

RESEARCH

Open Access

A distributed broadcast algorithm for duty-cycled networks with physical interference model

Dianbo Zhao and Kwan-Wu Chin*

Abstract

Broadcast is a fundamental operation in multi-hop wireless networks. Given a source node with a message to broadcast, the objective is to propagate the message to all nodes in an interference-free manner while incurring minimum latency. This problem, called Minimum-Latency Broadcast Scheduling (MLBS), has been studied extensively in wireless networks whereby nodes remain on all times and has been shown to be NP-hard. However, only a few studies have addressed this problem in the context of duty-cycled wireless networks, which unfortunately, remains NP-hard. In these networks, nodes do not wake up simultaneously, and hence, not all neighbors of a transmitting node will receive a broadcast message at the same time, meaning multiple transmissions may be necessary. Moreover, most of these studies addressed the MLBS problem over the idealistic protocol interference model. Henceforth, this paper considers MLBS for duty-cycled wireless networks under the physical interference model and presents an approximation algorithm called hexagon-based broadcast algorithm (HBA), which has a constant ratio in terms of broadcast latency and transmission times. We have evaluated HBA in different network configurations, and the results show that the latencies achieved by our algorithm are much lower than existing schemes. In particular, HBA manages to half the broadcast latency achieved by the state-of-the-art tree-based algorithm.

Keywords: Duty-cycled networks; Physical interference model; Distributed broadcast; Minimum latency

1 Introduction

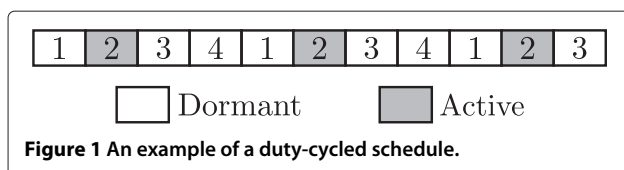
Wireless sensor networks (WSNs) consist of numerous nodes deployed in a field. These nodes may be resource constrained in terms of battery lifetime, memory, and processor speed. In addition, they use multi-hop communications whereby sensed data are forwarded to one or more sinks/gateways via one or more nodes. Upon receiving the data, a sink/gateway may transmit a command to sensor nodes to affect their operating parameters; e.g., sampling frequency. Consequently, network-wide broadcast from a sink is a fundamental operation. Apart from that, broadcast is relied upon by several network protocols, such as routing [1], information dissemination [2], and resource/service discovery [3]. These protocols in turn help applications in disaster relief, military communication, rescue operation, and object detection [4].

For these applications, time is critical, and hence, a Minimum-Latency Broadcast Scheduling (MLBS) algorithm/protocol will be of great importance to their operation. Like many other communication protocols, any developed MLBS solution must deal with interference. Unfortunately, the MLBS problem for multi-hop wireless networks has been proven to be NP-hard [5], and researchers have proposed many approximation algorithms; see [4-10]. These algorithms, however, assume all nodes are always active and adopt highly theoretical disk graph models, in which the transmission and interference range is assumed to be a unit disk centered at each node. These works deal with interference through the RTS/CTS model or protocol interference model [11]. RTS/CTS model only considers the interference within a node's transmission range; that is, interference and transmission range are equal. On the other hand, the protocol interference model assumes that the interference range is larger than transmission range. Specifically, interference is assumed when nodes have overlapping interference range.

*Correspondence: kwanwu@uow.edu.au
School of Electrical, Computer and Telecommunications Engineering,
University of Wollongong, Wollongong 2500, Australia

These ‘interfered nodes’ must therefore be scheduled in different time slots. The main drawback of these interference models is that they cannot model the case where many far-away nodes could still have non-negligible effect on reception. To this end, the physical interference model, also called SINR-based interference model, is more realistic, where the cumulative interference of many nodes outside the interference range is not neglected.

To date, only a few broadcast algorithms [7,12,13] have been designed for the MLBS problem under the physical interference model. Notably, there are even fewer works that address the MLBS problem in duty-cycled wireless networks. Briefly, in these networks, nodes are powered by batteries and are only awake for a fraction of the time [14]. Here, the duty cycle of a node is defined as the ratio between its active time and scheduling period, denoted as T . We note that they can employ a synchronous wake-up schedule; that is, nodes wake up at the same time. This means existing MLBS solutions for always on nodes are applicable. Unfortunately, doing so would require sensor nodes to coordinate and synchronize their wake-up time *globally*. This incurs high signaling overheads and energy that may severely shorten their operational life-time. This paper, therefore, only considers wireless networks with an asynchronous schedule. Advantageously, Chin [15] showed that nodes only need to maintain the clock offset to each neighbor. To reduce signaling overheads even further, nodes can select a random sequence of wake-up times independently instead of having a coordinated wake-up schedule, which requires network-wide negotiation, where all nodes are awake. Figure 1 shows a sequence of four slots that repeat periodically. The grayed slots denote the wake-up times of a node. In this example, the node has picked slot ‘2’ as its active time slot from a scheduling period T of 4, and hence, its duty cycle is $\frac{1}{4}$. In fact, in a duty-cycled WSN, all nodes will pick their own wake-up time upon network boot-up; in this example, a random slot in the range [1,4]. Once selected, a node then wakes up in its chosen slot every T slots. Given the wake-up schedule of nodes, a broadcasting node may have to transmit a message multiple times; i.e., all neighbors are unlikely to select the same slot to wake up. Consequently, any solutions to the MLBS problem in duty-cycled networks must first determine the wake-up times of neighbors, and in each wake-up time, it needs to ensure transmissions do not collide.



Henceforth, this paper presents the design and evaluation of a novel approximation algorithm that is designed for duty-cycled networks under the physical interference model. Specifically, it contains the following contributions:

1. We propose the first distributed broadcast algorithm, called hexagon-based broadcast algorithm (HBA), for the problem at hand. It produces a constant approximation ratio^a of $9 \left[2 + \frac{2}{3} \left(\frac{8\beta}{1-(r/r_{\max})^\alpha} \left(\frac{2}{\alpha-2} + \frac{1}{\alpha-1} + 3 \right) \right)^{1/\alpha} \right]^2 T$ in terms of broadcast latency, where α is the path-loss exponent, β is the minimum SINR threshold required for a message to be decoded successfully, r_{\max} is the maximum transmission range, r is the transmission range of nodes, and T is the scheduling period.
2. The total number of transmissions in terms of broadcast messages produced by HBA is upper-bounded by $(T + 1)N_H$, where N_H is the number of hexagons required to cover the entire network.
3. We evaluate HBA under different network parameters via simulation and show that that on average, our proposed algorithm has a much better performance in terms of broadcast latency than the tree-based algorithm [7]. The key reason is because our algorithm is able to schedule transmissions in multiple layers as opposed to layer by layer, as is done by [7].

The remainder of this paper is organized as follows. Section 2 lists related work. In Section 3, we introduce the network model, definitions, theories, and the problem formulation. In Section 4, we introduce our approximation algorithm, followed by its analysis in Section 5. Section 6 presents our research methodology and results. Lastly, Section 7 concludes the paper and presents future works.

2 Related works

To date, there are many approaches to address the MLBS problem in multi-hop wireless networks. The simplest by far is flooding [16], where each node simply re-transmits a received message to its neighbors unscrupulously. However, this causes broadcast storms [17] and is thus very costly and causes long latencies. Consequently, a number of researchers, e.g., [18-20], have proposed methods that improve the efficiency of broadcast. In this paper, we address a variant of the MLBS problem, which aims to find an efficient, interference-free schedule that yields the minimum broadcast latency.

For the MLBS problem under the RTS/CTS model, where it only considers interference or collision from senders inside a receiver’s transmitting range, Gandhi et al. [5] presented an approximation algorithm with a

constant approximation ratio of more than 400 for one-to-all broadcast. They then improve this ratio to 12 in [4]. Huang et al. [8] outlined three approximation algorithms for MLBS with latency of at most $24R$, $16R$, and $R + \mathcal{O}(\log_2 R)$, respectively, and the omitted constant in $\mathcal{O}(\log_2 R)$ exceeds 150 [4], where R is the theoretical lower-bound.

To the best of our knowledge, Chen et al. [6] are the first to study the MLBS problem under the protocol interference model, where the interference range is larger than transmission range. Huang et al. [7] propose a constant approximation algorithm with a ratio of $6 \lceil \frac{2}{3}(\epsilon + 2) \rceil^2$ for the MLBS problem, where ϵ is the ratio between the interference range and transmission range. Tiwari et al. [10] extend Huang et al.'s method to consider different transmission ranges and dimensions, i.e., 2D and 3D, and presented an approximation algorithm with a constant ratio of $2 \lceil \frac{4}{3}(\epsilon + 1)^2 \gamma^2 + \frac{8\gamma(\epsilon+1)}{3} + \frac{4}{3} \rceil$ for the 2D space, where γ is the ratio between the maximum and minimum transmission range. Tiwari et al. [10] also propose the first distributed algorithm for the MLBS problem under the protocol interference model. However, their distributed algorithm requires a node to exchange a significant number of control messages until all its neighbors receive the broadcast message, leading to an increase in energy consumption. Mahjourian et al. [9] study the conflict-aware broadcast problem whereby apart from the transmission and interference range, they also consider the carrier sensing range. They propose a constant approximation algorithm that has a ratio of $\mathcal{O}(\max(\epsilon, \delta)^2)$, where δ is the approximation ratio between the carrier sensing and transmission range.

There are only handful works that have considered the MLBS problem with the physical interference model. Huang et al. [7] propose the first approximation algorithm for the MLBS problem under the physical interference model assuming each node is aware of its geographical location. Wan et al. [13] convert the network under the physical interference model into a disk graph and apply coloring methods to schedule simultaneous transmissions. Huang et al. [12] then extend the approach in [7] to consider a more realistic interference model, where a message is received successfully as per its SINR level.

Thus far, the aforementioned works assume an always-on network, whereby all nodes remain awake indefinitely, meaning they do not employ any duty-cycle regime. To this end, only a handful of papers [21-23] have tried to address the MLBS problem in duty-cycled wireless networks, and even worse, all of these works only study the problem under the RTS/CTS model. Hong et al. [21] prove that the MLBS problem in duty-cycled wireless networks is NP-hard and proposed two approximation algorithms with an approximation ratio of $\mathcal{O}((\Delta^2 + 1)T)$ and $24T + 1$ respectively, where Δ is the maximum degree.

In [22], Jiao et al. propose an algorithm called OTAB and prove that OTAB has an approximation ratio of $17T$. Also, they showed that the total number of transmissions scheduled by OTAB is at most 15 times larger than the minimum number of transmissions. Recently, Xu et al. [23] extended the pipelined broadcast scheme in [8] to consider duty-cycled WSNs. Their broadcast algorithm produces a latency of at most $TR + T\mathcal{O}(\log_2 R)$, where the omitted constant in $T\mathcal{O}(\log_2 R)$ also exceeds 150.

The aforementioned approaches rely on a tree constructed using breadth-first search (BFS), which in turn is used to schedule interfering transmissions layer by layer. A key observation, however, is that transmissions in the next layer can only start when all nodes in the current layer have finished. In contrast, in our approach, HBA considers the set of all nodes holding a broadcast message at any point in time as potential transmitters. Consequently, HBA is able to schedule simultaneous transmissions in multiple layers of a BFS tree, which helps to reduce broadcast latency. Furthermore, HBA uses a coloring technique based on checking individual transmitters to ascertain whether they violate the interference-free conditions, which we will be precise in Theorem 1. These design features thus help increase the number of simultaneous interference-free transmissions in each time slot. In addition, except for [10], the aforementioned works are centralized and for always-on networks. However, the distributed algorithm in [10] is designed for always-on networks under the protocol interference model. More importantly, to the best of our knowledge, HBA is the first distributed solution for the MLBS problem in duty-cycled networks under the physical interference model.

3 Preliminaries

3.1 Network model

We assume nodes are placed on an Euclidean plane. Let $d(u, v)$ be the Euclidean distance between node u and v . We also assume the power level assignment is uniform, whereby all senders transmit with power level P . We adopt the following standard SINR-based interference model [24]. Here, a node v receives a message successfully from a sender u if and only if the following condition holds:

$$\frac{P d(u, v)^{-\alpha}}{\sum_{w \in V \setminus \{u, v\}} P d(w, v)^{-\alpha} + N} \geq \beta \quad (1)$$

where V denotes the set of nodes in the network, α is the path-loss exponent that is normally between 2 and 6, β denotes the minimum SINR required for a message to be received successfully which is greater than 1, N is the ambient noise; and $\sum_{w \in V \setminus \{u, v\}} P d(w, v)^{-\alpha}$ is the interference experienced by node v from nearby nodes.

We assume time is slotted. Each time slot is assumed to be of sufficient duration to transmit or receive a broadcast message. To this end, the network is locally synchronized

at the slot level. As shown in [25], this can be achieved by local synchronization techniques, such as FTSP [26], which can yield an accuracy of $2.24 \mu s$ using only a few small packet exchanges among neighboring nodes every 15 min. It is important to note that this accuracy is sufficient as the active duration of each node is typically above $10,000 \mu s$ [27,28]. Moreover, transmissions are not required to start at the beginning of each slot, meaning nodes do not need strict synchronization in order to communicate.

A scheduling period has T slots of fixed, equal length. We will index each slot by $1, 2, 3, \dots, T$. The duty cycle is thus defined as the ratio between active time and T . For example, if $T = 10$, a 10% duty cycle means nodes are only awake in one slot. Similar to [29,30] and [22], each node v selects one active time slot from $[1, 2, 3, \dots, T]$ randomly and independently and wakes up at its chosen time slot to receive a message; i.e., sensor nodes operate using an *asynchronous schedule*. If node v wants to transmit a message, it wakes up at the slot in which the receiver is awake.

We remark that the process of determining slot boundaries and a schedule, see Section 3.2, requires only *local* communication. This means nodes only need to communicate with their direct neighbors, as opposed to *all* nodes, in order to determine the start and end of a slot and also to learn the wake-up time of their neighbors. These design decisions thus help improve the scalability of the proposed solution, especially with respect to communication cost.

Lastly, we assume that each node is aware of its location. This can be achieved by using localization methods such as those in [31]. Note, we assume localization errors are bounded by ϵ . Also, nodes know the location of the base station, which is located at position $(0, 0)$.

3.2 Definitions and theories

We define the *transmission graph* $G_T = (V_T, E_T(r))$, where $E_T(r) = \{(u, v) \mid d(u, v) \leq r\}$. We assume G_T is connected. Let $r_{\max} = (\frac{P}{N\beta})^{1/\alpha}$, which is the maximum transmission range in the absence of interference from other simultaneous transmissions. Let r_{\min} be the length of the longest edge in the minimum spanning tree of G_T . In other words, r_{\max} and r_{\min} are the maximum and minimum r such that transmission graph G_T is connected, i.e., $r_{\min} \leq r \leq r_{\max}$. Let $N_1(u)$ denote the set of *one-hop neighbors* of node u , i.e., $N_1(u) = \{v \mid d(u, v) \leq r\}$. Accordingly, for a set V of nodes, $N_1(V)$ denotes the set of one-hop neighbors of nodes in V , i.e., $N_1(V) = \bigcup_{u \in V} N_1(u)$. $N_2(u)$ denotes the *two-hop neighbors* of u , where node $v \in N_2(u)$ should share at least one common one-hop neighbor with u and $r < d(u, v) \leq 2r$.

In a distributed environment, we assume each node knows the ID, position, and active time slot of its two-hop neighbors. This information can be gathered readily from any local broadcast techniques, e.g., [32,33] or [34],

in the network establishment stage. Incidentally, these techniques are the first to achieve local broadcast under the SINR-based interference model. Note, in practice, the required information can be embedded in HELLO messages sent out by nodes during neighbor discovery. That is, a node 'A' only needs to include its ID and selected wake-up slot in its HELLO message. A neighbor that receives this message records the embedded information and included this in its own HELLO message. Hence, node A's information has propagated two hops. One way to schedule the transmission of HELLO messages is to use the method in [15]. Each node has a broadcast slot whereby it transmits a HELLO message and all its neighbors receive. If each node has k neighbors, in the worst case, a node needs to transmit at most two times. That is, it first transmits its own ID and selected slot. The next HELLO message is transmitted only when it has received the HELLO message from all neighbors.

A link (u, v) is defined as the transmission from sender u to receiver v , where $(u, v) \in E_T(r)$. Let L denote a set of links in G_T . Set L is said to be *independent* in the SINR-based interference model if all senders in L transmit simultaneously; all transmissions can be received successfully by receivers in L . The next theorem gives the sufficient condition for L to be independent, and its proof can be found in the Appendix.

Theorem 1. *In order for set L to be independent, it is sufficient for one of the following to be true:*

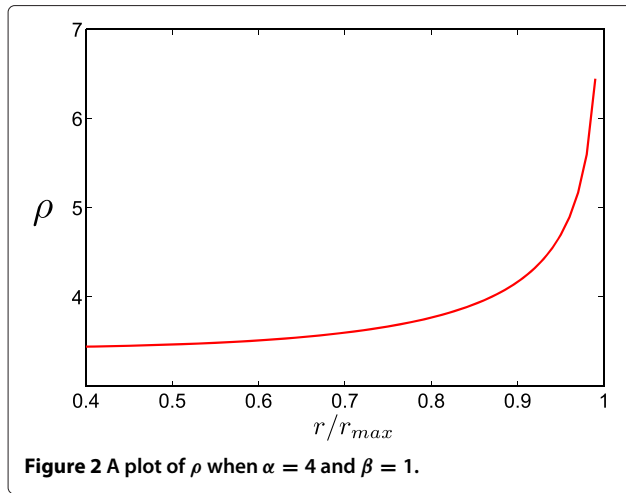
1. *The mutual distances of senders are all greater than ρr ;*
2. *The mutual distances of receivers are all greater than ρr .*

where $\rho = 1 + \left(\frac{8\beta}{1-(r/r_{\max})^\alpha} \left(\frac{2}{\alpha-2} + \frac{1}{\alpha-1} + 3 \right) \right)^{1/\alpha}$.

In practice, ρ is a small constant. Consider $\alpha = 4$ and $\beta = 1$. Figure 2 indicates the relationship between r/r_{\max} and ρ . We see that when $r/r_{\max} \leq 0.8$, ρ is smaller than 4.

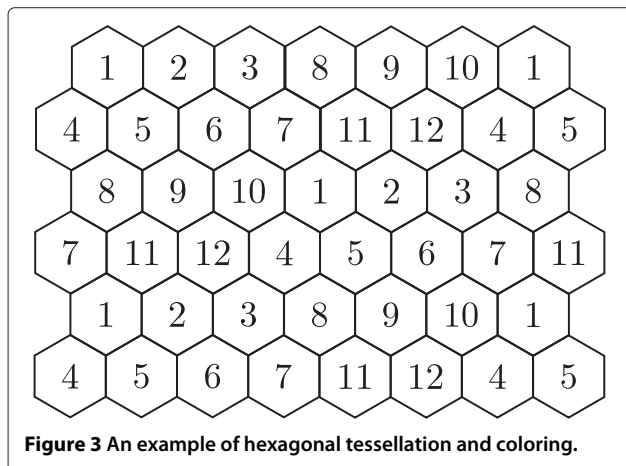
We remark that in practice, there will be localization errors. They can be incorporated by requiring the mutual distance between nodes to be at least $\rho r + \epsilon$. Recall that ϵ is the maximum localization error. The incorporation of localization error also increases the distance between hexagons (explained later) with the same color and thereby ensuring the transmissions from nodes in these hexagons are interference free.

Tessellation is a way to partition a plane into equal (or similar) pieces. In Figure 3, we see a regular hexagonal tessellation. We assume sensor nodes are pre-programmed with this information before deployment and then using their location determine the hexagon they are in. Given this tessellation, we can now proceed to color it using



a number of methods; examples include [7,12] and [10]. Without loss of generality, in this paper, we will employ the following $3k^2$ coloring method when scheduling broadcast. As we will see later, the color of a hexagon will be used to achieve interference-free transmissions in Section 4.2, where nodes located in hexagons with a different color are not allowed to transmit or receive simultaneously.

We are given a natural number k and a hexagonal tessellation with a hexagon radius of $r/2$, where r is the transmission range in G_T . Define two vectors $\vec{x} = (3\sqrt{3}r/4, 3r/4)$ and $\vec{y} = (3\sqrt{3}r/4, -3r/4)$. These vectors have a length of $3r/2$. Repeat the following process, see Algorithm 1, for all colors; here, $color$ is an integer in the range $[1, 3k^2]$; i.e., $1 \leq color \leq 3k^2$. Start from an uncolored hexagon with centre \vec{h} and then assign all hexagons with center at $\vec{h} + ak\vec{x} + bk\vec{y}$ with $color$, where $a, b \in \mathbb{Z}$. For instance, suppose $k = 2$. We need to assign the same value $color$ to all hexagons with center $\vec{h} + 3a\vec{x} + 3b\vec{y}$ where $a, b \in \mathbb{Z}$. This process repeats until $color = 3k^2 = 12$. The result



of 12 coloring is shown in Figure 3. Note, $\vec{h} + ak\vec{x} + bk\vec{y}$ is a function of a and b ; both of which can be arbitrary integers, e.g., $a = 1, b = -1$, if there exists a hexagon in the network with center $\vec{h} + ak\vec{x} + bk\vec{y}$.

Algorithm 1 $3k^2$ -coloring method

```

1: input:  $r, k, \vec{x}$  and  $\vec{y}$ 
2: output:  $3k^2$ -colored hexagons
3:  $color \leftarrow 1$ 
4: for  $i \leftarrow 1$  to  $2k$  do
5:   if  $i \leq k$  then
6:     for  $j \leftarrow 1$  to  $k + i$  do
7:        $\vec{h} \leftarrow ((2j - i - 1)\sqrt{3}r/4, -(i - 1)3r/4)$ 
8:     end for
9:   else
10:    for  $j \leftarrow 1$  to  $3k - i$  do
11:       $\vec{h} \leftarrow ((2j + i - 1 - 2k)\sqrt{3}r/4, -(i - 1)3r/4)$ 
12:    end for
13:  end if
14:  Assign  $color$  to hexagons with center  $\vec{h} + ak\vec{x} + bk\vec{y}$ 
   for all  $a, b \in \mathbb{Z}$ ,
15:   $color \leftarrow color + 1$ 
16: end for

```

Lemma 1. (Huang et al. [7]). *Algorithm 1 results in a $3k^2$ coloring, and hexagons of the same color are separated by at least a distance of $(3k - 2)r/2$.*

According to Theorem 1 and Lemma 1, in order to apply Algorithm 1 under the SINR-based interference model, we need to set $(3k - 2)r/2 = \rho r$. In other words, $k = \lceil 2(\rho + 2)/3 \rceil$. Based on the transmission graph G_T , a hexagon is said to be covered by a node $v \in V_T$, if and only if node v 's neighbors are in the said hexagon, excluding v . To distinguish nodes on the edges of hexagons, we assume each hexagon is left half open and right half close, with the topmost node included and the bottommost node excluded.

3.3 Problem formulation

Let $(B_i, R_i, t_i + k_i T)$ denote the i th transmission, $i, k_i \in \mathbb{N}$, where each B_i (respectively, R_i) is the set of nodes that send (respectively, receive) the message at time slot $t_i + k_i T$, where t_i is the active time slot of nodes in R_i . Given a wireless network with duty cycle and a source node $s \in V$, the broadcast problem is to find a forwarding schedule,

$$B = \{(B_1, R_1, t_1 + k_1 T), \dots, (B_m, R_m, t_m + k_m T)\}$$

that satisfies the following constraints: (i) each node $v \in B_i$ must have been in R_j , with $j < i$; that is, node v must receive the message before it sends, (ii) all transmissions from set B_i to R_i must be interference free under

SINR-based interference model, (iii) $|\bigcup_{i=1}^m R_i| = |V|$ and $t_m + k_m T - (t_1 + k_1 T)$ 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.

4 A distributed broadcast scheduler

We now present our distributed hexagon broadcast scheduling algorithm (HBA). We first describe HBA followed by our theoretical analysis confirming its $O(1)$ -approximation ratio in terms of broadcast latency.

4.1 Broadcast structure

HBA starts by constructing a broadcast structure T_b before using the color of hexagons to derive a broadcast schedule such that nodes located in hexagons with a different color are not allowed to transmit or receive simultaneously.

Firstly, we describe the construction of T_b ; see Algorithm 2 for details. Each node first tessellates the network into equal hexagons with a radius of $r/2$ and then gives a $3(\lceil 2(\rho + 2)/3 \rceil)^2$ coloring to all hexagons (lines 5 and 6 in Algorithm 2). Note, as the radius of each hexagon is $r/2$, the maximum distance in each hexagon is r ; that is, nodes located in the same hexagon are one-hop neighbors of each other.

Algorithm 2 Broadcast structure T_b

- 1: **input:** Transmission graph $G_T = (V_T, E_T(r))$
 - 2: **output:** $T_b = (V_b, E_b)$
 - 3: $V_b \leftarrow \emptyset, E_b \leftarrow \emptyset$
 - 4: **for** each node v in V_T **do**
 - 5: Tessellate the plane into equal hexagons with radius $r/2$, one of which is centered at $(0, 0)$
 - 6: Apply Algorithm 1 to color all hexagons by setting $k = \lceil 2(\rho + 2)/3 \rceil$
 - 7: $\mathcal{H}_1(v) \leftarrow \{u \mid u \text{ lies in the same hexagon as } v\} \cup \{v\}$
 - 8: $\mathcal{H}_2(v) \leftarrow \{u \mid u \in N_1(\mathcal{H}_1(v)) \text{ and } u \notin \mathcal{H}_1(v)\}$
 - 9: Mark all hexagons as *uncovered*
 - 10: **while** $\mathcal{H}_2(v) \neq \emptyset$ **do**
 - 11: $u \leftarrow$ a node in $\mathcal{H}_1(v)$ covering most *uncovered* hexagons (break ties based on smaller ID)
 - 12: **for** each *uncovered* hexagon covered by u **do**
 - 13: $w \leftarrow$ a node with smallest ID among nodes in this *uncovered* hexagon and $N_1(u)$
 - 14: $\mathcal{H}_2(v) \leftarrow \mathcal{H}_2(v) \setminus \{\mathcal{H}_1(w) \cap \mathcal{H}_2(v)\}$
 - 15: $E_b \leftarrow E_b \cup \{(u, w)\}$
 - 16: Mark this *uncovered* hexagon as *covered*
 - 17: **end for**
 - 18: **end while**
 - 19: **end for**
 - 20: $V_b \leftarrow$ nodes in E_b
-

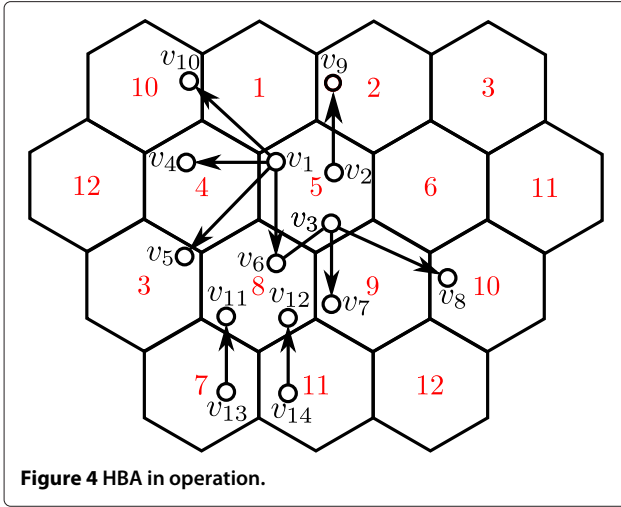
Next, based on two-hop neighbors information, node v places into set $\mathcal{H}_1(v)$ neighbors that are in the same hexagon as itself and adds into set $\mathcal{H}_2(v)$ nodes in the other hexagons that are the one-hop neighbors of nodes in $\mathcal{H}_1(v)$ (lines 7 and 8 in Algorithm 2). Note that set $\mathcal{H}_1(v)$ includes node v . Since nodes in $\mathcal{H}_1(v)$ are one-hop neighbors with each other and they are aware of two-hop neighbors information, nodes in the same hexagon will produce the same \mathcal{H}_1 and \mathcal{H}_2 sets.

To reduce the number of transmissions, HBA selects a set of nodes from $\mathcal{H}_1(v)$ that covers all neighboring hexagons containing a neighbor; i.e., the selected nodes are neighbors of nodes in $\mathcal{H}_2(v)$. Ideally, we want the chosen set to have a small cardinality. Specifically, HBA, applies line 9 to 18 in Algorithm 2 to produce the broadcast structure $T_b = (V_b, E_b)$, where V_b contains nodes used to relay broadcast messages, i.e., *providers* and *receptors*, and E_b indicates the set of links between a *provider* and its corresponding *receptor*. Here, *provider* is a node selected from $\mathcal{H}_1(v)$ and is used for relaying a broadcast message to its corresponding *receptors*; while a *receptor* is a node chosen from set $\mathcal{H}_2(v)$ and is used to transmit a broadcast message to all other nodes in its hexagon.

Initially, HBA marks all hexagons as *uncovered* and then repeats the following iterations until $\mathcal{H}_2(v)$ is empty. It first picks a node $u \in \mathcal{H}_1(v)$ that covers the most *uncovered* hexagons (line 11 in Algorithm 2), and labels u as a *provider*. The next step is to select one corresponding receptor of u from each *uncovered* hexagon. Specifically, for each *uncovered* hexagon, HBA will choose as the corresponding receptor a node w with the smallest ID among nodes in $N_1(u)$; see line 13 in Algorithm 2. Then, it includes link (u, w) in the set E_b and removes nodes in $\mathcal{H}_1(w)$ from $\mathcal{H}_2(v)$, i.e., $\mathcal{H}_2(v) \setminus \{\mathcal{H}_1(w) \cap \mathcal{H}_2(v)\}$ (lines 14 and 15 in Algorithm 2). It then marks the *uncovered* hexagon as *covered*. Note, provider u and its corresponding receptor w are located in different hexagons, and provider u (respectively, receptor w) has only one corresponding receptor w (respectively, provider u) in the hexagon including w (respectively, u).

After the execution of Algorithm 2, we have the broadcast structure T_b , where V_b contains providers and receptors, and E_b indicates the link of a provider and its corresponding receptors.

We now use Figure 4 as an example to illustrate the operation of Algorithm 2. We only consider the broadcast structure of the hexagon with color 5. Recall that nodes in the same hexagon produce the same broadcast structure T_b . Hence, in the following description, we illustrate Algorithm 2 from the perspective of node v_1 . It starts by constructing the set $\mathcal{H}_1(v_1)$ and $\mathcal{H}_2(v_1)$ based on its two-hop neighbors information. We thus have $\mathcal{H}_1(v_1) =$



$\{v_1, v_2, v_3\}$ and $\mathcal{H}_2(v_1) = \{v_4, v_5, v_6, v_7, v_8, v_9, v_{10}\}$. Node v_1 is first selected as the provider from $\mathcal{H}_1(v_1)$ as it covers the most *uncovered* hexagons, i.e., hexagons with node v_4, v_5, v_6 , and v_{10} . Next, node v_4, v_5, v_6 , and v_{10} are selected as the corresponding receptors of v_1 and are removed from $\mathcal{H}_2(v_1)$, i.e., $\mathcal{H}_2(v_1) = \{v_7, v_8, v_9\}$. Algorithm 2 then marks the hexagons covered by v_1 as *covered* and adds into the set E_b the links (v_1, v_4) , (v_1, v_5) , (v_1, v_6) , and (v_1, v_{10}) . The other nodes in $\mathcal{H}_1(v_1)$, i.e., v_2 and v_3 , are handled in a similar manner, and the final result is shown in Figure 4. For the hexagon with color 5, the set of providers is $\{v_1, v_2, v_3\}$.

The aforementioned T_b construction process yields the following property.

Lemma 2. *For each hexagon \mathcal{H} , there are at most 18 providers with corresponding receptors located in \mathcal{H} .*

Proof 1. *Recall that a provider u and its corresponding receptor v are located in different hexagons and only one link (u, v) exists in E_b between these two hexagons. This means we only need to prove the number of hexagons covered by receptors in a given hexagon \mathcal{H} is upper-bounded by 18. As shown in Figure 3, for a given hexagon \mathcal{H} with radius of $r/2$, it has at most 18 hexagons around it with a minimum distance less than r ; that is, nodes in \mathcal{H} can cover at most 18 hexagons. Hence, this lemma holds.*

4.2 Broadcast scheduling

In this section, we describe the protocol used to broadcast a message from the source node s to all other nodes in G_T , see Algorithm 3.

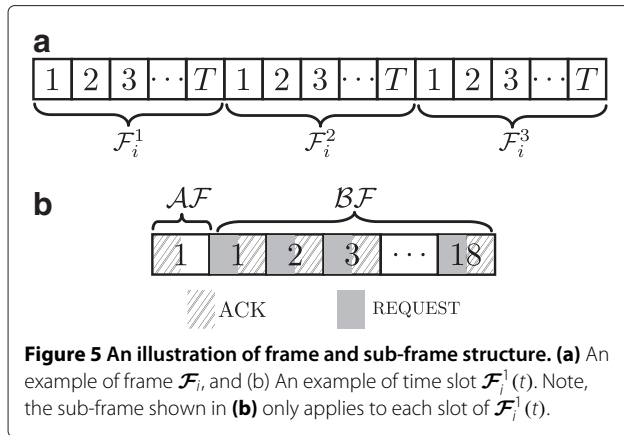
Algorithm 3 Broadcast scheduling

```

1: input:  $s, T_b = (V_b, E_b)$  and message  $m$ 
2: output: Broadcast latency  $Lat$ 
3: Schedule node  $s$  to transmit message  $m$  in frame  $\mathcal{F}_c$ 
4:  $Lat \leftarrow 0, V \leftarrow \mathcal{H}_1(s)$  and  $i \leftarrow (c + 1) \bmod 3k^2$ 
5: while  $V \neq V_T$  do
6:   // Phase 1-schedule nodes in  $V_b$ 
7:   for each hexagon  $\mathcal{H}$  assigned to frame  $\mathcal{F}_i$  do
8:      $w \leftarrow \text{NIL}$ 
9:     for  $t \leftarrow 1$  to  $T$  do
10:       $S_r(t) \leftarrow \{v \mid v \text{ is a receptor in } \mathcal{H} \text{ with active}$ 
11:         $\text{time slot of } t\}$ 
12:       $S_p(t) \leftarrow \{u \mid u \text{ is a provider with } m \text{ and yet to}$ 
13:         $\text{send } m \text{ to its receptor } v \in S_r(t)\}$ 
14:      Node  $w$  sends an ACK at  $\mathcal{AF}$  of  $\mathcal{F}_i^1(t)$ 
15:       $S_p(t)$  sends a REQUEST at the sub-time slot
16:        allocated by  $\mathcal{H}$  among  $\mathcal{BF}$  of  $\mathcal{F}_i^1(t)$ 
17:       $v \in S_r(t)$  sends an ACK after receiving a
18:        REQUEST from its provider  $u \in S_p(t)$ , if  $v$  has
19:        not heard an ACK from other receptors in  $\mathcal{H}$ 
20:      Node  $u$  sends message  $m$  to  $v \in S_r(t)$  at  $\mathcal{F}_i^2(t)$ 
21:       $w \leftarrow v$ 
22:    end for
23:    // Phase 2-schedule nodes in  $V_T \setminus V_b$ 
24:    Node  $w$  broadcasts message  $m$  in  $\mathcal{F}_i^3$ 
25:     $V \leftarrow V \cup \{v \mid v \text{ is in } \mathcal{H}\}$ 
26:  end for
27:   $Lat \leftarrow Lat + 3T$  and  $i \leftarrow (i + 1) \bmod 3k^2$ 
28: end while
    
```

HBA schedules the transmission of nodes in G_T in two phases. In phase 1, the algorithm only considers nodes in V_b . Specifically, for each hexagon, denoted by \mathcal{H} , HBA schedules the transmission from a provider u to its corresponding receptor v in \mathcal{H} , where $(u, v) \in E_b$. In phase 2, HBA allows a receptor v in \mathcal{H} to transmit a broadcast message received in phase 1 to all other nodes in \mathcal{H} . Furthermore, HBA schedules all transmissions based on hexagons' color, where those with a different color are not permitted to transmit or receive simultaneously.

We divide time into different frames. A hexagon with the color value of i is assigned to the i th frame, denoted by \mathcal{F}_i , where $1 \leq i \leq 3k^2$; recall that $3k^2$ is the number of colors used by Algorithm 1. As shown in Figure 5a, each frame \mathcal{F}_i consists of three sub-frames, i.e., $\mathcal{F}_i^1, \mathcal{F}_i^2$, and \mathcal{F}_i^3 , comprising of $3T$ time slots. For each hexagon \mathcal{H} , the first T time slot, i.e., sub-frame \mathcal{F}_i^1 , is used to determine which provider u from other hexagons is allowed to send a broadcast message m to its corresponding receptor v , in \mathcal{H} . Sub-frame \mathcal{F}_i^2 is used to transmit message m from provider u to receptor v in \mathcal{H} . The last sub-frame, \mathcal{F}_i^3 , is used by receptor v in \mathcal{H} to send the broadcast message to



all other nodes in \mathcal{H} . Specifically, phase 1 is conducted in \mathcal{F}_i^1 and \mathcal{F}_i^2 , and phase 2 is carried out in \mathcal{F}_i^3 .

Let c be the color of the hexagon containing source node s . Therefore, node s initiates the broadcast by transmitting a message m to all nodes in its hexagon in frame \mathcal{F}_c . After that, HBA, see Algorithm 3, starts from frame \mathcal{F}_i , where i is initially set to $(c + 1) \bmod 3k^2$; that is, it starts from the next frame of \mathcal{F}_c (line 4 in Algorithm 3). Then, HBA repeats the following iterations until all nodes in the network receive the broadcast message.

In phase 1, for each hexagon \mathcal{H} assigned to frame \mathcal{F}_i , i.e., they have color i , let $S_r(t)$ denote the set of receptors in \mathcal{H} with active time slot t , and $S_p(t)$ denotes the set of providers that have received broadcast message m before but have yet to send m to their corresponding receptors in $S_r(t)$ (lines 10 and 11 in Algorithm 3), where $1 \leq t \leq T$. Recall that these providers will have to wake up at time t to communicate with the receptors in $S_r(t)$. Let $\mathcal{F}_i^1(t)$ be the time slot t of sub-frame \mathcal{F}_i^1 , where $1 \leq t \leq T$. Denote by w a receptor in \mathcal{H} that received a REQUEST message from its corresponding provider before $\mathcal{F}_i^1(t)$, and w is initially set to null. For any receptor $v \in S_r(t)$, v first listens to the channel for an ACK message from a receptor w when it wakes up at time slot $\mathcal{F}_i^1(t)$ (line 12 in Algorithm 3). This ACK is sent in the \mathcal{AF} slot; see Figure 5b. Then, for any provider $u \in S_p(t)$, it will send a REQUEST message to its receptor $v \in S_r(t)$ asking it to receive a broadcast message in sub-frame \mathcal{F}_i^2 (line 13 in Algorithm 3).

When receptor v receives the REQUEST message from provider u , if v has not received any ACK message from other receptors in \mathcal{H} , node v replies with an ACK message to u . Otherwise, it does not respond to REQUEST messages (line 14 in Algorithm 3). As shown in line 16 of Algorithm 3, the selected receptor v is assigned to w , and it is responsible for sending an ACK message in subsequent \mathcal{AF} slots in $\mathcal{F}_i^1(t)$ (line 12 in Algorithm 3). This ensures all nodes waking up in subsequent slots are aware that a w node is available and thus stop responding to REQUEST messages.

We now discuss how providers in $S_p(t)$ transmit their REQUEST message in an interference-free manner. For each time slot in $\mathcal{F}_i^1(t)$, we further divide it into two parts, \mathcal{AF} and \mathcal{BF} ; see Figure 5b. As mentioned earlier, \mathcal{AF} is used by receptors in $S_r(t)$ to listen to the channel for an ACK message and for node w , if present, to transmit an ACK. The second part, namely \mathcal{BF} , is divided into 18 sub-time slots, which is equal to the number of hexagons around \mathcal{H} that have a minimum distance less than r ; cf. Lemma 2. We allocate these 18 sub-time slots to these neighboring hexagons according to their ID or color. Hence, a provider $u \in S_p(t)$ is only able to send a REQUEST message to its corresponding receptor $v \in S_r(t)$ in the sub-time slot corresponding to its hexagon. A receptor v is then required to reply with an ACK message in the same sub-time slot. As shown in Figure 5b, we assume each sub-time slot is sufficient to receive a REQUEST message and transmit an ACK message for receptor v .

During time slot t of sub-frame \mathcal{F}_i^2 , denoted by $\mathcal{F}_i^2(t)$, where $1 \leq t \leq T$, the broadcast message m is transmitted from the provider u to receptor v with active time slot t (line 15 in Algorithm 3). Note, only one provider u is selected in \mathcal{F}_i^1 to relay the broadcast message in sub-frame \mathcal{F}_i^2 to hexagon \mathcal{H} .

In phase 2, after receiving the broadcast message m , receptor v will broadcast message m to all other nodes in the same hexagon as v , i.e., \mathcal{H} , in sub-frame \mathcal{F}_i^3 . The broadcast is carried out when these nodes wake up (line 19 in Algorithm 3). Finally, HBA updates i to $(i + 1) \bmod 3k^2$ and repeats the above steps until all nodes receive the broadcast message m .

We now illustrate the operation of Algorithm 3 using Figure 4. Consider the hexagon with color 8. Assume receptors v_6, v_{11} , and v_{12} have the same active time slot of t , and their corresponding providers, v_1, v_{13} , and v_{14} , have received a broadcast message m , which it has yet to send to v_6, v_{11} , and v_{12} . Hence, as per Algorithm 3, we get $S_r(t) = \{v_6, v_{11}, v_{12}\}$ and $S_p(t) = \{v_1, v_{13}, v_{14}\}$. Nodes in $S_r(t)$, which are in hexagon 8, execute Algorithm 3 in frame \mathcal{F}_8 . In the \mathcal{AF} sub-time slot of $\mathcal{F}_8^1(t)$, receptors in $S_r(t)$ listen to the channel for an ACK message. Suppose that no ACK message is sent at \mathcal{AF} by a node w . Also, in this case, we assume that the transmitting order of providers is v_1, v_{13} , and v_{14} . As mentioned, the sub-time slots in \mathcal{BF} can be assigned as per hexagon ID or color. In this example, provider v_1 first sends a REQUEST to v_6 . On receiving this REQUEST, v_6 replies with an ACK immediately. After receiving this ACK from v_6 , provider v_1 knows receptor v_6 is ready to receive the broadcast message m and will transmit m to v_6 in sub-frame \mathcal{F}_8^2 . Other receptors, namely v_{11} and v_{12} , will also receive the ACK from v_6 , meaning they will not respond to any REQUEST from their respective provider; i.e., v_{13} and v_{14} . At $\mathcal{F}_8^2(t)$,

provider v_1 sends m to v_6 . In sub-frame \mathcal{F}_8^3 , receptor v_6 broadcasts m to nodes v_{11} and v_{12} .

4.3 Distance-based backoff

Recall that the \mathcal{BF} portion of $\mathcal{F}_i^1(t)$ is divided into 18 sub-time slots. A possible optimization to shorten \mathcal{BF} is by employing a distance-based backoff method. When a provider wants to send a REQUEST message, it first backs off for a period of time. This backoff duration depends on the distance between the provider and the hexagon containing its corresponding receptor. The smaller the distance, the shorter the backoff duration. Specifically, assume that a network operator decides to reduce the \mathcal{BF} duration to T_{backoff} . This so called *backoff time bound* can be divided into $W \leq 18$ sub-time slots. Note, each sub-time slot is sufficient for transmitting an ACK and receiving a REQUEST message. Let d be the distance between a provider u and the centre of hexagon \mathcal{H} containing u 's receptor. We get $d \geq \sqrt{3}r/4$ because u is not included in \mathcal{H} , and the distance between \mathcal{H} 's edge and \mathcal{H} 's centre is $\sqrt{3}r/4$. Denote by q the ratio between $\sqrt{3}r/4$ and d , i.e., $q = \frac{\sqrt{3}r}{4d}$, where $q \leq 1$. For provider u , it computes its backoff duration t_{backoff} using the following equation,

$$t_{\text{backoff}} = (\lfloor W(1 - q) \rfloor) \frac{T_{\text{backoff}}}{W} + X \quad (2)$$

where X is random period of the time generated from the range $\left[-\frac{T_{\text{backoff}}}{W}, \frac{T_{\text{backoff}}}{W}\right]$ for $1 \leq \lfloor W(1 - q) \rfloor \leq W - 1$ and from range $\left[0, \frac{T_{\text{backoff}}}{W}\right]$ for $\lfloor W(1 - q) \rfloor = 0$. The random value X reduces the chance of interference when two or more providers have the same q .

5 Analysis

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

Theorem 2. *HBA yields a correct and interference-free broadcast schedule.*

Proof 2. *According to Theorem 1, transmissions are interference free as long as the mutual distance between transmitters or receivers is larger than ρr . Hence, we only need to prove that simultaneous transmissions carried out by HBA are separated by ρr .*

Recall that in G_T , by design, the mutual distance between hexagons sharing the same color is larger than ρr . For each frame \mathcal{F}_i , only nodes in hexagons with the same color of i are scheduled by HBA. Considering sub-frames \mathcal{F}_i^2 and \mathcal{F}_i^3 , only providers and their corresponding receptors are allowed to send a broadcast message to nodes in hexagons with the same color of i . These receptors lie in hexagons with color value of i , and hence, their mutual

distance is larger than ρr . Thus, these simultaneous transmissions during sub-frames \mathcal{F}_i^2 and \mathcal{F}_i^3 are interference free by Theorem 1.

Then, we prove the transmissions in sub-frame \mathcal{F}_i^1 are also interference free. Only providers and their corresponding receptors are allowed to send a REQUEST and ACK during \mathcal{F}_i^1 . For hexagon \mathcal{H} with color value of i , the transmissions of providers and their receptors in \mathcal{H} are assigned with non-overlapping sub-time slots in \mathcal{F}_i^1 , and hence, transmissions in the same hexagon \mathcal{H} are interference free. For different hexagons which share the same color of i , the mutual distance of receptors lying in them is lower-bounded by ρr . According to Theorem 1, simultaneous transmissions in different hexagons during sub-frame \mathcal{F}_i^1 are also interference free.

Theorem 3. *HBA produces a constant approximation for the MLBS problem with a ratio of $9 \lceil 2(\rho + 2)/3 \rceil^2 T$ in terms of broadcast latency.*

Proof 3. *The theoretical lower-bound of the MLBS problem is R , i.e., the radius of the network with respect to the source node s . To compare the broadcast latency of HBA algorithm with the theoretical lower-bound R , we consider the BFS tree of the transmission graph G_T rooted at s . This tree divides the network into layers L_1, L_2, \dots, L_R . Let t_i denote the maximum reception time of nodes in L_i , where $1 \leq i \leq R$. Then, we prove for each layer L_i , we have $t_i \leq t_{i-1} + 9 \lceil 2(\rho + 2)/3 \rceil^2 T$. We prove this by induction.*

L_1 only contains source node s , and thus $t_1 = 0$. Nodes in L_2 are the one-hop neighbors of s . Thus, for L_2 , it holds true because it takes a frame, i.e., $3T$, for node s to broadcast the message m to L_2 , i.e., $t_2 = t_1 + 3T$. Then, we also prove this is true for layer i , where $3 \leq i \leq R$. Recall that nodes in L_i are the one-hop neighbors of L_{i-1} . After t_{i-1} , receptors in L_i will take at most $3 \lceil 2(\rho + 2)/3 \rceil^2$ frames to get the message m from providers in L_{i-1} and broadcast m to other nodes in L_i , where $3 \lceil 2(\rho + 2)/3 \rceil^2$ is the maximum color number used by Algorithm 1. Note, a frame contains $3T$ time slots. Thus, for each layer L_i , $t_i \leq t_{i-1} + 9 \lceil 2(\rho + 2)/3 \rceil^2 T$. After $(R - 1)3 \lceil 2(\rho + 2)/3 \rceil^2$ frames, nodes in L_R will receive the broadcast message m . Hence, the broadcast latency of HBA is upper-bounded by $(R - 1)9 \lceil 2(\rho + 2)/3 \rceil^2 T < 9 \lceil 2(\rho + 2)/3 \rceil^2 TR$.

Theorem 4. *The number of REQUEST, ACK, and broadcast messages in HBA is upper-bounded by $18N_H$, TN_H , and $(T + 1)N_H$ respectively, where N_H is the number of hexagons required to cover the entire network.*

Proof 4. *We first prove that the number of REQUEST messages is upper-bounded by $18N_H$. According to Lemma 2, for each hexagon \mathcal{H} , it has at most 18 providers with corresponding receptors that are in \mathcal{H} . Recall that a*

REQUEST message is only sent once from a provider to its corresponding receptor. It means, for each hexagon \mathcal{H} , its receptors receive at most 18 *REQUEST* messages. Hence, given N_H hexagons, the number of *REQUEST* messages is upper-bounded by $18N_H$. Next, we show that the number of *ACK* messages is upper-bounded by TN_H . For each hexagon \mathcal{H} , the *ACK* message is first sent by a receptor v in \mathcal{H} in response to a *REQUEST* message from v 's corresponding provider. An *ACK* message is sent once in each time slot until sub-frame \mathcal{F}_i^1 ends. Since \mathcal{F}_i^1 consists of T time slot, the maximum transmission time of *ACK* is T for each hexagon. To sum up, the maximum number of *ACK* sent during the broadcast is TN_H . Last, we show the maximum number of broadcast messages transmitted by HBA is $(T + 1)N_H$. As illustrated in Section 4.2, during sub-frame \mathcal{F}_i^2 for a hexagon \mathcal{H} , only one provider is allowed to transmit a broadcast message to its corresponding receptor v in \mathcal{H} . During each time slot of \mathcal{F}_i^3 , receptor v will transmit a broadcast message at most T times to its neighbors in \mathcal{H} . The maximum number of broadcast messages transmitted in a hexagon is thus $T + 1$, meaning the total number of broadcast messages is upper-bounded by $(T + 1)N_H$.

6 Evaluation

In this section, we outline the research methodology used to evaluate the performance of HBA. In our experiments, we measure each algorithm against two metrics: *broadcast latency* and *number of transmissions*. In our experiments, we fix $\alpha = 4$, $\beta = 1$ and $r_{\max} = 100$ m. We place wireless nodes in a square area of 700×700 m² randomly, while changing the number of nodes, transmission range r , and scheduling period T . In addition, we ensure that the resulting WSN is connected. For each experiment, we change one network configuration while the other two remain unchanged. Each experiment is conducted on 50 randomly generated topologies. Moreover, for each topology, we run the algorithms ten times; for each run, we select a source node uniformly and randomly. Hence, each result is an average of 50×10 simulation runs.

6.1 Performance of HBA

In Figure 6a,b, we delineate the broadcast latency and number of transmissions for different number of nodes, respectively. The value of r is fixed at 50 m, and T is set to 10. As shown in Figure 6a, we see that the broadcast latency of HBA decreases as the number of nodes increases. The reason is as follows. For a fixed area, the network becomes denser when the number of nodes becomes larger. As a result, when the network becomes denser, there are more links and path lengths become shorter. However, we observe that the number of transmissions increases with higher number of nodes in Figure 6b. The reason is that more hexagons will be filled with nodes when the number of nodes becomes larger,

and hence, more transmissions are required to propagate the broadcast message to these hexagons.

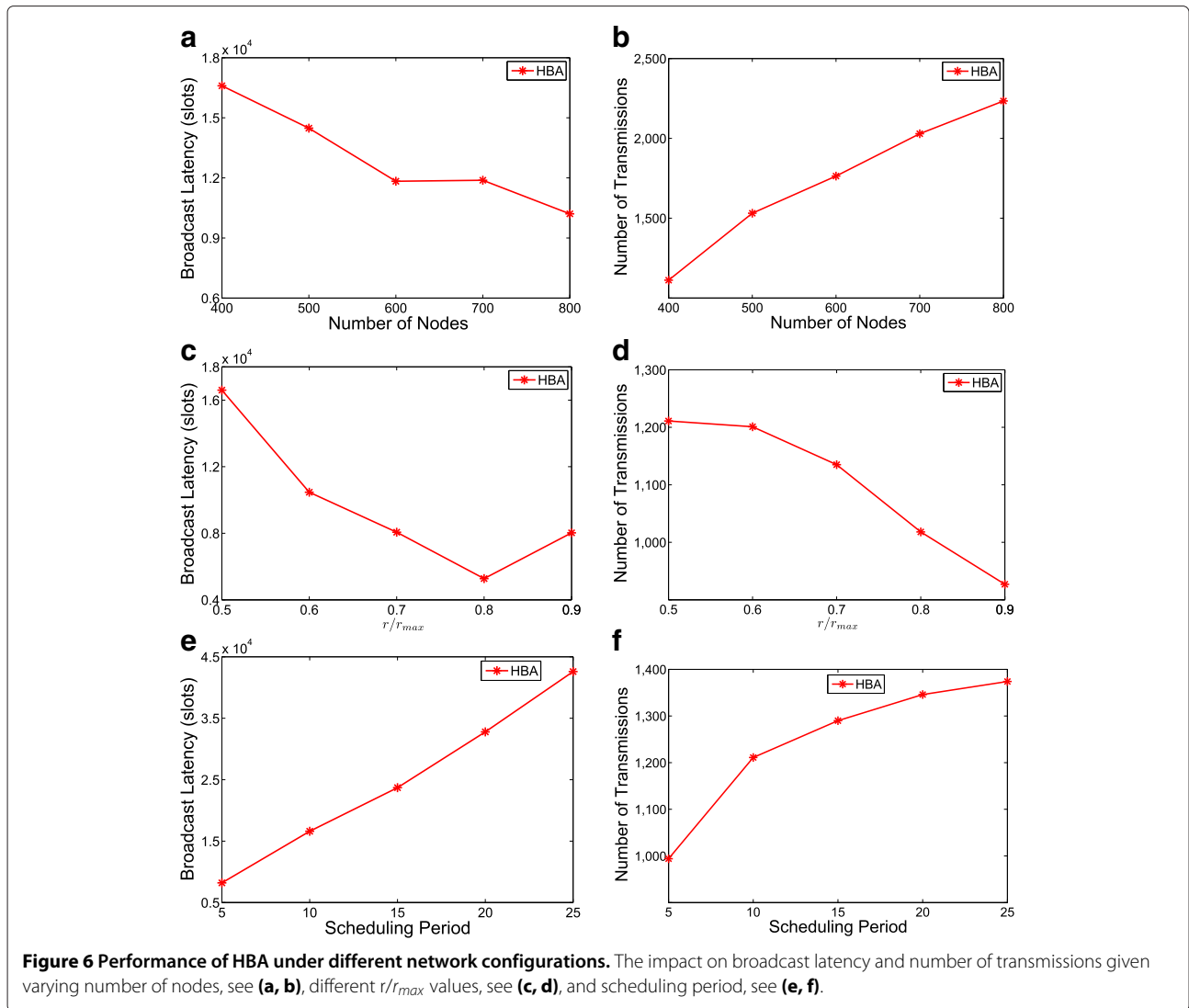
Figure 6c,d shows the performance of HBA under different r/r_{\max} , where r is the transmission range in G_T , and r_{\max} is the maximum transmission range with a fixed value of 100 m. In this experiment, the number of nodes is fixed to 400, T is set to 10, and r ranges from 50 to 90 m. As shown in Figure 6c, the broadcast latency decreases when r ranges from 50 to 80 m. This is because a larger transmission range r leads to more links and higher connectivity. As a result, HBA is able to find shorter broadcast paths. On the other hand, according to Theorem 1, a larger r also prevents more nodes from transmitting simultaneously, which will result in longer broadcast paths. Hence, when the transmission range r exceeds 80 m in Figure 6c, the broadcast latency starts to increase. We see that in Figure 6d the number of transmissions decreases with increasing r . This is because the number of hexagons used to cover the network becomes smaller when r becomes larger. Hence, there are fewer transmissions when r is large.

Figure 6e,f depicts the performance of HBA for different scheduling periods T . The value of T ranges from 5 to 25 ($r = 50$ m and the number of nodes is set to 400). Note that the broadcast latency and number of transmissions increase with higher T values. A larger T will result in larger frames and thereby leads to higher broadcast latency. Consequently, for each hexagon, a receptor needs to transmit more times to its neighbors with different active time slots.

6.2 Performance of HBA versus tree-based algorithm

In this section, we compare HBA against the tree-based algorithm [7] for always-on networks. Recall that the tree-based algorithm [7] is the first centralized method designed for always-on networks under SINR-based interference model. In this respect, we note that HBA is the first distributed algorithm designed for duty-cycle networks under the SINR-based interference model. In order to compare HBA faithfully against the tree-based algorithm [7], the scheduling period used by HBA is set to 1, i.e., $T = 1$, meaning HBA also works in the always-on mode.

As shown in Figure 7a, the broadcast latency of the tree-based algorithm is around two times larger than that of HBA. This is mainly because the tree-based algorithm is conducted layer by layer based on the BFS tree and nodes in lower layers are prevented from transmitting until all nodes in the current layer have finished their transmissions even though these transmissions do not cause any interference; instead, transmissions in HBA is handled in a greedy manner, which allow nodes to transmit as long as they do not result in interference. In addition, we observe that the number of transmissions experienced by



nodes using HBA is about 50% more than that of the tree-based algorithm. This is because the tree-based algorithm determines the transmitting nodes in the maximum independent set centrally. This, however, does not consider the cost of requiring all nodes to synchronize their wake-up schedule and collating such information centrally.

7 Conclusions

In this paper, we have studied the minimum latency broadcast scheduling problem in duty-cycled WSNs. To achieve interference-free broadcast, we designed a novel algorithm, called HBA, for nodes that employ a random duty-cycle schedule. We prove that HBA provides a correct and interference-free schedule, produces a low broadcast latency, and has low overheads. Our simulation results show HBA to have better performance in terms of broadcast latency than the tree-based algorithm [7]. As a future work, we are currently looking into implementing

HBA in the more realistic and probabilistic interference model where a message can be received successfully with varying probability as per SINR levels.

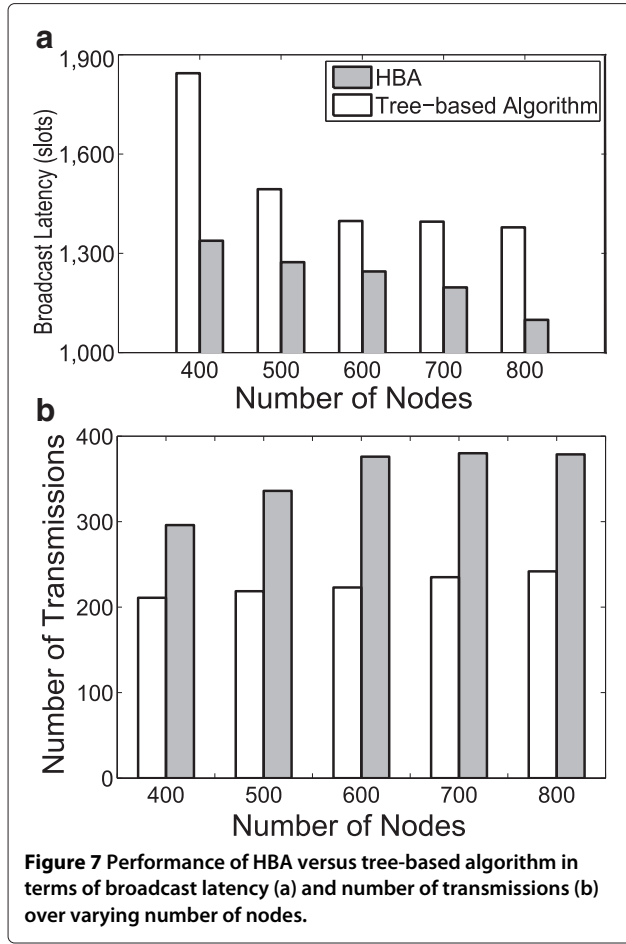
Endnote

^aA ρ -approximation algorithm, with approximation ratio $\rho > 1$, if given a problem instance I to minimize and with optimal solution $OPT(I)$, produces a solution that is bounded by $\rho \cdot OPT(I)$.

Appendix

Lemma 3. Given a set L of links, if the mutual distance of senders in L are greater than ρr , set L is independent, where $\rho = 1 + \left(\frac{8\beta}{1-(r/r_{max})^\alpha} \left(\frac{2}{\alpha-2} + \frac{1}{\alpha-1} + 3 \right) \right)^{1/\alpha}$.

Proof 5. Let (u, v) denote a link belonging to L . With sender u as the centre, we partition the senders in L into



concentric rings $ring^k$ with width ρr . Ring $ring^k$ contains all senders w of links in L satisfying $k\rho r \leq d(w, u) < (k+1)\rho r$. The first ring $ring^0$ only contains sender u . We now consider all senders $w \in ring^k$ for some integer $k > 0$.

First, we consider the distance between any senders w in $ring^k$ and u . As per the construction of rings, we have $d(w, u) \geq k\rho r$ for ring $ring^k$. Note that, $d(u, v) \leq r$ and $\rho > 1$. Applying the triangle inequality, the lower-bound of $d(w, v)$ for $ring^k$ is,

$$\begin{aligned} d(w, v) &\geq d(w, u) - d(u, v) \\ &\geq k\rho r - r \\ &> (\rho - 1)kr \end{aligned} \quad (3)$$

Next, observe that for any senders w in $ring^k$, the disk centered at w with a radius of $\frac{1}{2}\rho r$ is non-overlapping with other senders in $ring^k$, and such a disk is fully contained in an extended ring of $ring^k$, with an extra width of $\frac{1}{2}\rho r$ at each side of $ring^k$. Then, by referring to the ratio between the area of this extended ring and the disk with a radius

of $\frac{1}{2}\rho r$, the number of senders contained in $ring^k$ is upper-bounded by $8(2k+1)$ as per Equation 4.

$$\frac{\pi(k+3/2)^2(\rho r)^2 - \pi(k-1/2)^2(\rho r)^2}{\pi(1/2\rho r)^2} = 8(2k+1) \quad (4)$$

The total interference I_k emanating from $ring^k$ is bounded by

$$\begin{aligned} I_k &= \sum_{w \in ring^k} P d(w, v)^{-\alpha} \\ &\leq 8(2k+1)P((\rho-1)kr)^{-\alpha} \end{aligned} \quad (5)$$

Summing up the total interferences I over all rings yields

$$\begin{aligned} I &= \sum_{k=1}^{\infty} I_k \\ &= \sum_{k=1}^{\infty} 8(2k+1)P((\rho-1)kr)^{-\alpha} \end{aligned} \quad (6)$$

Recall that $d(u, v) \leq r$ and $N = P/\beta r_{max}^\alpha$, where r_{max} is the maximum transmission range in the absence of interference. If v successfully receives a message from u if and only if the following condition holds:

$$\begin{aligned} SINR &= \frac{P d(u, v)^{-\alpha}}{I + N} \\ &\leq \frac{P r^{-\alpha}}{\sum_{k=1}^{\infty} 8(2k+1)P((\rho-1)kr)^{-\alpha} + P/\beta r_{max}^\alpha} \\ &= \frac{\beta}{\sum_{k=1}^{\infty} 8\beta(2k+1)(\rho-1)^{-\alpha} k^{-\alpha} + (r/r_{max})^\alpha} \\ &\leq \beta \end{aligned} \quad (7)$$

According to inequality 7, such SINR is at least β if and only if

$$\sum_{k=1}^{\infty} 8\beta(2k+1)(\rho-1)^{-\alpha} k^{-\alpha} + (r/r_{max})^\alpha \leq 1 \quad (8)$$

According to Riemann zeta function, we know that $\sum_{k=1}^{\infty} k^{-\alpha} \leq \frac{1}{\alpha-1} + 1$, where $\alpha > 2$.

Plugging this in, we have

$$\sum_{k=1}^{\infty} 8(2k+1)k^{-\alpha} \leq 8 \left(\frac{2}{\alpha-2} + \frac{1}{\alpha-1} + 3 \right) \quad (9)$$

According to inequality 8 and 9, we have

$$\begin{aligned} &\sum_{k=1}^{\infty} 8\beta(2k+1)(\rho-1)^{-\alpha} k^{-\alpha} + (r/r_{max})^\alpha \\ &= 8\beta \left(\frac{2}{\alpha-2} + \frac{1}{\alpha-1} + 3 \right) (\rho-1)^{-\alpha} + (r/r_{max})^\alpha \\ &\leq 1 \end{aligned} \quad (10)$$

When $\rho = 1 + \left(\frac{8\beta}{1-(r/r_{max})^\alpha} \left(\frac{2}{\alpha-2} + \frac{1}{\alpha-1} + 3 \right) \right)^{1/\alpha}$, inequality 10 must hold; in other words, receiver v can

receive the message successfully in L . In conclusion, set L is independent.

Lemma 4. Given a set L of links, if the mutual distance of receivers in L is greater than ρr , set L is independent, where

$$\rho = 1 + \left(\frac{8\beta}{1-(r/r_{\max})^\alpha} \left(\frac{2}{\alpha-2} + \frac{1}{\alpha-1} + 3 \right) \right)^{1/\alpha}.$$

Proof 6. Suppose that a link (u, v) is in L . With the center as receiver v , we then divide the receivers in L into concentric rings ring^k . Recall that the length of each link is upper-bounded by r in L . For a sender w whose receivers lie in ring^k , $d(w, v)$ is lower-bounded by $\rho kr - r \geq (\rho - 1)kr$. That is, for ring^k , the distance between interfered sender w and receiver v is no smaller than $(\rho - 1)kr$. Next, using the same argument as Lemma 5, we conclude that L is also independent.

Theorem 5. Given a set L of links, in order for set L to be independent, it is sufficient to have:

1. The mutual distance of senders are all greater than ρr ; OR
2. The mutual distance of receivers are all greater than ρr .

$$\text{where } \rho = 1 + \left(\frac{8\beta}{1-(r/r_{\max})^\alpha} \left(\frac{2}{\alpha-2} + \frac{1}{\alpha-1} + 3 \right) \right)^{1/\alpha}.$$

Proof 7. This is proved by Lemma 5 and 6.

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

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

Received: 2 October 2014 Accepted: 12 February 2015

Published online: 28 March 2015

References

1. C Perkins, E Royer, in *IEEE WMCSA*. Ad-hoc on-demand distance vector routing (New Orleans, LA, USA, 1999)
2. W Nan, S Xue-li, in *IEEE IITAW*. Research on nodes location technology in wireless sensor network underground (Nanchang, China, 2009)
3. S Brown, C Sreenan, Updating software in wireless sensor networks: A survey (2006). Technical Report UCC-CS-2006-13-07, University College Cork, Ireland, (2006)
4. R Gandhi, Y-A Kim, S Lee, J Ryu, P-J Wan, in *IEEE INFOCOM*. Approximation algorithms for data broadcast in wireless networks (Rio de Janeiro, Brazil, 2009)
5. R Gandhi, A Mishra, S Parthasarathy, Minimizing broadcast latency and redundancy in ad hoc networks. *IEEE/ACM Trans. Netw.* **16**(4), 840–851 (2008)
6. Z Chen, C Qiao, J Xu, T Taekyeun Lee, in *IEEE INFOCOM*. A constant approximation algorithm for interference aware broadcast in wireless networks (Anchorage, Alaska, USA, 2007)
7. S-H Huang, P-J Wan, J Deng, Y Han, Broadcast scheduling in interference environment. *IEEE Trans. Mobile Comput.* **7**(11), 1338–1348 (2008)
8. S-H Huang, P-J Wan, X Jia, H Du, W Shang, in *IEEE INFOCOM*. Minimum-latency broadcast scheduling in wireless ad hoc networks (Anchorage, AK, USA, 2007)
9. R Mahjourian, F Chen, R Tiwari, M Thai, H Zhai, Y Fang, in *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*, MobiHoc '08. An approximation algorithm for conflict-aware broadcast scheduling in wireless ad hoc networks (ACM New York, NY, USA, 2008), pp. 331–340
10. R Tiwari, T Dinh, M Thai, On centralized and localized approximation algorithms for interference-aware broadcast scheduling. *IEEE Trans. Mobile Comput.* **12**, 233–247 (2013)
11. P Gupta, P Kumar, The capacity of wireless networks. *IEEE Trans. Inf. Theory.* **46**(2), 388–404 (2000)
12. S Huang, S-Y Chang, H-C Wu, P-J Wan, Analysis and design of a novel randomized broadcast algorithm for scalable wireless networks in the interference channels. *IEEE Trans. Wireless Commun.* **9**(7), 2206–2215 (2010)
13. P-J Wan, L Wang, O Frieder, in *IEEE MASS*. Fast group communications in multihop wireless networks subject to physical interference (Macau SAR, China, 2009)
14. W Ye, J Heidemann, D Estrin, in *IEEE INFOCOM*. An energy-efficient MAC protocol for wireless sensor networks (New York, NY, USA, 2002)
15. K-W Chin, Pairwise: A novel time hopping MAC for wireless sensor networks. *IEEE Trans. Consumer Electron.* **55**(4), 288–304 (2000)
16. H Lim, C Kim, Flooding in wireless ad hoc networks. *Comput. Commun.* **24**(3), 353–363 (2001)
17. S Ni, YCYC Tseng, J Sheu, in *ACM MOBICOM*. The broadcast storm problem in a mobile ad hoc network (Seattle, WA, USA, 1999)
18. F Ferrari, M Zimmerling, L Thiele, O Saukh, in *IPSN*. Efficient network flooding and time synchronization with glossy (Chicago, IL, USA, 2011)
19. Y Sasson, D Cavin, A Schiper, in *IEEE WCNC*. Probabilistic broadcast for flooding in wireless mobile ad hoc networks (New Orleans, LA, USA, 2003)
20. F Stann, J Heidemann, R Shroff, MZ Murtaza, in *ACM SenSys*. Rbp: robust broadcast propagation in wireless networks (Boulder, CO, USA, 2006)
21. J Hong, J Cao, W Li, S Lu, D Chen, in *IEEE ICC*. Sleeping schedule-aware minimum latency broadcast in wireless ad hoc networks (Dresden, Germany, 2009)
22. X Jiao, W Lou, J Ma, J Cao, X Wang, X Zhou, Minimum latency broadcast scheduling in duty-cycled multi-hop wireless networks. *IEEE TPDS*. **23**(1), 110–117 (2012)
23. X Xu, J Cao, P-J Wan, in *WASA*. Fast group communication scheduling in duty-cycled multihop wireless sensor networks (Fusionopolis, Singapore, 2012)
24. P Gupta, PR Kumar, The capacity of wireless networks. *IEEE Trans. Inf. Theory.* **46**(2), 388–404 (2000)
25. Y Gu, T He, in *IEEE ICDCS*. Bounding communication delay in energy harvesting sensor networks (Genoa, Italy, 2010)
26. M Maróti, B Kusy, G Simon, A Lédeczi, in *SenSys*. The flooding time synchronization protocol (Baltimore, Maryland, 2004)
27. Y Gu, T He, in *ACM SenSys*. Data forwarding in extremely low duty-cycle sensor networks with unreliable communication links (ACM, Sydney, Australia, 2007)
28. P Dutta, D Culler, in *SenSys*. Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications (ACM Raleigh, NC, USA, 2008)
29. J Hong, W Li, S Lu, J Cao, D Chen, in *IEEE ICPADS*. Sleeping schedule aware minimum transmission broadcast in wireless ad hoc networks (Melbourne, Victoria, Australia, 2008)
30. C Hua, T-SP Yum, Asynchronous random sleeping for sensor networks. *ACM Trans. Sensor Netw.* **3**(3) (2007)
31. G Han, H Xu, TQ Duong, J Jiang, T Hara, Localization algorithms of wireless sensor networks: A survey. *Springer Telecommunication Syst.* **52**(4), 2419–2436 (2013)
32. O Goussevskaia, T Moscibroda, R Wattenhofer, in *ACM POMC*. Local broadcasting in the physical interference model (ACM, Toronto, Canada, 2008)
33. MM Halldórsson, P Mitra, in *ACM FOMC*. Towards tight bounds for local broadcasting (ACM, Madeira, Portugal, 2012)
34. D Yu, Y Wang, Q-S Hua, FCM Lau, in *IEEE DCOSS*. Distributed local broadcasting algorithms in the physical interference model (Barcelona, Spain, 2011)