform gap of $3k$ time units between two successive messages. Recall that k is the constant appearing in expression (1). Specifically, message j is transmitted by s at time $3(j-1)k+1$. Any retransmitting node u which receives j, delays the transmission of j by $delay(u)$ time units computed above.

Why should s introduce a gap of $3k$ time units between two successive message transmissions? To answer this question, we make the following observations. Let L_q and $L_{q'}$ be two BFS levels such that $q' > q$. Our first observation is that, in a network with only bi-directional edges, L_q and $L_{q'}$ have an edge between them only if $q' \leq q+1$. Specifically, transmissions from L_q to L_{q+1} and transmissions from L_{q+3} to L_{q+4} can proceed simultaneously without interfering with each other. Thus, it suffices to "separate" two different messages by three levels in order to allow simultaneous collision-free transmissions of these messages. Our next observation is that once all nodes in L_q receive a message, it takes atmost k time units for all nodes in L_{q+1} to receive the message (see proof of Theorem 5.6). Clearly, $3k$ time units is an appropriate choice of the gap between successive messages.

5.3.2 Multiple Source Multiple Messages

We present two algorithms for scheduling broadcasts of multiple messages with multiple sources. Recall that for every message $j \in M$, s_j is the source of the message j. In the first algorithm, called Intermediate Source Broadcast (ISB), messages get *unicast* from their sources s_j to the intermediate source s. These unicasts themselves are greedily scheduled to ensure collision-free transmissions (see section 5.3.3). After s receives all the messages, it broadcasts them using the algorithm presented in Figure 4. In the second algorithm called Multi-Source Broadcast (MSB), a broadcast tree T_j from each source s_j. Message j gets broadcast on tree T_j using the greedy scheduling algorithm presented in Figure 2. We now briefly explain the details of the algorithm below.

5.3.3 Intermediate Source Broadcast (ISB)

This algorithm consists of two stages. Consider the shortest path p_j from s_j to s. In the first stage, every message j, gets unicast along p_j to s. We ensure that these unicasts are collision-free by a greedy scheduling strategy such that message j_1 gets priority over j_2 iff $j_1 < j_2$. In the second stage, s uses the algorithm in Figure 4 to broadcast all the messages.

5.3.4 Multi-Source Broadcast (MSB)

In the multi-source broadcast algorithm, each source s_j of the message computes a broadcast tree rooted at itself using Figure 1. Message j is broadcast along tree rooted at s_j. Transmissions are scheduled using the algorithm in Figure 2. Message j_1 gets priority over j_2 iff $j_1 < j_2$. This priority is enforced by scheduling j_1 before j_2.

The two algorithms for multiple messages present interesting trade offs. The intermediate source algorithm (ISB) requires only one broadcast tree to be computed and maintained at s, apart from the unicasts from each s_j to s. The multi-source algorithm (MSB), on the other hand, requires computation and maintenance of potentially m trees from each s_j. However, intuitively, MSB should have a lower broadcast latency, since the messages are not routed through an intermediate source. Experimental studies (see section 7.2) confirm this intuition.

5.4 Analysis for Multiple Messages

5.4.1 Latency

We now state the latency bounds for SSMM and the ISB. Recall that k is the constant which appears in expression (1). Observe that m, the number of messages and l, the number of levels in the BFS tree rooted at s, are both lower bounds on the optimal latency OPT_{lat}. We state the following theorems.

Theorem 5.12. *Algorithm SSMM has a latency T such that*

$$3k(m-1)+k(l-2) \leq T \leq 3k(m-1)+k(l-1) \leq 4kOPT_{lat} \quad (2)$$

Theorem 5.13. *Algorithm ISB has a latency T such that*

$$3k(m-1)+k(l-2) \leq T \leq 4k(m-1)+2k(l-1) \leq 6kOPT_{lat} \quad (3)$$

5.4.2 Bound on the Number of Retransmissions

We state the following theorem without proof for all three multiple message broadcast algorithms. The bound below holds for networks with uni-directional edges also.

THEOREM 5.14. *The number of retransmissions in all the three multiple message broadcast algorithms is $O(r_{max}^2/r_{min}^2)$ times the number of retransmissions in the optimal algorithm.*

COROLLARY 5.15. *For networks with bounded maximum and minimum node transmission ranges, the number of retransmissions in all the three multiple message broadcast algorithms is $O(1)$ times the number of retransmissions in the optimal algorithm.*

6. DISTRIBUTED IMPLEMENTATION

All the algorithms presented in this paper have easy distributed implementations when all the edges in the network are bi-directional. Our distributed implementations are inspired by the technique presented in [8]. The basic building blocks for our distributed implementation are the Depth First Search (DFS) and the Breadth First Search (BFS) graph traversal algorithms. There are many distributed DFS and BFS implementations available and we could use any of them for our purpose.

In order to build the broadcast tree for a message, the source of the message creates a token. The purpose of this token is to record the information about the set of primaries and their parents. This token visits each node of the graph during its DFS traversal. When a node receives the token, it knows the current set of primaries and their parents. In particular, it can determine if it is within the range of any current primary. If not, it adds itself to the set of primaries, chooses its parent, updates this information in the token, and passes the token to the next node in the DFS traversal. Since each edge is traversed twice in a DFS traversal, once forward and once backward, a node u also computes its set of children as the token visits u through the back edges.

In order to compute the schedule for a message, the source creates another token. The purpose of this token is to record the information about the receive and transmit times of the nodes in the graph. This token does a DFS traversal of the non-leaf nodes of the broadcast tree created above. When a node receives this token, it knows the transmit times of all the internal nodes visited so far. In particular, it knows the transmit times of the nodes which could interfere with its own transmission. The node now computes its minimum possible collision-free transmit time, updates this information in the token and sends it to the next node to be visited. In other words, the single message broadcast algorithm can be implemented using two DFS traversals, independent of the number of messages m.

We now discuss the distributed implementations of our multiple message broadcasting algorithms. We note that essentially the same technique discussed above can be applied for the distributed implementation of the single source multiple message (SSMS) algorithm. The first DFS computes the broadcast tree from the source s. In the second DFS, every retransmitting node u computes the $delay(u)$ by which it should delay the transmission of any message after receiving it. After this step, s starts transmitting all the messages with a uniform gap of $3k$ time units between two successive messages (see section 5.3.1). In the intermediate source broadcast (ISB), the second stage is the same as SSMS. In the first stage, ISB computes the shortest path from the source s_j of each message j to s and greedily schedules the unicast of the messages along these paths. We note that this stage can be efficiently implemented using one BFS (for computing shortest paths) and one DFS (for computing the schedule). We omit the details involved due to lack of space. Finally, for the multiple source broadcast, we perform a DFS from each source s_j to compute its broadcast tree and for each message j to compute its schedule. This is precisely the tradeoff referred to earlier between ISB and MSB: ISB involves three DFS traversals and one BFS traversal which is independent of the number of messages m, whereas MSB involves potentially $2m$ DFS traversals. However, MSB has a much better broadcast latency compared to ISB (see section 7.2).

There are several other issues which arise in the presence of uni-directional edges. In this case, many back edges during a DFS traversal might not even exist. Instead, A single back edge may need to be replaced by a directed path. Another basic issue is that of network connectivity. All the distributed implementations presented above assume strong network connectivity, i.e., the existance of a directed path from any node u to any other node v. In the presence of uni-directional edges, the network may not be strongly connected. We simply note that routing in ad hoc networks in the presence of uni-directional edges is a very involved and interesting research issue. A complete treatment of this issue is beyond the scope of this paper.

7. EXPERIMENTAL EVALUATION

This section deals with the experimental performance evaluation of our algorithms through simulations. The experiments focus on the effect of various network conditions on broadcast latency and the number of retransmissions. In all the experiments, the network nodes were placed uniformly at random within a square. All the experiments were performed on strongly connected graphs. All data points were averaged over 10 simulation runs and the error bars indicate the maximum and minimum deviation of any observed value from the mean, i.e, a confidence of 100% that the mean is within the range indicated by the error bars. The choice of parameters made in our experiments were inspired in part by the recent survey by Williams and Camp [30] and by the need for completing the simulations within reasonable time. All our simulations were performed using the Mathematica and Combinatorica packages.

7.1 Single Message Broadcast

7.1.1 Uniform Transmission Ranges

We present the results of our study of the single message broadcast algorithm with uniform node transmission range. In these experiments, we placed the nodes uniformly at random in a square of length 350 meters and chose the trans-

mission range to be 100 meters. We varied the number of nodes from 20 to 110 in steps of 10, leading to higher node densities and higher average node degree. We studied the effect of this on the latency and the number of retransmissions. One node was chosen uniformly at random to be the source. We now present our results.

Figure 5 is a plot of the broadcast latency vs. the number of network nodes. Broadcast latency is the maximum time taken by any node to receive the message. The plot indicates that latency does not vary much with the number of nodes in the graph. This is intuitive, since the latency is mostly influenced by the depth of the BFS tree from the source. This depth does not depend much upon the number of nodes, but only on the length of the square and the transmission range.

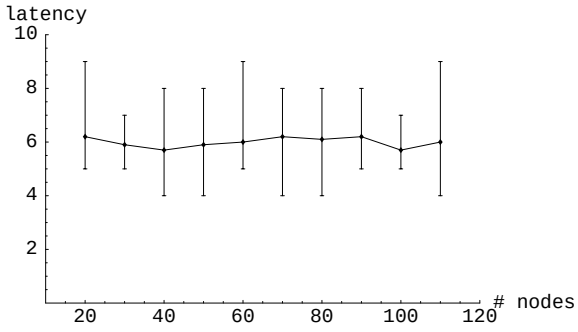

Figure 5: Latency vs. # nodes

Figure 6 is a plot of the approximation ratio vs. the number of network nodes, where ratio is $(latency/BFSDepth)$. In reality, the plotted ratio is a conservative estimate of the real approximation ratio. As is seen from the plot, even this ratio is very close to 1 most of the times. In fact, the worst case ratio observed for all the runs is no worse than 1.75.

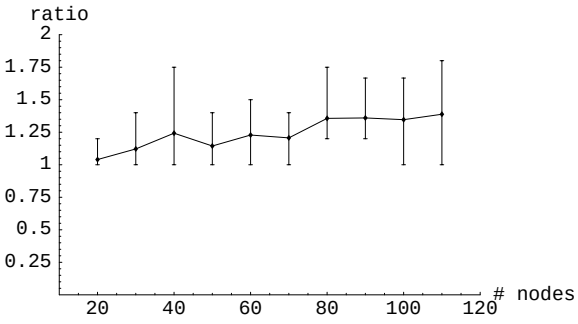

Figure 6: ratio vs. # nodes. ratio = latency/BFS depth

Figure 7 is a plot of the number of retransmitting nodes vs. the number of network nodes. While the number of retransmissions seems to increase steadily initially, it stabilizes once the network size grows beyond 70 nodes. On an average 11% of the nodes retransmit while the worst case among all the runs is 15%.

7.1.2 Non-uniform Transmission Ranges

We now present the results of the study of the single message broadcast algorithm with non-uniform transmission ranges.

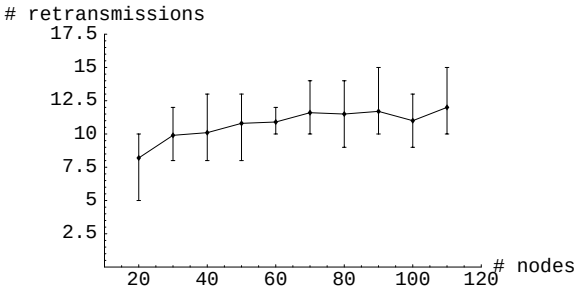

Figure 7: # retransmissions vs. # nodes

In these experiments, we placed 200 nodes in a square of length 1500 meters. The transmission ranges were chosen uniformly at random from the interval $[r_{min}, r_{max}]$, where $r_{min} = 200$ meters for all experiments, and r_{max} was varied from 200 to 1000 in steps of 100.

Figure 8 plots latency vs. r_{max} and Figure 10 plots the number of retransmissions vs. r_{max}. Note that both these values drop down with increasing r_{max} values. Intuitively, as r_{max} increases, the average range of the nodes also increases. This leads to increasing number of nodes being covered for each retransmission leading to a reduction in the total number of retransmissions. Note that even though the latency drops down with r_{max}, in Figure 9 the ratio $(Latency/BFSDepth)$ stabilizes around 1.5. This is in contrast with the analytically predicted approximation ratio of (r_{max}^2/r_{min}^2). We interpret these results as an indication of a much better average case performance than the guaranteed worst case performance.

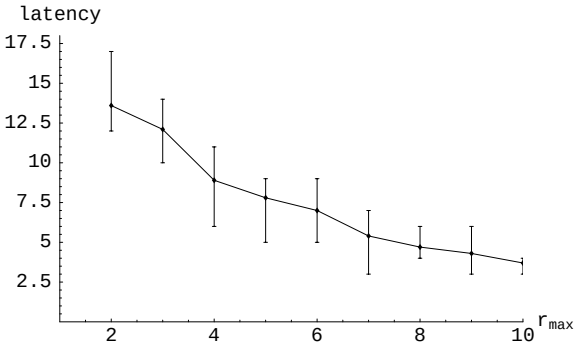

Figure 8: Latency vs. r_{max}

7.2 Multiple Messages

We now present the results of performance studies of the MSB algorithm. In these experiments, 100 nodes were placed in a square area of length 350 meters. All nodes had a uniform transmission range of 100 meters. The number of messages were doubled in each step from 2 to 128. All the messages originate at the same time. We studied the approximation ratio of the algorithm. The lower bound used in computation of the ratio is $\max\{BFSDepth, \#messages\}$, since both the $BFSDepth$ from any source and the number of messages are lower bounds for latency.

Figure 11 plots the approximation ratio vs. lower bound. Note how the ratio increases initially and then drops down

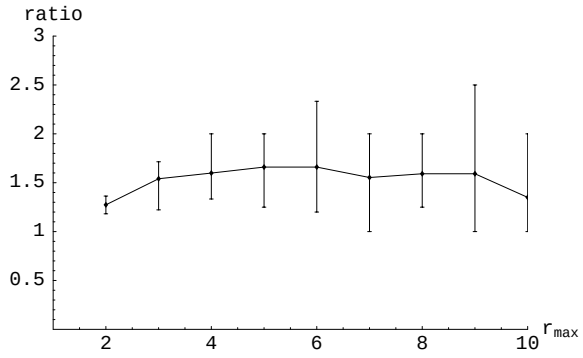

Figure 9: ratio vs. r_{max}. ratio = latency/BFS depth

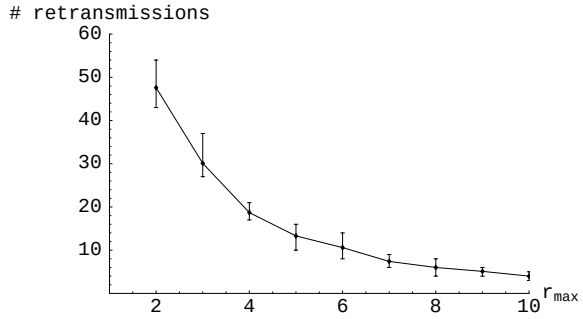

Figure 10: # retransmissions vs. r_{max}

to a constant value as the number of messages increase. In particular, the ratio reaches the peak when the number of messages is around 10. This can be explained by observing that the $BFSDepth$ is a very poor lower bound when multiple messages are transmitted. Once the number of messages exceeds the $BFSDepth$, which happens at around 10, the lower bound is strengthened leading to a better approximation ratio. This plot also implies that for every additional message, the marginal increase in latency is only 4 time units.

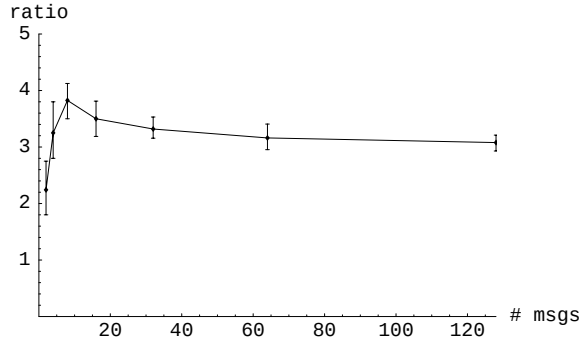

Figure 11: ratio vs. # messages. ratio = latency/lower bound

In the case of the intermediate source broadcast (ISB), recall that equation (3) provides a tight lower bound of $3k$ for the broadcast of multiple messages. This indicates that when the number of messages m is large, ISB will have a marginal latency of about 200 times units for every additional message. In contrast, MSB performs much better with a marginal latency of only 4 for every additional message. However, while ISB can be implemented with three DFS and one BFS traversal which is independent of the number of messages, MSB requires potentially $2m$ DFS traversals. This is precisely the trade off between MSB and ISB which was referred to in Section 5.3.4.

8. CONCLUSIONS AND FUTURE WORK

In this paper, we studied the problem of efficient collision-free broadcasting in ad hoc networks. We showed that minimum latency collision-free broadcasting in ad hoc networks is NP-hard. We presented simple, efficient and distributed collision-free broadcast algorithms which have provably good approximation bounds for both latency and the number of retransmissions. Specifically, for single message broadcast, for networks with bounded node transmission ranges, the latency of our algorithm is within a factor of $O(1)$ from the optimal value. For multiple messages, for networks with uniform transmission ranges, our algorithms have a latency which is within a factor of $O(1)$ from the optimal. For networks with bounded node transmission ranges, the number of retransmissions for all our algorithms is within a factor of $O(1)$ from the optimal. We studied the performance of our algorithms experimentally and showed that our algorithms perform much better in practice than what is analytically guaranteed.

We would like to extend our study in many ways. An interesting future work is to extend our current algorithms to networks with dynamic topologies. Creating and maintaining a broadcast tree in dynamic topologies is inefficient. Existing deterministic, collision-free broadcasting approaches involve pre-computing the schedule of all the nodes independent of the network topology in the beginning. However, in general, such schemes do not guarantee good latency and retransmission bounds. Randomized schemes which do not rely on complete topology information but locally guarantee collision-free transmissions with high probability seem more attractive for dynamic topologies. Another research issue for the future is extending our work to support multicast operations. Finally, integrated protocols for supporting a mixture of unicast, multicast, and broadcast traffic is a very interesting and challenging research issue.

Acknowledgments. We thank Samir Khuller, Sriram Pemmaraju, Aravind Srinivasan and Yung-Chun (Justin) Wan for useful discussions. Special thanks to Sriram Pemmaraju for answering various questions about Mathematica and Combinatorica and for pointing out and correcting a bug in the original version of our algorithm. Thanks also to Aleksi Penttinen who pointed out the same bug.

9. REFERENCES

[1] Alon, N., Bar-Noy, A., Linial, N., and Peleg, D. A lower bound for radio broadcast. *Journal of Computer and System Sciences 43*, 2 (Oct. 1991), 290–298.

[2] Bar-Yehuda, R., Goldreich, O., and Itai, A. On the time-complexity of broadcast in multi-hop radio networks: An exponential gap between determinism and randomization. *Journal of Computer and System Sciences 45* (1992), 104–126.

[3] Basagni, S., Chlamtac, I., and Bruschi, D. A mobility-transparent deterministic broadcast mechanism for

ad hoc networks. *IEEE/ACM Transactions on Networking (TON) 7*, 6 (1999), 799–807.

[4] Bruschi, D., and Pinto, M. D. Lower bounds for the broadcast problem in mobile radio networks. *Distributed Computing* (1997), 129–135.

[5] Cheng, X., Huang, X., Li, D., and Du, D.-Z. Polynomial-time approximation scheme for minimum connected dominating set in ad hoc wireless networks. Tech. rep.

[6] Chlamtac, I., and Farago, A. Making transmission schedule immune to topology changes in multi-hop packet radio networks. *IEEE/ACM Transactions on Networking* (1994), 23–29.

[7] Chlamtac, I., and Kutten, S. On broadcasting in radio networks - problem analysis and protocol design. *IEEE Transactions on Communications 33*, 12 (1985), 1240–1246.

[8] Chlamtac, I., and Kutten, S. Tree-based broadcasting in multihop radio networks. *IEEE Transactions on Computers 36*, 10 (1987), 1209–1223.

[9] Chlebus, B., Gasieniec, L., Gibbons, A., Pelc, A., and Rytter, W. Deterministic broadcasting in unknown radio networks. *Distributed Computing* (2002), 27–38.

[10] Clark, B., Colbourn, C., and Johnson, D. Unit disk graphs. *Discrete Mathematics 86* (1990), 165–177.

[11] Das, B., and Bharghavan, V. Routing in ad-hoc networks using minimum connected dominating sets. In *ICC (1)* (1997), pp. 376–380.

[12] Das, B., Sivakumar, R., , and Bharghavan, V. Routing in ad-hoc networks using a virtual backbone. In *6th International Conference on Computer Communications and Networks (IC3N '97)* (Sept. 1997), pp. 1–20.

[13] Elkin, M., and Kortsarz, G. Combinatorial logarithmic approximation algorithm for directed telephone broadcast problem. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing* (2002), ACM Press, pp. 438–447.

[14] Guha, S., and Khuller, S. Approximation algorithms for connected dominating sets. *Algorithmica* (1998), 374–387.

[15] Ho, C., Obraczka, K., Tsudik, G., and Viswanath, K. Flooding for reliable multicast in multi-hop ad hoc networks. In *Proceedings of the International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM)* (1999), pp. 64–71.

[16] Hung, P.-K., Sheu, J.-P., and Hsu, C.-S. Scheduling of broadcasts in multihop wireless networks. In *European Wireless* (2002).

[17] Jetcheva, J., Hu, Y., Maltz, D., and Johnson, D. A simple protocol for multicast and broadcast in mobile ad hoc networks, July 2001.

[18] Kowalski, D. R., and Pelc, A. Deterministic broadcasting time in radio networks of unknown topology. In *43rd Annual Symposium on Foundations of Computer Science (FOCS)* (2002), pp. 63–72.

[19] Lim, H., and Kim, C. Multicast tree construction and flooding in wireless ad hoc networks. In *Proceedings of the 3rd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems* (2000), ACM Press, pp. 61–68.

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

[21] MILCOM: Military Communications Conference, 2002. http://www.milcom.org/2002.

[22] Ni, S.-Y., Tseng, Y.-C., Chen, Y.-S., and Sheu, J.-P. The broadcast storm problem in a mobile ad hoc network. In *Proceedings of the fifth annual ACM/IEEE international conference on Mobile computing and networking* (1999), ACM Press, pp. 151–162.

[23] Peng, W., and Lu, X. Efficient broadcast in mobile ad hoc networks using connected dominating sets. *Journal of Software - Beijing, China* (1999).

[24] Peng, W., and Lu, X. Ahbp: An efficient broadcast protocol for mobile adhoc networks. *Journal of Science and Technology - Beijing, China* (2000).

[25] Peng, W., and Lu, X.-C. On the reduction of broadcast redundancy in mobile adhoc networks. In *Proceedings of First Annual Workshop on Mobile Ad Hoc Networking Computing. MobiHOC* (Aug. 11, 2000).

[26] Qayyum, A., Viennot, L., and Laouiti, A. Multipoint relaying: An efficient technique for flooding in mobile wireless networks. Tech. Rep. Research Report RR-3898, INRIA, Feb. 2000.

[27] Sivakumar, R., Das, B., and Bharghavan, V. Spine routing in ad hoc networks. *ACM/Baltzer Cluster Computing Journal (special issue on Mobile Computing)* (1998).

[28] Sucec, J., and Marsic, I. An efficient distributed network-wide broadcast algorithm for mobile adhoc networks. Tech. rep., Rutgers University, September 2000.

[29] Wan, P.-J., Alzoubi, K., and Frieder, O. Distributed construction of connnected dominating set in wireless ad hoc networks. In *IEEE INFOCOM* (2002).

[30] Williams, B., and Camp, T. Comparison of broadcasting techniques for mobile ad hoc networks. In *Proceedings of the third ACM international symposium on Mobile ad hoc networking and computing* (2002), ACM Press, pp. 194–205.

[31] WoWMoM: ACM Workshop on Wireless Mobile Multimedia, 2002. http://www.wowmom.com.