Low-Latency Broadcast Scheduling in Ad Hoc Networks

Scott C.-H. Huang[1], Peng-Jun Wan[1,2], Xiaohua Jia[1], and Hongwei Du[1]

[1] City University of Hong Kong
{shuang, pwan, jia, hongwei}@cs.cityu.edu.hk
[2] Illinois Institute of Technology
wan@cs.iit.edu

Abstract. Broadcast is a fundamental operation in wireless network, and naïve flooding is simply not practical. Previous results showed that although broadcast scheduling can achieve constant approximation ratios in respect of latency, the current state-of-the-art algorithm's ratio is still overwhelmingly large (≈ 650). In this paper we present two basic broadcast scheduling algorithms that both achieve small ratios 51 and 24, while preserving low redundancy 1 and 4 (in terms of number of retransmissions a node has to make). Moreover, we also present a highly efficient algorithm whose latency is $R + O(\sqrt{R}\log^{1.5} R)$ (where R is the network radius) and each node only has to transmit up to 5 times. This result, in a sense of approximation, is nearly optimal since $O(\sqrt{R}\log^{1.5} R)$ is negligible when R is large. Moreover, R is itself a lower bound for latency, so the approximation ratio is nearly 1 and this algorithm is nearly optimal.

1 Introduction

Among many operations of mobile ad hoc networks, broadcast is probably the most fundamental yet challenging operation since [25] tells us naïve flooding is simply not practical. Our first objective is to find a good scheduling algorithm that can mitigate the impact of potential collision and have a low broadcast latency. In addition, we also want our algorithm efficient such that nodes only have to transmit the message very few times. Redundancy is measured by how many times a node has to retransmit in order to guarantee collision-free reception. We want to balance latency and redundancy in this work.

It is known that broadcast in ad hoc networks has a constant approximation algorithm [18]. However, it is still not practical because the approximation ratio is overwhelmingly large (it was estimated to be near 648). In this paper, we present two basic broadcast scheduling algorithms that significantly reduce this ratio. One of our algorithms has ratio 51 and another has 24, and, more importantly, they do not increase redundancy much. The above two algorithms guarantee that each node only has to retransmit 1 and 4 times to guarantee proper reception, respectively. Moreover, we also present a highly efficient algorithm whose latency is $R + O(\sqrt{R}\log^{1.5} R)$ (where R is the network radius) and each node only has to transmit up to 5 times. This result, in a sense of approximation, is nearly optimal

X. Cheng, W. Li, and T. Znati (Eds.): WASA 2006, LNCS 4138, pp. 527–538, 2006.

since $O(\sqrt{R}\log^{1.5} R)$ is negligible when R is large. Moreover, R is itself a lower bound for latency, so the approximation ratio is nearly 1 and this algorithm is nearly optimal.

2 Related Work

Sheu *et al* [28] did empirical studies about the efficiency of broadcasting schemes in terms of collision-free delivery, number of retransmissions and latency. Basagni *et al* [4] presented a mobility transparent broadcast scheme for mobile multi-hop radio networks by using mobility-transparent schedule that guarantees bounded latency. Chlamtac and Kutten [8] first showed that the problem of finding an optimal deterministic broadcasting scheme for general graphs is NP-hard. Chlamtac and Weinstein [9] used undirected bipartite graphs to model this problem and gave an $O(\log^2 n)$ approximation algorithm (which gave a $O(R\log n)$ upper bound on broadcast latency where R is the radius and n is the number of nodes). Kowalski and Pelc [21] later reduced it to $O(R\log n + \log^2 n)$. Bar-Yehuda *et al* [3] obtained the same result earlier, but their solution was a randomized algorithm of Las Vegas type. Gaber and Mansour [17] employed clustering techniques to reduce this broadcast latency upper bound to $O(R + \log^5 n)$. Elkin and Kortsarz [16] refined this method and obtained a bound of $R + O(\sqrt{R} \cdot \log^2 n)$, thus reducing it to $O(R + \log^4 n)$. Alon *et al* [1] proved that there exists a family of radius-2 networks for which any broadcast schedule requires at least $\Omega(\log^2 n)$ time slots. Bruschi and Del Pinto [5] considered distributed protocols and obtained a lower bound of $\Omega(D\log n)$ with the assumption that no nodes know the identities of their neighbors. Kushilevitz and Mansour [22] proved that for any randomized broadcast protocol there exists a network whose latency is $\Omega(D\log(N/D))$. Chlebus *et al* [10] studied deterministic broadcasting without a-priori knowledge of the network. Elkin and Kortsarz showed in [14] that the radio broadcast problem is $\Omega(\log n)$-inapproximable unless $NP \subset BPTIME(n^{O(\log\log n)})$. Also, they showed in another work [15] that this problem can not be approximated within an additive term $c\log^2 n$ for some constant c unless $NP \subset BPTIME(n^{O(\log\log n)})$.

Gandhi *et al* [18] proved that constant approximation exists in disk graphs, which was impossible in general graphs according to [14]. In [18], an important technique of finding a *Connected Dominating Set* (CDS) as a virtual backbone is used. CDS plays an important role and has been used extensively in broadcast [18], routing [29] [12] [13], as well as many other areas of networking. Guha and Khuller [19] studied the minimum connected dominating set problem in general graphs and proved its NP-hardness. Clark *et al* [11] showed that this problem remains NP-hard even in unit disk graphs (UDGs) There are also some results on approximating this problem. Marathe *et al* [24] gave some heuristics for UDGs. Cheng *et al* [6] designed a polynomial-time approximation scheme for MCDS problem. Wan *et al* [30] and Alzoubi *et al* [2] studied the approximation of CDS, MCDS problem in terms of its size as well as time and message complexity.

3 Preliminaries

3.1 Network Model

An ad hoc network can be modelled as a unit disk graph $G = (V, E)$ along with a source node $s \in V$. Each node has transmission range 1, and two nodes u, v are neighbors if and only if their Euclidean distance is less than 1. The *radius* of G is defined as $\max_{v \in V} dist(s, v)$, where $dist(u, v)$ is the hop distance between u and v. For any node $v \in V$, its *layer* $l(v)$ is defined as the hop distance between v and s. The source node's layer $l(s)$ is 0. We can group V by layers as follows. L_i denotes the nodes whose layers are equal to i (i.e. $L_i = \{v | l(v) = i\}$). Time is assumed to be discrete and we use time slots to represent it throughout this paper.

3.2 Problem Definition

Given $G = (V, E)$ and $s \in V$, the objective is to find a *schedule* satisfying the following requirements. (1) It is represented as a function, called $TransmitTime$, from V to subsets of natural numbers. A subset of $S \subset V$ is mapped to a subset of $T \subset \mathbb{N}$ if and only if all nodes in S are scheduled to transmit in time slots indicated in T. (2) A node receives the message *collision-free* at some time if and only if exactly one of its neighbors is transmitting at this time. (3) A node cannot transmit unless it has already received the message collision-free earlier.

Latency is the first time slot such that all nodes have received the message collision-free, and *redundancy* is the maximum size of these subsets of $\mathbb{N}$ mapped by some subset in V, namely $\max_{S \subset V} |TransmitTime(S)|$. The goal this paper is to find a scheduling algorithm that minimizes this latency while having low redundancy.

3.3 Layered MIS and Virtual Backbone Construction

We will use the concept of *Maximal Independent Sets* (MIS) throughout this paper. An independent set (IS) is a subset $S \subset V$ such that no two nodes in S are adjacent to each other (i.e. all nodes are independent of each other). A maximal independent set M is an independent set with maximality, which means adding any other node will destroy the independence property. Formally, M is an MIS if it is an independent set and for all $M' \supset M$, M' is not independent. We construct MIS in a layered manner, which ensures a *2-hop separation property* as follows. Starting from the first layer L_1, we choose a MIS, denoted by $BLACK_1 \subset L_1$ first. Then we move on to L_2 and select independent nodes $BLACK_2 \subset L_2$ as well. Not only are $BLACK_2$ nodes independent of each other, they are independent of all nodes in $BLACK_1$ as well. We follow this method and select $BLACK_3, BLACK_4, \ldots$ until we finish the last layer and finally we get a layered MIS denoted by $BLACK = \cup_i BLACK_i$. Such set $BLACK$ we construct in this manner has a stronger property than arbitrarily constructed ones. Let's construct a breadth-first search tree T_{BFS} for G and pick a node $v \in BLACK_i$. Observe that v's parent $p(v)$ in T_{BFS} must be in $L_{i-1} - BLACK_{i-1}$ (since

$p(v)$ is in L_{i-1} and not independent of v), and therefore $p(v)$ must be adjacent to some node in $BLACK_{i-1}$. In short, for all i and any node in $BLACK_i$, it must have a 2-hop neighbor in $BLACK_{i-1}$. This is called the 2-hop separation property. Because of this property, if G is connected, we can construct a virtual backbone consisting of the set $BLACK$ along with another set $BLUE$ defined as $BLUE = \{v|v \text{ is the parent of some nodes in } BLACK \text{ in } T_{BFS}\}$ Obviously $BLACK \cup BLUE$ form a connected dominating set as described above. For simplicity we also use the term "black nodes" to refer to the set $BLACK$, and the term "blue nodes" to refer to $BLUE$. This virtual backbone is actually a tree T_{br} formally defined as follows. $T_{br} = (BLACK \cup BLUE, E_{br})$ where $(u, p(u)) \in E_{br}$ ($p(u)$ represents u's parent in T_{br}) if and only if

$$\begin{cases} u \in BLACK_i, p(u) \text{ is } u\text{'s parent in } T_{BFS} \\ u \in BLUE_i, p(u) \in BLACK_i \cup BLACK_{i-1}, (u, p(u)) \in E \end{cases}$$

The second condition is always valid since $BLACK$ is constructed layer by layer. Any blue node at layer i must be adjacent to a black node either at layer i or $i-1$. However, there may be more than one black node adjacent to it. In this case, we just pick one of them arbitrarily. T_{br} has the following properties. (1) A black node's parent is always blue. (2) A blue node's parent is always black. (3) If $u \in BLACK_i$ then $p(p(u)) \in BLACK_{i-1}$ where $p(p(u))$ is the parent of u's parent (i.e. u's grandparent).

3.4 Coloring of MIS Nodes

If some black nodes get the broadcast message, then we are sure that all of their neighbor nodes will get the message from them within very few time slots. Similarly, if all black nodes get the message, then all nodes will get the message from them in a short time. We are going to show that 12,13 time slots are enough. The time required for black nodes to pass message to their neighbors depends on a *coloring* of them.

If two nodes are separated by at least 3 hops (this essentially means their Euclidean distance is greater than 2), then if they broadcast at the same time, there will be no collision at all since their transmission ranges do not overlap. In other words, two black nodes can be scheduled to transmit for the same time slot as long as they are 2-independent (not 2-hop neighbors). We can define a new graph H whose vertices are black nodes and an edge exists between two nodes if they are 2-hop neighbors, and consider the coloring of H. Nodes of the same color will be scheduled for the same time slot, and the time required for black nodes to broadcast the message to all nodes (using this scheduling method) is equal to the number of colors used. We show that 12 colors are enough if all nodes know about their locations and 13 colors are enough if not.

Lemma 3.1. *If all nodes know about their own locations, 12 colors are enough to color H (H is defined as above).*

Proof. We partition the entire plane into half-open, half-closed hexagons and give a 12-coloring, as shown in Figure 1. Each hexagon has radius 1/2 (and

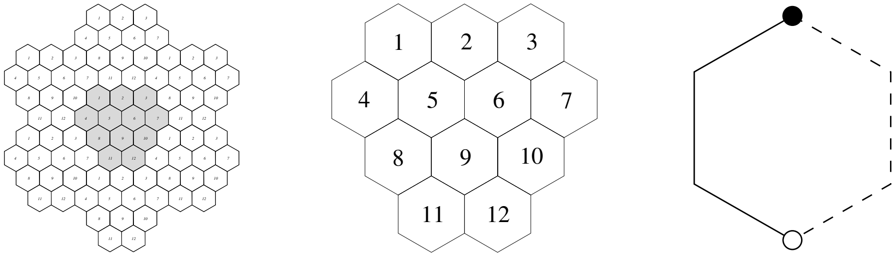

Fig. 1. Partition of the entire plane into half-open, half-closed hexagons

therefore its diameter is equal to 1). According to its location, each node belongs to exactly one hexagon and is assigned to a corresponding color. Since each hexagon has radius 1/2, there can be at most one black node in each hexagon and the distance between any different hexagons of the same color is at least 2. □

The above coloring takes great advantage of unit disk graphs geometrical properties. Note that this way of coloring does not need global information and can be implemented locally. If location information is not known but the global topology is known (by some server), then 13 time slots are enough. However, this is a centralized approach.

Lemma 3.2. *If the global topology is known, 13 colors are enough to color H.*

Lemma 3.3. *Given a black node v. There can be at most 12 black nodes other than v itself in any half-disk centered at v with radius 2.*

Proof. Consider a half disk shown in Figure 2(a). Since black nodes are independent, they cannot appear in the unit circle centered at v. In other words, all black nodes can only appear in the half-annulus ($A_1 \cup A_2$). We divide this region into A_1 and A_2 (where g is the golden ratio) as shown in the figure. This lemma can be proved by showing that (1)there can be at most 5 black nodes in A_1 (2)there can be at most 7 black nodes in A_2. To prove (1), we assume there are 6 or more black nodes in A_1. If we draw a line from v to each of them, then there must be two black nodes u_1, u_2 such that $\angle u_1 v u_2$ is at most 36°. Since $1 < \overline{u_1 v}, \overline{u_2, v} < g$, we know that the distance between these two black nodes is less than 1. Figure 2(b) shows the extreme case where $\overline{u_1 v} = g$, $\overline{u_2, v} = 1$, $\angle u_1 v u_2 = 36°$, and $\overline{u_1 u_2} = 1$ but equality does not hold and $\overline{u_1, u_2}$ has to be less than 1. (2) can be proved similarly. Suppose there are 8 or more black nodes in A_2, then there must be two black nodes u_1, u_2 such that $\angle u_1 v u_2$ is at most $180/7 \doteq 25.71°$. This also implies $\overline{u_1, u_2} < 1$ since the extreme case happens when $\overline{u_1 v} = g$, $\overline{u_2, v} = 2$. $\overline{u_1, u_2} > 1$ would imply $\angle u_1 v u_2 > 29.77°$ as shown in Figure 2(b), which is impossible. □

Proof of Lemma 3.2. Let's give an ordering of vertices as follows. Since the global topology is known, we can find the node v_1 of smallest degree in H first. Then we consider the new graph with v_1 and all its incident edges removed from

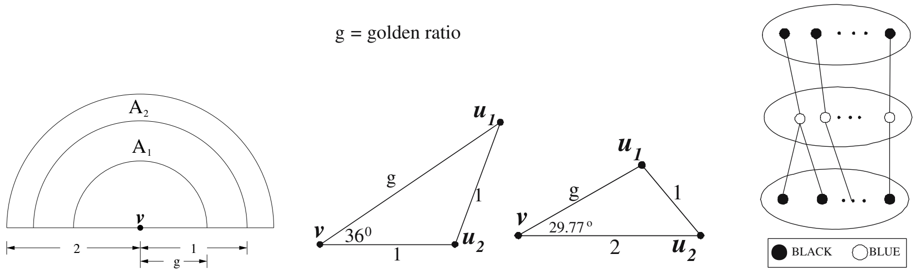

Fig. 2. (a) Half disk (b) In A_1 (c) In A_2 (d) Tripartite model

H. We pick another node v_2 of smallest degree in this new graph, delete it, and repeat this process until all nodes have been picked and deleted. When v_i is picked, $v_1, v_1, \ldots, v_{i-1}$ have already been removed and v_i's degree in the new graph, denoted by d_i, equals $d_i = |N(v_i) \cap \{v_n, v_{n-1}, \ldots, v_{i+1}\}|$, where $N(v_i)$ is the set of v_i's neighbors in H. This gives an ordering of nodes $\{v_1, v_2, \ldots, v_n\}$. Note that $d_i \leq 12$ for all i. This is due to the fact that, at any time, there is always a vertex w that is geographically located at the leftmost position, which implies all of its neighbors in H are located in a half annulus centered at w. According to Lemma 3.3, there are at most 12 such neighbors.

We now reverse this ordering and claim that if we use first-fit coloring in the reverse order $\{v_n, v_{n-1}, \ldots, v_1\}$, 13 colors are enough. When we color a node v_i, we need to look at its neighbors in $\{v_n, v_{n-1}, \ldots, v_{i+1}\}$ and avoid using any color appeared there. Some of those neighbors may use the same color, but the worst case may happen in which none of those neighbors use the same color. In this case, $|N(v_i) \cap \{v_n, v_{n-1}, \ldots, v_{i+1}\}| = d_i$ colors have been used, and we have to use another color for v_i. Since we already know earlier that $d_i \leq 12$ for all i, 13 colors are enough to color H. □

3.5 Tripartite Graph Model

We introduce a basic model called the *Tripartite Graph Model* for analyzing approximation ratios of broadcast latency. We focus on interactions between neighboring layers $BLACK_{i-1}$ and $BLACK_i$, for all $1 \leq i \leq R$ where R is the radius of G. Consider T_{br} again. Since we know that each black node's grandparent must be a black node of the previous layer, it suffices to consider the interaction between $BLACK_{i-1}$, $BLACK_i$ and those blue nodes in-between as connectors alone. If all black nodes have received the message, it only takes 12 or 13 more time slots to pass the message to all nodes. We know earlier that $BLACK_{i-1}$ and $BLACK_i$ are connected through blue nodes in between, so we only focus on the tripartite graph consisting of $BLACK_{i-1}$, $BLACK_i$, and blue nodes in between and the edges are taken directly from T_{br}, as shown in Fig 2(d). Note that it also takes 12 or 13 time slots from $BLACK_{i-1}$ to blue connectors, since $BLACK_{i-1}$ must contain all blue connectors. Broadcast latency thus depend solely on the schedule from blue connectors to $BLACK_i$, called *BLUE-to-BLACK*. Using this model, if BLUE-to-BLACK can be scheduled within ζ time slots, then we can repeat this process to pass the message to all black nodes within $(13+\zeta)R$ time slots (where R is the network radius) and finally to all nodes within $(13+\zeta)R$ time slots (we can regard the source s as blue and start from $BLACK_1$). We also know that R is itself a lower bound for transmission latency (since the message needs at least R time slots to reach the farthest node). This actually means that the approximation ratio of any broadcast algorithm using this model is at most $13+\zeta$ (or $12+\zeta$ if location is known).

4 First and Second Broadcast Algorithms

4.1 First Broadcast Algorithm - Least Redundancy

First algorithm has approximation ratio 51 and each node is only required to broadcast the message at most once. Using the tripartite graph model, we claim that applying First-Fit scheduling on BLUE-to-BLACK yields a result in which $\zeta = 39$ and the approximation ratio will be $39 + 13 = 52$ (or 51 if location is known).

Observe that for two blue nodes to be able to interfere with each other, their distance must be at most 2. Therefore, if we regard one blue node u as the origin, any potential interfering blue nodes must lie inside the circle centered at the origin with radius 2. Every interfering blue node must have at least one black child, otherwise this blue node wouldn't need to transmit and wouldn't be selected as a connector at all. The number of blue nodes lying inside a circle with radius 2 cannot exceed the number of black nodes lying inside the concentric circle with radius 3. We can apply Wegner's Theorem [31] with proper scaling to show that this number is at most 41 as follows.

Let S denote the set of black nodes that are at most three hops away from u. Then the convex hull of S is contained in the disk of radius 3.5 centered at u. Let $k = |S|$. Apply scaled Wegner Theorem, we know $k \le 42$, and apparently 43 time slots are enough to schedule BLUE-to-BLACK. Now, we are going to show that 39 time slots are enough. First, we know that u's parent must have been counted, but u certainly does not have to transmit to its parent. Moreover, u must have a black child that it has to transmit to. This has been over-counted as an interfering blue node as well. So far, we know that 41 time slots are enough.

We will show that we only have to consider the case in which u has two or more children. The reason is that if u has only one child, we actually need way less than 39 time slots for the following reason. Assume u has only one child w, then any blue node that may interfere with u's transmission must be a neighbor of w. This essentially means the number of interfering blue nodes cannot exceed the number of black nodes inside the disk of radius 2 centered at w. Applying Wegner's Theorem and using the same argument, we can prove that the number of independent nodes lying inside a disk of radius 2 is at most 21 (the proof is omitted here for simplicity). Since w is itself a black node and has been counted already, w has at most 20 interfering blue neighbors and 39 time slots are certainly more than enough.

The above argument shows that we only have to consider the case in which u has 2 or more children. For this reason we know that we have over-counted one black node, and 40 time slots are enough so far. Finally, we are going to show that we still have over-counted one more black node and our desired result will be proved.

We look at u's parent $p(u)$ now. We claim that we only need to consider the case in which there exists some interfering blue node whose parent is not $p(u)$, otherwise it will result in an even tighter time bound. Assume the contrary that all interfering blue nodes' parent is $p(u)$, then the number of interfering blue nodes cannot exceed the number of black nodes lying inside the disk of radius 2 centered at $p(u)$. Again, by Wegner's Theorem, this results in a bound of 21 time slots and, indeed, we don't have to consider this case. Finally, since there exists some blue node whose parent is not $p(u)$, we can further subtract 1 from 40 and 39 time slots are enough to schedule BLUE-to-BLACK.

Note that all transmissions are scheduled in such a way that all possible collisions are avoided, so In the first algorithm each node needs to broadcast the message at most once.

4.2 Second Broadcast Algorithm - Tradeoff Between Latency and Redundancy

In this section we present another algorithm that achieves a better approximation ratio (24 if location is known and 26 if global topology is known) at the price of increasing redundancy by a factor of 4. Note that redundancy is measured as the maximum number of transmissions a single node has to make rather than the total number of transmissions in the entire network. We believe our metric makes the most sense in measuring redundancy, as it most precisely reflects the battery lifetime of a sensor node in sensor networks. The lifetime of a sensor network should be characterized as the lifetime of a single node rather than the number of all transmissions, since the failure of few nodes may disconnect the network and a small number of total transmissions does not guarantee network connectivity. We will study the tradeoff between latency and redundancy in detail in this subsection.

Again, we start from the tripartite graph model and schedule the part BLUE-to-BLACK. Instead of First-Fit, we now use the coloring-H method similar to the BLACK-to-BLUE part. Note that we can color H with 12 colors if location is known and with 13 colors if global topology is known. We schedule the blue nodes in the following way. Each blue nodes has at most 4 black children (since each blue node has at most 5 black neighbors and one of them must be its parent), so each blue node simply looks at its black children's colors and transmit in the corresponding time slots. For example, if a blue node has 4 children that are colored with colors #3, #5, #6, #11, this blue node simply transmit in time slots $\{3, 5, 6, 11\}$. This way of scheduling may cause collisions, but it ensures that, in any time slot, all receiving black nodes with corresponding color will receive the message collision-free. This can be proved by the following arguments. Assume that in a time slot, there is a black node w receiving from two blue nodes u, v and assume without loss of generality that u is the parent of w. Since u, v must have different black children that are their destinations, v must have a child w' other than w. Now, v is blue and w, w' are black. Then w and w' must be neighbors in H (H is defined in the tripartite graph model) since v is adjacent to both of them. Therefore, they wouldn't have been colored with the same color. On the other hand, since u, v are transmitting at the same time, w, w' must be of the same color and we finally get a contradiction.

5 Third Broadcast Algorithm - Theoretically the Best

Third algorithm achieves a highly efficient latency $R + O(\sqrt{R} \cdot \log^{1.5} R)$, where R represents the radius of the radio network. Since R is also a lower bound of the optimal solution, the above latency essentially means the approximation ratio is nearly 1, when R is large. In some sense, this means our algorithm is *nearly optimal.* This algorithm combines the first and second algorithms with the broadcast algorithm in [16] as well as some new elements to achieve this result. In addition to its high efficiency, it also has very low redundancy: Each node only needs to broadcast at most once to achieve this result if the first algorithm is used as one of its subroutines.

Phase 1: Virtual backbone construction
We construct MIS layer by layer with the 2-hop separation property and get the set $BLACK$. Then we construct the *Shortest Path Tree* (in G) of this set $BLACK$. Let $T_{sp} = (V_{sp}, E_{sp})$ denote the shortest path tree. We denote the set of non-black nodes in the shortest path tree by $GRAY$, and we also call they gray nodes as well. Note that $V_{sp} = BLACK \cup GRAY$ is a connected dominating set in G, since they form a tree and are therefore connected. V_{sp} can be regarded as a virtual backbone for G.

Phase 2: Broadcast inside virtual backbone
Let H_{vb} denote the induced subgraph of V_{sp} in G. We simply apply the broadcast algorithm in [16] to H_{vb}, s.

Phase 3: Broadcast from virtual backbone to others
Now, since all nodes in V_{sp} have already received the message, all black nodes must have received the message. By the technique of MIS coloring described in § 3.4, we know that we need 12 or 13 more time slots to pass this information to all nodes.

The latency of this algorithm is actually $R_{vb} + O(\sqrt{R_{vb}} \log^2 |V_{sp}|) + 13$ where R_{vb} is the radius of H_{vb}. Since $R_{vb} \leq R$, the broadcast latency is thus $R_{vb} + O(\sqrt{R_{vb}} \log^2 |V_{sp}|)$, using asymptotic notations. The following lemma gives us an upper bound for $|V_{sp}|$.

Lemma 5.1. *$|V_{sp}| = O(R^3)$ where R is the radius of G.*

Proof. Since $V_{sp} = BLACK \cup GRAY$, let $m_1 = |BLACK|$ and $m_2 = |GRAY|$. Since R is the radius, the hop distance between any node and s is at most R, which essentially means the Euclidean distance between them is at most R too. If we consider a disk with radius R centered at s, then all nodes must lie within this disk. We know all black nodes are independent. If, for each black node, we draw a disk with radius $1/2$ center at it, then none of these disks will overlap since the distance between any two black nodes is at least 1. These disks altogether should lie within the disk with radius $R + 1/2$ centered at s, and the sum of their areas is bounded by the area of this big disk. Therefore, $m_1\pi(1/2)^2 \leq \pi(R + 1/2)^2$ and $m_1 = O(R^2)$. To bound m_2, we know that T_{sp} is a shortest path tree. Since the hop distance between any black node and s is at most R, there are at most $R - 2$ intermediate nodes between any shortest path from s to this black node. This means black node contributes at most $R-2$ gray nodes, and m_2 is therefore bounded by $(R - 2)m_1$. Therefore $m_2 = O(R^3)$ and $|V_{sp}| = m_1 + m_2 = O(R^3)$. Actually, $|V_{sp}| \leq 4R^3$ (This can be proved by applying Wegner's Theorem and is omitted here for simplicity.) □

This lemma tells us that by constructing a virtual backbone, the broadcast latency of unit disk graphs can be significantly improved. The latency of third algorithm is $R_{vb} + O(\sqrt{R_{vb}} \log^2 |V_{sp}|)$. Since $R_{vb} \leq R$ and $|V_{sp}| \leq 4R^3$. The latency is at most $R + O(\sqrt{R} \log^2 R)$. If we compare this bound with the latency bound of [16], which is $R + O(\sqrt{R} \log^2 n)$, although they look very similar, there is a significant difference. The latency of our third algorithm depends **solely on R**, which implies that when R is large $O(\sqrt{R} \log^2 R)$ is negligible and the latency is nearly R. Since R is a also a lower bound for broadcast latency, this essential means the approximation ratio is nearly 1 and our algorithm is **nearly optimal**. However, in [16], we still know nothing about their approximation ratio. Actually, if we modify the phase 2 of [16], we can further reduce this latency bound to $R + O(\sqrt{R} \log^{1.5} R)$. Because of the limitation of space, we omit the proof here.

6 Conclusions and Future work

In this paper we presented three broadcast scheduling algorithms in ad hoc networks modeled by unit disk graphs, and all of them have latency significantly lower than any other scheduling algorithms in the literature. In addition, they all have low redundancy as well. The first two algorithms clearly showed that there is some trade-off between latency and redundancy and we believe that we have already balanced them and achieved the maximum overall benefits.

One might think that we took great advantage in the geometrical properties of unit disk graphs, and they might be too ideal to be practical. For example, in real cases, the transmission topology and interference topology are not the same, the range a node could interfere with other nodes may be several times large than its transmission area. Actually, all three of our algorithms can be modified to suit this scenario, except that their approximation ratios may be increased. Moreover, though UDG model is ideal, it is a good place to start with. If we jump into complex models directly without know what advantages we could take in the ideal case, then we cannot get any good results.

Another important point regarding these algorithms is whether they can be implemented in a distributed manner. For first and second algorithms, if location is known, they there can be distributed implementation. However, if location is not known, we need global topology to color MIS nodes in order to get comparable results. Otherwise, the approximation ratios are not that good any more. Our third algorithm cannot be implemented in a distributed fashion either. The reason is that it involves some graph partitioning techniques which entail the knowledge of the whole topology. Although it seems to be a nearly optimal approximation algorithm, this requirement makes it excellent only from a theoretical point of view. How to find an algorithm that can avoid this strong requirement while having comparable performance is the main goal of our follow-up research topic.

References

1. N. Alon, A. Bar-Noy, N. Linial, and D. Peleg. A lower bound for radio broadcast. *Journal of Computer and System Sciences*, 43(2):290–298, 1991.
2. K. M. Alzoubi, P.-J. Wan, and O. Frieder. Message-optimal connected dominating sets in mobile ad hoc networks. In *3rd ACM international symposium on Mobile ad hoc networking & computing–MobiHoc'02*, pages 157–164, New York, NY, USA, 2002. ACM Press.
3. R. Bar-Yehuda, O. Goldreich, and A. Itai. 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(1):104–126, 1992.
4. S. Basagni, I. Chlamtac, and D. Bruschi. A mobility-transparent deterministic broadcast mechanism for ad hoc networks. *IEEE/ACM Transactions on Networking*, 7(6):799–807, 1999.
5. D. Bruschi and M. Del Pinto. Lower bounds for the broadcast problem in mobile radio networks. *Distributed Computing*, 10(3):129–135, 1997.
6. X. Cheng, X. Huang, D. Li, and D. Du. Polynomial time approximation scheme for minimum connected dominating set in ad hoc wireless networks. Technical Report. To appear in *Networks*.
7. I. Chlamtac and A. Faragó. Making transmission schedules immune to topology changes in multi-hop packet radio networks. *IEEE/ACM Transactions Networking*, 2(1):23–29, 1994.
8. I. Chlamtac and S. Kutten. On broadcasting in radio networks–problem analysis and protocol design. *IEEETransactions on Communications*, 33:1240–1246, 1985.
9. I. Chlamtac and O. Weinstein. The wave expansion approach to broadcasting in multihop radio networks. *IEEETransactions on Communications*, 39:426–433, 1991.
10. B. S. Chlebus, L. Gąsieniec, A. Gibbons, A. Pelc, and W. Rytter. Deterministic broadcasting in unknown radio networks. In *Symposium on Discrete Algorithms*, pages 861–870, 2000.
11. B. N. Clark, C. J. Colbourn, and D. S. Johnson. Unit disk graphs. *Discrete Math.*, 86(1-3):165–177, 1990.
12. B. Das and V. Bharghavan. Routing in ad-hoc networks using minimum connected dominating sets. In *ICC (1)*, pages 376–380, 1997.
13. B. Das and V. Bharghavan. Routing in ad-hoc networks using minimum connected dominating sets. In *ICC (1)*, pages 376–380, 1997.

14. M. Elkin and G. Kortsarz. Logarithmic inapproximability of the radio broadcast problem. *Journal of Algorithms*, 52:8–25, 2004.
15. M. Elkin and G. Kortsarz. Polylogarithmic additive inapproximability of the radio broadcast problem. In *7th Int'l Workshop on Approximation Algoorithms for Combinatorial Optimization Problems–APPROX'04*, 2004.
16. M. Elkin and G. Kortsarz. An improved algorithm for radio networks, 2005. An earlier version appeared in *SODA'05*.
17. I. Gaber and Y. Mansour. Centralized broadcast in multihop radio networks. *Journal of Algorithms*, 46(1):1–20, 2003.
18. R. Gandhi, S. Parthasarathy, and A. Mishra. Minimizing broadcast latency and redundancy in ad hoc networks. In *ACM MobiHoc'03*, pages 222–232, 2003.
19. S. Guha and S. Khuller. Approximation algorithms for connected dominating sets. In *European Symposium on Algorithms*, pages 179–193, 1996.
20. J.-H. Ju and V. O. K. Li. An optimal topology-transparent scheduling method in multihop packet radio networks. *IEEE/ACM Transactions on Networking*, 6(3):298–306, 1998.
21. D. R. Kowalski and A. Pelc. Centralized deterministic broadcasting in undirected multi-hop radio networks. In *7th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems–APPROX-RANDOM'04*, pages 171–182, 2004.
22. E. Kushilevitz and Y. Mansour. An $\Omega(D \log(N/D))$ lower bound for broadcast in radio networks. *SIAM Journal on Computing*, 27:702–712, 1998.
23. N. Linial and M. Saks. Decomposing graphs into regions of small diameter. In *2nd annual ACM-SIAM symposium on Discrete algorithms–SODA'91*, pages 320–330, Philadelphia, PA, USA, 1991. Society for Industrial and Applied Mathematics.
24. M. V. Marathe, H. Breu, H. B. Hunt III, S. S. Ravi, and D. J. Rosenkrantz. Simple heuristics for unit disk graphs. *Networks*, 25:59–68, 1995.
25. S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu. The broadcast storm problem in a mobile ad hoc network. In *5th annual ACM/IEEE international conference on Mobile computing and networking–MobiCom'99*, pages 151–162, New York, NY, USA, 1999. ACM Press.
26. S. Ramanathan and E. L. Lloyd. Scheduling algorithms for multihop radio networks. *IEEE/ACM Transactions on Networking*, 1(2):166–177, 1993.
27. A. Sen and M. L. Huson. A new model for scheduling packet radio networks. *Wireless Networks*, 3(1):71–82, 1997.
28. J.-P. Sheu, P.-K. Hung, and C.-S. Hsu. Scheduling of broadcasts in multihop wireless networks. In *The handbook of ad hoc wireless networks*, pages 483–495. CRC Press, Inc., Boca Raton, FL, USA, 2003.
29. R. Sivakumar, B. Das, and V. Bharghavan. Spine routing in ad hoc networks. *Cluster Computing*, 1(2):237–248, 1998.
30. P.-J. Wan, K. M. Alzoubi, and O. Frieder. Distributed construction of connected dominating set in wireless ad hoc networks. *Mobile Networks and Applications*, 9(2):141–149, 2004.
31. G. Wegner. Über endliche kreispackungen in der ebene. *Studia Scientiarium Mathematicarium Hungarica*, 21:1–28, 1986.