Broadcast Scheduling in Interference Environment

By: Scott C.-H. Huang, Peng-Jun Wan, Jing Deng, and Yunghsiang S. Han

S. C.-H. Huang, P.-J. Wan, J. Deng, and Y. S. Han, "Broadcast Scheduling in Interference Environment," *IEEE Transactions on Mobile Computing*, vol. 7, no. 11, pp. 1338-1348, November 2008. DOI: 10.1109/TMC.2008.56

Abstract:

Broadcast is a fundamental operation in wireless networks, and naïve flooding is not practical, because it cannot deal with interference. Scheduling is a good way of avoiding interference, but previous studies on broadcast scheduling algorithms all assume highly theoretical models such as the unit disk graph model. In this work, we reinvestigate this problem by using the 2-Disk and the signal-to-interference-plus-noise-ratio (SINR) models. We first design a constant approximation algorithm for the 2-Disk model and then extend it to the SINR model. This result, to the best of our knowledge, is the *first result* on broadcast scheduling algorithms in the SINR model.

Index Terms: SINR, broadcast, TDMA.

Article:

1 INTRODUCTION

BROADCAST is probably the most fundamental yet challenging operation among all operations of wireless ad hoc networks. The broadcast storm problem [26] tells us that naïve flooding is simply not practical, because it causes severe contention, collision, and congestion. When two or more nodes are transmitting to a node, their signals will interfere with each other, resulting in the receiving node's inability to recognize anything. In the literature, broadcast is often studied in the highly theoretical *Disk Graph* model, in which the transmission and interference range of a node equipped with an omnidirectional antenna is thought of as a disk centered at this node with some radius. Disk graphs in this case are defined as follows: The node set is the set of all transceivers. A directed edge exists from *u* to *v* if *v* lies in *u*'s disk. In addition, if all nodes have the same radius, then the resulting graph is bidirectional, and we can thus use an undirected graph to represent it. This is called the *Unit Disk Graph* model, which has been widely used in the literature. Others use a more generalized *General Graph* model, in which the transmission and interference topology is modeled as a general graph. However, these three models are all overly simplified, and they do not match what actually happens in reality. For example, a node can interfere with a far-away node, and the interference range of a node is generally much larger than its transmission range [16], [17]. None of the three models described earlier can address this issue.

In this paper, we investigate the broadcast problem by using two new models that are much more realistic. First, we use the *2-Disk* model, in which two disks are employed to represent the transmission and interference range, respectively. Then, we use the signal-to-interference-plus- noise-ratio (SINR) model, which deals directly with transmission laws in general physics. SINR is more realistic, as it actually models the case where many far-away nodes could still have the effect of interfering some nodes if they are transmitting simultaneously. This case cannot be dealt with in the 2-Disk model, as no interference whatsoever is assumed when nodes are located outside the interference range. The SINR model gives a more precise analysis in this case, in which the accumulative interference of many nodes outside the interference range should not be neglected. Surprisingly, we found that we can still use the 2-Disk model to deal with this case by carefully selecting the transmission and interference radii. This result, to the best of our knowledge, is **the first result** on broadcast scheduling algorithms in the SINR model.

The rest of this paper is organized as follows: Related work is introduced in Section 2. In Section 3, we formally present our interference models, assumptions, and the definition of the broadcast scheduling problem in both models. We give the preliminaries of tessellation in Section 4, which will be used extensively in later sections. We present our broadcast scheduling algorithms in Section 5, give an example of them in Section 6, and analyze them in Section 7. Simulation results are given in Section 8.

2 RELATED WORK

Broadcast was studied extensively in the literature. Sheu et al. [27] did empirical studies about the efficiency of broadcasting schemes in terms of collision-free delivery, number of retransmissions, and latency. They also designed a centralized distributed broadcast algorithm. Basagni et al. [4] presented a mobility-transparent broadcast scheme for mobile multihop radio networks by using a mobility-transparent schedule that guarantees bounded latency.

Minimum-latency broadcast schedule has been extensively studied in the literature. The prevailing network model in the literature is an arbitrary undirected graph. Let n be the number of nodes in the graph, Δ be the maximum node degree in the graph (i.e., the maximum number of neighbors of a node), and R be the radius of the source in the graph (i.e., the number of hops from the source to the farthest node). Obviously, R is a trivial lower bound on the latency of any broadcast schedule. Alon et al. [1] proved the existence of a family of n-node networks of radius 2, for which any broadcast schedule has latency $\Omega(\log^2 n)$. Chlamtac and Kutten [6] established the NP-hardness of the minimum-latency broadcast schedule in general graphs. Recently, Elkin and Kortsarz have investigated the hardness of approximation for the same problem. In [10], they proved a logarithmic multiplicative inapproximability: unless $NP \subseteq BPTIME(n^{O(\log \log n)})$, $\Omega(\log n)$-approximation of the radio broadcast problem is impossible. In [11], they also proved a polylogarithmic additive inapproximability: unless $NP \subseteq BPTIME(n^{O(\log \log n)})$, there exists a constant c such that there is no polynomial-time algorithm that produces, for every n-node graph G, a broadcast schedule with a latency less than $opt(G) + \log^2 n$, where $opt(G)$ is the optimal broadcast latency for G. Several multiplicative approximation algorithms for minimum-latency broadcast schedule have been proposed in [6], [7], and [20]. Chlamtac and Kutten [6] proposed a broadcasting schedule of latency $O(R\Delta)$. Chlamtac and Weinstein [7] gave the first broadcast schedule whose latency is $O(R \log^2 (n/R))$, where R (the radius of the source) is the lower bound of the broadcast latency. This algorithm is of the best possible order for networks with a constant diameter due to the lower bound obtained in [1]. Kowalski and Pelc [20] improved this result by constructing a broadcast schedule with latency $O(R \log n + \log^2 n)$. For $R = Q(\log n)$, the approximation ratio is $O(\log_2(n = R))$, which is of the best possible order, unless $NP \subseteq BPTIME(n^{O(\log \log n)})$ due to the inapproximability result in [11]. Bar-Yehuda et al. [3] obtained the same result as in [20] earlier, but their solution was a randomized algorithm of Las Vegas type (which means that they cannot guarantee a 100 percent success). Although this is a serious problem in some scenarios, it has great advantage in distributed implementation. A couple of additive approximation algorithms for minimum-latency broadcast schedule have been proposed in [13] and [12]. Gaber and Mansour [13] presented a method consisting of partitioning the underlying graph into clusters. This method improves the time of broadcast, because the existing broadcast schemes can be applied in each cluster separately, and the diameters of clusters are smaller than the diameter of the graph. This method can be used to construct (in polynomial time) a deterministic broadcast scheme working in $O(R + \log^6 n)$ steps by using the broadcast schedule in [7].

It can produce a broadcast scheme with latency $O(R + \log^5 n)$ by using the schedule in [20]. Recently, the clustering method in [13] has been improved by Elkin and Kortsarz [12]. This new clustering method can be used for constructing (in polynomial time) a deterministic broadcast scheme working in $O(R + \log^5 n)$ steps by using the broadcast schedule in [7], and it can produce a broadcast scheme with latency $O(R + \log^4 n)$ if the schedule in [20] is used. This result was reduced to $O(R + \log^3 n)$ by Gasieniec et al. [15]. Very recently, Kowalski and Pelc [19] have further reduced it to $O(R + \log^2 n)$ in [20], which is asymptotically optimal, unless

$NP \subseteq BPTIME(n^{O(\log \log n)})$.

The minimum-latency broadcast schedule in wireless ad hoc networks that are represented by unit disk graphs was only considered in [14] and [9]. Dessmark and Pelc [9] presented a broadcast schedule with a latency of at most 2,400 R. Bruschi and Del Pinto [5] considered distributed protocols and obtained a lower bound of $\Omega(R \log n)$, with the assumption that no nodes know the identities of their neighbors. Kushilevitz and Mansour [21] proved that for any randomized broadcast protocol, there exists a network whose latency is $\Omega(R \log (N/R))$. Chlebus et al. [8] studied deterministic broadcasting without a priori knowledge of the network. They considered two models (with and without collision detection) and designed algorithms for the two models separately. They also established a lower bound $\Omega(R \log n)$ for the scheme without collision detection. Apart from these results on upper or lower bounds, there are also some results on the hardness of approximation of this problem. Gandhi et al. [14] established the NP-hardness of minimum-latency broadcast schedule restricted to unit disk graphs and presented an improved broadcast schedule with a latency of at most 648 R. Huang et al. [18] studied the unit disk graph model and designed two scheduling algorithms that improved the approximation ratio in [14]. In their work, these two algorithms have approximation ratios of 52 and 24, respectively. They also designed a theoretically near-optimal scheduling algorithm, whose latency is bounded by $O(R + R \log^{1.5} R)$. If R is large, then the approximation ratio is nearly 1. This algorithm is nearly optimal for all broadcast scheduling algorithms (in unit disk graphs).

Our work uses the SINR model, so it is also related to those who used this model. Moschibroda et al. [25] considered the problem of scheduling a given topology by using the SINR model. In a network, for any given topology, we may not be able to realize this topology in one time slot if interference is considered. In other words, we need to do scheduling in order to make a topology feasible, and Moscibroda et al. [25] focused on the latency issue. This problem is not directly related to our work, as scheduling a topology is always a one-hop concept, in which there is no relay. In broadcast, a nonsource node cannot transmit a message, unless it has already received from another node. This property makes our work fundamentally different from [25]. Zheng and Barton [28] investigated the theoretical limits of data aggregation. They proved that the data aggregation rates $\Theta((\log n)/n)$ and $\Theta(1)$ are optimal for systems with path-loss exponent α satisfying $2 < \alpha < 4$ and $\alpha > 4$, respectively.

3 INTERFERENCE MODELS, ASSUMPTIONS, AND PROBLEM DEFINITION

In this section, we introduce two interference models, namely, the *2-Disk* and *SINR models*. The descriptions of the 2-Disk model are given as follows: A wireless network is modeled as a set of nodes V arbitrarily located in a 2D euclidean space. Each node is associated with two radii: the transmission radius r_T and the interference radius r_I (where $r_I \geq r_T$). The transmission range of a node v is a disk of radius r_T centered at v, and the interference range of v is a disk of radius r_I centered at v. However, the transmission range is a concept with respect to the transmitting nodes, while the interference range is a concept with respect to the receiving nodes. A node u receives a message successfully from v if and only if u is within v's transmission range and no other nodes are within u's interference range. For simplicity, we assume that all nodes have the same r_T and r_I in the 2-Disk model throughout this paper.[1] Note that the transmission range can now be considered from the receivers' point of view and the interference range can be considered from the transmitters' point of view, since they are equivalent this way.

In the SINR model, a wireless network is also regarded as a set V in a 2D euclidean space. Each node is associated with a transmission power P. For simplicity, we assume that all nodes have the same P. According to

[1] This, of course, limits the proposed algorithms to homogenous networks, where each node has the same transmission range and the same interference range. Interestingly, as we will show later, the same algorithms and transmission schedules can be used in the SINR model, in which the received signal's power is compared to the overall interference and noise level, and no fixed interference range r_I is assumed.

general physics, we know that if a node u is transmitting with power P, the theoretically received signal strength P_v at another node v is given by

$$P_v = \frac{P}{r^\alpha},$$

where r is the distance between u and v, and α is a constant called the path-loss exponent. As commonly assumed [16], the path-loss exponent is greater than two (i.e., $\alpha > 2$). A node v receives a message successfully in a time slot from another node u if and only if the SINR at v is at least a given constant β, where β is called the minimum SINR. The SINR at v is given by

$$SINR_v = \frac{P_v}{N + I_v},$$

where N is the background noise, and I_v is the total interference at v. P_v and I_v are given by

$$P_v = \frac{P}{d(u,v)^\alpha}, I_v = \sum_{w \in T - \{u\}} \frac{P}{d(v,w)^\alpha}.$$

In the above expressions, $d(u; v)$ is the euclidean distance between u and v, and $T \subset V$ is the set of nodes scheduled to transmit in the current time slot. Note that in order for the SINR to make sense, we need to assume that $N + I_v > 0$.

In practice, we further consider the generalized physical model, in which the actually received signal strength P_A can deviate from the theoretical value by a factor of $\theta > 1$ [25], i.e.,

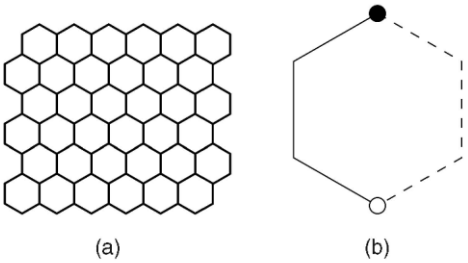

Fig. 1. (a) Hexagonal tessellation. (b) One hexagon.

$$\frac{1}{\theta} \cdot \frac{P}{r^\alpha} < P_A < \theta \cdot \frac{P}{r^\alpha}.$$

We assume that the network is connected. This fundamental assumption has different representations in different models. In the 2-Disk model, it only means that the disk graph generated by V and r_T (i.e., an edge exists between u, v, ⇔ $d(u, v) < r_T$) is connected. However, in the SINR model, it means more. Let u and v be any two nodes with an edge between them in V that is connected. Any successful received message at v means that $SINR_v \geq \beta$. Thus, we have $\frac{P}{d(u,v)^\alpha} > \beta(N + I_v) \geq \beta N$: Equivalently, we can say that there exists a $\gamma > 1$ such that $\frac{\theta P}{\gamma N \beta} = d(u,v)^\alpha$. Letting r' be any distance between two nodes with an edge on them in V, we can make the following assumption on connectivity:

Connectivity assumption. There exists a constant $\gamma > 1$ such that the disk graph generated by V and $r' = \sqrt[\alpha]{\frac{\theta P}{\gamma N \beta}}$ is connected.

Finally, we also make an assumption that every node knows its location. This assumption is strong but essential, since we are considering the SINR model, which is a geometrical concept.

The problem definition for either model is given as follows: Given a set of nodes V and a source $s \in V$, the objective is to find a schedule $\{U_1, U_2, ...\}$satisfying the following requirements: 1) for all i, $U_i \subset V$ represents the set of nodes scheduled to transmit in time slot i, 2) a node cannot be scheduled to transmit, unless it has already received successfully in an earlier time slot (note that the conditions of successful reception are different in those two models), and 3) in the end, all nodes in V receive successfully. *Latency* is the first time slot such that this happens.

4 TESSELLATION OF HEXAGONS

Before presenting the proposed broadcast algorithm, we introduce a tessellation/coloring technique. This technique will be used in our algorithm.

A tessellation of the entire plane is a way of partitioning into equal (or similar) pieces. We partition the plane into hexagons, as shown in Fig. 1a. Each hexagon has radius 1/2 and is half open half closed, with the topmost point included and the bottommost point excluded, as shown in Fig. 1b. We can give many different colorings to this tessellation.

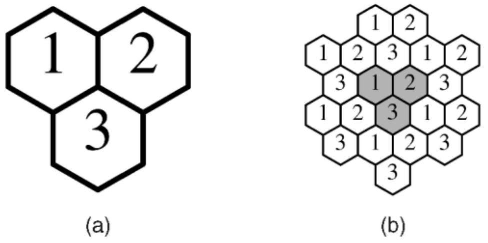

Fig. 2. (a) Three-coloring ($k = 1$). (b) Three-coloring filling up the plane.

Three-coloring is shown in Figs. 2a and 2b. Three hexagons are grouped together, as shown in Fig. 2a, and they can fill up the entire plane, as shown in Fig. 2b. Now, let us look at the three hexagons in Fig. 2a again. If we enclose another layer of hexagons, we get 12 hexagons grouped together, as shown in Fig. 3a. This introduces a 12-coloring, and they fill up the plane, as shown in Fig. 3b. Similarly, we can further enclose much more layers and get a 27-coloring, a 48-coloring, a 75-coloring, and a general 3k2-coloring, as shown in Figs. 4a and 4b and Figs. 5a, 5b, and 5c.

Note that hexagons of the same color in a 3-coloring are separated by at least the distance of one radius, which is 1/2. In a 12-coloring, they are separated by at least the distance of four radii, which is 2. They are separated by 7, 10, and 13 radii in 27-, 48-, and 75-coloring, respectively. In general, hexagons of the same color are separated by at least $3k – 2$ radii (or euclidean distance $\frac{3k-2}{2}$) in a $3k^2$-*coloring*. This can be easily proven by mathematical induction. There are many different ways of coloring these hexagons, and we just consider one of them [22].

5 BROADCAST SCHEDULING ALGORITHM

In this section, we first look at the 2-Disk model and design a broadcast scheduling algorithm of approximation ratio $6\left[\frac{2}{3}\left(\frac{r_I}{r_T}+2\right)\right]^2$, which is a constant. Later, we will show that the SINR model can be reduced to the 2-Disk model, and the same scheduling algorithm can be applied.

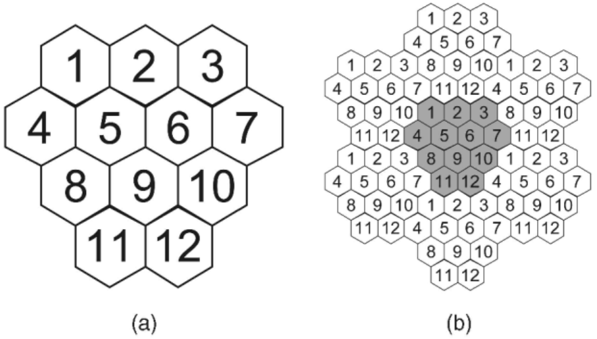

Fig. 3. (a) Twelve-coloring ($k = 2$). (b) Twelve-coloring filling up the plane.

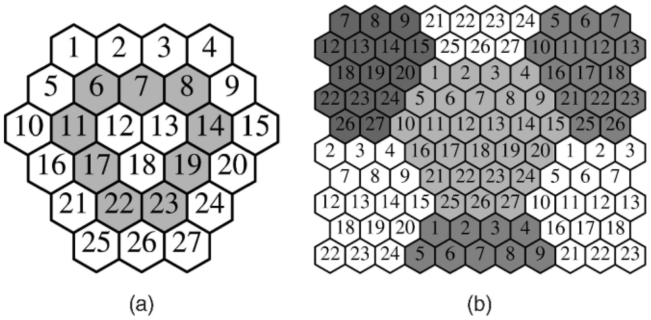

Fig. 4. (a) Twenty-seven-coloring ($k = 3$). (b) Twenty-seven-coloring filling up the plane.

We consider the transmission graph $G_T = (V; E_T)$ generated by rT and V. To define the broadcast schedule, we first need to construct a virtual backbone as follows: We look at G_T and its Breadth First Search (BFS) tree and then divide V into layers L_0; L_1; L_2; . . .; L_R (where R is the radius of G_T and source s). All nodes of layer i are thus i hops away from the root. Then, we construct a layered maximal independent[2] set, called BLACK, as follows: Starting from the zeroth layer, which contains only s, we pick up a maximal independent set (MIS), which contains only s as well. Then, at the first layer, we pick up an MIS in which each node is independent of each other and those nodes at the zeroth layer. Note that this is empty, because all nodes in L_1 (layer 1) must be adjacent to s. Then, we move on to the second layer, pick up an MIS, and mark these nodes black again. Note that the black nodes of the second layer also need to be independent of those of the first layer. We repeat this process until all layers have been worked on. Nodes that are not marked black are marked white at last. Those black nodes are also called the *dominators*, and we will use these two terms interchangeably throughout this paper. The pseudocode of layered MIS construction is given in Algorithm 1.

Algorithm 1: Construct an MIS in G_T layer by layer.
Input: V, s, and G_T

```
$BLACK \leftarrow \emptyset$
**for** $i \leftarrow 0$ to $R$ **do**
   Find an MIS $BLACK_i \subset L_i$, independent of $BLACK$
   $BLACK \leftarrow BLACK \cup BLACK_i$
**end for**
**return** $BLACK$.
```

Now, we construct the virtual backbone as follows: We pick some of the white nodes and color them blue to interconnect all black nodes. Note that $L_0 = \{s\}$ and all nodes in L_1 must be white. We simply connect s to all nodes in L_1. To connect L_1 and L_2, we look at L_2's black nodes. Each black node must have a parent on L_1, and this parent node must be white, since black nodes are independent of each other. We color this white node blue and add an edge between them. Moreover, we know that this blue node must be dominated by a black node either on L_1 or L_0 (in this case, L_0). We then add an edge between this blue node and its dominator.[3] We repeat this process layer by layer and finally obtain the desired virtual backbone (which is a tree) in this manner. Note that in this tree, each black node has a blue parent at the upper layer and each blue node has a black parent at the same layer or the layer right next to it above.

[2] The term independent means "nonadjacent" with respect to GT.
[3] If there is more than one dominator of the blue node, only one needs to be chosen to connect to the blue node.

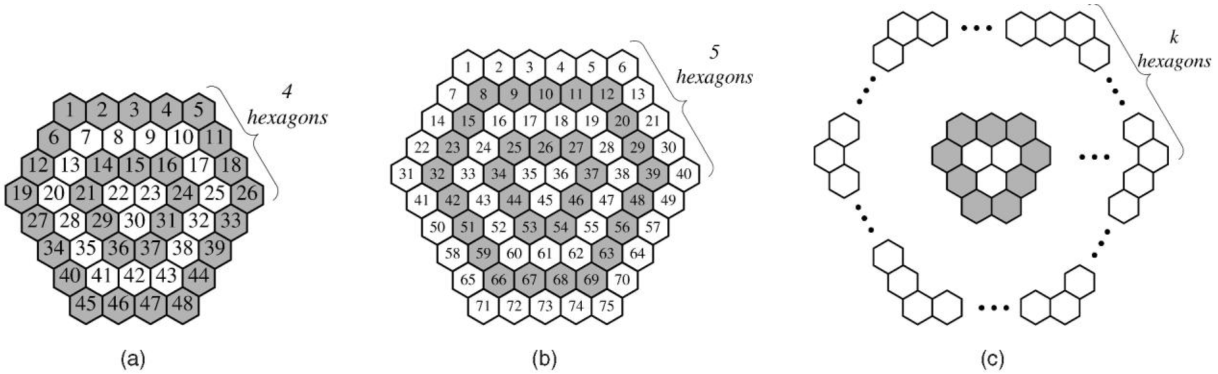

Fig. 5. (a) Forty-eight-coloring ($k = 4$). (b) Seventy-five-coloring ($k = 5$). (c) General $3k^2$-coloring.

The pseudocode is given in Algorithm 2. Note that until now, the construction of the virtual backbone is not related to the 2-Disk model and only the concept of transmission range is used. The concept of interference range is used when we schedule the time slot for each node, which will be explained next, according to the tessellation of hexagons, where enough colors must be used in order to avoid interference.

Algorithm 2: Virtual backbone construction.
Input: V, s, and G_T

```
$T_{vb} = (V; E_{vb})$, $E_{vb} \leftarrow \emptyset$
      ⊳/* Connect the black nodes layer by layer */
$\forall u \in L_1$ add an edge between $u$, $s$
for $i \leftarrow 1$ to $R - 1$ do
    for all black nodes v C BLACKi+1 do
      Find its parent $p(v)$ in $G_T$'s BFS tree
      Color $p(v)$ blue and find its dominator $d_{p(v)}$ in $BLACK_i \cup BLACK_{i-1}$
    Add an edge between $p(v)$, $v$ to $E_{vb}$
    Add an edge between $d_{p(v)}$, $p(v)$ to $E_{vb}$
  end for
end for
      ⊳/* Connect the remaining white nodes */
for all the remaining white nodes $u$ do
Find $u$'s dominator $d_u$
Add an edge between $u$ and $d_u$ to $E_{vb}$
end for
return $T_{vb}$.
```

The broadcast scheduling algorithm based on the virtual backbone in the 2-Disk model is described as follows: Note that the layers of the BFS tree and the virtual backbone may be different. Starting from the zeroth layer containing only the source s, we schedule s to transmit in the first time slot, and obviously, this transmission causes no collision. After the first time slot, all nodes of the first layer will receive successfully. We will design a schedule such that all nodes of the $(i + 1)$th layer receive from the ith layer successfully for $i = 1; 2; \ldots ;$R. We partition the plane into half-open half-closed hexagons of radius[4] $\frac{r_T}{2}$ and give a $3\left\lceil\frac{2}{3}\left(\frac{r_I}{r_T}+2\right)\right\rceil^2$-coloring with proper scaling, as described in Section 4 (in which $k = \left\lceil\frac{2}{3}\left(\frac{r_I}{r_T}+2\right)\right\rceil$). Then, the distance between two hexagons of the same color is at least $r_T + r_I$, which guarantees the validity of the proposed schedule. This schedule has two parts, and in the first part, we schedule each blue node of layer i to transmit in the time slot

[4] The size of hexagons is determined by guaranteeing that not more than one black node is in the same hexagon. rT/2 is thus the largest radius of each hexagon that we can have.

according to its targeted black nodes' colors. If there is more than one targeted black node with the same color, those blue nodes will need to transmit multiple times.[5] For example, suppose that the starting time of the *i*th layer is T_i. If a blue node has three black children of colors 4, 9, and 13, then we schedule it to transmit in time slots $T_i + 4$, $T_i + 9$, and $T_i + 13$. In the second part, we schedule each black node of layer $i + 1$ to transmit in the time slot according to its own color. After these two parts complete, all nodes at layer $i + 1$ receive the broadcast message. The pseudocode of this part is given in Algorithm 3.

Algorithm 3: Broadcast scheduling.
Input: V, s, and virtual backbone T_{vb}

```
1: Tessellate the plane and give a $3\left\lceil\frac{2}{3}\left(\frac{r_I}{r_T}+2\right)\right\rceil^2$-coloring by setting $k=\left\lceil\frac{2}{3}\left(\frac{r_I}{r_T}+2\right)\right\rceil$
2: Schedule $s$ to transmit in time slot 1.
3: $T_{start} \leftarrow 1$
4: for $i \leftarrow 1$ to $R - 1$ do
5:     $\forall u \in BLUE_i$, $\forall w \in \{u\text{'s children}\}$, schedule $u$ to transmit in time slots $T_{start} + color(w)$
6:     $T_{start} \leftarrow T_{start} + 3\left\lceil\frac{2}{3}\left(\frac{r_I}{r_T}+2\right)\right\rceil^2$
7:     $\forall v \in BLACK_{i+1}$, schedule $v$ to transmit in time slot $T_{start} + color(v)$
8:     $T_{start} \leftarrow T_{start} + 3\left\lceil\frac{2}{3}\left(\frac{r_I}{r_T}+2\right)\right\rceil^2$
9: end for.
```

Note that in line 5 of Algorithm 3, each blue node has at most four black children, and therefore, we need at most four time slots. This is because those black children are all independent of each other in G_T, and in the transmission range of any blue node u (i.e., in the disk centered at u with radius r_T), there can be at most five independent nodes, and one of them must be u's parent. Note that the source s does not have any parent, but s is black. So, u cannot be the source. For this reason, each blue node can only have at most four black children.

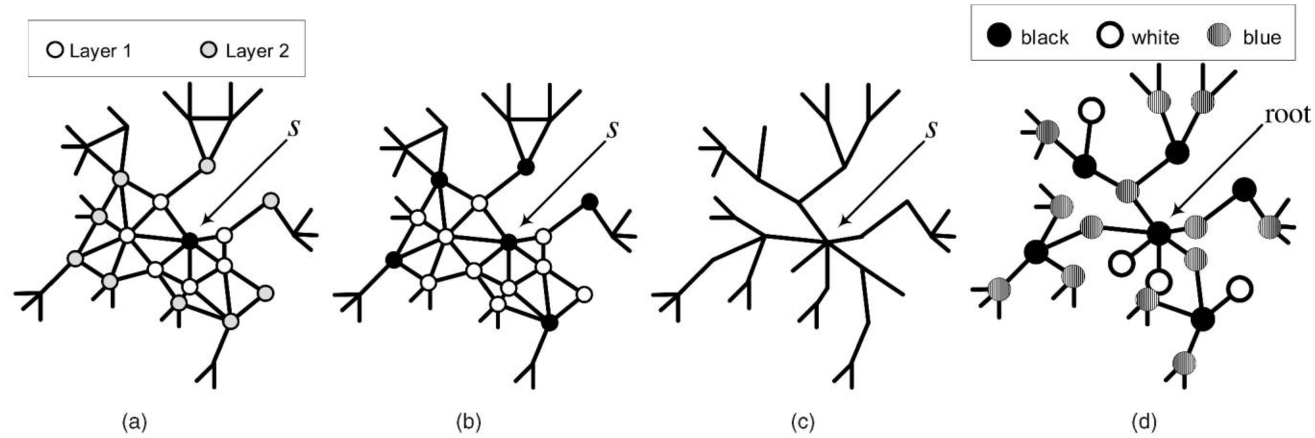

Fig. 6. (a) G_T's topology. (b) Layered MIS. (c) BFS tree. (d) Virtual backbone.

In the SINR model, we simply set

$$r_T = \sqrt[\alpha]{\frac{P}{\gamma\beta\theta N}}, r_I = \sqrt[\alpha]{\frac{24\theta P}{(\gamma-1)N}\left(\frac{2}{\alpha-1}+\frac{1}{\alpha-1}+3\right)}$$

[5] When a blue node sends a message, only the targeted black node is guaranteed to receive successfully, although other children may still be able to receive.

and apply the broadcast scheduling algorithm for the 2-Disk model.[6]

Note that since the proposed algorithm is a centralized algorithm, the source needs to inform each node its time slot to forward the message. However, this initial message forwarding is only performed once in the whole network lifetime. Any inefficient forwarding can be used without increasing the overhead significantly.

6 AN EXAMPLE

Figs. 6a and 6b show the layered construction of MIS, as described in Algorithm 1. Fig. 6a shows the topology of GT. In the first step, the source s is selected in the MIS and is colored black. Note that layer 2 is represented with a light gray color for ease of understanding (this color has nothing to do with the black-blue coloring scheme). In the second step, since the source is black, all nodes at layer 1 must all be white; otherwise, it will not be independent of s. In the third step, we will select an independent set at layer 2, which must also be independent of the nodes at the previous layer, i.e., layer 1, though there is no black node at layer 1, and this does not have any effect. Fig. 6b shows that five more black nodes were selected at layer 2. We keep doing this and select black nodes until all layers have been worked on. The black-node selection depends on GT only, and it has nothing to do with the BFS tree. Not until blue nodes are being selected do we need to consider the BFS tree, as shown in Fig. 6c. In Algorithm 2, we are trying to add appropriate blue nodes to interconnect all black ones. Since the source does not have an upper layer and there are no black nodes at layer 1, we start from layer 2 directly. For each black node at layer 2, we color it blue and connect to its parent in the BFS tree, as shown in Fig. 6d. In Fig. 6d, we see that four nodes at layer 1 are colored blue and are connected to some black nodes at layer 2. Nodes that are not colored blue remain white, and there are two white nodes. We also connect these four blue nodes and two white nodes to the source *s*, since they are dominated by *s*. We keep working on layer 3. For simplicity, suppose that we have already found the black nodes at layer 3 and their corresponding blue nodes at layer 2. Fig. 6d shows that there are three blue nodes at layer 2 that are connected to their black children at layer 3. Note that there are nine nodes at layer 2, in which five are black, three are blue, and the remaining node is still white. Now, for each blue or white node at layer 2, we know that it must be adjacent to at least one black node either at layer 2 or layer 1, since $BLACK_2$ is an MIS. Because of its maximality, all nodes at layer 2 must be adjacent to at least one black node at the same layer or the previous layer. Therefore, for each blue/ white node at layer 2, we find a black node either at layer 1 or layer 2 and connect to it, as shown in Fig. 6d. We keep doing this for all layers, and the virtual backbone will be constructed this way.

We present an example of broadcast scheduling in the 2-Disk model, as shown in Fig. 7. Assume that $r_I/r_T = 3$. $3\left\lceil\frac{2}{3}\frac{r_I}{r_T}+2)\right\rceil^2 = 48$ colors should be used to separate the transmission schedules of these hexagon cells (k = 4) and we give a 48-coloring. In Fig. 7, a virtual backbone has already been constructed according to Algorithm 2. The root (source) is black, and all nodes at layer 1 are either blue or white (four are blue, and two are white). The blue nodes at layer 1 are chosen to connect the black nodes at layer 2, and the remaining are white. At layer 3, there are five black nodes, two blue nodes, and one white node. We explain the broadcast schedule of our scheme, according to Algorithm 3, as follows:

1. The source transmits in time slot 1 and sets $T_{start} – 1$.
2. The four blue nodes at layer 1 are scheduled according to their black children's color. Therefore, the first node transmits in time slots $T_{start} + 24 = 25$ and $T_{start} + 25 = 26$, the second node transmits in time slot $T_{start} + 26 = 27$, the third node transmits in $T_{start} + 31 = 32$, and the last node transmits in $T_{start} + 39 = 40$. Note that the first node transmits in two time slots, because it has two black children. The white nodes do not transmit at all. All other time slots between [$T_{start} + 1$; $T_{start} + 48 + 1$] are idle.
3. T_{start}, $T_{start} + 48 = 49$.

[6] It will be explained in detail in Section 7.

4. At layer 3, there are five black nodes of colors 24, 25, 26, 31, and 39. Their transmission time slots are T_{start} + 24 = 73, T_{start} + 25 = 74, T_{start} + 26 = 75, T_{start} + 31 = 80, and T_{start} + 39 = 88, respectively.
5. Set T_{start}, T_{start} + 48 = 97, and by this time, all nodes at layer 3 should have already received the message successfully.
6. We keep scheduling in this manner until all nodes at layer R receive the message successfully and the broadcast finishes.

7 ANALYSIS

Theorem 7.1. *Algorithm* 3 *is a valid scheduling algorithm.*

Proof. We prove two assumptions: 1) each node will receive successfully before it is scheduled to transmit and 2) in the end, all nodes receive successfully. Algorithm 3 begins with the source's transmission, and since there is only one node transmitting, there will be no collision, and all nodes of L_1 will receive successfully. Now, we prove that all nodes of L_{i+1} will receive successfully from L_i for all $1 < i < R - 1$. First, we show that all nodes of $BLACK_{i+1}$ will receive successfully from $BLUE_i$. This is straightforward. Assume the contrary if there exists a receiver $v \in BLACK_{i+1}$ such that another node $w \in BLUE_i$ is interfering with the sender $u \in BLUE_i$. If this happens, we know that $d(u; v) < r_T$ and $d(w; u) < r_I$. This implies $d(v; w) < r_T + r_I$, contradicting to the fact that any two hexagons of the same color must be at least $r_T + r_I$ apart. Second, we show that all nodes of $L_{i+1} - BLACK_{i+1}$ must receive successfully from $BLACK_{i+1}$. This is also straightforward by using similar arguments. Assume the contrary: if there is a node $v \in L_{i+1} - BLACK_{i+1}$ such that another node $w \in BLACK_{i+1}$ is interfering with the sender $u \in BLACK_{i+1}$, then similarly, $d(u; v) < r_T$ and $d(w; u) < r_I$, implying $d(v; w) < r_T + r_I$, and we get a contradiction. ■

Theorem 7.2. *Algorithm* 3 *has latency (the total number of time slots for completing the broadcast procedure)* $1 + \left(6\left\lceil\frac{2}{3}\left(\frac{r_I}{r_T} + 2\right)\right\rceil^2\right)(R - 1)$.

Proof. We study the "for" loop in Algorithm 3. Inside the loop, first, we schedule the blue nodes according to their black children's colors, which takes $3\left\lceil\frac{2}{3}\left(\frac{r_I}{r_T} + 2\right)\right\rceil^2$ time slots, since we use $3\left\lceil\frac{2}{3}\left(\frac{r_I}{r_T} + 2\right)\right\rceil^2$ colors to construct the tessellation. Then, we schedule the black nodes to transmit according to their colors. Therefore, it takes $3\left\lceil\frac{2}{3}\left(\frac{r_I}{r_T} + 2\right)\right\rceil^2$ time slots as well. As a result, each iteration of the for loop takes $6\left\lceil\frac{2}{3}\left(\frac{r_I}{r_T} + 2\right)\right\rceil^2$time slots, and there are R — 1 iterations. Along with the source's time slot in the beginning, the overall latency is $1 + \left(6\left\lceil\frac{2}{3}\left(\frac{r_I}{r_T} + 2\right)\right\rceil^2\right)(R - 1)$. ■

Having the above latency bound and that R is itself a lower bound for any broadcast schedule, we can get the following corollary.

Corollary 7.1. *The broadcast scheduling algorithmfor the* 2*-Disk model is a constant approximation algorithm with ratio* $6\left\lceil\frac{2}{3}\left(\frac{r_I}{r_T} + 2\right)\right\rceil^2$.

It is easy to see that the approximation ratio of the proposed algorithm is only related to the physical transmission characters. That is, the approximation ratio of the proposed algorithm only depends on the ratio of the interference range to the transmission range. When these two ranges are similar, the approximation ratio becomes 24, no matter how many nodes are in the network. In a large network, the proposed algorithm can broadcast the message efficiently.

It is obvious that there are many idle time slots in the proposed scheduling algorithm. In practice, we can delete all idle time slots and reindex all scheduling of nodes. We will show by simulation that it can reduce the latency by up to 86 percent.

Theorem 7.3. *In the SINR model, if we set r_T, r_I as*

$$r_T = \sqrt[\alpha]{\frac{P}{\gamma\beta\theta N}}, r_I = \sqrt[\alpha]{\frac{24\theta P}{(\gamma - 1)N}\left(\frac{2}{\alpha - 2} + \frac{1}{\alpha - 1} + 3\right)}$$

and we use Algorithm 3 *to schedule the transmissions, then the overall interference at any intended receiver* (*i.e., the node that is scheduled to receive at this time*) *at any time is strictly less than* (γ – 1)*N.*

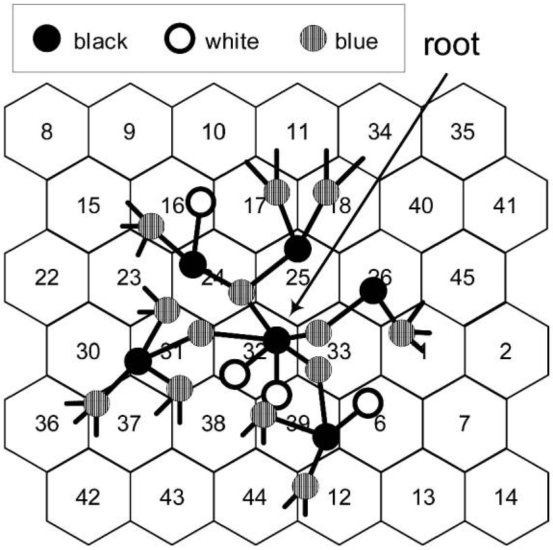

Fig. 7. An example of broadcast scheduling in the 2-Disk model, with $r_I/r_T = 3$.

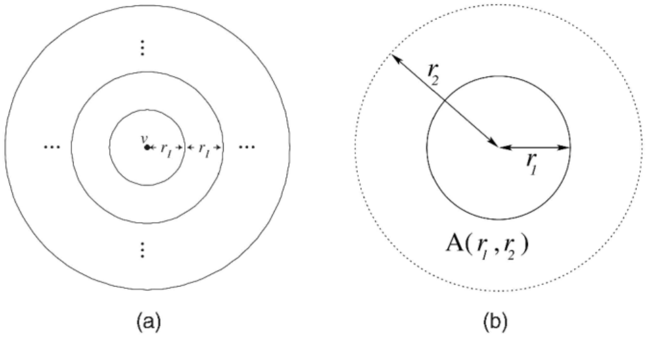

Fig. 8. (a) Concentric disks at v. (b) Annulus $A(r_1, r_2)$.

Proof. Since we use Algorithm 3, we know that at any time, the distance between two simultaneously transmitting nodes is at least $r_T + r_I$, because any two hexagons of the same color must be at least $r_T + r_I$ apart. Moreover, let u be a sender and let v be its intended receiver at any time in Algorithm 3. Then, there will be no other sender that is transmitting simultaneously and whose distance to v is less than r_I. This is true, because r_I is the interference radius, and we have avoided this situation in Algorithm 3. Now, let us pick up an intended receiver v and consider its concentric circles of radii r_I; $2r_I$; $3r_I$;..., as shown in Fig. 8a. Here, we use $A(r_1; r_2)$ to denote the annulus between two concentric circles of radii r_1 and r_2 ($r_1 < r_2$), as shown in Fig. 8b. We define $A(r_1; r_2)$ to be inner closed outer open (i.e., $A(r_1; r_2)$ contains the circle of radius r_1 but does not contain the circle of radius r_2). Now, we consider $A((i — 1)r_I; ir_I)$. We also consider the senders scheduled to transmit simultaneously at a fixed time. Let M_i be the number of these senders in $A((i – 1)r_I; ir_I)$. We know that the distance between any two black nodes is at least $r_T + r_I$. Moreover, since each blue sender is at most r_T from its black receiver, the distance between any two blue senders is at least $r_I – r_T$. Therefore, the distance between any two senders is at least $r_I – r_T$. If we draw an open disk of radius r_I 2 r_T at each sender in $A((i – 1) r_I; ir_I)$, then these disks will not overlap at all. Moreover, all of these disks will be completely contained in $A\left((i - 1)r_I - \frac{r_I - r_T}{2}, ir_I + \frac{r_I - r_T}{2}\right)$. Therefore, by comparing their areas, we know that

$$\pi\left(\frac{r_I - r_T}{2}\right)^2 \cdot M_i < \pi\left\{\left[ir_I + \frac{r_I - r_T}{2}\right]^2 - \left[(i - 1)r_I - \frac{r_I - r_T}{2}\right]^2\right\}$$

and that

$$M_i < \frac{4(2i-1)r_I(2r_I-r_T)}{(r_I-r_T)^2}. \qquad (1)$$

Since the distance between v and any point in $A((i-1)r_I; ir_I)$ is at least $(i-1)r_I$, the cumulative interference caused by sender in $A((i-1)r_I; ir_I)$ is bounded by $M_i \frac{\theta P}{((i-1)r_I)^\alpha}$ and the overall interference I_{total} at v caused by all senders on the entire plane is bounded by

$$I_{total} \leq \sum_{i=2}^{\infty} M_i \frac{\theta P}{\left((i-1)r_I\right)^\alpha}$$

Here, i starts from 2, because, except for the intended sender, no other interfering senders are within the disk centered at v with radius r_I. Plugging in (1), we know that I_{total} is less than

$$\sum_{i=2}^{\infty} \frac{4(2i-1)r_I(2r_I-r_T)}{(r_I-r_T)^2} \frac{\theta P}{\left((i-1)r_I\right)^\alpha}. \qquad (2)$$

Now, let q be defined as follows:

$$q = \frac{r_I}{r_T} = \sqrt[\alpha]{\frac{24\gamma\beta\theta^2}{\gamma-1}\left(\frac{2}{\alpha-2}+\frac{1}{\alpha-1}+3\right)}.$$

Then, (2) becomes

$$\begin{aligned} I_{total} &< \sum_{i=2}^{\infty} \frac{4(2i-1)q(2q-1)}{(q-1)^2} \cdot \frac{\theta P}{(i-1)^\alpha r_I^\alpha} \qquad (3) \\ &= \frac{4q(2q-1)}{(q-1)^2} \cdot \frac{\gamma\beta\theta^2 N}{q^\alpha} \sum_{i=2}^{\infty} \frac{2i-1}{(i-1)^\alpha}. \end{aligned}$$

Equation (3) is obtained by plugging in

$$r_I = q \cdot r_T = q \sqrt[\alpha]{\frac{P}{\gamma\beta\theta N}}.$$

In (3)

$$\begin{aligned} \sum_{i=2}^{\infty} \frac{2i-1}{(i-1)^\alpha} &= \sum_{i=2}^{\infty} \left[\frac{2(i-1)}{(i-1)^\alpha} + \frac{1}{(i-1)^\alpha}\right] \\ &= 2\sum_{i=2}^{\infty} \frac{1}{(i-1)^{\alpha-1}} + \sum_{i=2}^{\infty} \frac{1}{(i-1)^\alpha} \end{aligned}$$

$$= 2\sum_{j=1}^{\infty}\frac{1}{j^{\alpha-1}} + \sum_{j=1}^{\infty}\frac{1}{j^{\alpha}}.$$

From elementary calculus, we know that

$$\sum_{j=1}^{\infty}\frac{1}{j^{\alpha}} \leq \frac{1}{\alpha-1} + 1, \text{plugging this in} \Rightarrow$$

$$\sum_{i=2}^{\infty}\frac{2i-1}{(i-1)^{\alpha}} \leq \frac{2}{\alpha-2} + \frac{1}{\alpha-1} + 3. \qquad (4)$$

Also, in (3), the term

$$\frac{4q(2q — 1)}{(q-1)^2} \text{ is strictly increasing in } (1;\ \infty).$$

In practice, q, i.e., the ratio of interference radius to transmission radius, is 3 ~ 5, and we could assume $q \geq 2$ to obtain

$$\frac{4q(2q — 1)}{(q-1)^2} \leq 6.$$

Plugging in (4) and the above expression into (3), we obtain

$$I_{total} < \frac{24\gamma\beta\theta^2 N}{q^{\alpha}}\left(\frac{2}{\alpha-2} + \frac{1}{\alpha-1} + 3\right) = (\gamma-1)N,$$

since $q = \sqrt[\alpha]{\frac{24\gamma\beta\theta^2 N}{q^{\alpha}}\left(\frac{2}{\alpha-2} + \frac{1}{\alpha-1} + 3\right)}$. This theorem is thus proven. ■

Corollary 7.2. *The SINR at any intended receiver at any time is strictly greater than* 0.

Proof. At any intended receiver, the signal strength is at least $\frac{P}{\theta r^{\alpha}}$, where r is the distance between the designated sender and its intended receiver, and $r < r_T$. Therefore, the signal strength is at least $\frac{P}{r_T^{\alpha}} = \frac{P}{\theta(P/\gamma\beta\theta N)} = \gamma\beta N$. Theorem 7.3 tells us that the overall interference is strictly less than $(\gamma - 1)N$, so the SINR at any intended , receiver is strictly greater than $\frac{\gamma\beta N}{(\gamma-1)N+N} = \gamma\beta$. Remember that we have made the connectivity assumption in Section 3, in which the disk graph generated by V and $\sqrt[\alpha]{\frac{\theta P}{\gamma N\beta}}$ is connected. ■

Corollary 7.2 tells us that Algorithm 3 is also a valid scheduling algorithm for the SINR model.

Corollary 7.3. *Our broadcast algorithmfor the SINR model has a latency that is bounded by*

$$1 + 6\left\lceil\frac{2}{3}\left(\sqrt[\alpha]{\frac{24\gamma\beta\theta^2 N}{q^{\alpha}}\left(\frac{2}{\alpha-2} + \frac{1}{\alpha-1} + 3\right)} + 2\right)\right\rceil^2 (R-1).$$

Note that the number of colors depends on $r_I = r_T$ and not on the number of nodes. Also, broadcast latency is invariant of the number of nodes. This is because we applied the technique of constructing a virtual backbone, which plays a vital role in coloring. The number of nodes in this virtual backbone directly affects the latencies, and it is not affected by the number of nodes in the whole network.

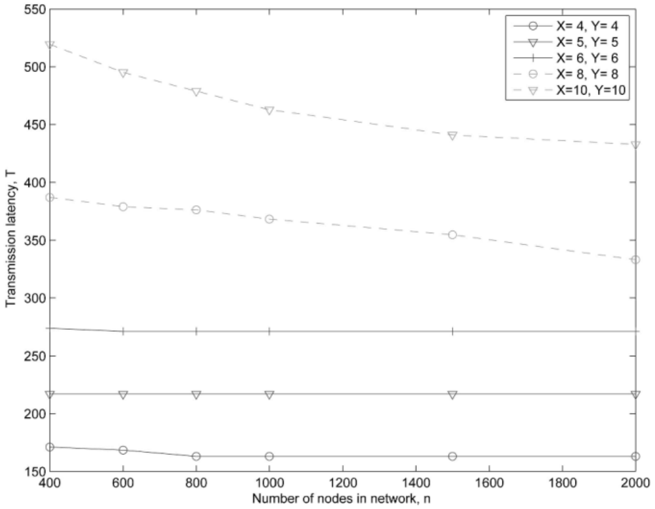

Fig. 9. Transmission latency for different network area sizes ($k = 3$).

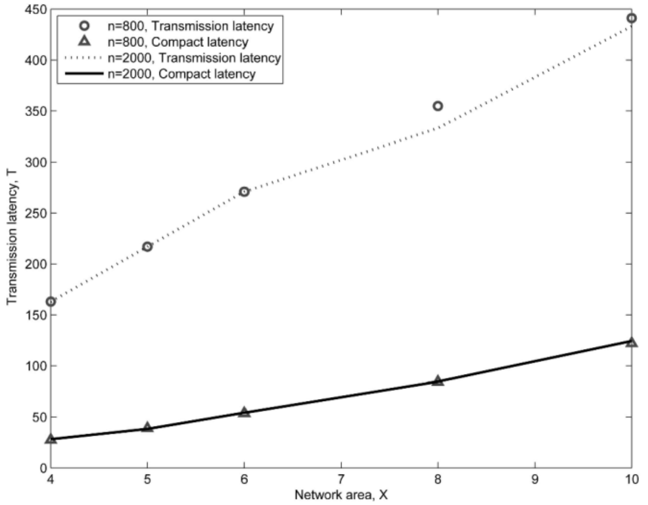

Fig. 10. Transmission latency for different numbers of nodes ($k = 3$).

8 SIMULATION RESULTS

Simulations have been performed in Matlab to evaluate the latency of our proposed scheme. In these simulations, n nodes were distributed randomly into a square region of size *X* by *Y*, where *X* and *Y* are normalized to the transmission range r_T. The transmission latency was then measured after our proposed scheme is employed. We measured two different latencies in our simulations:

- The *transmission latency*, based on Theorem 7.3, can be easily found when the maximum depth of the BFS tree is identified.
- The *compact transmission latency* is a shorter latency in which all idling time slots are removed.

Note that the compact latency measurements were based on the assumption that such removal of idling time slots is possible, which requires some extra communication between nodes in different BFS tree depths.

Fig. 9 shows the transmission latency as a function of the number of nodes in the network *n* for different network area sizes *X*. The value of *k* was set to 3 in these simulations. In Fig. 9, the transmission latencies remain almost the same when the number of nodes in network *n* is larger than 1,000 for each set of *X* and *Y*. This is actually expected: the increase in *n* does not change the transmission tessellation and its depth significantly (as discussed in Section 7). As the network size increases, the transmission latency becomes longer. This is because of the increased depth of the virtual backbone.

Fig. 10 shows the two types of transmission latency as a function of network area sizes *X* for different numbers of nodes in the network *n*. The value of *k* was set to 3 in these simulations. It can be seen that compact latency is much shorter than the transmission latency due to the existence of many idling time slots in this setting.

We compare the compact transmission latency in different network regions in Fig. 11. As the network region size increases, the compact transmission latency increases as well.

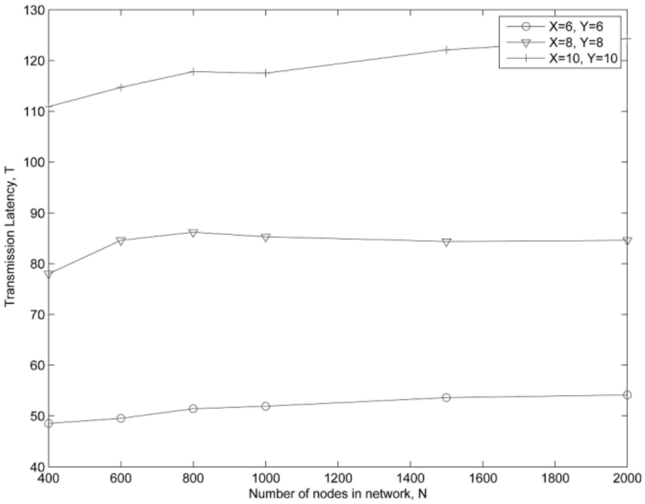

Fig. 11. Comparing the compact transmission latency ($k = 3$).

Remarks on distributed implementation. Our algorithm can be modified into a distributed version for the following reason. It makes use of the following centralized information:

1. layer information in Algorithm 1,
2. MIS in Algorithm 1,
3. BFS tree in Algorithm 2, and
4. color information in Algorithm 3.

In 1, each node only needs to know its layer number. In 2, each node only needs to know whether or not it is in the MIS. In 3, each node only needs to know its parent in the BFS tree. In 4, each node only needs to know its color. Lists 1 and 2 have distributed algorithms, because there are distributed BFS algorithms [2]. List 3 is related to the MIS, and there are also distributed MIS algorithms in the literature [23], [24]. However, we need to modify those algorithms slightly and apply them layer by layer. List 4 could have distributed implementations, provided that each node knows its location. This may be possible if each node has a GPS device, for example, or each node is given the location information when it is deployed.

Remarks on varying r_T and r_I. Varying the values of transmission/interference ranges does not affect our algorithm; it only affects the following: 1) graph topologies G_T and G_I and 2) coloring (since $k = \left\lceil \frac{2}{3}(\frac{r_I}{r_T} + 2) \right\rceil$ depends on them). From a practical point of view, varying the values of transmission/interference ranges only affects certain system parameters; it does not affect any algorithms/subroutines.

9 CONCLUSION AND FUTURE WORK

Many highly theoretical models were used in all previous works on broadcast scheduling. Instead, we have used two more practical models for reinvestigating this problem. Surprisingly, we have found that we can apply the same method to both models and obtain low-latency schedules. Although our proposed algorithms are centralized, we did not formulate the minimum latency problem as an optimization problem (such as linear programming) and find the optimal solution for the following reasons. First, this problem in general graph model was proposed in [6] in 1985, and so far, there is still no good formulation for representing it as a linear programming problem. The main reason for that is the difficulty of representing the condition that “a node can only transmit if it has successfully received from another node.” So far, there is still no good formulation for representing this condition, even in the general graph model, so we believe that it is more difficult to represent it in our more complicated 2-Disk and SINR models. Second, the broadcast latency problem in the disk graph has proven to be NP-hard [14], and this problem in our 2-Disk and SINR models can be regarded as a more general case and is therefore also NP-hard. For this reason, finding an optimal solution is difficult.

For future work, there are two promising directions as follows: The first direction is to apply our techniques to directional antennae. We believe that most techniques developed here can be applied to the case of directional antennae by reinvestigating their geometrical properties, although the models may need to be redefined accordingly. The second direction is to apply these techniques to data aggregation (or converge cast) scheduling. In such a scenario, all nodes wish to transmit their data back to a fixed sink node. This could be regarded as a reverse- direction broadcast. The major difference is that in a broadcast, a node can transmit to many nodes at the same time, while in a data aggregation, many nodes cannot transmit to one sink in one time slot. This property makes data aggregation fundamentally different from broadcast, but we believe that we can still apply several techniques that have been developed in this work. For these reasons, we believe this work will be an important start that bridges the gap between theory and practice.

ACKNOWLEDGMENTS

The authors would like to thank the associate editor, Prof. M. Liu, and the three anonymous reviewers for their insightful comments. The work of Scott C.-H. Huang was supported in part by the Research Grant Council of Hong Kong under the CERG CityU 113807 Project. The work of Peng-Jun Wan is supported in part by the Research Grants Council of Hong Kong under the CityU 114307, CityU 122105, and CityU 122807 Projects. The work of Jing Deng was supported in part by Louisiana Board of Regents RCS under Grant LEQSF (2005-08)-RD-A-43.

References:

[1] N. Alon, A. Bar-Noy, N. Linial, and D. Peleg, “A Lower Bound for Radio Broadcast,” J. Computer and System Sciences, vol. 43, no. 2, pp. 290-298, 1991.

[2] B. Awerbuch and R. Gallager, “A New Distributed Algorithm to Find Breadth First Search Trees,” IEEE Trans. Information Theory, vol. 33, no. 3, pp. 315-322, 1987.

[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,” J. Computer and System Sciences, vol. 45, no. 1, pp. 104-126, 1992.

[4] S. Basagni, I. Chlamtac, and D. Bruschi, “A Mobility-Transparent Deterministic Broadcast Mechanism for Ad Hoc Networks,” IEEE/ ACM Trans. Networking, vol. 7, no. 6, pp. 799-807,1999.

[5] D. Bruschi and M. Del Pinto, “Lower Bounds for the Broadcast Problem in Mobile Radio Networks,” Distributed Computing, vol. 10, no. 3, pp. 129-135, 1997.

[6] I. Chlamtac and S. Kutten, “On Broadcasting in Radio Networks: Problem Analysis and Protocol Design,” IEEE Trans. Comm., vol. 33, pp. 1240-1246, 1985.

[7] I. Chlamtac and O. Weinstein, “The Wave Expansion Approach to Broadcasting in Multihop Radio Networks,” IEEE Trans. Comm., vol. 39, pp. 426-433, 1991.

[8] B.S. Chlebus, L. Gasieniec, A. Gibbons, A. Pelc, and W. Rytter, “Deterministic Broadcasting in Unknown Radio Networks,” Proc. 11th Ann. ACM-SIAM Symp. Discrete Algorithms (SODA ’00), pp. 861-870, 2000.

[9] A. Dessmark and A. Pelc, “Tradeoffs between Knowledge and Time of Communication in Geometric Radio Networks,” Proc. 13th Ann. ACM Symp. Parallel Algorithms and Architectures (SPAA ’01), pp. 59-66, 2001.

[10] M. Elkin and G. Kortsarz, “Logarithmic Inapproximability of the Radio Broadcast Problem,” J. Algorithms, vol. 52, pp. 8-25, 2004.

[11] M. Elkin and G. Kortsarz, “Polylogarithmic Additive Inapproxim- ability of the Radio Broadcast Problem,” Proc. Seventh Int’l Workshop Approximation Algorithms for Combinatorial Optimization Problems (APPROX-RANDOM), 2004.

[12] M. Elkin and G. Kortsarz, “An Improved Algorithm for Radio Networks,” Proc. 16th Ann. ACM-SIAM Symp. Discrete Algorithms (SODA), 2005.

[13] I. Gaber and Y. Mansour, “Centralized Broadcast in Multihop Radio Networks,” J. Algorithms, vol. 46, no. 1, pp. 1-20, 2003.

[14] R. Gandhi, S. Parthasarathy, and A. Mishra, “Minimizing Broad- cast Latency and Redundancy in Ad Hoc Networks,” Proc. ACM MobiHoc ’03, pp. 222-232, 2003.

[15] L. Gasieniec, D. Peleg, and Q. Xin, “Faster Communication in Known Topology Radio Networks,” Proc. 24th Ann. ACM Symp. Principles of Distributed Computing (PODC ’05), pp. 129-137, 2005.

[16] P. Gupta and P.R. Kumar, “Capacity of Wireless Networks,” IEEE Trans. Information Theory, vol. 46, no. 2, pp. 388-404, 2000.

[17] P. Gupta and P.R. Kumar, “Internets in the Sky: The Capacity of Three-Dimensional Wireless Networks,” Comm. Information and Systems, vol. 1, no. 1, pp. 33-50, 2001.

[18] S.C.-H. Huang, P.-J. Wan, X. Jia, and H. Du, “Low-Latency Broadcast Scheduling Schemes in Ad Hoc Networks,” Proc. First Int’l Conf. Wireless Algorithms, Systems, and Applications (WASA ’06), Aug. 2006.

[19] D.R. Kowalski and A. Pelc, “Optimal Deterministic Broadcasting in Known Topology Radio Networks,” Distributed Computing, 2007.

[20] D.R. Kowalski and A. Pelc, “Centralized Deterministic Broad- casting in Undirected Multi-Hop Radio Networks,” Proc. Seventh Int’l Workshop Approximation Algorithmsfor Combinatorial Optimiza- tion Problems (APPROX-RANDOM ’04), pp. 171-182, 2004.

[21] E. Kushilevitz and Y. Mansour, “An Q*(*D logðN=DÞÞ Lower Bound for Broadcast in Radio Networks,” SIAM 1. Computing, vol. 27, pp. 702-712, 1998.

[22] R.A. Leese, “A Unified Approach to the Assignment of Radio Channels on a Regular Hexagonal Grid,” IEEE Trans. Vehicular Technology, vol. 46, no. 4, pp. 968-980, 1997.

[23] M. Luby, “A Simple Parallel Algorithm for the Maximal Independent Set Problem,” SIAM 1. Computing, vol. 21, no. 1, pp. 1036-1053, 1986.

[24] T. Moscibroda and R. Wattenhofer, “Maximal Independent Sets in Radio Networks,” Proc. 24th Ann. ACM Symp. Principles of Distributed Computing (PODC ’05), pp. 148-157,2005.

[25] T. Moscibroda, R. Wattenhofer, and A. Zollinger, “Topology Control Meets SINR: The Scheduling Complexity of Arbitrary Topologies,” Proc. ACM MobiHoc ’06, pp. 310-321, 2006.

[26] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu, “The Broadcast Storm Problem in a Mobile Ad Hoc Network,” Proc. ACM MobiCom ’99, pp. 151-162,1999.

[27] J.-P. Sheu, P.-K. Hung, and C.-S. Hsu, “Scheduling of Broadcasts in Multihop Wireless Networks,” The Handbook of Ad Hoc Wireless Networks. CRC Press, pp. 483-495, 2003.

[28] R. Zheng and R. Barton, “Toward Optimal Data Aggregation in Random Wireless Sensor Networks,” Proc. IEEE INFOCOM ’07, pp. 249-257, 2007.