

Approximation Algorithms for Interference Aware Broadcast in Wireless Networks

Dianbo Zhao
University of Wollongong
Email: dz985@uowmail.edu.au

Kwan-Wu Chin
University of Wollongong
Email: kwanwu@uow.edu.au

Abstract—Broadcast is a fundamental operation in wireless networks and is well supported by the wireless channel. However, the interference resulting from a node's transmission pose a key challenge to the design of any broadcast algorithms/protocols. In particular, it is well known that a node's interference range is much larger than its transmission range and thus limits the number of transmitting and receiving nodes, which inevitably prolong a broadcast. To this end, a number of past studies have designed broadcast algorithms that account for this interference range with the goal of deriving a broadcast schedule that minimizes latency. However, these works have only taken into account interference that occurs within the transmission range of a sender. Therefore, the resulting latency is non-optimal given that collision occurs at the receiver. In this paper, we address the Interference-Aware Broadcast Scheduling (IABS) problem, which aims to find a schedule with minimum broadcast latency subject to the constraint that a receiver is not within the interference range of any senders. We study the IABS problem under the protocol interference model, and present a constant approximation algorithm, called IABBS, and its enhanced version, IAEBS, that produces a maximum latency of at most $2 \left\lceil \frac{\pi}{\sqrt{3}}(\alpha + 1)^2 + (\frac{\pi}{2} + 1)(\alpha + 1) + 1 \right\rceil R$, where α is the ratio between the interference range and the transmission range, i.e., $\alpha \geq 1$, and R is the radius of the network with respect to the source node of the broadcast. We evaluated our algorithms under different network configurations and confirmed that the latencies achieved by our algorithms are much lower than existing schemes. In particular, compared to CABS, the best constant approximation broadcast algorithm to date, the broadcast latency achieved by IAEBS is $\frac{5}{8}$ that of CABS.

I. INTRODUCTION

The use of broadcast to disseminate a message from a source to all other nodes is a fundamental operation relied upon by a number of network protocols and applications, including route discovery, software updates and so forth [13]. Moreover, applications such as those used for military surveillance, emergency disaster relief and environmental monitoring may have time constraints that require strict bound on end-to-end latency. Alternatively, due to resource constraints, there may be a limit on the number of broadcast transmissions performed by nodes. However, due to interference, achieving low latency broadcast is a key challenge in wireless networks. When a node v receives a message from one of its neighbors, say node u , node v may fail to receive the message due to the interference from other transmitting nodes. More specifically, there are two types of interference scenarios. If the transmitting node u and w are both located within node v 's transmission range, then

node v will experience a collision. In such a case, this type of interference is called collision occurring at node v . The second scenario is where the interference is caused by transmitting nodes that are outside the receiver's transmission range. For example, transmitting node w is located outside receiving node v 's transmission range but inside v 's interference range. In both cases, the resulting interference will raise the noise floor and hence reduce the signal-to-noise ratio required to receive the message correctly.

Unfortunately, the problem of finding a broadcast scheduling with minimum broadcast latency is NP-hard [5]. To date, there have been numerous studies that aim to approximate the broadcast schedule with minimum latency [4] [5] [9] [14]. However, these works have mainly considered interference from a sending node's perspective and do not consider interference caused by nodes that are outside a receiving node's transmission range.

Henceforth, in this paper, we address the Interference-Aware Broadcast Scheduling (IABS) problem which aims to find a schedule with minimum broadcast latency, whilst also considering the two aforementioned types of interference. We present the design and evaluation of two centralized broadcast algorithms. In a nutshell, this paper contains the following contributions:

- 1) We provide two constant approximation algorithms: IABBS and IAEBS. These algorithms produce a broadcast schedule with a latency of at most $2 \left\lceil \frac{\pi}{\sqrt{3}}(\alpha + 1)^2 + (\frac{\pi}{2} + 1)(\alpha + 1) + 1 \right\rceil R$, which is the best latency bound without knowledge of geographical information. In particular, when α is set to 2, they bound the broadcast latency to $50R$. Here, α is the ratio between the interference range r_I and transmission range r_T , i.e., $\alpha = \frac{r_I}{r_T}$, and R is the radius of the topology with respect to the source.
- 2) We prove that the total number of transmissions produced by IABBS and IAEBS is at most eight times that of the minimum total number of transmissions.
- 3) We evaluate IABBS and IAEBS under different network parameters through simulations. Extensive experimental results show that on average, our proposed algorithms have a better performance than the state of the art algorithm CABS [11].

The remainder of this paper is organized as follows. Section

2 lists related work. In Section 3, we introduce the network model, the problem formulation and some graph definitions and theories. In Section 4, we introduce our approximation algorithms: IABBS and IAEBS. Section 5 presents the research methodology and results. Lastly, Section 6 concludes the paper, and presents further works.

II. RELATED WORK

To date, there have been several studies [2] [8] [10] [11] [15] on the IABS problem with the objective of minimizing broadcast latency. Most of them assume Disk Graphs (DGs), where the graph radius R with respect to the source of the broadcast serves as a lower bound for broadcast latency [1].

Chen et al. [2] are the first to study the IABS problem for Disk Graphs (DGs), where they consider the interference and transmission range to be different with $\alpha > 1$. They create a broadcast schedule based on a Breadth-First-Search (BFS) tree. To minimize the broadcast latency, the forward and backward transmissions are utilized to accelerate the broadcast transmissions of nodes in the same layers of the BFS tree. They prove their algorithm has a claimed approximation ratio of $O(\alpha^2)$. In particular, when $\alpha = 2$, they show that the broadcast latency of their algorithm is bounded by $26R$. However, as shown in [11], the bound of $26R$ is incorrect because of the unbridgeable gaps in Lemma 4 and Corollary 7 of [2].

To the best of our knowledge, Huang et. al [8] are the first to propose a correct constant approximation algorithm assuming each node is not aware of its geographical location. In their method, they partition the network plane into equal hexagons and assign them with different colors with the constraint that two hexagons with mutual distance of less than $(\alpha+1)r_T$ must not share the same color. They schedule nodes' transmission times based on hexagons' color and BFS tree. For instance, to avoid interference, transmitting nodes with the same color are not allowed to send simultaneously. Their proposed algorithm has an approximation ratio of $6 \lceil \frac{2}{3}(\alpha+2) \rceil^2$. Moreover, when $\alpha = 2$, they prove that the broadcast latency of their algorithm is bounded by $54R$.

Tiwari et al. [15] extend the method of [8] to consider different transmission ranges and spaces, i.e., 2D and 3D, and presented an approximation algorithm with a constant ratio of $2 \lceil \frac{4}{3}(\alpha+1)^2\gamma^2 + \frac{8\gamma(\alpha+1)}{3} + \frac{4}{3} \rceil$ for the 2D space, where γ is the ratio between the maximum and minimum transmission range. Similar to [8], they also apply hexagon coloring method to schedule nodes' transmission time. Notably, when $\gamma = 1$, Tiwari et al.'s algorithm is also suitable for UDG. Furthermore, when $\alpha = 2$ and $\gamma = 1$, their algorithm has a bounded broadcast latency of $44R$, which is the state-of-the-art latency bound assuming nodes are aware of their location information.

Mahjourian et. al [11] study the conflict-aware broadcast problem whereby apart from the transmission and interference range, they also consider the carrier sensing range under UDGs. They propose a constant approximation algorithm called CABS which has a ratio of $O(\max(\alpha, \beta)^2)$, where β is the approximation ratio between the carrier sensing and

transmission range. CABS is also a tree-based method, with a color schedule constructed along the BFS tree, where senders with the same color are not allowed to transmit concurrently. Interestingly, for $\beta = 0$, where carrier sensing range is not taken into account, CABS becomes an approximation algorithm for the IABS problem.

One key limitation of tree-based algorithms is higher broadcast latency because they prevent nodes in lower layers of the BFS tree from transmitting until all nodes in the current layer have finished their transmissions even though transmissions from lower layers do not interfere with the transmissions of nodes in the current layer. To overcome this disadvantage, Jiang et al. [10] propose a greedy heuristic algorithm by adopting a greedy coloring scheme. They show that their algorithm produces a much lower broadcast latency compared with [2] via simulations. However, they do not give any theoretical bound of the broadcast latency in the worst case.

All the works reviewed thus far, except [2] and [10], apply the same geometrical constraint to avoid interfering transmissions. For example in [8], [11] and [15], nodes with a distance less than $(\alpha+1)r_T$ must not be scheduled to transmit or receive at the same time. However, this geometrical constraint is in general stronger than what is needed to avoid interfering transmissions. That is, it is possible for two parallel transmissions to receive a message correctly despite not satisfying this geometrical constraint.

In the ensuing sections, we address the aforementioned limitations by proposing two constant approximation algorithms. Unlike past works, our algorithms do not use geometrical constraint nor require nodes to have their location information. Instead, we allow parallel transmissions to proceed only if they do not interfere with each other. Furthermore, IAEBS allows a node in a lower layer to receive or transmit a message earlier than a node in an upper layer. As a result, as shown in Section V, our algorithms yield a lower latency than current broadcast algorithms.

III. PRELIMINARIES

A. Network Model

We assume that all nodes have an equal transmission and interference range. Therefore, the network is represented by a UDG, $G = (V, E)$, where V is the set of nodes, and E represents the set of edges/links that exist between two nodes if their Euclidean distance is no more than the transmission range. Unlike [8] and [15], we do not require nodes in the network to know their geographical location. We denote the transmission range of nodes as r_T , their interference range as r_I , and α is the ratio between r_I and r_T , with $\alpha \geq 1$. We will use $N(v)$ and $N_\alpha(v)$ to denote the set of nodes that are within the transmission and interference range of node $v \in V$ respectively, $N(v) \subseteq N_\alpha(v)$.

We assume time is discrete and every message transmission occupies one unit time. In this paper, we adopt the protocol interference model, which is widely used because of its generality and tractability [7]. In the protocol interference model, two simultaneous transmissions, i.e., ' $u_1 \rightarrow v_1$ ' and

' $u_2 \rightarrow v_2$ ', are said to be interference-free if none of one sender's receivers locate within the other's interference range; that is, $d(u_1, v_2) > r_I$ and $d(u_2, v_1) > r_I$, where $d(u_1, v_2)$ (respectively, $d(u_2, v_1)$) is the Euclidean distance between u_1, v_2 (respectively, u_2, v_1).

B. Graph Definitions and Theories

Let $G = (V, E)$ be a connected and undirected UDG with $|V| = n$, and node s is a fixed node in G . The subgraph of G induced by $U \subseteq V$ is denoted by $G[U]$. The minimum degree of G is denoted by $\delta(G)$. The *inductivity* of G is defined as $\delta^*(G) = \max_{U \subseteq V} \delta(G[U])$. The *depth* of a node $v \in V$ is the distance between v and s , and the *radius* of G with respect to s , denoted by R , is the maximum distance of all nodes from s . The depth of a node v can be computed by constructing a BFS tree T_{BFS} from G . For $0 \leq i \leq R$, the layer i of T_{BFS} consists of all nodes at depth i , denoted by L_i .

An *Independent Set* (IS) I in $G(V, E)$ is defined as a subset of V such that $u, v \in I, (u, v) \notin E$. A *Maximal Independent Set* (MIS) U is an independent set which is not a subset of any other independent sets. A subset U of V is a *dominating set* of G if each node not in U is adjacent to at least a member of U . Clearly, every MIS of G is also a dominating set of G . If set U is a dominating set of G and $G[U]$ is connected, then U is called a *Connected Dominating Set* (CDS) of G . It is known that the size of MIS does not exceed $4opt + 1$, where opt denotes the minimum size of a CDS of G [16].

A proper node *coloring* of G is an assignment of colors, labeled by natural numbers, to the nodes in V such that any pair of adjacent nodes receive different colors. Any node ordering v_1, v_2, \dots, v_n of V induces a proper node coloring of G in the first-fit manner. Specifically, for $i = 1$ to n , assign node v_i the least assigned color that is not used by any neighbor v_j , where $j < i$. A particular node ordering of interest is the *smallest-degree-last* ordering. For $i = n$ to 1, it sets v_i to the node with the smallest degree in $G[U]$, where $U \subseteq V$ and initially $U = V$. After that, v_i is removed from U , and the process repeats until U is empty. It is well-known that the node coloring of G induced by a smallest-degree-last ordering uses at most $1 + \delta^*(G)$ colors [12].

Theorem 1: (Groemer Inequality [6]). Suppose that C is a compact convex set and U is a set of points with mutual distance at least one. Then,

$$|U \cap C| \leq \frac{area(C)}{\frac{\pi}{\sqrt{3}}/2} + \frac{peri(C)}{2} + 1,$$

where $area(C)$ and $peri(C)$ are the area and perimeter of C respectively.

When the set C is a disk or a half-disk, we have the following corollary.

Corollary 1: Suppose that C (respectively, C') is a disk (respectively, half-disk) of radius r , and U is a set of points with mutual distances at least one. Then,

$$|U \cap C| \leq \frac{2\pi}{\sqrt{3}}r^2 + \pi r + 1,$$

$$|U \cap C'| \leq \frac{\pi}{\sqrt{3}}r^2 + \left(\frac{\pi}{2} + 1\right)r + 1,$$

Theorem 2: (Mahjourian et. al [11]). In order for two simultaneous transmissions ' $u_1 \rightarrow v_1$ ' and ' $u_2 \rightarrow v_2$ ' to be

interference-free according to the protocol interference model, it is sufficient to have,

$$d(u_1, u_2) > (\alpha + 1)r_T \vee d(v_1, v_2) > (\alpha + 1)r_T$$

where $d(u_1, u_2)$ (respectively, $d(v_1, v_2)$) is the Euclidean distance between u_1, u_2 (respectively, v_1, v_2).

C. Key Observation

Note that the conditions in Theorem 2 are the geometrical constraint used by [8], [11] and [15], and is in general stronger than what is needed for avoiding interfering transmissions. For example in Fig. 1, assume that two transmissions ' $u_1 \rightarrow v_1$ ' and ' $u_2 \rightarrow v_2$ ' have the following geometrical property, $d(u_1, u_2) < 3r_T$, $d(v_1, v_2) < 3r_T$, $d(u_1, v_2) > 2r_T$ and $d(u_2, v_1) > 2r_T$ with $\alpha = 2$. According to Theorem 2, ' $u_1 \rightarrow v_1$ ' and ' $u_2 \rightarrow v_2$ ' cannot be scheduled simultaneously by algorithms in [8][11] and [15]. However, node v_1 and v_2 are outside the interference range of node u_1 and u_2 respectively, and hence transmissions ' $u_1 \rightarrow v_1$ ' and ' $u_2 \rightarrow v_2$ ' are interference-free. That is, they can be scheduled simultaneously.

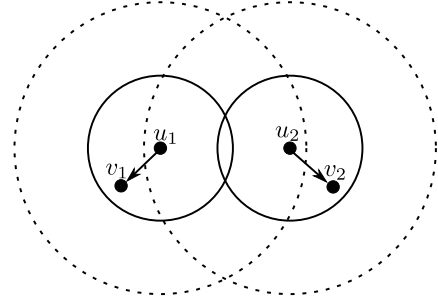


Fig. 1. An example of two simultaneous transmissions

D. Problem Formulation

The problem at hand, IABS, can be modeled as follows. Let (S_i, R_i) denote the i -th transmission, $i \in \mathbb{N}$, where each S_i (respectively, R_i) is the set of nodes that send (respectively, receive) the message in the i -th round. Given a wireless network $G(V, E)$ with a source node $s \in V$, the IAB problem is to find a forwarding schedule,

$$S = \{(R_1, S_1), (S_2, R_2), \dots, (S_m, R_m)\}$$

that satisfies the following constraints: (i) each node $u \in S_i$ must have been an R_j , with $j < i$; that is, node u must receive the message before it sends, (ii) all transmissions from set S_i to R_i must be interference-free, (iii) $|\bigcup_{i=1}^m R_i| = |V|$ and $|S|$ is minimum. In other words, find an interference-free broadcast schedule that guarantees all nodes in V receive the broadcast message interference-free in minimum time.

IV. PROPOSED ALGORITHMS

In this section, we present IABBS before presenting its enhanced version called IAEBS, which has a near optimal performance.

A. IABBS

The main idea is to schedule transmissions layer by layer based on the rule that a transmission is interference-free if there is no other senders within a receiving node's interference range. In our explanation to follow, we will use Fig. 2 and Fig. 3 to illustrate key aspects of IABBS. The network in Fig. 2 consists of 13 nodes randomly deployed in a 4×5 rectangle area, and node s is the source node.

Given $G = (V, E)$ and a source node s , IABBS starts by constructing a broadcast tree T_b rooted at node s , where if a node u is a parent of node v , then node u is responsible for transmitting the message to v . Then using T_b , IABBS schedules transmissions layer by layer such that every node receives the message interference-free.

Firstly, IABBS constructs a BFS tree T_{BFS} rooted at node s , and computes the depth of all nodes in the resulting T_{BFS} . Hence, this tree yields the radius R of G ; for the topology in Fig. 2, we have $R = 5$, see Fig. 3. With this tree in hand, let L_i , where $0 \leq i \leq R$, be the set of nodes at depth i of T_{BFS} .

The next step is to construct the broadcast tree T_b . For example, the resulting T_b for the network depicted in Fig. 2 is shown in Fig. 3. This tree will be used to determine the transmitting nodes and their transmission schedule. The construction of T_b has two key features: (i) deriving a MIS U , and (i) selecting nodes, called *dominators*, from the set U , and their parents, also called *connector* nodes, such that dominators together with connectors form a CDS.

Algorithm 1 constructs the MIS U layer by layer, starting from L_0 in T_{BFS} (line 4 to 9). Specifically, for each layer L_i , it selects nodes that are not adjacent to nodes in U greedily. Let $U_i = U \cap L_i$, and $M_i = L_i \setminus U_i$; in our example, we have $U_2 = \{v_1, v_2, v_5\}$ and $M_3 = \{v_{10}, v_{11}\}$. Note that, $U_0 = \{s\}$ and $U_1 = \emptyset$ because the source node s is the first node to be considered and nodes in L_1 must be adjacent to node s .

Given the sets U_i and M_i , the next step is to select parent nodes. At each layer i , where $0 < i \leq R$, Algorithm 1 greedily selects parents from U_i that cover the most nodes in the current and other layers that have yet to be assigned a parent; see lines 15 to 18. Let $P(v)$ and $C(v)$ be the parent of node v , and the set of children of node v respectively. For instance, in Fig. 3, for nodes in layer 4, initially, node $v_8 \in U_4$ covers the most nodes in $M_4 \cup M_5$, i.e., $M_4 = \emptyset$ and $M_5 = \{v_6, v_{12}\}$, so it is first chosen as the parent of node v_6 and v_{12} , i.e., $P(v_6) = P(v_{12}) = v_8$ and $C(v_8) = \{v_6, v_{12}\}$. To identify the connector nodes, lines 19–23 in Algorithm 1 process nodes in a similar manner; i.e., it selects as connectors nodes in M_i that cover the most dominators in the lower layer that have yet to be assigned a parent, whereby nodes in M serve as parents to nodes in U .

After constructing T_b , the next step is to schedule nodes' transmissions using Algorithm 4. For each layer of T_b , dominators first transmit followed by connectors. For each layer L_i , IABBS schedules transmissions with the help of two conflict graphs G_t and G_r , where an edge between two nodes indicate interference and hence must not transmit or receive

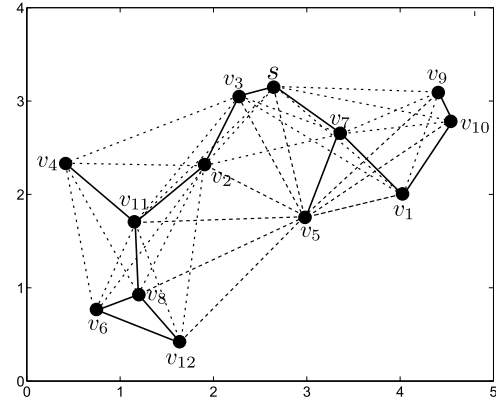


Fig. 2. An example wireless network with $r_T = 1$ and $r_I = 2$. There are 13 nodes randomly deployed in a 4×5 rectangle area. Solid lines denote transmission range. Dotted lines denote the interference range.

Algorithm 1 Broadcast Tree T_b

```

1:  $T_b \leftarrow (V_b, E_b)$ ,  $V_b \leftarrow V$ ,  $E_b \leftarrow \emptyset$ 
2:  $U \leftarrow U_0 \leftarrow \{s\}$ 
3:  $P(v) \leftarrow \text{Nil}$ ,  $\forall v \in V$ 
4: for  $i \leftarrow 1$  to  $R$  do
5:   for each  $w \in L_i$  do
6:     if  $(U \cap N(w)) = \emptyset$  then
7:        $U_i \leftarrow U_i \cup \{w\}$ ;  $U \leftarrow U \cup \{w\}$ 
8:     end if
9:   end for
10:   $M_i = L_i \setminus U_i$ 
11: end for
12: for  $i \leftarrow 0$  to  $R$  do
13:   $U'_i \leftarrow U_i$ ,  $M'_i \leftarrow M_i$ 
14:  while  $U'_i \neq \emptyset$  do
15:     $u \leftarrow$  a node in  $U'_i$  with maximum
       $|\{v | v \in N(u) \cap L_{j \geq i} \text{ and } P(v) = \text{NIL}\}|$  value
16:     $C(u) \leftarrow \{v | v \in N(u) \cap L_{j \geq i} \text{ and } P(v) = \text{NIL}\}$ 
17:     $U'_i \leftarrow U'_i \setminus \{u\}$  and  $P(v) \leftarrow u$ ,  $\forall v \in C(u)$ 
18:  end while
19:  while  $M'_i \neq \emptyset$  do
20:     $u \leftarrow$  a node in  $M'_i$  with maximum
       $|\{v | v \in N(u) \cap U_{j > i} \text{ and } P(v) = \text{NIL}\}|$  value
21:     $C(u) \leftarrow \{v | v \in N(u) \cap U_{j > i} \text{ and } P(v) = \text{NIL}\}$ 
22:     $M'_i \leftarrow M'_i \setminus \{u\}$  and  $P(v) \leftarrow u$ ,  $\forall v \in C(u)$ 
23:  end while
24: end for
25:  $E_b \leftarrow \{(u, v) | u = P(v)\}$ 
26: return  $T_b = (V_b, E_b)$ 

```

simultaneously. These two conflict graphs are constructed based on the rule that an edge exists between two transmitting nodes in G_t (respectively, two receiving nodes in G_r) if any of their children (respectively, parents) lie in the interference range of one another. The following paragraphs explain these conflict graphs in more detail.

Graph G_t is constructed as per Algorithm 2 which is used to ensure all dominators' transmissions in U_i are interference

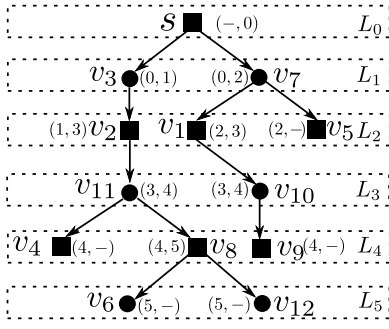


Fig. 3. Broadcast tree T_b of IABBS. The label (rec, tr) denotes the reception and transmission time of a node, and a square indicates a dominator.

free. Algorithm 2 constructs graph G_t by taking as input U_i and T_b , and outputs $G_t(V_t, E_t)$. First, all nodes in U_i that have children in layer i or $i+1$ are added into V_t (lines 3 to 7 in Algorithm 2). In the next step, Algorithm 2 will connect two nodes in V_t with an edge if dominators in U_i have children that lie in the interference range of one another (lines 8 to 14 in Algorithm 2). This means two sending nodes that are connected by an edge in G_t must not be scheduled to transmit simultaneously because a subset of one sending node's children lies within the interference range of the other.

Algorithm 2 Conflict Graph $G_t(V_t, E_t)$

```

1: Procedure Conflict-Graph- $G_t$  ( $U_i, T_b$ )
2:  $V_t \leftarrow E_t \leftarrow \emptyset$ 
3: for each  $u \in U_i$  do
4:   if  $C(u) \neq \emptyset$  then
5:      $V_t \leftarrow V_t \cup \{u\}$ 
6:   end if
7: end for
8: for each  $u \in V_t$  do
9:   for each  $v \in V_t$  do
10:    if  $C(u) \cap N_\alpha(v) \neq \emptyset$  then
11:       $E_t \leftarrow E_t \cup (u, v)$ 
12:    end if
13:  end for
14: end for
15: return  $G_t = (V_t, E_t)$ 

```

IABBS constructs the conflict graph G_r using Algorithm 3. G_r is then used to ensure that the reception of dominators in U_{i+1} is interference free. In other words, IABBS ensures the transmissions of connectors in M_i are interference free because the parent of nodes in U_{i+1} are connectors in M_i . Algorithm 3 takes as input of U_{i+1} and T_b , and outputs a subgraph $G_r(V_r, E_r)$. Specifically, all nodes in U_{i+1} are added into V_r (line 2 in Algorithm 3), and then two nodes in V_r are connected with an edge if they do not share as their parent the same connector, and neither of them is located in the interference range of the other's parent (lines 3 to 8 in Algorithm 3). That is, an edge in G_r indicates at least one receiving node is interfered by the other's parent node, hence

neither of them can be scheduled to receive at the same time.

Algorithm 3 Conflict Graph $G_r(V_r, E_r)$

```

1: Procedure Conflict-Graph- $G_r$  ( $U_{i+1}, T_b$ )
2:  $V_r \leftarrow U_{i+1}$ ,  $E_r \leftarrow \emptyset$ 
3: for each  $u \in V_r$  do
4:   for each  $v \in V_r$  do
5:     if  $P(u) \neq P(v)$  and  $P(u) \cap N_\alpha(v) \neq \emptyset$  then
6:        $E_r \leftarrow E_r \cup (u, v)$ 
7:     end if
8:   end for
9: end for
10: return  $G_r = (V_r, E_r)$ 

```

After constructing the conflict graphs of layer i , IABBS proceeds to color the nodes in G_t and G_r , where nodes in G_t (respectively, G_r) that share the same color are scheduled to transmit (respectively, receive) at the same time. To minimize their chromatic index, IABBS takes advantage of the smallest-degree-last ordering method to color nodes in the first-fit manner; see Algorithm 4 (lines 6 and 15). Denote by $color(v)$ the color number of node v , i.e., $color(v) = 0, 1, 2, \dots$. Let m_t and m_r be the maximum number of colors required by nodes in graph G_t and G_r respectively.

IABBS schedules the transmissions from parents at layer i to their children as follows. Specifically, the transmission of a dominator u in U_i to its children $C(u)$ is scheduled at time $T_{start} + color(u)$ based on graph $G_t(V_t, E_t)$, where T_{start} is the current time (lines 7 to 10 of Algorithm 4). The current time t_{start} increases by m_t to ensure all transmissions from dominators in U_i completes (line 11 in Algorithm 4). Then, transmissions from connectors in M_i to dominators in U_{i+1} are scheduled in a similar manner. Note that, $V_r = U_{i+1}$. The reception of dominator u in U_{i+1} is also scheduled at time $T_{start} + color(u)$ based on graph $G_r(V_r, E_r)$, and accordingly the transmission time $tr(P(u))$ of node u 's parent $P(u)$ is set to node u 's reception time $rec(u)$, i.e., $tr(P(u)) = rec(u) = T_{start} + color(u)$ (lines 16 to 19 in Algorithm 4). The current time T_{start} increases by m_r (lines 20 in Algorithm 4) so that all transmissions from connectors in M_i finishes before the next layer is considered. All subsequent layers are then scheduled in a similar manner.

We now use Fig. 3 as the example to illustrate the operation of Algorithm 4. It starts by constructing graph G_t and G_r for layer 0. Recall that $U_0 = \{s\}$, and $U_1 = \emptyset$, hence graph G_r is skipped. Graph G_t only contains one node s , therefore, we get $color(s) = 0$ and $m_t = 1$. Then, the transmission time $tr(s)$ of node s is set to $T_{start} + color(s) = 0$, where initially $T_{start} = 0$. Next, the current time T_{start} increases by 1, $T_{start} = 1$. For layer 1, $U_1 = \emptyset$, and thus graph G_t is empty. It only needs to construct graph G_r for layer 1 with $U_2 = \{v_1, v_2, v_5\}$. Recall that node v_1 and v_5 share the same parent v_7 , i.e., $P(v_1) = P(v_5) = v_7$, both node v_1 and v_5 lie in the interference range of node v_3 which is the parent of node v_2 , and node v_2 is within the interference range of node v_7 ,

Algorithm 4 Broadcast Scheduling of IABBS

```

1:  $tr(v) \leftarrow rec(v) \leftarrow -1, \forall v \in V$ 
2:  $T_{start} \leftarrow 0$ 
3: for  $i \leftarrow 0$  to  $R$  do
4:   if  $U_i \neq \emptyset$  then
5:      $G_t(V_t, E_t) \leftarrow \text{Conflict-Graph-}G_t(U_i, T_b)$ 
6:     Color nodes in  $G_t$  by smallest-degree-last ordering
7:     for each  $v \in V_t$  do
8:        $tr(v) \leftarrow T_{start} + color(v)$ 
9:        $rec(u) \leftarrow tr(v), \forall u \in C(v)$ 
10:    end for
11:     $T_{start} \leftarrow T_{start} + m_t$ 
12:  end if
13:  if  $U_{i+1} \neq \emptyset$  then
14:     $G_r(V_r, E_r) \leftarrow \text{Conflict-Graph-}G_r(U_{i+1}, T_b)$ 
15:    Color nodes in  $G_r$  by smallest-degree-last ordering
16:    for each  $v \in V_r$  do
17:       $tr(P(v)) \leftarrow T_{start} + color(v)$ 
18:       $rec(v) \leftarrow tr(P(v))$ 
19:    end for
20:     $T_{start} \leftarrow T_{start} + m_r$ 
21:  end if
22: end for
23: return  $rec(v), \forall v \in V$ 

```

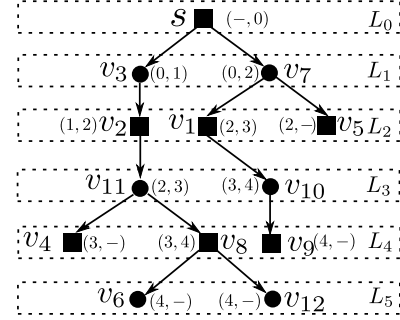
subsequently, in G_r , there is a link between node v_1 and v_2 , v_5 and v_2 respectively, i.e., $(v_1, v_2) = (v_5, v_2) = 1$. Then sort the nodes in G_r as per smallest-degree-ordering to yield a new scheduling order $\{v_2, v_1, v_5\}$. Next, color these nodes in the first-fit manner in the following order: $\{v_2, v_1, v_5\}$. Then we have $color(v_2) = 0$, $color(v_1) = color(v_5) = 1$ and $m_r = 2$. Finally, the reception time of node v_2 , v_1 and v_5 is set to 1, 2 and 2 respectively, i.e., $rec(v_2) = T_{start} + color(v_2) = 1 + 0$. Correspondingly, the transmission time of node v_3 and v_7 is also set to 1 and 2, i.e., $tr(v_3) = rec(v_2) = 1$ and $tr(v_7) = rec(v_1) = rec(v_5) = 2$. The other layers are handled in a similar manner and the final result is shown in Fig. 3.

B. IAEBS

In this section, we present an enhancement to IABBS. It differs from IABBS only in terms of how transmissions are scheduled. Instead of scheduling transmissions layer-by-layer in a top-down manner, IAEBS is able to schedule the transmissions across multiple layers. This means, in IAEBS, nodes in lower layers may receive or transmit the broadcast message earlier than nodes in an upper layer. As we will confirm in Fig. 4, this helps reduce broadcast latency.

Similar to IABBS, IAEBS starts by constructing the broadcast tree T_b using Algorithm 4. After that, for each L_i , it uses Algorithm 2 and 3 to construct G_t and G_r . Instead of coloring nodes in G_t and G_r as IABBS, IAEBS first sorts nodes in G_t and G_r according to the smallest-degree-last ordering, and records them in a new set Q_t and Q_r respectively. Then, it schedules the transmissions of nodes in L_i based on G_t and G_r greedily.

More specifically, any transmitting node u in G_t transmits at the minimum time t that satisfies the following interference-free constraints –(i) u must receive the message interference-free before time t , i.e., $tr(u) > rec(u)$ (lines 10 in Algorithm 5); (ii) no node in $C(u)$ hears the message from nodes in its interference range at time t (lines 7 in Algorithm 5); (iii) no node in $N_\alpha(u) \setminus C(u)$ is receiving a message from its parent at time t (lines 8 in Algorithm 5). Likewise, any receiving node v in G_r receives at the minimum time t that satisfies the following similar constraints –(i) the reception time t of node v must be larger than $rec(P(v))$ (lines 21 in Algorithm 5); (ii) node v is not hearing a message from nodes in its interference range at time t (lines 18 in Algorithm 5); (iii) no node in $N_\alpha(P(v)) \setminus C(P(v))$ is receiving a message from its parent at time t (lines 19 in Algorithm 5). In Algorithm 5, we use set $I_1(v)$ and $I_2(v)$ to record the time in constraint (ii) and (iii) for node v respectively. Denote by $I(v)$ the set of $I_1(v) \cap I_2(v)$. Note that, nodes in G_t are scheduled before nodes in G_r because all parents of nodes in G_r are assigned a reception time only when nodes in G_t are scheduled to transmit.

Fig. 4. Broadcast tree T_b of IEBBS.

We use Fig. 4 as the example to illustrate the operation of Algorithm 5. Recall that IAEBS constructs the same broadcast tree T_b , conflict graph G_t and G_r for each layer i , and the final broadcast tree T_b is shown in Fig. 4. In the next step, IAEBS will schedule the transmissions for each layer. It starts by sorting the nodes in G_t and G_r for layer 0. $U_1 = \emptyset$, hence G_r for layer 0 is empty. We have $tr(s) = 0$, i.e., $I_1 = I_2 = \emptyset$, $rec(v_3) = rec(v_7) = 0$ because G_t only contains node s , $Q_t = \{s\}$. For layer 1, since U_1 is empty, IAEBS only needs to consider nodes in U_2 . Hence, it will sort the nodes in G_r as per the smallest-last-degree ordering, and yields $Q_r = \{v_2, v_1, v_5\}$ for layer 1. Node v_2 is first considered in Q_r . As node s is the only transmission at time 0, $I_1(v_2) = \{0\}$, $I_2(v_2) = \{0\}$ and $rec(v_3) = 0$. Thus, $tr(v_3) = rec(v_2) = 1$. Next, node v_1 is considered. Node v_1 hears a message from s and v_3 at time 0 and 1 respectively, therefore $I_1(v_1) = \{0, 1\}$. For node v_7 as node v_1 's parent, among its interference range, node v_3 and v_2 are scheduled to receive the broadcast message at time 0 and 1 respectively; That is, $I_2(v_1) = \{0, 1\}$. Thus, we get $tr(v_7) = rec(v_1) = 2$, as $rec(v_7) = 0$ and $I(v_1) = \{0, 1\}$. Node v_5 is the last node to be scheduled. Since $rec(v_7) = 0$, $I_1(v_5) = \{0, 1\}$

and $I_2(v_5) = \{0, 1\}$, we have $tr(v_7) = rec(v_5) = 2$, i.e., $\min\{t|t > 0 \text{ and } t \notin \{0, 1\}\} = 2$. The other layers are handled in a similar manner, and the final result is shown in Fig. 4. Note that, the reception time of node v_{11} in layer L_3 is equal to that of node v_1 and v_5 in layer 2.

Algorithm 5 Broadcast Scheduling of IAEBS

```

1:  $tr(v) \leftarrow rec(v) \leftarrow -1, \forall v \in V$ 
2: for  $i \leftarrow 0$  to  $R$  do
3:   if  $U_i \neq \emptyset$  then
4:      $G_t(V_t, E_t) \leftarrow \text{Conflict-Graph-}G_t(U_i, T_b)$ 
5:     Sort nodes in  $G_t$  by smallest-degree-last ordering and
     Use  $Q_t$  to denote nodes in  $V_t$  with the new order
6:     for each  $u \in Q_t$  do
7:        $I_1(u) \leftarrow \{t|\exists w \in C(u) \text{ that hears a message at}$ 
         time  $t$  from nodes in  $N_\alpha(w)\}$ 
8:        $I_2(u) \leftarrow \{t|\exists w \in N_\alpha(u) \setminus C(u) \text{ that is scheduled}$ 
         to receive a message at time  $t\}$ 
9:        $I(u) \leftarrow I_1(u) \cup I_2(u)$ 
10:       $tr(u) \leftarrow \min\{t|t > rec(u) \text{ and } t \notin I(u)\}$ 
11:       $rec(w) \leftarrow tr(u), \forall w \in C(u)$ 
12:     end for
13:   end if
14:   if  $U_{i+1} \neq \emptyset$  then
15:      $G_r(V_r, E_r) \leftarrow \text{Conflict-Graph-}G_r(U_{i+1}, T_b)$ 
16:     Sort nodes in  $G_r$  by smallest-degree-last ordering and
     Use  $Q_r$  to denote nodes in  $V_r$  with the new order
17:     for each  $v \in Q_r$  do
18:        $I_1(v) \leftarrow \{t|\exists v \text{ that hears a message at time } t$ 
         from nodes in  $N_\alpha(v) \setminus \{P(v)\}\}$ 
19:        $I_2(v) \leftarrow \{t|\exists w \in N_\alpha(P(v)) \setminus C(P(v)) \text{ that is}$ 
         scheduled to receive a message at time  $t\}$ 
20:        $I(v) \leftarrow I_1(v) \cup I_2(v)$ 
21:        $tr(P(v)) \leftarrow \min\{t|t > rec(P(v)) \text{ and } t \notin I(v)\}$ 
22:        $rec(v) \leftarrow tr(P(v))$ 
23:     end for
24:   end if
25: end for
26: return  $rec(v), \forall v \in V$ 

```

C. Analysis

The following set of theorems assert the correctness, and approximation ratio of IABBS and IAEBS in terms of broadcast latency and transmission times.

Theorem 3: IABBS yields a correct and interference-free broadcast schedule.

Proof: Recall that IABBS processes nodes' transmissions layer by layer in a top-down manner, and the transmissions of each layer are only scheduled after all those in upper layers have completed. Thus we only need to prove all nodes in each layer can be scheduled interference-free. That is, for each layer, nodes in the conflict graph G_t and G_r are interference free when they are scheduled to transmit or receive. We prove the correctness for each layer by contradiction. Assume node

u and v in G_t transmit at the same time t to their respective children. Assume node u 's children are in the interference range of node v . This means there is a link between node u and v in G_t , according to Algorithm 2. Thus, node u and v must not share the same color. That is, node u and v must not be scheduled to transmit simultaneously. This is contradictory to our assumption. So for each layer, nodes in G_t transmit interference-free. Similarly, we can prove all nodes in G_r receive the broadcast message interference-free. Consequently, this theorem holds true. ■

Theorem 4: IAEBS yields a correct and interference-free broadcast schedule.

Proof: We prove the correctness of this theorem by contradiction. We assume node v cannot be scheduled to receive interference-free because there are two or more parallel transmissions to node v at the same time. Assume that node v 's parent $P(v)$ and one of the nodes in $N_\alpha(v)$, i.e., u , are scheduled to transmit at t . Furthermore, we consider two different cases. In the first case, node $P(v)$ is scheduled before node u . If node $P(v)$ selects time t as $P(v)$'s transmission time, node u will not choose t again because by the third constraint (lines 8 and 19 in Algorithm 5), when node v 's reception time is set to t , i.e., $rec(v) = tr(P(v)) = t$, node u must not choose t . In the second case, assume node u is scheduled before node $P(v)$. According to the second constraint (lines 7 and 18 in Algorithm 5), after node u selects time t as its transmission time, node $P(v)$ will not choose it again, because node v will hear node u 's transmission at time t , i.e., $v \in N_\alpha(u)$. This is contradictory to our assumption, so this theorem is true. ■

Theorem 5: IABBS produces a constant approximate solution for the IABS problem with latency at most $2 \left\lfloor \frac{\pi}{\sqrt{3}}(\alpha + 1)^2 + (\frac{\pi}{2} + 1)(\alpha + 1) + 1 \right\rfloor R$.

Proof: Recall that for each layer i , where $0 \leq i \leq R$, it takes $m_t + m_r$ unit time to finish all transmissions, thus we only need to prove the maximum value of m_t and m_r to obtain the maximum latency for IABBS. Recall that m_t and m_r are defined as the maximum number of colors required by dominators in G_t and G_r respectively. IABBS applies the smallest-degree-last ordering to color nodes in G_t and G_r respectively, hence $m_t = \delta^*(G_t) + 1$ and $m_r = \delta^*(G_r) + 1$ by [12]. In the worst case, Algorithm 2 and 3 will add a link between any two dominators whose distance is no larger than $(\alpha + 1)r_T$ by Theorem 2. The maximum minimum degree of any node u in G_t or G_r is bounded by the number of nodes which lie in a half-disk of radius $(\alpha + 1)r_T$ centered at u . All nodes in G_t and G_r are dominators. Therefore, by Theorem 1, the maximum minimum degree of node u is bounded by $\left\lfloor \frac{\pi}{\sqrt{3}}(\alpha + 1)^2 + (\frac{\pi}{2} + 1)(\alpha + 1) + 1 \right\rfloor - 1$. That is, $\delta^*(G_t), \delta^*(G_r) \leq \left\lfloor \frac{\pi}{\sqrt{3}}(\alpha + 1)^2 + (\frac{\pi}{2} + 1)(\alpha + 1) + 1 \right\rfloor - 1$. As a result, each layer will take at most $m_t + m_r = \delta^*(G_t) + \delta^*(G_r) + 2 \leq 2 \left\lfloor \frac{\pi}{\sqrt{3}}(\alpha + 1)^2 + (\frac{\pi}{2} + 1)(\alpha + 1) + 1 \right\rfloor$ unit time to finish

all transmissions. Hence, the maximum broadcast latency of IABBS is $2 \left\lfloor \frac{\pi}{\sqrt{3}}(\alpha + 1)^2 + (\frac{\pi}{2} + 1)(\alpha + 1) + 1 \right\rfloor R$. ■

Theorem 6: IAEBS yields a constant approximate solution for the IABS problem with latency at most $2 \left\lfloor \frac{\pi}{\sqrt{3}}(\alpha + 1)^2 + (\frac{\pi}{2} + 1)(\alpha + 1) + 1 \right\rfloor R$.

Proof: Recall that the transmission schedule of nodes is derived in a top-down manner greedily. Assume the maximum transmission time of nodes in layer i , where $0 \leq i < R$, is T_i . Suppose that node u in L_{i+1} and G_t is scheduled to transmit after T_i . We only need to consider nodes in G_t which have been scheduled before node u because all nodes in layer i finish their transmissions after T_i . The scheduling order of nodes in G_t is determined as per smallest-last-order, and thus when node u is considered, at most $\delta^*(G_t)$ nodes have been considered before it. After T_i , at most $\delta^*(G_t)$ nodes will interfere with node u 's transmission, i.e., $|I(u)| \leq \delta^*(G_t)$. Consequently, the maximum transmission time of nodes in G_t for layer $i + 1$ is $T_i + \delta^*(G_t) + 1$.

Next, nodes in G_r are scheduled after nodes in G_t according to IAEBS. Suppose that node v in G_r is scheduled to receive the broadcast message after $T_i + \delta^*(G_t) + \delta^*(G_r) + 2$, and only transmissions from the parents of nodes in G_r interfere with node v 's reception. Similar to nodes in G_t , the scheduling order of nodes in G_r is also determined by smallest-last-order, and thus when node v is considered, at most $\delta^*(G_r)$ nodes have been scheduled to receive. Hence, the maximum reception time of nodes in G_r for layer $i + 1$ is $T_i + \delta^*(G_t) + 1$, and the maximum transmission time of parents of nodes in G_r is $T_i + \delta^*(G_t) + \delta^*(G_r) + 2$. We get the maximum transmission time T_{i+1} of nodes in layer $i + 1$ is $T_i + \delta^*(G_t) + \delta^*(G_r) + 2$.

By theorem 5, $\delta^*(G_t), \delta^*(G_r) \leq \left\lfloor \frac{\pi}{\sqrt{3}}(\alpha + 1)^2 + (\frac{\pi}{2} + 1)(\alpha + 1) + 1 \right\rfloor - 1$, we get for each layer $i + 1$, where $0 \leq i < R$, its maximum transmission time T_{i+1} is bounded by $T_i + 2 \left\lfloor \frac{\pi}{\sqrt{3}}(\alpha + 1)^2 + (\frac{\pi}{2} + 1)(\alpha + 1) + 1 \right\rfloor$.

Thus we get the maximum transmission time T_i for each layer i is bounded by $2 \left\lfloor \frac{\pi}{\sqrt{3}}(\alpha + 1)^2 + (\frac{\pi}{2} + 1)(\alpha + 1) + 1 \right\rfloor i$, where $0 \leq i \leq R$. Hence, the maximum latency yielded by IAEBS is $2 \left\lfloor \frac{\pi}{\sqrt{3}}(\alpha + 1)^2 + (\frac{\pi}{2} + 1)(\alpha + 1) + 1 \right\rfloor R$. ■

Theorem 7: IABBS and IAEBS are 8-approximate solutions in terms of the number of transmissions.

Proof: Recall that IABBS and IAEBS use the same method, i.e., Algorithm 1, to construct the broadcast tree T_b . In T_b , only dominators and connectors are allowed to transmit a message. Each dominator transmits at most once and a connector may transmit several times to inform all of its dominator children. Given that each connector is a parent node of dominators in U , the number of transmissions by all connectors is equal to the number of dominators in U except the source node s which does not have a parent. The number of dominators is $|U|$, and thus, the total number of transmissions

of dominators and connectors is $|U| + |U| - 1 = 2|U| - 1$. Recall that the size of U does not exceed $4opt + 1$ [16], where opt is the minimum number of transmissions. IABBS and IAEBS are thus a $2(4opt + 1) - 1 = 8opt + 1$ solution.

It is known that for a UDG, a node can be adjacent to at most five dominators [3]. Therefore, each connector is adjacent to at most five dominators in U , and one of these dominators is assigned as its parent. A connector may transmit at most four times, because for any connector it has at most four children in U . ■

V. EVALUATION

We now outline the research methodology used to evaluate the performance of IABBS and IAEBS. We compare them against BFS and CABS [11], which are known to have the best performance to date. Note that BFS outputs the depth of the BFS tree and can be used to obtain the lower broadcast latency bound, assuming no interference. In particular, for CABS, β is set to 0; i.e., we do not consider the carrier sensing range in our experiments. It is worth pointing out that the main goal of our simulations is to compare the theoretical and experimental broadcast latency performance of our algorithms. To this end, our focus is on the effect of various network configurations, explained below, on broadcast latency.

In our experiments, all nodes are stationary and randomly deployed in a 700×700 m² square area. We study the effect of different network configurations including number of nodes and transmission radius. The number of nodes ranges from 100 to 300. The transmission radius ranges from 70 to 160 meters. Every experiment is conducted with one change to the network configuration whilst the other are fixed. Each experiment is conducted on 20 randomly generated topologies. Moreover, for each topology, we carry out the experiment for 10 runs, and in each run, an arbitrary node is selected as the source node. Each result is the average of 200 simulation runs. Our simulations were performed using MATLAB.

Fig. 5 is a plot of broadcast latency versus the number of nodes. Broadcast latency is the maximum time taken by any node to receive the message. This figure indicates that broadcast latency does not vary very much with the number of nodes. This is because the broadcast latency is mostly influenced by the depth of the BFS tree, which does not depend much on the number of nodes. As shown in Fig. 5, the depth of BFS tree does not fluctuate very much. Moreover, IABBS and IAEBS have a better performance than CABS, i.e., the broadcast latency produced by CABS in this experiment is about 40 time units; in contrast, IABBS and IAEBS perform much better with a broadcast latency of 30 and 25 respectively. This is because instead of adopting Theorem 2 to schedule nodes' transmissions, IABBS and IAEBS schedule two parallel transmissions if the corresponding children do not lie in one another's interference range. This means two senders or receivers with distance less than $(\alpha + 1)r_T$ but satisfying the condition that their children or parents are not within the interference range of one another can be scheduled

to transmit or receive simultaneously, and thereby, leading to a lower latency than CABS. Additionally, IAEBS performs better than IABBS because IAEBS schedules transmissions in more than one layer; that is, nodes in a lower layer may transmit or receive earlier.

Fig. 6 is a plot of the broadcast latency versus the transmission radii. It shows that the broadcast latency of all algorithms decreases with increasing transmission radius. It is because as the transmission radius increases, the number of nodes being covered by each transmission also increases, leading to a reduction in the depth of the BFS tree, which is the key factor that influences the performance of CABS, IABBS and IAEBS. As shown in Fig. 6, the depth of BFS tree reduces with increasing transmission radius. Furthermore, IABBS and IAEBS perform better than CABS.

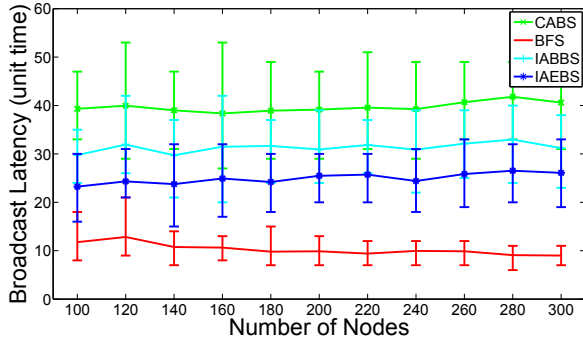


Fig. 5. Broadcast latency under different number of nodes

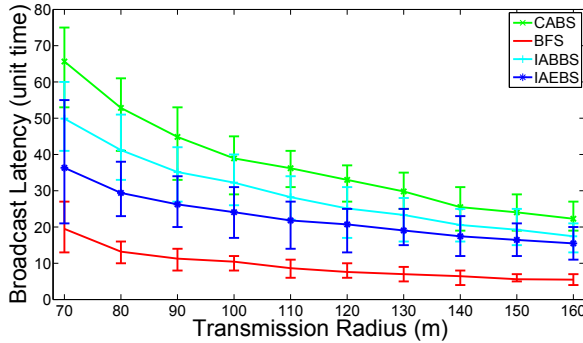


Fig. 6. Broadcast latency under different transmission radii

VI. CONCLUSION

In this paper, we have studied the interference aware broadcast scheduling problem in multi-hop wireless networks. To overcome this problem, we designed two novel algorithms, IABBS and IAEBS, for nodes use the protocol interference model. We prove that our algorithms provide correct and interference-free schedule, and produce a low broadcast latency. Furthermore, IABBS and IAEBS achieve a constant approximation ratio of $2 \left\lfloor \frac{\pi}{\sqrt{3}}(\alpha + 1)^2 + (\frac{\pi}{2} + 1)(\alpha + 1) + 1 \right\rfloor$. Our simulation results show that our algorithms have better performance in terms of broadcast latency as compared to CABS under different network scenarios.

As a future work, we are currently looking into probabilistic methods to model the trade-off between latency, redundancy and reliability, thus extending our solution to support QoS demands, and studying methods to improve reliability. The use of our solution in all-to-all broadcast is another possible future work.

ACKNOWLEDGMENT

This project is supported by the CSC-UoW joint scholarship program.

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] Z. Chen, C. Qiao, J. Xu, and T. Taekyeun Lee. A constant approximation algorithm for interference aware broadcast in wireless networks. In *IEEE INFOCOM*, Anchorage, Alaska, USA, May 2007.
- [3] A. V. Fishkin. Disk graphs: A short survey. In *WAOA*, Budapest, Hungary, September 2003.
- [4] R. Gandhi, Y.-A. Kim, S. Lee, J. Ryu, and P.-J. Wan. Approximation algorithms for data broadcast in wireless networks. In *IEEE INFOCOM*, Rio de Janeiro, Brazil, April 2009.
- [5] R. Gandhi, A. Mishra, and S. Parthasarathy. Minimizing broadcast latency and redundancy in ad hoc networks. *IEEE/ACM Transactions on Networking*, 16(4):840–851, 2008.
- [6] H. Groemer. Über die Einlagerung von Kreisen in einen konvexen Bereich. *Mathematische Zeitschrift*, 73:285–294, 1960.
- [7] P. Gupta and P. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2):388–404, 2000.
- [8] S.-H. Huang, P.-J. Wan, J. Deng, and Y. Han. Broadcast scheduling in interference environment. *IEEE Transactions on Mobile Computing*, 7(11):1338–1348, 2008.
- [9] S.-H. Huang, P.-J. Wan, X. Jia, H. Du, and W. Shang. Minimum-latency broadcast scheduling in wireless ad hoc networks. In *IEEE INFOCOM*, Anchorage, AK, USA, May 2007.
- [10] Z. Jiang, D. Wu, M. Guo, J. Wu, R. Kline, and X. Wang. Minimum latency broadcasting with conflict awareness in wireless sensor networks. In *ICPP*, Pittsburgh, PA, USA, September 2012.
- [11] R. Mahjourian, F. Chen, R. Tiwari, M. Thai, H. Zhai, and Y. Fang. An approximation algorithm for conflict-aware broadcast scheduling in wireless ad hoc networks. In *ACM MobiHoc*, Hong Kong, China, May 2008.
- [12] D. W. Matula and L. L. Beck. Smallest-last ordering and clustering and graph coloring algorithms. *Journal of the ACM*, 30(3):417–427, 1983.
- [13] S. Ni, Y. C. Y.C. Tseng, and J. Sheu. The broadcast storm problem in a mobile ad hoc network. In *ACM MOBICOM*, Seattle, WA, USA, October 2009.
- [14] J.-P. Sheu, P.-K. Hung, and C.-S. Hsu. *Scheduling of broadcasts in multihop wireless networks*, pages 483–495. CRC Press, Inc., Boca Raton, FL, USA, 2003.
- [15] R. Tiwari, T. Dinh, and M. Thai. On centralized and localized approximation algorithms for interference-aware broadcast scheduling. *IEEE Transactions on Mobile Computing*, PP(99):1, 2011.
- [16] P.-J. Wan, K. Alzoubi, and O. Frieder. Distributed construction of connected dominating set in wireless ad hoc networks. In *IEEE INFOCOM*, New York, NY, USA, June 2002.