

Efficient Broadcasting in Wireless Sensor Networks

Min Kyung An, Hyuk Cho

Abstract—In this paper, we study the Minimum Latency Broadcast Scheduling (MLBS) problem in wireless sensor networks (WSNs). The main issue of the MLBS problem is to compute schedules with the minimum number of timeslots such that a base station can broadcast data to all other sensor nodes with no collisions. Unlike existing works that utilize the traditional *omni-directional* WSNs, we target the *directional* WSNs where nodes can collaboratively determine and orientate their antenna directions. We first develop a 7-approximation algorithm, adopting directional WSNs. Our ratio is currently the best, to the best of our knowledge. We then validate the performance of the proposed algorithm through simulation.

Keywords—Broadcast, collision-free, directional antenna, approximation, wireless sensor networks.

I. INTRODUCTION

WIRELESS Sensor Networks (WSNs) consist of wireless sensor devices whose power source is usually in the form of a battery. These sensor devices are called wireless nodes, and they are scheduled to turn on their power to emit radio signals, or to turn it off to conserve energy for specific time duration.

One important task of a WSN is to disseminate data from a *base station* to all other nodes in the network periodically. This type of application is commonly known as *broadcasting* in the literature. During the process of broadcasting, a node sends (forwards) data to other nodes by emitting its radio signal. However, a *collision* can occur at a node if the data transmission is interfered by signals concurrently sent by other nodes. In this case, the data should be re-transmitted. As the small-sized sensors still have limited energy resources, it is desirable to reduce the unnecessary retransmissions in order to prolong the network's lifetime.

An interesting approach for broadcasting is to assign *timeslots* to sensor nodes to obtain a good *schedule* through which data can be disseminated without any collisions. As the broadcasting occurs periodically, reducing the *latency* of the schedule, that is, constructing a schedule with a minimum number of timeslots, becomes a fundamental problem. Such problem, whose objective is to construct a collision-free broadcasting schedule with minimum latency, is called the *Minimum Latency Broadcast Scheduling (MLBS)* problem.

In the literature, the MLBS problem has been actively investigated with two different power models: *non-uniform power model*, where each node can have various transmission power levels, and *uniform power model*, where every node is assigned a uniform transmission power level. With the non-uniform power model, Gandhi et al. [1] showed the first result of the NP-hardness of the problem, and a

$O(1)$ -approximation algorithm with latency bounded by $648R$, where R is the network radius. Later, they improved the latency bound to be less than $400R$ in [2]. With the uniform-power model, Dessmar et al. [3] proposed 2400-approximation algorithm with latency bounded by $2400R$, and later Huang et al. [4] proposed two approximation algorithms with the ratios of 51 and 24, whose latencies are bounded by $51R$ and $24R$, respectively. Then they improved the ratio to be 16 in [5]. An et al. [6] also proposed another approximation algorithm with the same ratio of 16. Huang et al. [7] again proposed a cell-based algorithm, which partitions a network into several hexagonal cells, and showed that the latency of the algorithm is bounded by $24R$ with the ratio of 24. Later, Krzywdzinski [8] proposed a distributed algorithm with latency bounded by $258R$ with the ratio of 258. Currently, Gandhi et al. [9], [10] achieved the best approximation ratio of 12 for the MLBS problem with uniform-power model.

Note that these existing works have studied the MLBS problem in the traditional *omni-directional* WSNs, where every node is equipped with omni-directional antenna with a beam-width $\theta = 360^\circ$. The omni-directional WSNs are commonly modeled as *undirected graphs*, where any two nodes are connected via an undirected communication edge if they are covered by each other's *broadcasting range*. Let us assume that a sender node v sends data to its receiver node u . If there is any other simultaneously sending node w covering u in its broadcasting range as well, then a *collision* occurs at u , and the data sent from v is not delivered to u successfully and should be re-transmitted (See Fig. 1 (a)).

Recently, advances in networks have led to the development of new wireless sensor devices equipped with directional antenna with a beam-width $\theta \in (0, 360^\circ]$. WSNs, consisting of such nodes that can collaboratively determine and orientate their antenna directions, are called *directional* WSNs. The directional WSNs are commonly modeled as *directed graphs*, where a directed edge exists from nodes v to u if u resides in *broadcasting sector* of v . In such networks, if we orient the antennas of nodes v and w as seen in Fig. 1 (b), then the two nodes v and w can send data simultaneously without causing any collisions.

In this paper, we study the MLBS problem in the directional wireless sensor networks with the uniform power model. We first develop an approximation algorithm with 7-approximation ratio whose latency is bounded by $7R$, and then validate the performance of the proposed algorithm through simulation.

The rest of this paper is organized as follows. In Section II, we introduce the two main classes of techniques to equip directional antennas in wireless sensor networks, describe our network models, and then define the MLBS problem. In Section III, we describe our constant factor approximation algorithm for the MLBS problem, and analyze the algorithm.

M. K. An is with Sam Houston State University, Huntsville, TX 77341, USA (e-mail: an@shsu.edu).

H. Cho is with Sam Houston State University, Huntsville, TX 77341, USA (e-mail: hyukcho@shsu.edu).

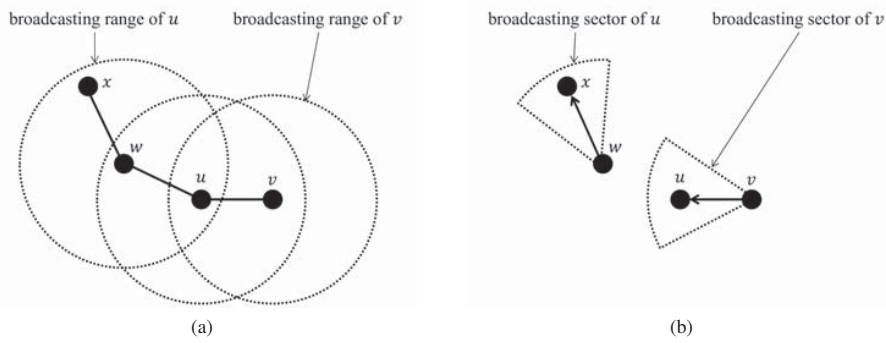


Fig. 1 (a) Omni-directional WSN. Senders v and w cannot simultaneously send data to their receivers u and x , respectively. (b) Directional WSN. Senders v and w can simultaneously send data to their receivers u and x , respectively

In Section IV, we evaluate the latency performance of the algorithm with simulated networks. Finally, we conclude with some remarks in Section V.

II. PRELIMINARIES

In this section, we introduce the two main classes of techniques to equip directional antennas in wireless sensor networks. Then, we describe our network model, and define the MLBS problem.

A. Directional Antenna Models

First, we review the two classes of techniques to equip directional antennas in wireless sensor networks: *steer beam directional antenna system* and *switch beam directional antenna system*.

- Steer beam directional antenna model [11]: In this model, each node is equipped with directional sending antenna and omni-directional receiving antenna. The *sensing range* sec (i.e., a broadcasting sector) with a beam-width $\theta \in (0, 360^\circ]$ is steered to some direction for transmission.
- Switch beam directional antenna model [11]: In this model, each node is equipped with switch beam directional sending antenna and omni-directional receiving antenna. Each node has K fixed *transmission directions* (i.e., a broadcasting sector, denoted by sec_k , $1 \leq k \leq K$, whose central angle is θ), and it can *switch on* one sector for transmission (i.e., the node is scheduled to be *active to work to* the sec_k direction.) Commercially available sectored antennas are typically designed for beam-widths of 180° , 120° , 90° , 60° and 45° [12].

B. Network Model

In this paper, we consider a wireless sensor network (WSN) that consists of a set V of sensor nodes deployed in a plane. Each node $u \in V$ is assigned a transmission power level $p(u)$ and equipped with a switch beam directional antenna with a fixed beam-width θ and omni-directional receiving antenna. Accordingly, a directed edge (u, v) exists from node u to

node v , if v resides in $sec_k(u)$, where $sec_k(u)$ denotes a broadcasting sector with an angle of θ centered at u with radius $p(u)$. Let $C_{sec_k(u)} = \{v \mid v \in V \text{ and } v \text{ resides in } sec_k(u)\}$ denote the set of nodes that reside in $sec_k(u)$. Then, the *collision* is said to occur at node w if there exist concurrently sending nodes u and u' such that $w \in C_{sec_k(u)}$ and $w \in C_{sec_k(u')}$.

C. Problem Definition

The MLBS problem is defined as follows. Given a set of nodes in a plane, we assign each node a timeslot and activate its one of antenna sectors. The goal is to compute a schedule with the minimum number of timeslots such that a base station can broadcast data to all other sensor nodes with no collisions. A *schedule* is defined as a sequence of such timeslots at each of which several sender nodes s_{t_i} are scheduled to successfully broadcast data to their neighbors that reside in $sec_k(s_{t_i})$ using transmission power $p(s_{t_i})$ with no collisions. Formally, at each timeslot t , we have an *assignment set* $\pi_t = \{(s_{t_1}, sec_k(s_{t_1}), p(s_{t_1})), \dots, (s_{t_m}, sec_k(s_{t_m}), p(s_{t_m}))\}$, $1 \leq i \leq m$, where m denotes the number of nodes scheduled at timeslot t .

A *broadcast schedule* is a sequence of assignment sets $\Pi = (\pi_1, \pi_2, \dots, \pi_M)$, where M is the length of the schedule, also called *latency*. A broadcast schedule Π is *successful* if data sent from a base station $c \in V$ is broadcasted to every node $v \in V \setminus \{c\}$. The MLBS problem is formally defined as follows:

Input: A set V of nodes, and transmission power level $p(v)$ for every node $v \in V$

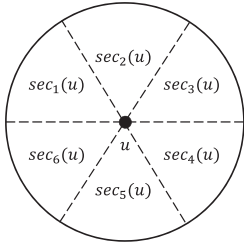
Output: A successful minimum latency schedule Π

III. CONSTANT FACTOR APPROXIMATION ALGORITHM

In this section, we describe our new constant factor approximation algorithm for the MLBS problem. We assume the uniform power model, where all nodes are initially assigned a uniform power level r , i.e., $p(u) = r$, for every $u \in V$. We also assume that every node $u \in V \setminus \{c\}$, i.e., except the base station c , is equipped with a switch beam directional antenna with a fixed beam-width $\theta = 60^\circ$.

TABLE I
NOTATIONS

Symbol	Definition
V	The set of nodes
E	The set of edges
G	A directed graph with V and E
T	A broadcast tree
E_T	The set of edges in T
θ	Beam-width of a sensor node
R	Network (graph) radius
D	Network (graph) diameter
K	The number of sectors of a node
$sec_k(u)$	A broadcasting sector k of a node u , $1 \leq k \leq K$
$p(u)$	Transmission power level of u
(u, v)	A directional edge from u to v
$C_{sec_k(u)}$	The set of nodes which reside in $sec_k(u)$
t	A timeslot
π_t	An assignment set at timeslot t
Π	A schedule
M	The length of schedule Π
$\ell(v)$	The level of v on T
S_i	The set of sender nodes whose ℓ is i
$d(u, v)$	The Euclidean distance between two nodes u and v
n	The number of nodes
$parent(v)$	A parent node of v on T

Fig. 2 Node u 's broadcasting disk partitioned into 6 equal-sized sectors, each of which has an equal beam-width $\theta = 60^\circ$

Accordingly, broadcasting sector of each node $u \in V \setminus \{s\}$ is assumed to be partitioned into 6 sectors, each of which is identified as sec_k , $k = 1, \dots, 6$, and the central angle of each sector is 60° , as shown in Fig. 2. Note that the base station has much more energy and communication resources than other sensor nodes. Therefore, we assume the base station is exceptionally equipped with the omni-directional antenna with beam-width 360° .

See Table I for notations.

A. Algorithm

We start this section by introducing some standard notations that are used subsequently [13] (cf. [14]):

- **Graph Center:** Given a communication graph $G = (V, E)$, we call a node c a *center* node if the hop distance from c to the farthest node from c is minimum.
- **Maximal Independent Set (MIS):** A subset $V' \subseteq V$ of the graph G is said to be *independent* if for any vertices $u, v \in V'$, $(u, v) \notin E$. An independent set is said to be *maximal* if it is not a proper subset of another independent set.
- **Connected Dominating Set (CDS):** A *dominating set* (DS) is a subset $V' \subseteq V$ such that every vertex v is either in V' or adjacent to a vertex in V' . A DS is said to be *connected* if it induces a connected subgraph.

In our proposed algorithm shown in Algorithm 1, we assume that the initial communication graph, modeled as a bidirectional unit disk graph $G = (V, E)$, whose $E = \{(u, v) \mid d(u, v) \leq r\}$, is connected. We further assume that a center node of G is the base station c in order to obtain a latency bound in terms of the network radius R rather than its diameter D as the lower bound of the MLBS problem is R . (See Lemma 3 in Section III-B.)

Algorithm 1 MLBS-Dir

Input: A set V of nodes in a plane, $p(u) = r$ for every node $u \in V$, and a base station $c \in V$

Output: Schedule Π

- 1: Construct a UDG $G = (V, E)$, where
 $E = \{(u, v) \mid d(u, v) \leq r\}$.
 - 2: Broadcast tree $T(V, E_T) \leftarrow$ Data aggregation tree of G rooted at c as constructed in [15].
 - 3: Starting timeslot $t \leftarrow 1$
 - 4: $\pi_t \leftarrow \pi_t \cup (c, \text{broadcasting range of } c, r)$
 - 5: $t \leftarrow t + 1$
 - 6: **for** $i = 1$ to R **do**
 - 7: $S_i \leftarrow \{u \mid u \text{ is a dominator or connector, } \ell(u) = i, \text{ and } u \in V\}$
 - 8: **if** $S_i \neq \emptyset$ **then**
 - 9: **for** $k = 1$ to 6 **do**
 - 10: $isTimeslotUsed \leftarrow false$
 - 11: **for every** $u \in S_i$ **do**
 - 12: **if** $|\{(u, v) \mid v \text{ resides in } sec_k(u), \ell(v) = \ell(u) + 1, \text{ and } (u, v) \in E_T\}| > 0$ **then**
 - 13: $isTimeslotUsed \leftarrow true$
 - 14: $\pi_t \leftarrow \pi_t \cup (u, sec_k(u), r)$
 - 15: **end if**
 - 16: **end for**
 - 17: **if** $isTimeslotUsed$ **then**
 - 18: $t \leftarrow t + 1$
 - 19: **end if**
 - 20: **end for**
 - 21: **end if**
 - 22: **end for**
 - 23: $M \leftarrow t - 1$
 - 24: **return** $\Pi \leftarrow (\pi_1, \pi_2, \dots, \pi_M)$
-

Algorithm 1 describes the details of the *Minimum Latency Broadcast Scheduling with Directional Antennas (MLBS-Dir)* algorithm which works as follows.

The algorithm starts by constructing *broadcast tree* T (Step 2), which is a data aggregation tree constructed as in [15]. The details of the construction of T is as follows. We first construct a breadth-first-search (BFS) tree (cf. [14]) on G rooted at the center node, i.e., base station, c . Based on the BFS tree obtained, T is constructed by first finding an MIS level by level on the BFS tree using the algorithm in [16]. Let us call the nodes in the MIS *dominators*, and the remaining nodes *dominatees*. In Fig. 3 (a), the dominators in the MIS are denoted by black nodes, and the dominatees are denoted by gray nodes. Here, the constructed MIS guarantees that the shortest hop-distance between any two complementary pairs,

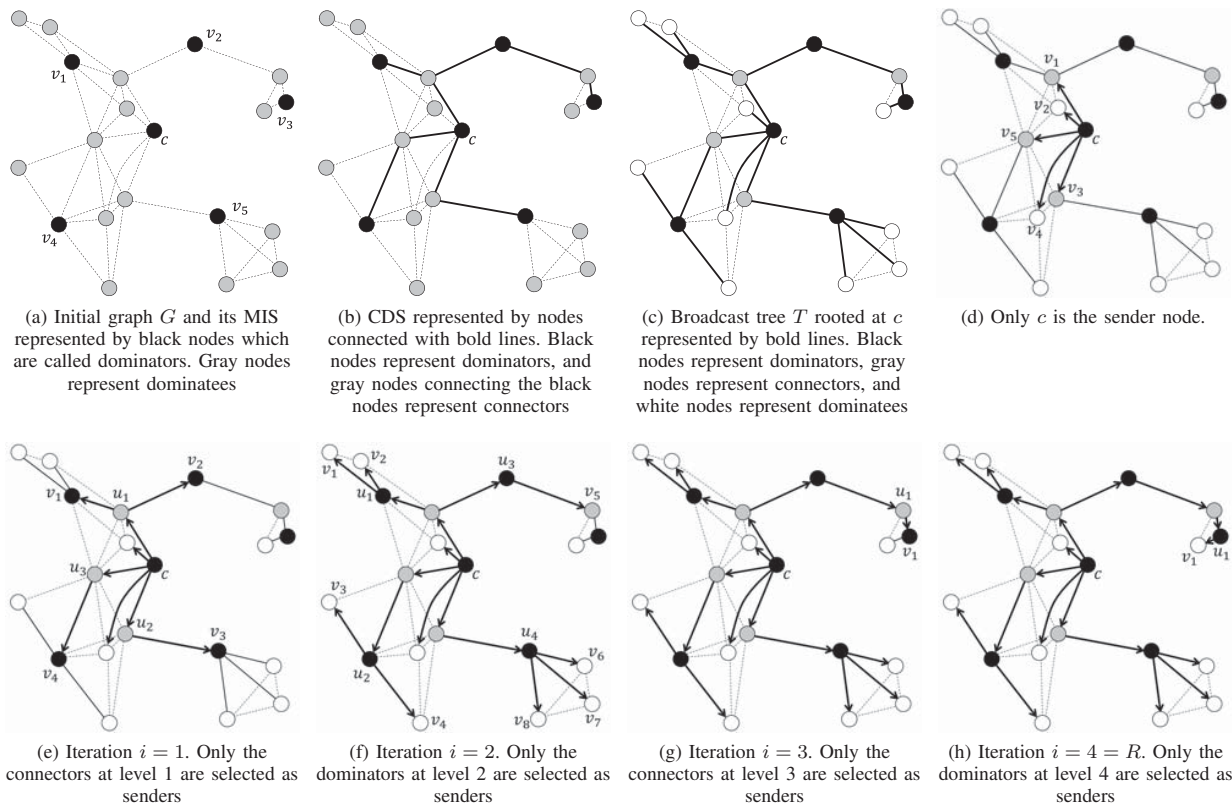


Fig. 3 Illustration of MLBS-Dir Algorithm. (a)-(c) illustrate the steps of constructing broadcast tree T . (d)-(h) illustrate the steps of scheduling nodes at each level

A and $MIS \setminus A$, where $A \subseteq MIS$, is exactly two hops. For example, in Fig. 3 (a), consider one pair $A = \{v_1, v_2\}$ and $B = MIS \setminus A = \{v_3, v_4, v_5, c\}$. The shortest hop-distance between the sets A and B on G is exactly two hops. Then, as Li et al. [15] proposed, we obtain a CDS of G by connecting the dominators using dominatees. The dominatees used to connect dominators are renamed *connectors*. If there exist some remaining dominatees that are not connected to the CDS, then each of the dominatees is connected to its neighboring dominator. For example, in Fig. 3 (b), the bolded edges represent the CDS, and in Fig. 3 (c), white nodes represent dominatees and the bolded edges represent T . We then use this newly constructed tree T as the broadcast tree to guide to find a minimum latency schedule Π in our algorithm.

Then, the root c broadcasts its data to its neighbors at the timeslot 1 (Step 4).

Next, Steps 6-22 are iterated R times (from level 1 to level R) to schedule the nodes at each level, as follows.

- **Scheduling sender nodes at level i at iteration i – Connector to Dominator:**

During the iteration, every node in the sender set S_i is scheduled, where the set S_i consists of connectors at level i . For every connector $u \in S_i$, only the lower level dominators of u need to receive data. So, if any connectors have their dominators at sec_1 , then

the algorithm assigns the same timeslot, say t , to them to activate sec_1 . Next, if any connectors have their dominators at sec_2 , then the algorithm assigns the next timeslot, $t + 1$, to them to activate sec_2 , and repeat until all sectors of connectors are examined thereby assigning appropriate timeslots (Steps 9-20).

- **Scheduling sender nodes at level $i + 1$ at iteration $i + 1$ – Dominator to Connectors and Dominatees:**

During the next iteration, every node in the sender set S_{i+1} is scheduled, where the set S_{i+1} consists of dominators at level $i + 1$. So, a dominator u at level $i + 1$ is assigned the timeslot, say t , to activate sec_1 if there exist any lower level neighbors in sec_1 , the timeslot $t + 1$ to activate sec_2 if there exist any lower level neighbors in sec_2 , and so on. Lastly, u is assigned the timeslot $t + 5$ if there exist any neighbors in sec_6 (Steps 9-20).

In order to avoid using timeslots to activate sectors in which no neighbors reside, the algorithm activates timeslots only for the sectors in which at least one receiver resides (Step 12). The algorithm repeats the above iterations until the last level's nodes are scheduled (Steps 6-22).

Figs. 3 (d)-(h) illustrates the steps of scheduling nodes at each level.

- **Base station scheduling (Fig. 3 (d)):**

At the very first step, the base station c is assigned

timeslot 1 for its receivers $v_1, v_2, v_3, v_4,$ and v_5 .

- Iteration $i = 1$ (Fig. 3 (e)): The sender set $S_1 = \{u_1, u_2, u_3\}$ consists of only the connectors whose level is 1. First, u_1 is assigned timeslot 2 for its receiver v_1 in $sec_1(u_1)$, and timeslot 3 for v_2 in $sec_3(u_1)$. Next, u_2 is assigned the timeslot 4 for its receiver v_3 in $sec_4(u_2)$. Then, u_3 is assigned the next timeslot 5 for receiver v_4 in $sec_5(u_3)$.
- Iteration $i = 2$ (Fig. 3 (f)): The sender set $S_2 = \{u_1, u_2, u_3, u_4\}$ consists of only the dominators whose level is 2. First, u_1 and u_2 are assigned the same timeslot 6 for their receivers v_1 and v_2 in $sec_1(u_1)$ and v_3 in $sec_1(u_2)$, respectively. Next, u_2, u_3 and u_4 are assigned the same timeslot 7 for their receivers v_4 in $sec_4(u_2)$, v_5 in $sec_4(u_3)$, and v_6 and v_7 in $sec_4(u_4)$, respectively. Then, u_4 is assigned the next timeslot 8 for receiver v_8 in $sec_5(u_4)$.
- Iteration $i = 3$ (Fig. 3 (g)): The sender set $S_3 = \{u_1\}$ consists of a connector whose level is 3. u_1 is assigned the next timeslot 9 for v_1 in $sec_5(u_1)$.
- Iteration $i = 4$ ($= R$) (Fig. 3 (h)): The sender set $S_4 = \{u_1\}$ consists of a dominator whose level is 4. u_1 is assigned the next timeslot 10 for v_1 in $sec_6(u_1)$.

B. Analysis

In this section, we analyze the MLBS-Dir algorithm (Algorithm 1) and bound the latency of the resulting broadcast schedules.

Lemma 1: No collision occurs when two dominators u and v , whose level $\ell(u) = \ell(v) = i$, simultaneously send their data to their lower level receivers, which reside in $sec_k(u)$ and $sec_k(v)$, respectively.

Proof: Consider two dominators u and v whose level $\ell(u) = \ell(v) = i$, and their receivers which are lower level dominators and connectors in $sec_k(u)$ and $sec_k(v)$, respectively. Here, a collision occurs only when

- u and v sends data simultaneously (i.e., $sec_k(u)$ and $sec_k(v)$ are activated at the same time),
- $sec_k(v)$ and $sec_k(u)$ are overlapped, and
- at least one receiver node must reside in the overlapped area.

The above conditions imply that u resides in v 's broadcasting range and/or v resides in u 's broadcasting range, and thus $d(u, v) \leq r$, which is a contraction because the distance between two dominators must be greater than r . Therefore, no collision occurs at any receivers when any two dominators u and v at the same level $\ell(u) = \ell(v) = i$ simultaneously.

Lemma 2: No collision occurs when two connectors u and v , whose level $\ell(u) = \ell(v) = i$, simultaneously send their data to their lower level dominators, which reside in $sec_k(u)$ and $sec_k(v)$, respectively.

Proof: Consider two connectors u and v , and their receivers which are lower level dominators, u' in $sec_k(u)$ and v' in $sec_k(v)$, respectively. Here, a collision occurs only when

- u and v sends data simultaneously (i.e., $sec_k(u)$ and $sec_k(v)$ are activated at the same time), and
- $sec_k(v)$ covers u' and/or $sec_k(u)$ covers v' .

As illustrated in Fig. 4, the above conditions imply that $d(u', v') \leq r$, which is contraction because the distance between two dominators must be greater than r . Therefore, no collision occurs at u' and v' when u and v send data simultaneously.

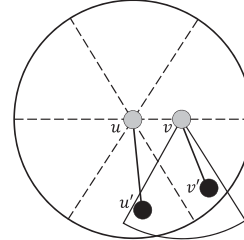


Fig. 4 No collision

Lemma 3 (Lower Bound [7]): Every broadcast schedule has at least R timeslots, where R is the network radius.

Theorem 4: The MLBS-Dir algorithm (Algorithm 1) produces a successful schedule whose latency is bounded by $7R$, and it is therefore a constant-factor approximation algorithm with the factor of 7.

Proof: To schedule the base station, we need only 1 timeslot (Step 4). Next, we repeat several iterations with the levels from 1 to R (Steps 6-22). At each iteration, we need at most 6 timeslots to schedule nodes, and it does not cause any collisions by Lemmas 1 and 2. As these steps repeat at most R times, the broadcast takes at most $1 + 6R \leq 7R$ timeslots. Therefore, it is a constant factor approximation with the ratio of 7 by Lemma 3.

IV. SIMULATION

In this section, we examine the latency performance of our proposed algorithm, the MLBS-Dir (Algorithm 1) in terms of the number of nodes, network (graph) radius, and transmission power level.

In this simulation, we generated a set $\mathbb{G} = \{\mathcal{G}_n \mid n = 50, 100, 150, 200, \dots, 500\}$, where $\mathcal{G}_n = \{\mathcal{G}_n^i \mid 1 \leq i \leq 100\}$ consists of 100 different networks, $\mathcal{G}_n^1, \mathcal{G}_n^2, \dots, \mathcal{G}_n^{100}$, each of which has n nodes. All networks were generated randomly in the Euclidean plane of dimension 500×500 . For each $\mathcal{G}_n \in \mathbb{G}$, we averaged the latencies produced by the MLBS-Dir algorithm over all the 100 networks. For the initial power assignment of nodes in the networks, we first computed a minimum spanning tree T_{MST} using edge weights defined as the distance between every two nodes. Then, we set the maximum transmission range r to be the distance of the longest edge in T_{MST} , and the uniform power level $p(u)$ for every $u \in V$ to be r . Notice that r is the minimally required power level to get a connected graph.

First, the Fig. 5 (a) shows the latency performance of the MLBS-Dir algorithm, in which as the network size increases the latency also increases. In the boxplot, red circles represent

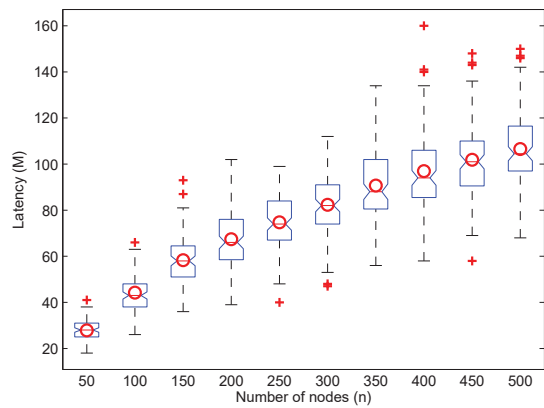
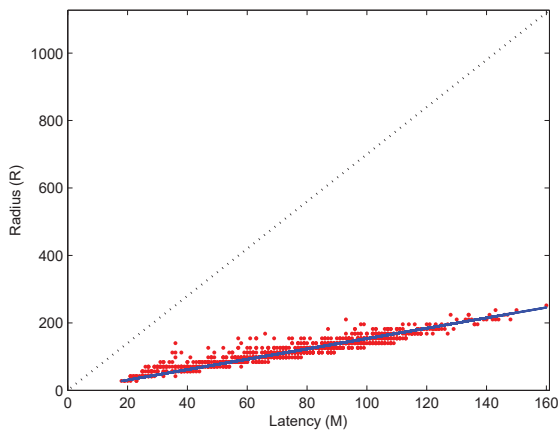
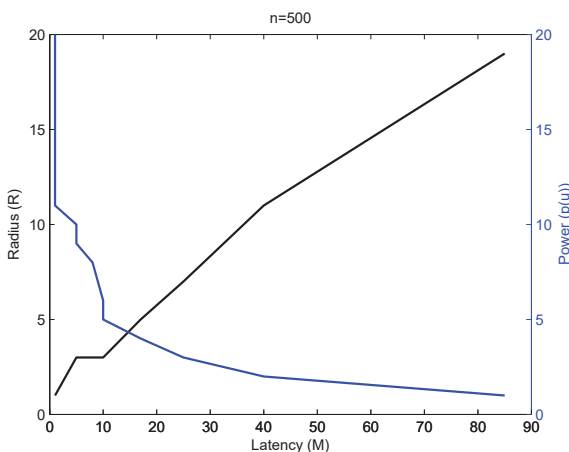
(a) Latencies with various n (b) Theoretical latency bound $7R$ vs. the latencies of the randomly generated different networks(c) The effect of the transmission power level $p(u)$ and R on latency

Fig. 5 Simulation results

the average latency for each node size n , and the horizontal line in the middle in each box represents the median latency.

The + symbols represent outliers.

Next, the Fig. 5 (b) shows our theoretical latency bound $7R$ (black dotted diagonal line) and the latencies obtained for the randomly generated 2000 different networks (red dots) in \mathbb{G} . For every network, we computed the ratio of the latency to the lower bound R . The minimum ratio computed is 1.1667, the mean ratio is 1.6044, and the maximum ratio is 3.8889. Also, the slope of the simple regression line (the blue line over the red dots) is only 1.5352. Here, it can be observed that the ratios of the latencies to R in the simulation are much lower than our theoretical ratio 7 (See Theorem 4).

Lastly, in order to evaluate the effect of the transmission power level $p(u)$, $u \in V$, we fixed $n = 500$, and varied $p(u)$. Note that as $p(u)$ increases, the network radius R decreases. It also implies that as $p(u)$ increases, the network becomes denser. Fig. 5 (c) with a single network shows the latency performance of the MLBS-Dir algorithm with the fixed $n = 500$ and various $p(u) = \{r, 2r, \dots, 20r\}$, $u \in V$. Here, as $p(u)$ increases, R decreases, and thus the latency M also decreases as it can be observed from the Lemma 3 and the Theorem 4.

V. CONCLUSION

In this paper, we focused on the Minimum Latency Broadcast Scheduling (MLBS) problem in the directional wireless sensor networks. We developed a 7-approximation algorithm yielding schedules, whose latency is bounded by $7R$, where R is the network radius. We then evaluated the performance of the proposed algorithm with simulated networks and discussed experimental results. For future work, we plan to study other related problems such as data collection and data aggregation for directional wireless sensor networks.

ACKNOWLEDGEMENT

This work is supported by Faculty Research Grant (FRG), Sam Houston State University (SHSU), Huntsville, TX, USA.

REFERENCES

- [1] R. Gandhi, S. Parthasarathy, and A. Mishra, "Minimizing Broadcast Latency and Redundancy in Ad Hoc Networks," in *MobiHoc*, 2003, pp. 222–232.
- [2] R. Gandhi, A. Mishra, and S. Parthasarathy, "Minimizing Broadcast Latency and Redundancy in Ad Hoc Networks," *IEEE/ACM Trans. Netw.*, vol. 16, no. 4, pp. 840–851, 2008.
- [3] A. Dessmark and A. Pelc, "Tradeoffs Between Knowledge and Time of Communication in Geometric Radio Networks," in *Proceedings of the Thirteenth Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)*. ACM, 2001, pp. 59–66.
- [4] S. C.-H. Huang, P.-J. Wan, X. Jia, and H. Du, "Low-Latency Broadcast Scheduling in Ad Hoc Networks," in *WASA*, 2006, pp. 527–538.
- [5] S. C.-H. Huang, P.-J. Wan, X. Jia, H. Du, and W. Shang, "Minimum-Latency Broadcast Scheduling in Wireless Ad Hoc Networks," in *INFOCOM*, 2007, pp. 733–739.
- [6] M. K. An, N. X. Lam, D. T. Huynh, and T. N. Nguyen, "Minimum Latency Gossiping in Wireless Sensor Networks," in *Proceedings of the International Conference on Wireless Networks (ICWN)*, 2012, pp. 406–412.
- [7] S. C. Huang, P. Wan, J. Deng, and Y. S. Han, "Broadcast Scheduling in Interference Environment," *IEEE Trans. Mob. Comput.*, vol. 7, no. 11, pp. 1338–1348, 2008.
- [8] K. Krzywdzinski, "Fast Construction of Broadcast Scheduling and Gossiping in Dynamic Ad Hoc Networks," in *IMCSIT*, 2010, pp. 879–884.

- [9] R. Gandhi, Y.-A. Kim, S. Lee, J. Ryu, and P.-J. Wan, "Approximation Algorithms for Data Broadcast in Wireless Networks," in *INFOCOM*, 2009, pp. 2681–2685.
- [10] R. Gandhi, Y. Kim, S. Lee, J. Ryu, and P. Wan, "Approximation Algorithms for Data Broadcast in Wireless Networks," *IEEE Trans. Mob. Comput.*, vol. 11, no. 7, pp. 1237–1248, 2012.
- [11] R. Ramanathan, "On the Performance of Ad Hoc Networks with Beamforming Antennas," in *MobiHoc*, 2001, pp. 95 – 105.
- [12] S. Roy, Y. C. Hu, D. Peroulis, and X.-Y. Li, "Minimum-Energy Broadcast Using Practical Directional Antennas in All-Wireless Networks," in *INFOCOM*, 2006.
- [13] M. K. An, N. X. Lam, D. T. Huynh, and T. N. Nguyen., "Minimum Latency Aggregation Scheduling in Interference-Aware 3-Dimensional WSNs," in *International Conference on Wireless Networks (ICWN)*, 2013.
- [14] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd ed. The MIT Press, 2009.
- [15] X. Li, X. Xu, S. Wang, S. Tang, G. Dai, J. Zhao, and Y. Qi, "Efficient Data Aggregation in Multi-hop Wireless Sensor Networks under Physical Interference Model," in *IEEE MASS*, 2009, pp. 353–362.
- [16] P. Wan, K. M. Alzoubi, and O. Frieder, "Distributed Construction of Connected Dominating Set in Wireless Ad Hoc Networks," in *INFOCOM*, 2002, pp. 1597 – 1604.