EURASIP Journal on Wireless Communications and Networking
a SpringerOpen Journal

RESEARCH Open Access

# Approximation algorithm for data broadcasting in duty cycled multi-hop wireless networks

Dianbo Zhao* and Kwan-Wu Chin

**Abstract**

Broadcast is a fundamental operation in wireless networks. To this end, many past studies have studied the NP-hard, broadcast problem for always-on multi-hop networks. However, in wireless sensor networks, nodes are powered by batteries, meaning, they have finite energy. Consequently, nodes are required to have a low duty cycle, whereby they switch between active and sleep state periodically. This means that a transmission from a node may not reach all of its neighbors simultaneously. Consequently, any developed broadcast protocols must consider collisions and the wake-up times of neighboring nodes. Henceforth, this paper studies the minimum latency broadcast scheduling problem in duty cycled multi-hop wireless networks (MLBSDC), which remains NP hard. The MLBSDC problem aims to find a collision-free schedule that minimizes the time in which the last node receives a broadcast message. We propose a novel algorithm called CFBS that allows nodes in different layers of the broadcast tree to transmit simultaneously. We prove that CFBS produces a latency of at most $(T+1)H + T\mathcal{O}(\log_2 H)$. Here, $T$ denotes the number of time slots in a scheduling period, and $H$ is the optimal broadcast latency obtained from the shortest path tree algorithm assuming no collision. We also show that the total number of transmissions is at most $4(T+2)$ times larger than the optimal value. The results from extensive simulation show that CFBS has a better performance than OTAB, the best broadcast scheduling algorithm to date. In particular, the broadcast latency achieved by CFBS is up to $\frac{3}{20}$ that of OTAB.

## 1 Introduction

Wireless sensor networks (WSNs) consist of numerous sensor nodes deployed in a field. These nodes are usually resource constrained in terms of battery lifetime and computation, and are equipped with a number of sensing elements. Moreover, they have one or more radios and communicate with each other via multi-hop communications because these radios have a bounded and short transmission range. In addition, there exist one or more sinks to collect sensed data and to issue commands that affect the operation of sensor nodes. To date, WSNs have found a myriad of applications. For example, precision agriculture [1], monitoring of pests [2], and volcanology [3] to name a few.

Network-wide broadcast is a fundamental operation in wireless networks, where a message needs to be propagated from a source node, e.g., a sink, to all other nodes. It is relied upon by several network protocols, such as routing [4], information dissemination [5], and resource/services discovery [6]. These protocols in turn help applications in disaster relief, military communication, rescue operation, and object detection [7]. 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 collision. Unfortunately, the MLBS problem for multi-hop wireless networks has been proven to be NP hard [8], and researchers have proposed many approximation algorithms. These algorithms, however, assume that all nodes are always active. They typically make use of neighborhood information to determine whether a node needs to transmit a message. Specifically, collisions can be detected by identifying the common neighbors of two or more transmitting nodes via topological information and ensuring the interfering nodes transmit in different time slots.

*Correspondence: dz985@uowmail.edu.au
School of Electrical, Computer & Telecommunications Engineering, University of Wollongong, Wollongong, 2500, Australia

In contrast, the MLBS problem is quite different in duty cycled WSNs. Briefly, in these networks, nodes are powered by batteries and are only awake for a fraction of the time [9]. Here, the duty cycle of a node is defined as the ratio between its active time and the scheduling period, i.e., $T$. We note that WSNs can employ a synchronous wake-up schedule, that is, nodes wake up at the same time. However, nodes will have to coordinate and synchronize their wake-up time globally and, hence, incur high signaling overheads. This paper, therefore, only considers WSNs with asynchronous schedule, where nodes determine their wake-up time independently and randomly.

As an example, consider Figure 1. Node $S$ needs to broadcast a message to nodes $A$, $B$, $C$, and $D$. All of which have different wake-up times, i.e., time slot '1', '3', '5', and '5', respectively. Here, node $S$ may transmit the message at least three times because its neighbors $A$, $B$, and $C$ have different wake-up times. Moreover, assuming node $A$ has received the message from $s$ at time slot '1' and $B$ received the message from $S$ at time slot '3', nodes $S$, $A$, and $B$ may try to forward the message to their neighbors at time slot '5'. However, this will cause a collision at nodes $C$ and $D$. Considering the fact that $B$ is adjacent to nodes $C$ and $D$, both with the same wake-up time of '5', one feasible way to conduct the broadcast is for $S$ to send the message to $A$ and $B$ at time slot '1' and '3', respectively, after which $B$ transmits it to $C$ and $D$ at time slot '5'. As we can see, both topology and wake-up schedule of nodes are key issues to consider when solving the MLBSDC problem. In fact, this consideration renders the MLBS problem more complex, meaning, existing algorithms for always-on wireless networks are no longer applicable.

Henceforth, this paper presents the design and evaluation of a novel approximation algorithm that has significantly better performance than prior solutions. Specifically, it contains the following contributions:

1. A novel algorithm called centralized collision-free broadcast scheduling (CFBS) that is suitable for both always-on and duty cycled networks. CFBS produces a broadcast latency of at most $(T+1)H + T\mathcal{O}(\log_2 H)$, where the constant before $T\mathcal{O}(\log_2 H)$ does not exceed 108. In particular, for always-on networks, i.e., $T = 1$, the broadcast latency of CFBS is bounded by $2R+\mathcal{O}(\log_2 R)$, where $R$ is the maximum hop distance from the source to any node.
2. The total number of transmissions produced by CFBS is at most $4(T + 2)$ times that of the minimum total number of transmissions. For always-on networks, this approximation ratio is 12.
3. We evaluate CFBS 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 state of the art algorithm OTAB [10]. The key reason is our algorithm is able to schedule transmissions in multiple layers as opposed to layer by layer, as is done by OTAB. Moreover, it allows non-interfering nodes in lower layers to transmit even though nodes in the current layer have not finished their transmission.

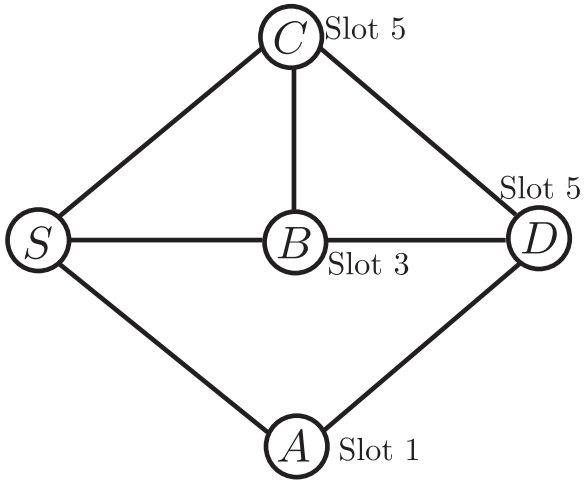


**Figure 1 Broadcast in duty cycled WSNs.**

## 2 Related works

To date, there are many approaches to carry out broadcast in multi-hop wireless networks. The simplest by far is flooding [11], where each node simply re-transmits a received message to its neighbors unscrupulously. However, this causes broadcast storms [12] and is thus very costly and causes long latencies. Consequently, a number of researchers, e.g., [13-15], 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, collision-free schedule that yields the minimum broadcast latency.

Gandhi et al. [8] 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 [7]. Huang et al. [16] 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 [7].

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, there are only a handful of works related to broadcast in duty cycled wireless networks. Lai and Ravindran [17] and Hong et al. [18] designed centralized and distributed broadcast algorithms for duty cycled networks that aim to reduce the number of transmissions.

In particular, the two methods proposed in [19] have an approximation ratio of $3(\ln\Delta + 1)$ and 20 in terms of the number of transmissions, respectively, where $\Delta$ is the maximum degree. These works, however, have not addressed the MLBSDC problem in duty cycled networks.

To date, there are only a handful of directly related works. Hong et al. [20] proved that the MLBSDC problem is NP hard and proposed two approximation algorithms: SLAC and ELAC. Their algorithms achieve an approximation ratio of $\mathcal{O}((\Delta^2+1)T)$ and $24T+1$, respectively, where $\Delta$ is the maximum degree, and $T$ denotes the number of time slots in a scheduling period. Both algorithms apply the D2-coloring approach [21] to schedule transmissions on a shortest path tree. In [10], Jiao et al. show that ELAC can be improved further by using D2-coloring twice at each layer of the shortest path tree. They 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. Duan et al. [22] provide a generalized algorithm for the MLBSDC problem with an approximation ratio of $\Delta T$. They transform the MLBSDC problem into the conventional maximum independent set problem and try to find a maximum set of non-interfering senders in each time slot. Recently, Xu et al. [23] extended the pipelined broadcast scheme in [16] to consider duty cycled WSNs. Their broadcast algorithm produces a latency of at most $TH + T\mathcal{O}(\log_2 H)$, where the omitted constant in $T\mathcal{O}(\log_2 H)$ also exceeds 150; in contrast, our solution has 108 as a constant in $T\mathcal{O}(\log_2 H)$.

The key limitation of [20] and [10] is that transmissions are scheduled layer by layer based on a shortest path tree, which prevent non-interfering nodes in lower layers from transmitting until all nodes in the current layer finish their transmissions. The broadcast latency performance of [22] is mainly influenced by the maximum degree of nodes, i.e., $\Delta$, which produces a large bound for dense networks. Unlike [20] and [10], our proposed algorithm is able to schedule nodes' transmissions in more than one layer, leading to a lower latency. The broadcast latency of CFBS is mainly influenced by $H$, which does not rely on the number of nodes or maximum degree. All these features constitute key advantages over [22] and also result in an algorithm that is suitable for dense networks.

## 3 Preliminaries

### 3.1 Network model

We consider a duty cycled WSN which has a scheduling period that is divided into $T$ slots of fixed and equal length, and is indexed by $0, 1, 2, \cdots, T-1$. Each time slot is assumed to be of sufficient duration to receive a message. We assume that the network is locally synchronized at a slot level. As shown in [24], this can be achieved using local synchronization techniques, such as Flooding Time Synchronization Protocol (FTSP) [25]. The duty cycle of a node is defined as $\frac{1}{T}$, where the numerator corresponds to one active time slot. Similar to [10,20,26], each node $v$ selects to wake up at a time slot in the range $[0, 1, \cdots, T-1]$ randomly and independently in order to receive a message. We will denote node $v$'s wake-up slot as $\tau(v)$. If node $v$ wants to transmit a message, it will wake up at the corresponding receiver's wake-up slot. Here, we assume there is no message or bit error, and links are bidirectional. This is reasonable because any retransmissions due to bit errors can be accounted for by dimensioning the slot size accordingly. However, a message is considered lost if there is a collision, i.e., two or more simultaneous transmissions to a common node. A node must not receive and send a message at the same time. We will use $N(v)$ to denote the set of one-hop neighbors of node $v \in V$, and $n$ is the cardinality of $V$, i.e., $n = |V|$.

The duty cycled WSN is modeled as a weighted unit disc graph (UDG) $G = (V, E)$, where $V$ is the set of nodes, and $E$ represents the set of edges/links that exist between two nodes if their Euclidean distance is no more than a given transmission range. Furthermore, each edge in $V$ has an associated numerical value, called a weight or cost. This corresponds to the time interval between two nodes' active time slots. Specifically, for each edge $(u, v) \in E$, its *cost*, denoted as $edc(u, v)$, is defined as per Equation 1, where $s$ is the source node.

$$edc(u,v) = \begin{cases} \tau(v) + 1, & \text{if } u = s; \\ \tau(v) - \tau(u), & \text{if } u \neq s \text{ and } \tau(v) - \tau(u) > 0; \\ \tau(v) - \tau(u) + T, & \text{otherwise} \end{cases} \tag{1}$$

### 3.2 Graph definitions and theories

Given a weighted UDG $G = (V, E)$, we designate node $s$ to be the source of a broadcast message and set the cost of each edge as per Equation 1. We will denote the subgraph of $G$ induced by $U \subseteq V$ as $G[U]$. The shortest path tree $T_{\text{spt}}(G)$ of $G$ with respect to node $s$ is the spanning tree obtained by applying Dijkstra's algorithm from $s$. The *depth* of a node $v \in V$ is the total cost of the path from $s$ to $v$ in $T_{\text{spt}}(G)$, and the *radius* of $G$ with respect to $s$, denoted by $\text{Rad}(G, s)$, is the maximum depth/cost of the paths in $T_{\text{spt}}(G)$. In our solution, $T_{\text{spt}}(G)$ is divided into different layers according to the depth of nodes in increasing order. This means that each layer $i$ of $T_{\text{spt}}(G)$ consists of all nodes with the same depth/cost. Let $\text{Depth}(G, i)$ be the depth of nodes at layer $i$. Note that node $s$ is at layer 0. Let $R$ be the maximum hop distance from the source node $s$ to any other nodes. We thus have $\text{Rad}(G, s) \geq R$. Note,

in an always-on network, which can be modeled by setting $T = 1$, we have $\text{Rad}(G, s) = R$.

An *independent set* (IS) $I$ of $G(V, E)$ is defined as a subset of $V$, such that if $u, v \in I$, then $(u, v) \notin E$. A *maximal independent set* (MIS) $U$ is an independent set which is not a subset of any other independent sets. A subset $U$ of $V$ is a *dominating set* of $G$ if each node not in $U$ is adjacent to at least a member of $U$. Clearly, every MIS of $G$ is also a dominating set of $G$. If set $U$ is a dominating set of $G$ and $G[U]$ is connected, then $U$ is called a *connected dominating set* (CDS) of $G$. The authors of [27] showed that the MIS size of a UDG graph is bounded by $\mathcal{O}(R^2)$. It is also known that the size of MIS does not exceed $4opt + 1$, where *opt* denotes the minimum size of a CDS of $G$ [28].

A proper *D2-coloring* [21] of $G$ is an assignment of colors, labeled by natural numbers to the nodes in $V$, such that any pair of nodes within two-hop neighborhood receives different colors. Any node ordering $v_1, v_2, \cdots, v_n$ of $V$ induces a proper node coloring of $G$ in the first-fit manner. Specifically, for $i = 1$ to $n$, assign node $v_i$ the least assigned color that is not used by any neighbor $v_j$, where $j < i$. For example, consider a line topology $A - B - C$. A proper coloring results in the color assignments 1, 2, and 1 to nodes $A$, $B$, and $C$, respectively. A particular node ordering of interest is the *smallest-degree-last ordering* [29]. For $i = n$ to 1, it sets $v_i$ to the node with smallest degree in $G[U]$, where $U \subseteq V$. Initially, set $U$ to $V$, and then repeat the following iteration until $U$ becomes empty: for $i = n$ to 1, set $v_i$ to the node with the smallest degree in $U$, and remove it from $U$. Consider the line topology $A - B - C$. The smallest-degree-last ordering will be $C - B - A$. A summary of notations used in this paper can be found in Table 1.

**Table 1 Commonly used notations**

| Notation | Meaning |
|---|---|
| $G(V,E)$ | Network graph |
| $N(v)$ | $v$'s one-hop neighbors |
| $T$ | Scheduling period |
| $H$ | Broadcast latency bound |
| $\tau(v)$ | $v$'s active slot |
| $T_{\text{spt}}(G)$ | Shortest path tree (SPT) |
| $P_{ij}$ | Transmissions from $S_{ij}$ to $V_{ij}$ |
| $\text{Rad}(G,s)$ | Maximum depth of paths in $T_{\text{spt}}(G)$ |
| $S_{ij}$ | Nodes with rank $j$ that are parents of $V_{ij}$ |
| $G[U]$ | Subgraph of graph $G$ |
| $U_i$ | Dominators in layer $i$ |
| $C$ | Set of connectors |
| $L$ | Maximum layer number |
| $edc(u,v)$ | Cost of edge $(u,v)$ |
| $rank(v)$ | Rank of node $v$ |
| $t_{ij}$ | Starting transmission time of $P_{ij}$ |
| $\text{Depth}(G,i)$ | Depth of nodes in layer $i$ of $T_{\text{spt}}(G)$ |
| $V_{ij}$ | Nodes in layer $i$ whose parents have rank of $j$ |

### 3.3 Problem formulation

Our problem, called MLBSDC, concerns the broadcast of a message from a source node $s \in V$ to all other nodes. The goal is to minimize the time in which the last node receives the message. Without loss of generality, we define the start time of node $s$'s broadcast to be slot zero, and the broadcast latency is the maximum time taken by a message to reach all nodes.

We model the MLBSDC problem as follows. Let $(S_i, t_i + k_i T)$ denote the $i$th transmission, and $i, k_i \in \mathbb{N}$. At the $i$th transmission, the nodes in the set $S_i$ transmit the message to nodes in $R_i$ at time $t_i + k_i T$, where $R_i$ denotes the set of nodes that received the message from nodes in $S_i$ collision free, and $t_i$ is the active time slot of nodes in $R_i$. The MLBSDC problem is then to find a forwarding schedule

$$S = \{(s, 0), (S_1, t_1 + k_1 T), \cdots, (S_m, t_m + k_m T)\} \quad (2)$$

that satisfies the following constraints: (1) $t_1 + k_1 T < t_2 + k_2 T < \ldots < t_m + k_m T$, (2) any node in $S_i$ cannot be scheduled to transmit the message until it receives the message, (3) all transmissions from $S_i$ to $R_i$ must be collision free, (4) $|\bigcup_{i=1}^{m} R_i| = |V|$, and $t_m + k_m T$ is minimum. In other words, find a collision-free broadcast schedule that guarantees that all the nodes in $V$ receive the message collision free in minimum time.

## 4 Proposed algorithm

In this section, we present CFBS, a collision-free broadcast algorithm with a latency of at most $(T + 1)H + T\mathcal{O}(\log_2 H)$, where the omitted constant in $T\mathcal{O}(\log_2 H)$ is 108. Different from OTAB, where transmissions are processed layer by layer. CFBS is able to schedule transmissions in more than one layer, that is, it allows a node in a lower layer to transmit or receive earlier than a node in an upper layer.

### 4.1 Inner-layer broadcast scheduling

Before outlining CFBS, we first describe the inner-layer broadcast scheduling (ILBS) algorithm, which is responsible for scheduling the broadcast of two disjoint sets of nodes with a latency of at most 17. As we will see in the following section, ILBS is used to schedule the broadcast between nodes in the same layer. We like to note that ILBS is similar to the algorithm outlined in [10]. However, their algorithm, which schedules transmissions layer by layer, leads to longer broadcast latency.

Let $X$ and $Y$ be two disjoint subsets of $G$. The set $X$ is a cover of $Y$, where each node in $Y$ is adjacent to some

nodes in $X$. ILBS takes as input $G[X \cup Y]$ and outputs a broadcast schedule from $X$ to $Y$. ILBS starts by constructing a MIS $U$ from $G[Y]$. This ensures that the minimal number of nodes is used to broadcast a message. It then assigns a parent to nodes in $U$ from the set $X$. Then, a subset of nodes in $U$ are chosen as the parents of nodes in $Y \setminus U$. Specifically, the *selection order* is such that a node becomes a parent if it covers the most nodes in $U$ (respectively, $Y \setminus U$) that have yet to be assigned a parent. These nodes will then receive the message from their designated parent.

The next step is to determine a collision-free transmission schedule for parent nodes. This is carried out as follows. First, ILBS collects the parents of nodes in $U$ and $Y \setminus U$ into two corresponding subsets $W_1$ and $W_2$ according to the said selection order. Then to schedule interfering parent nodes, it uses two D2-coloring methods: (1) *front-to-back ordering*, whereby the coloring proceeds from the first to the last node and (2) *smallest-degree-last ordering*, with the rule being that two parent nodes must not share the same color if a subset of a parent's children is adjacent to another, i.e., a parent node's transmission interferes with the reception of another parent's children.

ILBS first colors parent nodes in $W_1$ using *front-to-back ordering* and divides them into a sequence $\langle W_1(i) : 1 \le i \le f \rangle$ based on nodes' color, that is, the set $W_1(i)$ contains nodes with color $i$ and, hence, are able to transmit simultaneously. Then, it assigns the color of nodes in $W_2$ using *smallest-degree-last ordering* and collects nodes with color $i$ into $W_2(i)$ for $1 \le i \le c$. This thus yields the broadcast schedule $\langle W_1(i) : 1 \le i \le f \rangle$ and $\langle W_2(i) : 1 \le i \le c \rangle$.

As proven in [10], $f \le 5$ and $c \le 12$, and hence, the latency by ILBS is at most 17. By letting $W = W_1 \cup W_2$, the broadcast schedule can be presented as $\langle W(1), W(2), \cdots, W(l) \rangle$, where $l = f + c \le 17$.

We illustrate the operation of ILBS on the subgraph shown in Figure 2; note that the said subgraph is extracted from Figure 3, which we will revisit later. First, we collect nodes $v_2$ and $v_3$ into set $X$, and nodes $v_5$, $v_6$, $v_7$, $v_8$, and $v_9$ into set $Y$. As shown in Figure 2, there is an edge between nodes $v_5$ and $v_6$, and thus the MIS $U$ of $Y$ is set to $\{v_6, v_7, v_8, v_9\}$. Node $v_3$ covers the maximum number of nodes in $U$, and therefore, it is first selected to transmit the message to $v_7$, $v_8$, and $v_9$. Accordingly, node $v_6 \in U$ will get the message from node $v_2$ which is adjacent to node $v_6$ and $v_7$ of $U$. Node $v_5$ will receive the message from a dominator in $U$ such as $v_6$.

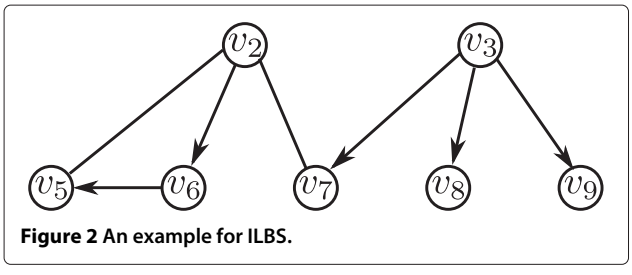


**Figure 2 An example for ILBS.**

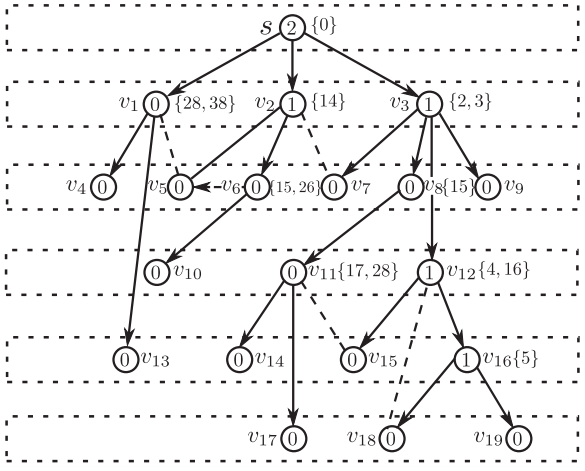


**Figure 3 The ranking and broadcast scheduling of shortest path tree $T_{\text{spt}}(G[U \cup C])$ consisting of solid edges.** The number inside each circle is a node's rank, and the numbers in curly braces correspond to a node's transmission times.

ILBS then applies *front-to-back ordering* to color parent nodes in $W_1$, i.e., $W_1 = \{v_3, v_2\}$. As one of node $v_3$'s children, node $v_7$ is also adjacent to node $v_2$, two colors will be needed to color them, i.e., $v_3$ is colored 1, and $v_2$ is colored 2. That is, $W_1(1) = \{v_3\}$ and $W_1(2) = \{v_2\}$. Node $v_5$ only gets the message from $v_6 \in W_2$, sand $v_6$ is colored 1 as per *smallest-degree-last ordering*, i.e., $W_2(1) = \{v_6\}$. The broadcast schedule can be presented as $W(1) = \{v_3\}$, $W(2) = \{v_2\}$, and $W(3) = \{v_6\}$.

### 4.2 CFBS algorithm

Recall that the main idea of CFBS is to schedule transmissions in more than one layer to speed up the broadcast. This is achieved using three key steps: (1) computing a CDS of $G$, (2) associating a *rank* to nodes in the CDS, and (3) scheduling transmissions based on the constructed CDS and nodes' ranks.

#### *4.2.1 CDS construction*

The NP-hard problem of computing a minimum CDS of $G$ is well studied, see [28,30,31], and references therein, and there are many approximation algorithms. However, for our problem, we not only require a small-size CDS but also one that has a small radius. To this end, we propose a new heuristic solution that achieves both objectives.

CFBS starts by constructing the shortest path tree $T_{\text{spt}}(G)$ via Dijkstra's algorithm. Then, it constructs the MIS $U$ of $G$ to form a backbone by considering one layer at a time starting from layer 0. In particular, source node

$s$ will be the first node to be added into $U$, and no nodes in layer 1 of $T_{\text{spt}}(G)$ will be selected because they must be adjacent to node $s$. The process then continues for layer 2 and so forth, whereby nodes at each layer which are not adjacent to those in $U$ are selected greedily. From hereon, we will refer to nodes in $U$ as *dominators*.

To ensure connectivity, the next step is to select *connector* nodes; recall that $G[U]$ is not connected as per the definition of MIS. Let $U_i$ be set of dominators in layer $i$, and $C$ be the set of selected connectors. The set $C$ is populated layer by layer in a top-down manner. Specifically, a connector is chosen from nodes in an upper layer $j$, where $j < i$, that covers the most dominators in $U_i$ that have yet to be covered by other connectors. Upon completion, we thus have $G[U \cup C]$, whereby $U \cup C$ is a CDS of $G$.

**Lemma 1.** *The resultant CDS satisfies the following properties:*

1. $|U \cup C| \leq 2|U| - 1 \leq 2\mathcal{O}(Rad(G,s)^2) - 1$
2. $Rad(G[U \cup C], s) \leq (T+1)Rad(G,s) - 2T$.

*Proof.* The first property is true because the connectors in $C$ are required to cover at least one dominator located in a lower layer. Hence, the number of connectors $|C|$ is bounded by $|U| - 1$, which excludes the source node $s$. The size of the CDS is thus bounded by $2|U| - 1$, which comprises $|U|$ dominators and at most $|U|-1$ connectors. As proven in [27], the size of CDS for graph $G$ is bounded by $\mathcal{O}(R^2)$. This yields the inequality $2|U|-1 \leq 2\mathcal{O}(R^2)-1$. Recall that $R \leq \text{Rad}(G,s)$, and thus we have $|U \cup C| \leq 2|U| - 1 \leq 2\mathcal{O}(\text{Rad}(G,s)^2) - 1$.

For the second property, we first count the number of edges for a path from the source node $s$ to the maximum layer number, denoted as $L$, of $T_{\text{spt}}(G)$. Observe that the dominators at layer $L$ of $T_{\text{spt}}(G)$ will remain at the lowest layer of $T_{\text{spt}}(G[U \cup C])$. The path from source node $s$ to a dominator at the lowest layer of $T_{\text{spt}}(G[U \cup C])$ consists of two kinds of edges: (1) the edge between two nodes in the same layer of $T_{\text{spt}}(G)$ and (2) the edge between two nodes from different layers of $T_{\text{spt}}(G)$. Therefore, in the worst case, there are $L-2$ edges of the first kind, i.e., from layer 2 to $L-1$ of $T_{\text{spt}}(G)$, and $L$ edges of the second kind.

Now, for the path cost, the edge cost between two nodes in the same layer is $T$ because both nodes have the same active time slot, and thus, the total cost of the $L-2$ edges of the first kind mentioned earlier is thus $(L-2)T$. For the other $L$ edges of the second kind, their total cost will not exceed the radius of $G$, i.e., $\text{Rad}(G,s)$.

The total depth or cost to a dominator at the lowest layer of $T_{\text{spt}}(G[U \cup C])$ is thus $\text{Rad}(G,s) + T(L-2)$. We know that the maximum layer number $L$ is no more than $\text{Rad}(G,s)$, and thus, the total cost to the said dominator cannot exceed $\text{Rad}(G,s) + T(\text{Rad}(G,s) - 2) = (T+1)\text{Rad}(G,s) - 2T$. As the said dominator lies at the lowest layer of $T_{\text{spt}}(G[U \cup C])$ and the depth of nodes in the lowest layer of the shortest path tree is equal to the radius of $G[U \cup C]$, we thus have the required property. □

#### 4.2.2 Ranking process

The next step is to rank the nodes in the CDS. After which, in Section 4.2.3, CFBS will use the resulting ranks to construct a broadcast schedule, whereby nodes with the greatest rank will be scheduled to transmit first. A key property of ranking is that nodes with a higher rank is able to cover more nodes or relay a message further quicker, and thus reducing broadcast latency.

The ranking process starts by constructing the shortest path tree $T_{\text{spt}}(G[U \cup C])$. Then, CFBS assigns each node in $G[U \cup C]$ with a rank layer by layer in a bottom-up manner. Initially, for any node $v \in U \cup C$, its rank is set to 0, i.e., $\text{rank}(v) = 0$. For each layer $i$ of $T_{\text{spt}}(G[U \cup C])$, collect all nodes in layer $i$ into set $M$ and repeat the following iteration until $M$ is empty. First, compute the maximum rank $r$ of nodes in $M$. Then, find a node $u$ from an upper layer that covers the most nodes with rank $r$ in $M$. If the rank of node $u$, i.e., $\text{rank}(u)$, is more than $r$, $\text{rank}(u)$ is unchanged; otherwise, it will be updated in the following way. If $u$ is adjacent to only one node in $M$ with rank $r$, then $\text{rank}(u) = r$; otherwise, $\text{rank}(u) = r + 1$. Mark node $u$ as the parent of the chosen nodes with rank $r$ in $M$ and remove it from $M$.

We now use Figure 3 to illustrate the ranking of $T_{\text{spt}}(G[U \cup C])$. In our example, $T_{\text{spt}}(G[U \cup C])$ consists of 20 nodes, and the scheduling period $T$ is set to $[0,1,2,3]$. Table 2 lists the active time slot, layer number, and depth information of all nodes in $T_{\text{spt}}(G[U \cup C])$.

Initially, all nodes in Figure 3 are assigned a rank of 0. Then, starting from the bottom layer, CFBS collects all nodes in layer 5 into set $M$, i.e., $M = \{v_{17}, v_{18}, v_{19}\}$. Next, node $v_{16}$ from layer 4 will be considered first because it covers the most number of nodes with rank 0 in layer 5, i.e., $v_{18}$ and $v_{19}$. Thus, node $v_{16}$'s rank will be updated to 1, i.e., $\text{rank}(v_{16}) = 1$ because it is adjacent to two nodes with rank 0, and its original rank is also 0. After that, nodes $v_{18}$ and $v_{19}$ are marked as the children of node $v_{16}$ and are removed from the set $M$ to yield $M = \{v_{17}\}$. Node $v_{17}$ is only covered by node $v_{11}$, and thus, node $v_{11}$ is set as

**Table 2 Active time, layers, and depths of all nodes in Figure 3**

| ID | $s$ | $v_1, v_2, v_3$ | $v_4, v_5, v_6, v_7, v_8, v_9$ | $v_{10}, v_{11}, v_{12}$ | $v_{13}, v_{14}, v_{15}, v_{16}$ | $v_{17}, v_{18}, v_{19}$ |
|---|---|---|---|---|---|---|
| $\tau$ | 1 | 0 | 2 | 3 | 0 | 1 |
| Layer | 0 | 1 | 2 | 3 | 4 | 5 |
| Depth | 0 | 1 | 3 | 4 | 5 | 6 |

the parent of $v_{17}$, and its rank remains at 0. The other layers are considered in a similar manner, and the maximum rank of $T_{\text{spt}}(G[U \cup C])$ is 2, i.e., rank$(s) = 2$.

**Lemma 2.** *For nodes in $U \cup C$, their rank holds the following properties.*

1. *For any node v and its parent node u in $G[U \cup C]$, rank$(u) \geq$ rank$(v)$;*
2. *Assume that nodes $v_1$ and $u_1$ belong to the same layer of $T_{spt}(G[U \cup C])$, with $v_2$ and $u_2$ as their parents, respectively, and all of them have the same rank, then neither $v_2$ and $u_1$ nor $u_2$ and $v_1$ are adjacent in $G[U \cup C]$;*
3. *The source node s has the maximum rank r, and $r \leq 1 + 2\mathcal{O}(\log_2(Rad(G[U \cup C], s)))$.*

*Proof.* The first property is true due to how nodes obtain their rank. To prove the second property, assume that node $v_2$ is ranked before $u_2$. When $v_2$ is ranked, nodes $v_1$ and $u_1$ are in set $M$ and have the same rank $r$. Hence, node $v_1$ must be the only neighbor of node $v_2$ with rank $r$ in the set $M$. Otherwise, if $v_2$ has two neighbors with rank $r$ in $M$, say node $v_1$ and $u_1$, the rank of node $v_2$ must be more than $r$. Therefore, the second property also holds true.

The first part of the third property is true because each node has a rank no more than its parent by the first property, and ranking is carried out in a bottom-up manner, and therefore, it follows that the source node $s$ has the maximum rank $r$. Next, we show that rank $r$ is bounded by $\mathcal{O}(\log_2(|U \cup C|))$. Denote by $N_i$ the number of nodes in layer $i$ of $T_{\text{spt}}(G[U \cup C])$ and by $r_i$ the maximum rank of nodes in layer $i$. Let $L$ be the maximum layer number of $T_{\text{spt}}[G[U \cup C]]$. First, observe that for any layer $i$, $r_i$ is no more than $r_L + (L - i)$. As ranking is carried out from layer $L$, each additional layer thereafter increases a node's rank by at most one, and thus for nodes in layer $i$, their rank increases by at most $1 \times (L - i)$, for a total of $r_L + (L - i)$. Furthermore, for any layer $i - 1$, the number of nodes with rank $r_L + (L - i) + 1$ does not exceed $N_i/2$ because in the worst case, every parent node in layer $i - 1$ with rank $r_L + (L - i) + 1$ has two children in layer $i$ that has the maximum rank $r_L + (L - i)$, which means each parent node picks at most two children in layer $i$ at a time, and the number of these said parent nodes is $N_i/2$. By induction, we have for any layer $i$ the number of nodes with rank $r_L + (L - i)$ does not exceed $N_L/2^{L-i}$, whereby $N_L/2^{L-i} \geq 1$.

Hence, we get $L - i \leq \log_2(N_L)$, and $r_L + (L - j) \leq r_L + \log_2(N_L)$. Recall that $r_L = 0$ and $N_L \leq |U \cup C|$. The rank $r_L + (L - i)$ for any layer $i$ is bounded by $\mathcal{O}(\log_2(|U \cup C|))$. That is, the maximum rank $r$ is bounded by $\mathcal{O}(\log_2(|U \cup C|))$. According to the first property of Lemma 1, we have $|U \cup C| \leq 2\mathcal{O}(\text{Rad}(G[U \cup C], s)^2) - 1$, which leads us to the inequality $r \leq \mathcal{O}(\log_2(2\mathcal{O}(\text{Rad}(G[U \cup C], s)^2) - 1)) \leq 1 + 2\mathcal{O}(\log_2(\text{Rad}(G[U \cup C], s))$. □

#### 4.2.3 Broadcast scheduling

After computing the ranks of all nodes in $G[U \cup C]$, transmissions are scheduled in two phases. In phase 1, CFBS schedules the transmission of nodes in $G[U \cup C]$, i.e., the CDS. In phase 2, it schedules transmissions from dominators in $U \cup C$ to all other nodes in $G$. The rationale for having two phases is that it is not necessary to send a message to non-CDS nodes early as they are not responsible for relaying the message further. On the other hand, by reducing the number of receiving nodes in phase 1, a transmitter will avoid a number of potential conflicts when sending a message to CDS nodes, thus reducing the broadcast latency.

In phase 1, transmissions are scheduled from the top to the bottom layer of $T_{\text{spt}}(G[U \cup C])$. Let $S_{ij}$ be the set of nodes with rank $j$ that are parent of nodes in layer $i$, and $V_{ij}$ be the corresponding set of children in layer $i$. A *pipe* with rank $j$, denoted as $P_{ij}$, is defined as the transmissions from nodes in $S_{ij}$ to $V_{ij}$. Let $t_{ij}$ be the starting transmission time of $P_{ij}$.

Initially, only node $s$ in layer 0 transmits a message at time slot 0. Then, for each layer $i$ of $T_{\text{spt}}(G[U \cup C])$, scheduling is carried out according to nodes' rank, whereby the pipe with the highest rank is scheduled first. For instance, for layer 2 of Figure 3, CFBS first schedules pipe $P_{21}$.

CFBS follows a greedy strategy to compute the minimum $t_{ij}$ for each pipe $P_{ij}$. The starting transmission time $t_{ij}$ must satisfy the following constraints:

(1) $t_{ij}$ is larger than the reception time of nodes in $S_{ij}$, meaning a parent node in $S_{ij}$ must have received the message collision-free before it is allowed to transmit;
(2) to avoid collisions within the same layer, $t_{ij}$ must be larger than the reception time of nodes in $V_{i(j+1)}$ of pipe $P_{i(j+1)}$ if it exists, that is, each pipe $P_{ij}$ starts after pipe $P_{i(j+1)}$ ends;
(3) to avoid collisions between different layers, we must have $|t_{ij} - (\text{Depth}(G[U \cup C], i) - 1)| \bmod 3T = 0$, where the time slot of $(\text{Depth}(G[U \cup C], i) - 1)$ is the minimum or optimal receiving time of nodes in layer $i$ of $T_{\text{spt}}(G[U \cup C])$; this constraint thus ensures that the interval between transmissions is $3T$, which guarantees that there are no inter-layer, interfering, and transmitting nodes.

It is worth pointing out that this greedy strategy helps nodes in lower layers to transmit or receive earlier than than the nodes in the upper layers. This is because each pipe's starting transmission time is only determined by the

reception time of parent nodes and other nodes that lie in the same layer, meaning, a parent node does not need to wait for all nodes in the upper layers to finish their transmission.

Next, CFBS schedules transmissions within pipe $P_{ij}$. Denote by $W_0^{'}$ the set of nodes in $V_{ij}$ with rank $j$, and $W_0$ is the set containing their respective parent, i.e., $W_0 \subseteq S_{ij}$. For each parent node $v$ in $W_0$, its transmission time is set to $t_{ij}$. Then, CFBS applies ILBS to generate a broadcast schedule $(\langle W(1), W(2), \cdots, W(l)\rangle)$ for nodes in $S_{ij}$ and $V_{ij} \setminus W_0^{'}$. For each $1 \leq k \leq l$, if $W_0$ or $W_0^{'}$ is non-empty, all nodes in $W(k)$ transmit at time slot $t_{ij} + 3kT$; otherwise, they transmit at time slot $t_{ij} + 3(k-1)T$. Moreover, given that we have $l \leq 17$, it follows that each pipe will take at most $51T$ time slots to finish transmission.

In phase 2, only a subset of dominators in $U$ send the message to nodes in $V \setminus (U \cup C)$. First, CFBS collects into a new subset $D_i$ all the dominators that have a neighbor with active time slot $T_i$ in set $V \setminus (U \cup C)$, where $0 \leq T_i \leq T-1$. Then, it computes a partition of $D_i$ into subsets $D_i(k)$ for $1 \leq k \leq c$ via D2-coloring with smallest-degree-last ordering based on the rule that if two dominators share the same neighbor(s) with active time slot $T_i$ in $V \setminus (U \cup C)$, they must not share the same color or be in the same subset. According to [10], we have $c \leq 12$. Let $T_{p1}$ be the maximum transmission time of Phase 1, and thus in Phase 2, the transmission time of nodes in $D(i)(k)$ is set to $\lfloor T_{p1}/T \rfloor T + kT + T_i$, where $1 \leq k \leq 12$ and $0 \leq T_i \leq T-1$. Denote by $T_{p2}$ the maximum transmission time of phase 2. Hence, we get $T_{p2} \leq T_{p1} + 12T$.

Referring to Figure 3, after determining the ranks in $T_{\text{spt}}(G[U \cup C])$, the next step is to determine the transmission time of nodes in $G[U \cup C]$. We start from pipe $P_{12}$, which consists of $S_{12} = \{s\}$ and $V_{12} = \{v_1, v_2, v_3\}$. Hence, the nodes in $V_{12}$ will receive the message from node $s$ at time slot 0. Then, it considers nodes in layer 2. Among the parents in layer 2, i.e., $v_1$, $v_2$, and $v_3$, nodes $v_2$ and $v_3$ have the maximum rank 1. Hence, CFBS first schedules pipe $P_{21}$, which comprises $S_{21} = \{v_2, v_3\}$ and $V_{21} = \{v_5, v_6, v_7, v_8, v_9\}$.

Both nodes in $S_{21}$ receive the message at time slot 0, and pipe $P_{21}$ is the first one to be considered for layer 2, and thus the starting transmission time $t_{21}$ must be larger than 0. Moreover, it must satisfy $|t_{21} - (\text{Depth}(G[U \cup C], 2) - 1)| \bmod 3T = 0$. Recall that $T = 4$ and $\text{Depth}(G[U \cup C], 2) = 3$; see Table 2. The minimum $t_{21}$ is set to 2, i.e., $t_{21} = \min\{t | t > 0 \text{ and } |t-2| \bmod 12 = 0\} = 2$. Set $V_{21}$ does not contain nodes with rank 1, i.e., $W_0 = \emptyset$, and thus the next step is to apply ILBS to schedule $P_{21}$. As illustrated in Section 4.1, since $W(1) = \{v_3\}$, $W(2) = \{v_2\}$, and $W(3) = \{v_6\}$ in pipe $P_{21}$, the transmission time of $v_3$, $v_2$, and $v_6$ is set to 2, 14, and 26, respectively.

Then, pipe $P_{20}$ is scheduled, whereby $S_{20} = \{v_1\}$ and $V_{20} = \{v_4\}$. Its starting transmission time $t_{20}$ must be larger than node $v_4$'s reception time, i.e., 0, and larger than $V_{21}$'s maximum reception time, i.e., 26. Hence, we have $t_{20} = \min\{t | t > 0,\ t > 26 \text{ and } |t-2| \bmod 12 = 0\} = 38$. The other layers are scheduled using a similar method, and the latency for $T_{\text{spt}}(G[U \cup C])$ is 39. Moreover, from Figure 3, node $v_{16}$ from layer 4 received the broadcast message from node $v_{12}$ at time slot 4, which is smaller than the reception time of node $v_1$ from layer 1, i.e., 38. This demonstrates the advantage afforded by CFBS in allowing a node in a lower layer to receive earlier than a node in an upper layer.

### 4.3 Analysis

The next set of theorems assert the correctness of CFBS and establish its upper bound in terms of the broadcast latency and number of transmissions.

**Theorem 1.** *CFBS provides a correct and collision-free broadcast schedule.*

*Proof.* Recall that CFBS performs transmissions in two phases. Thus, we only need to prove that all nodes in each phase are able to receive the broadcast message collision free. In phase 1, the broadcast is conducted pipe by pipe, and thus we need to prove that the transmissions in each pipe are collision free, and different pipes do not interfere with one another.

The theorem is true in phase 1 because CFBS schedules transmissions within each pipe using ILBS, which produces a collision-free schedule. Next, we show that the transmissions between different pipes are also collision free. We prove this by considering two cases. In the first case, we consider pipes belonging to the same layer, say $i$. Recall that for each layer $i$, pipe $P_{ij}$ starts after pipe $P_{i(j+1)}$ finishes. Therefore, the pipes from the same layer will not interfere with each other.

In the second case, pipes from different layers are considered. Assume that pipes $P_{i_1 j_1}$ and $P_{i_2 j_2}$ are from different layers, i.e., $i_1 \neq i_2$. According to Equation 1, the cost of two adjacent nodes does not exceed $T$, and hence, the cost between a node and its two-hop neighbors is no more than $2T$. For any node in $G[U \cup C]$, its reception can be affected by other transmitting nodes among its two hops range. Therefore, for nodes in layer $i$ of $T_{\text{spt}}(G[U \cup C])$, these interfering nodes can be located in layer $i^{'}$ with a depth of $|\text{Depth}(G[U \cup C], i) - \text{Depth}(G[U \cup C], i^{'})| \leq 2T$. If $|\text{Depth}(G[U \cup C], i_1) - \text{Depth}(G[U \cup C], i_2)| > 2T$, nodes in $i_1$ and $i_2$ are not within each other's two-hop range and hence do not interfere. Next, if $|\text{Depth}(G[U \cup C], i_1) - \text{Depth}(G[U \cup C], i_2)| \leq 2T$, it is also collision free because the reception time of nodes in the same layer is separated by an interval of $3T$ and starts according to their depth

in $T_{\text{spt}}(G[U \cup C])$, that is, the reception time of nodes in layer $i_1$ and $i_2$ will not overlap with each other. Hence, in the second case, the pipes' transmissions are also collision free. Hence, CFBS yields a correct and collision-free schedule in phase 1.

In phase 2, CFBS uses smallest-degree-last ordering D2-coloring method to divide dominators into different subsets; hence, as mentioned in [10], it is also collision free. Thus, the theorem is proven. □

**Lemma 3.** *For any pipe $P_{ij}$ of $T_{spt}(G[U \cup C])$, its starting transmission time $t_{ij}$ does not exceed Depth$(G[U \cup C], i) + 54(r - j)T$.*

*Proof.* We prove this lemma by induction. For layer 0 of $T_{\text{spt}}(G[U \cup C])$, it holds true because the transmission time of source node $s$ is zero. Assume this lemma is correct for all layers before layer $i$. We now prove that it also holds true for layer $i$. Recall that the starting transmission time of $t_{ij}$ is determined by two constraints: (1) maximum reception time of $S_{ij}$ and (2) maximum reception time of nodes in $V_{i(j+1)}$. Next, we analyze the correctness of this lemma based on these two constraints.

First, we compute the maximum reception time of nodes in $S_{ij}$. According to the definition of pipes, the nodes in $S_{ij}$ are the parent of nodes in layer $i$, and hence they lie higher than layer $i$. Assume that node $v \in S_{ij}$ lies in layer $i_1$, where $i_1 < i$. Note that the rank of node $v$'s parent, denoted by $j_1$, is no less than $v$'s rank $j$ by the first property of Lemma 2, i.e., $j_1 \geq j$. Lemma 3 is correct for layer $i_1$, and therefore, the starting transmission time $t_{i_1 j_1}$ of pipe $P_{i_1 j_1}$ is no more than Depth$(G[U \cup C], i_1) + 54T(r - j_1)$, i.e., $t_{i_1 j_1} \leq$ Depth$(G[U \cup C], i_1) + 54T(r - j_1)$; recall that $r$ is the maximum rank, i.e., $rank(s) = r$. Each pipe takes at most $51T$ to finish its transmission, and hence, when $j < j_1$, node $v$ will receive the message after pipe $P_{i_1 j_1}$ finishes at time Depth$(G[U \cup C], i_1) + 54T(r - j_1) + 51T \leq$ Depth$(G[U \cup C], i_1) + 54T(r - j)$. On the other hand, when $j = j_1$, node $v$ will receive the message from its parent at the starting transmission time $t_{i_1 j_1} =$ Depth$(G[U \cup C], i_1) + 54T(r - j)$. Hence, for node $v$, its maximum reception time is no more than Depth$(G[U \cup C], i_1) + 54T(r - j)$. Furthermore, since Depth$(G[U \cup C], i) >$ Depth$(G[U \cup C], i_1)$, node $v$'s maximum reception time is less than Depth$(G[U \cup C], i) + 54T(r - j)$.

Second, we analyze the maximum reception time of nodes in $V_{i(j+1)}$. Assume that the maximum rank of nodes in layer $i$ is $r_i$, i.e., $r_i \geq j$. For layer $i$, the transmission starts from the pipe with greatest rank, and hence, for pipe $P_{ir_i}$, its starting transmission time $t_{ir_i}$ is only determined by the maximum reception time of nodes in $S_{ir_i}$ because nodes with rank of $r_i + 1$ for layer $i$ do not exist. Recall that the maximum reception time of nodes in $S_{ir_i}$ is less than Depth$(G[U \cup C], i) + 54T(r - r_i)$, and thus in the worst case, for pipe $P_{ir_i}$, $t_{ir_i}$ is set to Depth$(G[U \cup C], i) + 54T(r - r_i)$. Since each pipe takes up at most $51T$ time slots, and the reception time of nodes in layer $i$ is separated by $3T$, we have $t_{i(j+1)} - t_{ir_i} \leq (r_i - (j + 1))54T$. Therefore, for nodes in $V_{i(j+1)}$, their maximum reception time is no more than Depth$(G[U \cup C], i) + 54T(r - (j + 1)) + 51T$, i.e., Depth$(G[U \cup C], i) + 54T(r - r_i) + 54T(r_i - (j + 1)) + 51T$.

By considering both reception time of nodes in $S_{ij}$ and $V_{i(j+1)}$, this means in the worst case, $t_{ij}$ is equal to Depth$(G[U \cup C], i) + 54T(r - j)$, which proves the required bound of $t_{ij} \leq$ Depth$(G[U \cup C], i) + 54T(r - j)$. Thus, this lemma is also true for layer $i$. Note that, $54T$ corresponds to $51T$ which is the number of time slots for each pipe to finish its transmission and $3T$ which is the interval used to separate the stating transmission time between adjacent pipes. □

**Corollary 1.** *Algorithm CFBS produces a broadcast schedule with latency $(T + 1)H + T\mathcal{O}(\log_2 H)$, where $H$ is $Rad(G[U \cup C], s)$.*

*Proof.* By Lemma 3, it is clear that the latency in phase 1 is at most Depth$(G[U \cup C], L) + 54rT$, where $L$ is the maximum layer number of $T_{\text{spt}}(G[U \cup C])$. As $H$ is also equal to Depth$(G[U \cup C], L)$ and is no more than $(T + 1)H - 2T$ by Lemma 1, the latency in phase 1 is no more than $(T + 1)H - 2T + 54rT$. According to Lemma 2, given that $r \leq 1 + 2\mathcal{O}(\log_2 H)$, the latency in phase 1 is therefore bounded by $(T + 1)H + 108T\mathcal{O}(\log_2 H) + 52T$. That is, in phase 1, the broadcast latency is bounded by $(T + 1)H + T\mathcal{O}(\log_2 H)$, whereby the omitted constant before $T\mathcal{O}(\log_2 H)$ is 108.

The second phase of CFBS takes at most $12T$ time slots, and hence, the broadcast latency of CFBS is bounded by $(T + 1)H + T\mathcal{O}(\log_2 H) + 12T = (T + 1)H + T\mathcal{O}(\log_2 H)$. □

**Theorem 2.** *CFBS is a $4(T + 2)$-approximate solution in terms of number of transmissions.*

*Proof.* Recall that only the nodes in CDS transmit and receive the message in phase 1. By Lemma 1, the size of CDS is bounded by $2|U| - 1$, and thus, the total number of transmissions in phase 1 is bounded by $2|U| - 1$. For phase 2, only dominators transmit the message, and hence, the number of transmitters does not exceed $|U|$. Furthermore, a dominator only needs to transmit once to its neighbors with the same active time slot in phase 2, and the neighbors of a dominator have at most $T$ different active time slots. Hence, the total number of transmissions in phase 2 does not exceed $T|U|$. Therefore, the total number of transmissions performed by CFBS does not exceed $(T + 2)|U| - 1$, i.e., $2|U| - 1 + T|U|$. Recall that the size

of $U$ does not exceed $4opt + 1$ [28], where *opt* denotes the minimum number of transmissions. CFBS is thus a $(T + 2)(4opt + 1) - 1$ solution. □

### 4.4 Remarks on always-on wireless networks

CFBS is also applicable for always-on wireless networks, where $T$ is set to one. Specifically, it starts by constructing a breadth-first search tree (BFS) rooted at the source node $s$. Here, the BFS tree is a special case of $T_{\text{spt}}$, where the cost of each edge in a given network is fixed to one. Then, CFBS builds the dominator set $U$ and connector set $C$ based on the BFS tree in the same way as illustrated in Section 4.2.1, where dominators in $U$ together with connectors in $C$ form a CDS. The next step is to build a new BFS tree rooted at the source based on graph $G[U \cup C]$, then followed by a ranking of the nodes in this new BFS tree layer by layer in a bottom-up manner via the same method in Section 4.2.2. Note, for a given always-on wireless network $G$, its radius with respect to the source node $s$, i.e., $\text{Rad}(G, s)$, is equal to $R$. sThis is because the cost of each edge in $G$ is one when $T = 1$. Also note that Lemmas 1 and 2 still hold true for always-on wireless networks. In particular, as stated in Lemma 1, $\text{Rad}(G[U \cup C], s) \leq 2R - 2$. As shown by Lemma 2, each node $v$ in $G[U \cup C]$ has a rank no more than its parent and $\text{rank}(v) \leq 1 + 2\mathcal{O}(\log_2(2R - 2))$.

In the third step, the broadcast scheduling process for always-on wireless networks also consists of two phases: (1) broadcast data to all nodes in the CDS and (2) broadcast data from dominators to remaining nodes. In the first phase, for each pipe $P_{ij}$, its staring transmission time $t_{ij}$ will be first calculated according to the same greedy method described in Section 4.2.3. Then, the parent whose corresponding child has a rank of $j$ in pipe $P_{ij}$ is scheduled to transmit at $t_{ij}$. For the other nodes in $P_{ij}$, CFBS applies the ILBS algorithm in Section 4.1 to generate a broadcast schedule. Note that during calculation, sthe scheduling period $T$ is always set to one. In the second phase, CFBS partitions the dominators into different subsets using D2-coloring with smallest-degree-last ordering, where the dominators in the same subset have the same color. Then, these dominators transmit based on their color.

Similar to Corollary 1, CFBS produces a $2R+\mathcal{O}(\log_2 R)$-approximate solution in terms of the broadcast latency. Note that for always-on wireless networks, the optimal broadcast latency is equal to $R$, that is, $H = R$. According to Theorem 2, we can see that CFBS is a 12-approximation solution with respect to the number of transmissions. Compared with the best multiplicative approximation algorithm to date for always-on networks, i.e., [7] that gives a broadcast latency bound of $12R$, our addictive approximation algorithm has a lower latency bound of $2R + \mathcal{O}(\log_2 R)$.

Furthermore, in CFBS, the omitted constant in $\mathcal{O}(\log_2 R)$ is less than 108. Compared with the addictive approximation algorithm in [16], which has a latency bound of $R + \mathcal{O}(\log_2 R)$, but with an omitted constant in $\mathcal{O}(\log_2 R)$ that exceeds 150, our broadcast bound will be smaller when $R$ becomes larger.

## 5 Evaluation

In this section, we outline the research methodology used to evaluate the performance of CFBS. We compare CFBS against OTAB [10], which is known to have the lowest constant approximation ratio to date. In our experiments, we measure each algorithm against two metrics:

- *Broadcast latency*: this is defined as the total time required by all nodes to receive a broadcast message;
- *Transmission ratio*: this is the ratio between the number of transmissions and the number of nodes.

That is, the transmission ratio represents the average number of messages retransmitted by each node in the network. It is worth pointing out that the main goal of our simulation is to compare the theoretical and experimental broadcast latency and transmission ratio performance of our algorithm. In particular, the latency is mainly determined by the nodes' interwake-up times, which are a few orders of magnitude higher than the length of a slot. Moreover, in Section 3.1, it is assumed that a message can be successfully delivered from a sender to a receiver within a time slot. In reality, as shown in [24], the maximum size of a typical TinyOS packet is 47 bytes, a time slot is usually set to 20 ms, and thus, a MicaZ node can attempt at least 13 transmissions in one time slot. In other words, although low-power wireless links are generally unreliable, we can still ensure that a message can be successfully transmitted within a time slot through multiple transmissions [24]. Therefore, in our simulations, we only consider the packet loss caused by collisions, and assume that unreliable links can be solved within a time slot through multiple transmissions. It is for this reason we do not employ a packet level simulator and any specific MAC protocols.

We place wireless nodes in a square area of $l \times l\text{m}^2$ uniformly and randomly while changing the square length $l$, number of nodes, transmission radius, and duty cycle. We study the performance of CFBS under different network configurations including the square length, number of nodes, transmission radius, and duty cycle, where the duty cycle is defined as the ratio of the duration of the active time slots to the scheduling period. The square length varies from 150 to 400 m. The number of nodes ranges from 200 to 1,000 with an interval of 200. The transmission radius ranges from 20 to 60 m. The duty cycle varies from 0.1 to 0.02. For each experiment, we change one network configuration while the other three

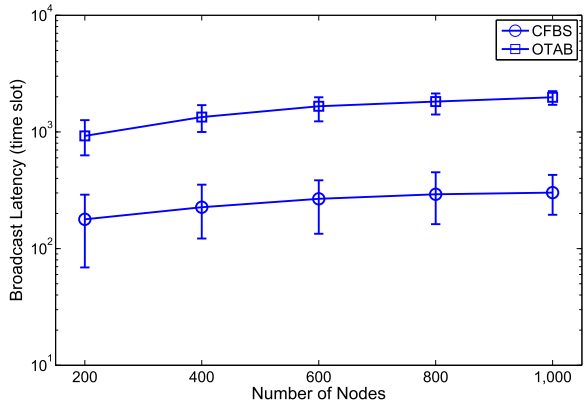


**Figure 4** Broadcast latency under different network sizes.

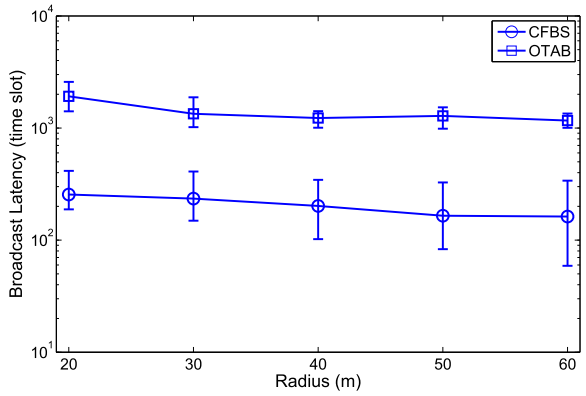


**Figure 6** Broadcast latency under different number of nodes.

remain unchanged. Each experiment is conducted on 20 randomly generated topologies. Moreover, for each topology, we carry out the experiment for 10 runs, and in each run, an arbitrary node is selected as the source node. Hence, each result is the average of 200 simulation runs.

### 5.1 Impact of network size

Figure 4 presents the average broadcast latencies of CFBS and OTAB when we vary the network size, which is denoted by the square length $l$. In this experiment, the number of nodes, transmission radius, and duty cycle are set to 400, 30 m, and 0.05, respectively. In Figure 4, we observe that the broadcast latency of both algorithms grows proportionally to the square of length $l$. The reason is as follows. The broadcast latency of CFBS and OTAB is mainly influenced by the number of layers in the SPT. For a fixed number of nodes and transmission radius, the network becomes sparse when we increase the network size. As a result, the network has fewer links and connectivity, and thus SPT has more layers. Furthermore, CFBS performs much better than OTAB, i.e., when the square length is set to 350 m, the broadcast latency of CFBS is only $\frac{1}{8}$ that of OTAB. This is because instead of scheduling transmissions layer by layer as in OTAB, CFBS is able to schedule nodes' transmission in more than one layer, which helps reduce the broadcast latency.

Figure 5 plots the transmission ratio versus the network size. We find that the transmission ratio for CFBS and OTAB grows with increasing network size. This is because the average degree decreases when we increase the network size; thereby, a node will inform fewer nodes after each transmission. This means a node requires more transmissions to cover its neighbors. Moreover, CFBS performs better than OTAB in terms of the transmission ratio. This is because CFBS selects transmitting nodes from a small CDS.

### 5.2 Impact of node numbers

In Figure 6, we present the average broadcast latencies of CFBS and OTAB when we change the number of nodes. In this experiment, the square length, transmission radius, and duty cycle are fixed at 200 m, 30 m, and 0.05, respectively. As shown in Figure 6, we find that the broadcast latency of both algorithms grows as the

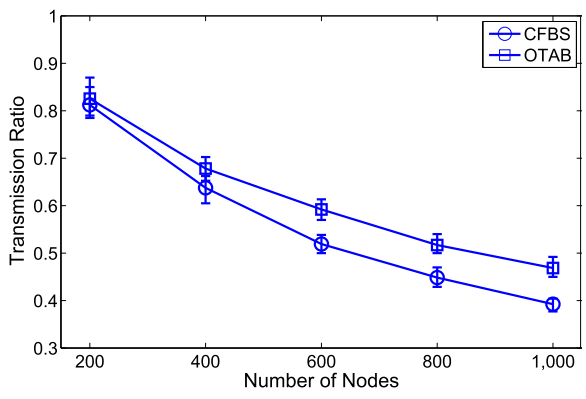


**Figure 5** Transmission ratio under different network sizes.

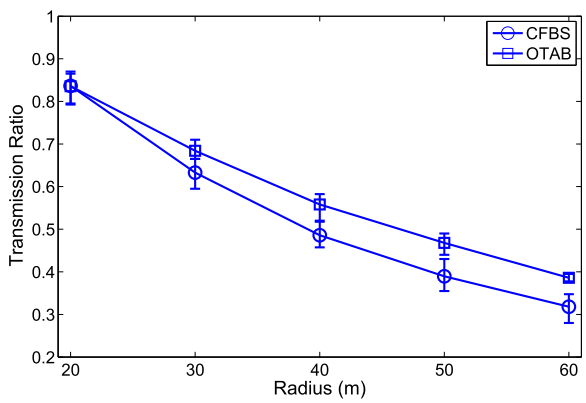


**Figure 7** Transmission ratio under different number of nodes.

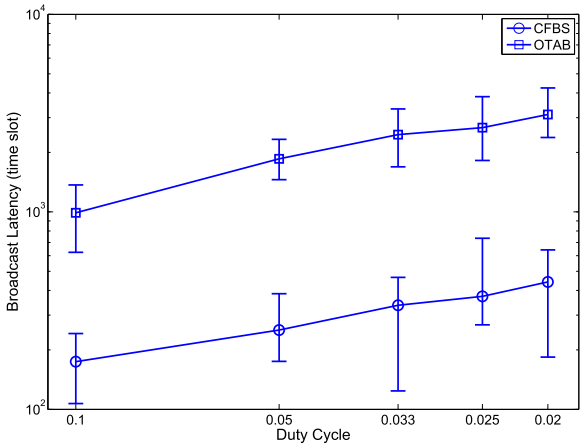


**Figure 8** Broadcast latency under different transmission radii.

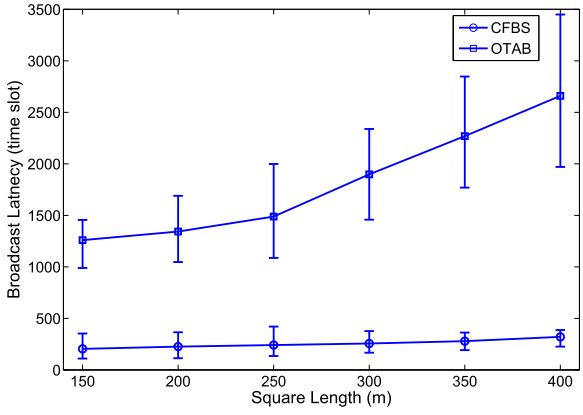


**Figure 10** Broadcast latency under different duty cycles.

number of nodes increases. This is because the network becomes denser when the number of nodes increases in a fixed network area. As a result, there are more links and richer connectivity, and thus, the SPT rooted at the source produces fewer layers. That is, less time will be required to inform all nodes. Furthermore, CFBS performs much better than OTAB, i.e., when the number of nodes is set to 1,000, the broadcast latency of CFBS is only $\frac{3}{20}$ that of OTAB, for the same reason as listed in Section 5.1.

Figure 7 shows the transmission ratio versus the number of nodes. We see that the transmission ratio for both algorithms decreases with increasing number of nodes. This is because the average degree grows with increasing number of nodes; thereby, a node can inform more neighbors via one transmission. This means a node requires fewer transmissions to cover its neighbors. Moreover, CFBS still performs better than OTAB in terms of transmission ratio.

### 5.3 Impact of transmission radius

In Figure 8, we plot the broadcast latencies of CFBS and OTAB under different transmission radii. In this experiment, we set the square length, number of nodes. and the duty cycle to 200 m, 400, and 0.05, respectively. We see that the broadcast latency of both algorithms decreases with increasing transmission radius. This is because the nodes with a larger transmission radius will have higher connectivity with other nodes, which helps reduce the number of layers in the SPT. Notably, CFBS performs much better than OTAB in terms of the broadcast latency under different transmission radii, i.e., the latency of CFBS is within 17% of the latency achieved by OTAB.

Figure 9 shows that the transmission ratio of CFBS and OTAB decreases as the transmission radius grows. This is due to nodes with larger transmission radius being able to inform more nodes in one transmission, and thus, fewer transmissions will be needed to inform its neighbors. Furthermore, CFBS has a better performance in terms of the transmission ratio as compared to OTAB.

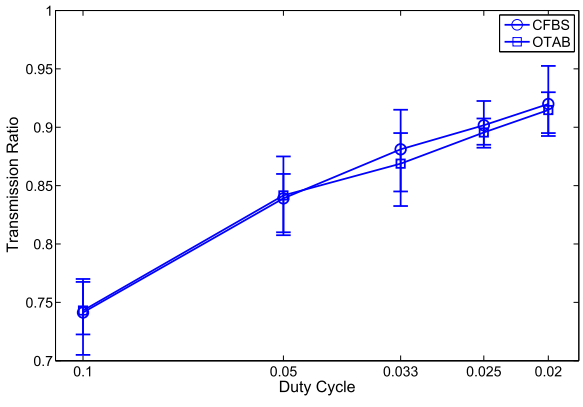


**Figure 9** Transmission ratio under different transmission radii.

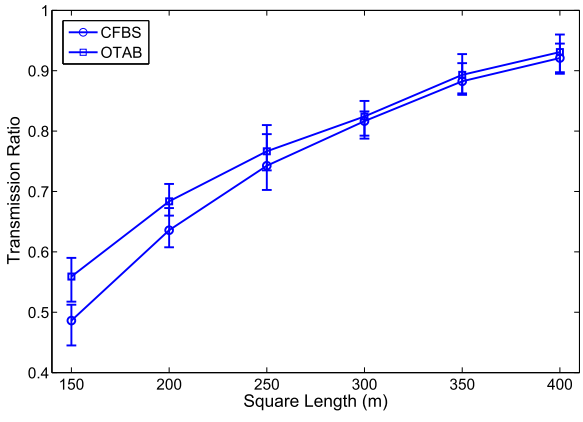


**Figure 11** Transmission ratio under different duty cycles.

### 5.4 Impact of duty cycle

Figure 10 is a plot of the broadcast latency versus duty cycle. We fix the square length to 200 m, the number of nodes to 400, and the transmission radius to 20 m. From Figure 10, we find that the broadcast latency of CFBS and OTAB increases with declining duty cycle. The reason is due to the scheduling period $T$ containing more time slots as the duty cycle decreases; a node will thus need to wait longer before forwarding a message to its neighbors. In addition, CFBS performs much better than OTAB in terms of the broadcast latency, i.e., CFBS's broadcast latency is around 15% that of OTAB when the duty cycle is set as 0.02.

Figure 11 shows that the transmission ratio for both algorithms increases with decreasing duty cycle. When the duty cycle decreases, the scheduling period $T$ will contain more time slots, and a node needs to transmit more times to inform its neighbors because they have a higher probability of choosing different active time slots from a larger $T$. Moreover, CFBS outputs a smaller transmission ratio than OTAB.

## 6 Conclusion

This paper has formally outlined the MLBSDC problem and presented a novel algorithm called CFBS with a broadcast latency of at most $(T+1)H + T\mathcal{O}(\log_2 H)$. In addition, we proved that CFBS provides a correct and collision-free broadcast scheduling and achieves a low latency and overhead in terms of the number of transmissions. Our simulation results indicate that CFBS has a better performance, in terms of the broadcast latency and transmission ratio, than OTAB under different network configurations.

As a future work, we are currently looking into implementing CFBS in distributed manner. The use of our method under the physical interference model is another possible future work. Under this model, we need to consider both collisions and total interference from nearby transmitters.

#### Competing interests

The authors declare that they have no competing interests.

#### Acknowledgements

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

Received: 21 January 2013 Accepted: 11 October 2013
Published: 23 October 2013

#### References

1. A Baggio, in *ACM REALWSN*. Wireless sensor networks in precision agriculture (Stokholm, Sweden, 20–21 Jun 2005)
2. W Hu, VN Tran, N Bulusu, C Chou, S Jha, A Taylor, in *ACM/IEEE IPSN*. The design and evaluation of a hybrid sensor network for cane toad monitoring (Los Angeles, CA, USA, 15 Apr 2005)
3. K Korincz, M Welsh, O Marcillo, J Johnson, M Ruiz, J Werner-Allen, Deploying a wireless sensor network on an active volcano. IEEE Internet Comput. **10**(2), 18–25 (2006)
4. C Perkins, E Royer, in *IEEE WMCSA*. Ad-hoc on-demand distance vector routing (New Orleans, LA, USA, 25–26 Feb 1999)
5. W Nan, S Xue-li, in *IEEE IITAW*. Research on nodes location technology in wireless sensor network underground (Nanchang, China, 21–22 Nov 2009)
6. S Brown, C Sreenan, Updating software in wireless sensor networks: a survey (2006) [Technical report UCC-CS-2006-13N-07, University College Cork, Ireland]
7. R Gandhi, YA Kim, S Lee, J Ryu, PJ Wan, in *IEEE INFOCOM*. Approximation algorithms for data broadcast in wireless networks (Rio de Janeiro, Brazil, 20–25 Apr 2009)
8. R Gandhi, A Mishra, S Parthasarathy, Minimizing broadcast latency and redundancy in ad hoc networks. IEEE/ACM Trans. Netw. **16**(4), 840–851 (2008)
9. W Ye, J Heidemann, D Estrin, in *IEEE INFOCOM*. An energy-efficient MAC protocol for wireless sensor networks (New York, NY, USA, 23–27 Jun 2002)
10. X Jiao, W Lou, J Ma, J Cao, X Wang, X Zhou, Minimum latency broadcast scheduling in duty-cycled multi-hop wireless networks. IEEE Trans. Parallel Distributed Syst. **23**, 110–117 (2012)
11. H Lim, C Kim, Flooding in wireless ad hoc networks. Comput. Commun. **24**(3), 353–363 (2001)
12. S Ni, YC Tseng, J Sheu, in *ACM MOBICOM*. The broadcast storm problem in a mobile ad hoc network (Seattle, WA, USA, 15–20 Aug 1999)
13. F Ferrari, M Zimmerling, L Thiele, O Saukh, in *IPSN*. Efficient network flooding and time synchronization with Glossy (Chicago, IL, USA, 12–14 Apr 2011)
14. Y Sasson, D Cavin, A Schiper, in *IEEE WCNC*. Probabilistic broadcast for flooding in wireless mobile ad hoc networks (New Orleans, LA, USA, 16–20 Mar 2003)
15. F Stann, J Heidemann, R Shroff, MZ Murtaza, in *ACM SenSys*. RBP: robust broadcast propagation in wireless networks (Boulder, CO, USA, 31 Oct–3 Nov 2006)
16. SH Huang, PJ Wan, X Jia, H Du, W Shang, in *IEEE INFOCOM*. Minimum-latency broadcast scheduling in wireless ad hoc networks (Anchorage, AK, USA, 6–12 May 2007)
17. S Lai, B Ravindran, in *IEEE DCOSS*. On multihop broadcast over adaptively duty-cycled wireless sensor networks (Santa Barbara, CA, USA, 21–23 Jun 2010)
18. 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, 8–10 Dec 2008)
19. J Hong, J Cao, W Li, S Lu, D Chen, Minimum-transmission broadcast in uncoordinated duty-cycled wireless ad hoc networks. IEEE Trans. Vehicular Technol. **59**, 307–318 (2010)
20. 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, 14–18 Jun 2009)
21. D Bozdag, U Catalyurek, AH Gebremedhin, F Manne, EG Boman, in *IEEE HPCC*. A parallel distance-2 graph coloring algorithm for distributed memory computers (Sorrento, Italy, 21–23 Sept 2005)
22. Y Duan, S Ji, Z Cai, in *IEEE IPCCC*. Generalized broadcast scheduling in duty-cycle multi-hop wireless networks (Orlando, FL, USA, 17–19 Nov 2011)
23. X Xu, J Cao, PJ Wan, in *WASA*. Fast group communication scheduling in duty-cycled multihop wireless sensor networks (Fusionopolis, Singapore, 26–27 Dec 2012)
24. Y Gu, T He, in *IEEE ICDCS*. Bounding communication delay in energy harvesting sensor networks (Genoa, Italy, 21–25 Jun 2010)
25. M Maróti, B Kusy, G Simon, A Lédeczi, in *SenSys*. The flooding time synchronization protocol (Baltimore, Maryland, 3–5 Nov 2004)
26. C Hua, TSP Yum, Asynchronous random sleeping for sensor networks. ACM Trans. Sensor Netw (2007). doi:10.1145/1267060.1267063
27. PJ Wan, SCH Huang, L Wang, Z Wan, X Jia, in *ACM MobiHoc*. Minimum-latency aggregation scheduling in multihop wireless networks (New Orleans, LA, USA, 18–21 May 2009)
28. PJ Wan, K Alzoubi, O Frieder, in *IEEE INFOCOM*. Distributed construction of connected dominating set in wireless ad hoc networks (New York, NY, USA, 23–27 Jun 2002)

29. DW Matula, LL Beck, Smallest-last ordering and clustering and graph coloring algorithms. J. ACM. **30**(3), 417–427 (1983)
30. S Guha, S Khuller, Approximation algorithms for connected dominating sets. Algorithmica. **20**(4), 374–387 (1998)
31. J Wu, M Gao, I Stojmenovic, in *ICPP*. On calculating power-aware connected dominating sets for efficient routing in ad hoc wireless networks (Valencia, Spain, 3–7 Sept 2001)

doi:10.1186/1687-1499-2013-248
**Cite this article as:** Zhao and Chin: **Approximation algorithm for data broadcasting in duty cycled multi-hop wireless networks.** *EURASIP Journal on Wireless Communications and Networking* 2013 **2013**:248.