Hindawi
Wireless Communications and Mobile Computing
Volume 2019, Article ID 5064109, 14 pages
https://doi.org/10.1155/2019/5064109

Research Article

Energy-Efficient Broadcast Scheduling Algorithm in Duty-Cycled Multihop Wireless Networks

Quan Chen,[1] Tao Wang,[1] Lianglun Cheng,[1] Yongchao Tao ,[2,3] and Hong Gao[3]

[1] *School of Computers, Guangdong University of Technology, Guangzhou, China*
[2] *Shenzhen Academy of Aerospace Technology, China*
[3] *School of Computer Science and Technology, Harbin Institute of Technology, China*

Correspondence should be addressed to Yongchao Tao; taoyongchao@chinasaat.com

Received 7 November 2018; Accepted 14 January 2019; Published 6 February 2019

Guest Editor: Wei Cheng

Broadcasting is a fundamental function for disseminating messages in multihop wireless networks. Minimum-Transmission Broadcasting (MTB) problem aims to find a broadcast schedule with minimum number of transmissions. Previous works on MTB in duty-cycled networks exploit a rigid assumption that nodes have only active time slot per working cycle. In this paper, we investigated the MTB problem in duty-cycled networks where nodes are allowed arbitrary active time slots per working cycle (MTBDCA problem). Firstly, it is proved to be NP-hard and $o(\ln \Delta)$-inapproximable, where Δ is the maximum degree in the network. Secondly, an auxiliary graph is proposed to integrate nodes' active time slots into the network and a novel covering problem is proposed to exploit nodes' multiple active time slots for scheduling. Then, a $\ln(\Delta + 1)$-approximation algorithm is proposed for MTBDCA and a $(\ln(\Delta + 1) + \Delta)$-approximation algorithm is proposed for all-to-all MTBDCA. Finally, extensive experimental results demonstrate the efficiency of the proposed algorithm.

1. Introduction

Broadcasting is a fundamental function for many services in wireless sensor networks (WSNs) [1, 2], such as data collection, code updating, and topology discovering [3–6]. Many effective broadcasting algorithms have been investigated to improve the network performance, while the total number of transmissions is often taken as a measured metric. Obviously, the smaller the number of transmissions is, the more energy will be saved. Therefore, the Minimum-Transmission Broadcasting problem (*MTB*), which tries to minimize the number of transmissions, has got a lot of attentions from researchers. In traditional wireless sensor networks where each node always keeps awake, many broadcasting algorithms [7–15] have been proposed. Since the MTB problem has been proved to be NP-hard [7] and the aforementioned algorithms [7–15] are approximate ones, they achieve high energy efficiency during broadcasting.

However, it is well known that the duty-cycled scheme is commonly adopted in wireless sensor networks to conserve energy [16, 17]. In such working mode, each node switches between the active state and sleep state cyclically. To save energy, all the functional modules (such as sensing the environment, sending, and receiving the messages) are turned off in sleep state. As this working mode is much different, the MTB problem in duty-cycled wireless sensor networks needs to be reinvestigated and the following two issues should be addressed: (1) A node may need several transmissions to inform all its neighbors since the neighbors may be active at different time slots. (2) How to construct the broadcast tree to exploit nodes' multiple active time slots for scheduling is also a big challenge. In duty-cycled wireless networks, the MTB problem is mainly studied by [18–21] currently. To reduce the number of transmissions, these algorithms try to organize the nodes with the same active time slot together, and they provided a solution for the aforementioned Issue 1. However, these methods exploit a strong assumption that all nodes can be active only once in a working cycle. When each node can be active arbitrary times per working cycle [16, 17, 22], these methods cannot solve the aforementioned Issue 2 efficiently and may result in a large redundancy.

Thus, to avoid the above shortcomings, the MTB problem in duty-cycled WSNs where nodes have arbitrary active time slots per working cycle (*MTBDCA* problem) is studied in this paper. To exploit nodes' whole active time slots, an auxiliary graph is designed and a novel kind of node covering problem, *i.e.*, minimum schedule node covering problem, is proposed. Finally, a $\ln(\Delta + 1)$-approximation algorithm is proposed for MTBDCA. The contributions of this paper are mainly listed as follows.

(1) The MTBDCA problem is firstly formulated and proved to be NP-hard and $o(\ln \Delta)$-inapproximable, where Δ is the maximum degree in the network. An auxiliary graph is designed to integrate nodes' active time slots into the network.

(2) In order to exploit nodes' whole active time slots for scheduling, a novel kind of node covering problem, *i.e.*, minimum schedule node covering problem, is proposed and proved to be NP-hard. An approximate algorithm is proposed to solve it.

(3) An approximation algorithm with ratio of $\ln(\Delta + 1)$ is proposed for MTBDCA, where Δ is the maximum degree in the network. An $(\ln(\Delta + 1) + \Delta)$-approximation algorithm is also proposed for all-to-all MTBDCA problem.

(4) The extensive simulations verify that the proposed algorithm can achieve high performance in terms of energy efficiency.

Compared to the conference version [23], this paper not only investigates the all-to-all MTBDCA problem and provide a $(\ln(\Delta + 1) + \Delta)$-approximation algorithm for it but also provides more details on the theoretical analysis, such as the proof of the NP-hardness of the Minimum Schedule Node Covering problem and the correctness of the broadcast tree construction algorithm, which are important for the completeness of the proposed methods. Additionally, the new experiments are also conducted to show some new findings of the proposed methods.

The rest of this paper is organized as follows. The related works are introduced in Section 2. Section 3 gives the network model and problem definition. The proposed approximate algorithm for MTBDCA is presented in Section 4. The simulation results and conclusion are given in Sections 5 and 6.

2. Related Works

The Minimum-Transmission Broadcasting (MTB) problem has drawn a lot of attentions from researchers. When each node always keeps awake, the MTB problem in WSNs are mainly studied by [7–15]. The NP-hardness of MTB problem was first proved in [7], where a tree-based structure is used to disseminate the messages. After that, the MCDS-based (Minimum Connected Dominate Set) scheme is used to further reduce the redundancy by minimizing the number of senders in the broadcast tree [8–10]. Lou *et al.* [11, 12] proposed a Dominant Pruning scheme and a quasi-local forward-node-set-based scheme to reduce the redundancy. The multipoint relay scheme is introduced for MTB in [13–15], which can determine a small set of forwarding nodes in a localized way. Since CDS is the core of the algorithms for MTB, Wan *et al.* proposed an 8-approximation algorithm in [24]. With a two-phase constructing approach, they reduced the approximation ratio to 6.8 in [25]. Recently, the CDS construction problem in battery-free WSNs and the weakly CDS construction problem are studied in [26–28]. However, these methods are unsuitable for duty-cycled wireless networks.

In duty-cycled wireless networks, the MTB problem are studied in [18–21]. The MTB problem in duty-cycled networks was first proved to be NP-hard in [18]. In [18], a centralized $3(\ln \Delta + 1)$-approximation algorithm is proposed by exploiting a set covering technique. After that, the authors in [19, 20] proposed a level-based scheduling framework and constructed the backbone according to nodes' level information to reduce the number of transmissions. Recently, the authors in [21] found that the number of transmissions can be further improved if a depth-first search is conducted on the forwarding nodes. Based on this, they proposed an efficient broadcasting algorithm outperforms the previous ones. However, these methods all assume that all nodes can active only once in a working cycle, which limits their application. When each node has multiple active time slot in a working cycle, the efficient algorithms have been proposed for multicasting [16, 29], flooding [22], data aggregation [17], and beaconing [30] in the duty-cycled networks, respectively. Thus, it is very necessary and meaningful to study the MTBDCA problem.

3. Problem Definition

Let $G = (V, E)$ denote a duty-cycled WSN, where $V = \{1, 2, \ldots, n\}$ denotes all nodes and $E = \{(u, v) \mid 1 \le u, v \le n \ \& \ u \ne v\}$ denotes the neighborhood relationship among nodes. As discussed before, each node owns two states in such network, *i.e.*, the sleep state and the active state, and switches between these two states cyclically. Let $\mathscr{W}$ denote a working cycle which includes $|\mathscr{W}|$ time slots with same length, *i.e.*, $\mathscr{W} = \{0, 1, 2, \ldots, |\mathscr{W}| - 1\}$. And assume $\mathscr{W}(u)$ denote the *working plan* of node u, which is defined as the set of the active time slots of u, *i.e.*, $\mathscr{W}(u) = \{t_1, t_2, \ldots, t_k\} \subseteq \{0, 1, \ldots, |\mathscr{W}| - 1\}$. If node u wants to receive a packet, it can only receive it when it is active (*i.e.*, the time slot $t \in \mathscr{W}(u)$). When it wants to send a message, it can choose a time slot when the receiver is awake to switch to the active state. The duty cycle is calculated as $|\mathscr{W}(u)|/|\mathscr{W}|$. Table 1 lists the major symbols used in this paper.

As for MTB problem in duty-cycled WSNs, one not only needs to construct a broadcast tree which is rooted at the source node s, but also computes the *Transmitting Schedule* of each nonleaf node to informs its children while the number of transmissions is minimized. Before giving the formal definition of *Transmitting Schedule*, some notations used in this paper should be clarified. Assume $NB(u)$ denote the set of u's one-hop neighbors in G. And let T denote the broadcast tree; $nl(T)$ and $ch(u)$ denote the set of nonleaf nodes and the set of u's children in the broadcast tree, respectively. Then we can have,

Definition 1 (transmitting schedule). Given any node $u \in V$, let $sch(u) = [u, t, ch(u, t)]$ denote node u's one transmitting

Table 1: Symbols and notations.

Notation	Description
$G = (V, E)$	a duty-cycled WSN
n	the number of nodes in the network
$\mathscr{W}$	a working cycle
$\mathscr{W}(u)$	node u's working plan
$sch(u)$	node u's transmitting schedule
$ch(u, t)$	the set of node u's children which can be reached at time t
$ch(u)$	the set of all child nodes of node u
$\mathcal{S}(u)$	all the transmitting schedules of node u
T	the broadcast tree of G
$\mathcal{S} = (T, \mathcal{S}(T))$	the broadcast schedules of G
$\mathcal{G} = (V', E')$	the auxiliary graph of G
$a_{u,i}$	the i-th schedule node of v
$a_{u,i}.p$	the schedule node $a_{u,i}$'s primary node
$a_{u,i}.t$	the schedule node $a_{u,i}$'s transmitting time slot
$R(a_{u,i})$	the set of primary nodes which can be reached by $a_{u,i}$
A_S	the set of schedule nodes for the MSNC problem
A_P	the set of all primary nodes of whose schedule node in A_S
$L(u)$	the level of node u
T_u	the subtree rooted at node u
$NB(u)$	the set of node u's neighbors
ψ	the average number of neighbors in G
Δ	the maximum node degree of G

schedule, in which for any node $v \in ch(u, t)$, we have $(u, v) \in E$ and $t\%|\mathscr{W}| \in \mathscr{W}(v)$.

Given a transmitting schedule $sch(u)$, it means that node u can deliver the message to all the nodes in $ch(u, t)$ through only one transmission at time slot t. To receive the message, each node $v \in ch(u, t)$ is able to wake up at time slot t ($t\%|\mathscr{W}| \in \mathscr{W}(v)$). Note that a node may have several transmitting schedules, then we can find that $ch(u) = \bigcup_{sch(u) \in \mathcal{S}(u)} sch(u).ch(u, t)$, where $\mathcal{S}(u)$ is the set of all transmitting schedules of u.

Thus, the MTB problem in duty-cycled WSNs where nodes have arbitrary active time slots per working cycle (*MTBDCA*) is to compute a broadcast tree T and calculate one or several transmitting schedules for nonleaf nodes in the broadcast tree, *i.e.*, $\mathcal{S}(T)$, with minimum number of transmissions. The broadcast tree T and the transmitting schedules of each nonleaf node $\mathcal{S}(T)$ are called a broadcast schedule. It can be formalized as follows.

Input:

(1) A duty-cycled network $G = (V, E)$ and a source node s.

(2) The working plans for all nodes, *i.e.*, $\{\mathscr{W}(v) \mid \forall v \in V\}$.

Output: The broadcast schedule $\mathcal{S}_{min} = \{T, \mathcal{S}(T)\}$, where $\mathcal{S}(T) = \{\mathcal{S}(u) \mid \forall u \in nl(T)\}$, where

(1) T is rooted at the source node s and spanning all of the nodes in V.

(2) For any $sch(u) \in \mathcal{S}(u)$ and $\mathcal{S}(u) \in \mathcal{S}(T)$, $sch(u)$ satisfies the conditions in Definition 1.

(3) The number of total transmissions, *i.e.*, $\sum_{\mathcal{S}(u) \in \mathcal{S}(T)} |\mathcal{S}(u)|$, is minimized.

Theorem 2. *The MTBDCA problem is NP-hard and there exists no polynomial-time approximation algorithm with ratio of* $(1 - o(1)) \ln \Delta$ *for MTBDCA unless* $NP \subseteq DTIME(n^{O(\log\log n)})$, *where* Δ *is the maximum degree in the network.*

Proof. We prove the NP-hardness of MTBDCA by showing that the Minimum Set Cover (MSC) problem can be mapped to one of its special cases. It has been proved that the MSC problem [29] is NP-hard and there exists no polynomial-time algorithm with ratio of $(1 - o(1)) \ln N$ unless $NP \subseteq DTIME(n^{O(\log\log n)})$, where N is the size of the MSC problem. Consider a special case of MTBDCA, where there is only one source node s along with its neighbors in the network and all the neighbors are only connected to s. Thus, source node s can choose a set of active time slots to cover all its neighbors. In this case, the MTBDCA problem is equivalent to the MSC problem. The theorem is proved. □

4. Approximation Algorithm for MTBDCA

In this section, we will introduce an algorithm with approximation ratio of $\ln(\Delta + 1)$ for MTBDCA, which includes the following steps: (1) The auxiliary graph is designed and

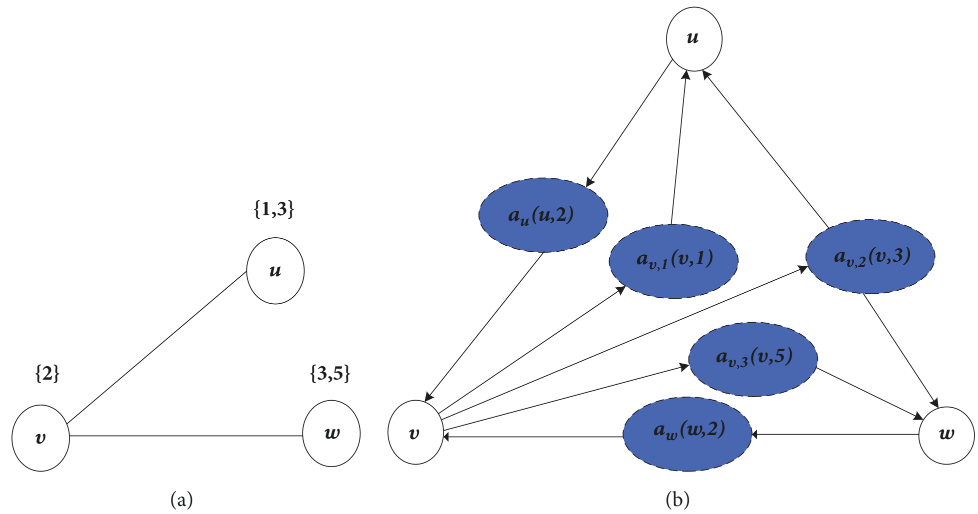


FIGURE 1: An example of the auxiliary graph. Figure 1 is reproduced from Chen et al. (2018) (under the Creative Commons Attribution License/public domain).

constructed to integrate nodes' active time slots into the network. (2) A *Minimum Scheduling Node Covering* problem is designed for multiple active time slot scheduling and an approximate algorithm is proposed for it. (3) A pseudo broadcast tree which contains nodes' schedule information is obtained on the constructed auxiliary graph. (4) A broadcast schedule for MTBDCA problem is computed according to the pseudo broadcast tree.

4.1. Constructing an Auxiliary Graph. In order to integrate nodes' working plan into the network and construct the broadcast tree, an *Auxiliary Graph* is first designed based on the duty-cycled network G. Different from the duty-cycled network, a new kind of node is introduced in the auxiliary graph, *i.e.*, *Schedule Node*. It is used to calculate the transmitting schedules for the nonleaf nodes.

In this paper, let a_u denote a schedule node of u ($u \in V$) (For simplicity, node u is called the primary node of a_u). Each schedule has two properties, *i.e.*, $(a_u.p, a_u.t)$, where $a_u.p$ denotes the schedule node's primary node and $a_u.t$ denotes its transmitting time slot. The auxiliary graph is defined as follows.

Definition 3 (auxiliary graph). Given a duty-cycled network $G = (V, E)$, the auxiliary graph $\mathscr{G} = (V', E')$ is a directed graph which contains nodes' active time slots. The set V' includes two kinds of nodes (primary node and schedule node) and E' is the set of all edges. They are constructed as follows:

(i) Initially, $V' = V$, $E' = \emptyset$.

(ii) For any node $u \in V$ and time $t' \in \bigcup_{v\in NB(u)} \mathscr{W}(v)$, we create a schedule node $a_{u,i}$ $(1 \le i \le |\bigcup_{v\in NB(u)} \mathscr{W}(v)|)$. Its two properties are set as (u, t') (*i.e.*, $a_{u,i}.p = u$ and $a_{u,i}.t = t'$). Let $\Upsilon(u)$ denote the set of u's schedule nodes, and then $V' = V' \bigcup_{u\in V} \Upsilon(u)$.

(iii) For any node $u \in V$ and schedule node $a_{u,i} \in \Upsilon(u)$, create an edge from u to $a_{u,i}$. Let $E'_u = \{(u, a_{u,i}) \mid a_{u,i} \in \Upsilon(u)\}$, then E' can be updated as $E' = \bigcup_{u\in V} E'_u$.

(iv) Let v be a one-hop neighbors of u in G. For each schedule node $a_{u,i} \in \Upsilon(u)$, add an directed edge $(a_{u,i}, v)$ in $\mathscr{G}$ if $a_{u,i}.t \in \mathscr{W}(v)$. Let $R(a_{u,i})$ denote the set of such node v. Actually, it denotes the nodes can be reached by $a_{u,i}.p$ at time slot $a_{u,i}.t$. And we can have $E' = E' \bigcup_{u\in V}\{\bigcup_{a_{u,i}\in\Upsilon(u)} E'_{a_{u,i}}\}$, where $E'_{a_{u,i}} = \{(a_{u,i}, v) \mid v \in R(a_{u,i})\}$.

Note that, the auxiliary graph is directed, where the edges are either started from the primary node u to its schedule nodes in $\Upsilon(u)$, or started from a schedule node $a_{u,i}$ to the primary nodes it can reach at time $a_{u,i}.t$, *i.e.*, $R(a_{u,i})$. For example, there is a simple duty-cycled network in Figure 1(a) and the number in the braces denotes nodes' working plan. According to Definition 3, its auxiliary graph is constructed as in Figure 1(b). As for node v, there are two neighbors and the union of the active time slots of these neighbors are $\{1, 3, 5\}$. Then, three schedule nodes $a_{v,1}$, $a_{v,2}$ and $a_{v,3}$ are created, and their properties are set as $(v, 1)$, $(v, 3)$ and $(v, 5)$. As we can see time slot 1 is only included in the working plan of node u, then there is an edge from node v to $a_{v,1}$ and an edge from $a_{v,1}$ to node u. The pseudo code for constructing the auxiliary graph is shown in Algorithm 1.

The size of the auxiliary graph is analyzed in Theorem 5.

Lemma 4. *Any schedule node in the auxiliary graph $\mathscr{G}$ can connect to at most Δ primary nodes.*

Proof. According to Definition 3, a schedule node $a_{u,i}$ is connected to u's neighboring nodes which have the active time slot $a_{u,i}.t$ in its working plan. Obviously, there are at most Δ such nodes. Since the auxiliary graph is directed, thus, the schedule node $a_{u,i}$ can connect to at most Δ primary nodes. □

Input: The duty-cycled network $G = (V, E)$, and the source node s;
Output: The auxiliary graph $\mathscr{G} = (V', E')$;

```
V' ⟵ V, E' ⟵ ∅;
for ∀u ∈ V do
    S ⟵ ⋃_{v∈NB(u)} 𝒲(v);
    i ⟵ 1;
    for ∀t' ∈ S do
        V' = V' ∪ {a_{u,i}};
        E' = E' ∪ {(u, a_{u,i})};
        a_{u,i}.p ⟵ u, a_{u,i}.t ⟵ t';
        R(a_{u,i}) ⟵ {v | v ∈ NB(u) & a_{u,i}.t ∈ 𝒲(v)};
        for ∀v ∈ R(a_{u,i}) do
            E' = E' ∪ {(a_{u,i}, v)};
        end for
        i ⟵ i + 1;
    end for
end for
return 𝒢;
```

ALGORITHM 1: Algorithm for constructing the auxiliary graph.

Theorem 5. *The number of nodes and edges in the auxiliary graph $\mathscr{G}$ are at most $(1 + |\mathscr{W}|) \times n$ and $(1 + \Delta) \times n \times |\mathscr{W}|$, respectively, where $|\mathscr{W}|$ denotes the length of a working period and $n = |V|$ is the number of nodes of the original graph.*

Proof. Firstly, according to Definition 3, a schedule node is created for each primary node u and each time slot $t \in \bigcup_{v\in NB(u)} \mathscr{W}(v)$. Since there are at most $|\mathscr{W}|$ time slots in $\bigcup_{v\in NB(u)} \mathscr{W}(v)$, then there are at most $|\mathscr{W}|$ schedule nodes for each primary node. Thus, the number of nodes in $\mathscr{G}$ can be calculated as

$$\begin{aligned} \left|V'\right| &= |V| + \left|V_s\right| = n + \sum_{u\in V} |\mathscr{W}| \leq n + n \times |\mathscr{W}| \\ &\leq (1 + |\mathscr{W}|) \times n \end{aligned} \tag{1}$$

where $|V_s|$ is the number of schedule nodes.

Secondly, for any primary node u, there is an edge between u and $a_{u,i}$ ($a_{u,i} \in \Upsilon(u)$) according to Definition 3, which is included in E'_u. And for any schedule node $a_{u,i}$, there are at most Δ edges from $a_{u,i}$ to the primary nodes in $R(a_{u,i})$. Then, we can have

$$\begin{aligned} \left|E'\right| &= \sum_{u\in V} \left|E'_u\right| + \left|\bigcup_{u\in V} \left\{ \bigcup_{a_{u,i}\in\Upsilon(u)} E'_{a_{u,i}} \right\}\right| \\ &\leq \sum_{u\in V} |\Upsilon(u)| + \sum_{u\in V} (|\Upsilon(u)| \times \Delta) \\ &\leq (1 + \Delta) \times n \times |\mathscr{W}| \end{aligned} \tag{2}$$

□

4.2. Minimum Schedule Node Covering Problem. In this subsection, we will introduce a new kind of problem, *i.e.*, *Minimum Scheduling Node Covering* problem. It tries to cover all the nonsource primary nodes in the auxiliary graph with minimum schedule nodes. The new problem can be defined as follows.

Definition 6 (minimum schedule node covering problem (MSNC)). Given a source node s and an auxiliary graph $\mathscr{G}$, the MSNC problem is objected to compute a set of schedule nodes, *i.e.*, A_S, which satisfies the following conditions:

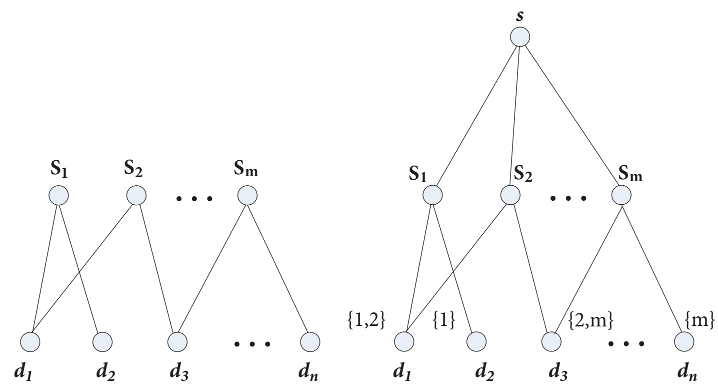
(1) For each node $u \in V \,\&\, u \neq s$, there must exist a schedule node in A_S, *i.e.*, a_v, which satisfies that $u \in R(a_v)$ ($R(a_v)$ is the set of primary nodes which can be reached by a_v).

(2) The size of A_S, *i.e.*, $|A_S|$, is minimized.

Theorem 7. *The MSNC problem is NP-hard and there exists no polynomial-time approximation algorithm with ratio of* $(1 - o(1)) \ln \Delta$ *for MSNC unless* $NP \subseteq DTIME(n^{O(\log\log n)})$.

Proof. We prove the NP-hardness of the MSNC problem by showing that the MSC problem can be mapped to one of its special cases. For example, Figure 2(a) gives an example of MSC, in which $U = \{d_1, d_2, \ldots, d_n\}$ is the universe of all n elements and set $S = \{S_1, S_2, \ldots, S_m\}$ denotes m subsets of U. An edge is created between d_i and S_k if d_i $(1 \leq i \leq n) \in S_k$ $(1 \leq k \leq m)$. The MSC problem is then to calculate a set of minimum nodes from S to cover all the nodes in U.

To prove the NP-hardness of MSNC, a simple graph with a source node s and its neighboring nodes d_i $(1 \leq i \leq n)$ is constructed. The working plan of each node is set as follows: (1) Initially, the working schedule of d_i $(1 \leq i \leq n)$ is set null; (2) If d_i $(1 \leq i \leq n) \in S_k$ $(1 \leq k \leq m)$, then $\mathscr{W}(d_i) = \mathscr{W}(d_i) \cup \{k\}$. Then, the corresponding auxiliary graph is shown in Figure 2(b), where S_k $(1 \leq k \leq m)$ denote the schedule nodes. Then the MSNC problem is equivalent to the MSC problem, which is proved to NP-hard. The theorem is proved. □



(a) The MSC problem (b) The MSNC problem

Figure 2: The example of MSNC problem. Figure 2 is reproduced from Chen et al. (2018) (under the Creative Commons Attribution License/public domain).

Input: The auxiliary graph $\mathscr{G}$, and the source node s;
Output: The set of schedule nodes A_S;
(1) $V_p \longleftarrow V - \{s\}, A_S \longleftarrow \emptyset$;
(2) $V_s \longleftarrow$ all the schedule nodes in $\mathscr{G}$;
(3) **while** $V_p \neq \emptyset$ **do**
(4) $a_v = \underset{a_u \in V_s}{\operatorname{argmax}} \left| R(a_u) \cap V_p \right|$;
(5) $A_S \longleftarrow A_S \cup \{a_v\}$;
(6) $V_p \longleftarrow V_p - (R(a_v) \cap V_p)$;
(7) $V_s \longleftarrow V_s - \{a_v\}$;
(8) **end while**
(9) **return** A_S;

Algorithm 2: Approximate algorithm for MSNC problem.

To solve the MSNC problem, we exploit the greedy set covering method [31] and choose the schedule node iteratively. Initially, A_S is set $\emptyset$ and V_p is set $V - \{s\}$. In each loop, the schedule node which has maximum adjacent uncovered primary nodes, *i.e.*, $a_v = \operatorname{argmax}_{a_u \in V_s} |R(a_u) \cap V_p|$, is found. Then, the schedule node a_v is added to A_S, i.e., $A_S = A_S \cup \{a_v\}$, and the covered primary nodes are removed from V_p. This process repeats until there are no uncovered primary nodes in V_p. The detailed procedure is shown in Algorithm 2. Since a schedule node can connect to at most Δ primary nodes, then Algorithm 2 can have an approximation ratio of $\ln(\Delta + 1)$ [31].

4.3. Generating the Pseudo Broadcast Tree. After obtaining the set of schedule nodes A_S, a *Pseudo Broadcast Tree* in the auxiliary graph is then constructed to connect all schedule nodes in A_S and these schedule nodes' primary nodes. The pseudo broadcast tree is mainly constructed by two steps: (1) The schedule nodes in A_S and their primary nodes are first organized into several subtrees; (2) These subtrees are then merged into one primary tree.

Assume $L(v)$ is the level of each node v (including the primary nodes and schedule nodes in the auxiliary graph). The above step (1) works as follows.

Initially, the level of source node s is set 0, *i.e.*, $L(s) = 0$. Then, a breadth-first search is conducted from s level by level.

Second, let $A_P = \{a_v.p \mid a_v \in A_S\}$ denote the set of primary nodes whose schedule nodes is in A_S. If s is not included in A_P, add s into A_P, *i.e.*, $A_P = A_P \cup \{s\}$.

Third, let u be the primary node with the smallest level in A_P and T_u denote the subtree rooted at u. Add u into T_u at the beginning, *i.e.*, $T_u = T_u \cup \{u\}$. Then we can construct T_u by the following:

(1) For any primary node m in $T_u \cap A_P$, let $A_S(m) = \{a_v \mid a_v \in A_S \,\&\, a_v.p = m\}$ be the set of schedule nodes which are contained in A_S and their primary node is m.

(2) For any schedule node $a_m \in A_S(m)$, add it into the subtree T_u, *i.e.*, $T_u = T_u \cup \{a_m\}$ and create an edge from m to schedule node a_m. Let n be a primary node in $R(a_m)$ and not included in any subtrees, add it into subtree T_u, *i.e.*, $T_u = T_u \cup \{n\}$ and create an edge from schedule node a_v to n.

(3) Delete the primary node m from A_P, *i.e.*, $A_P = A_P - \{m\}$.

The above three steps repeat until there are no nodes in $T_u \cap A_P$. Then, we will choose another primary node with the smallest level in A_P to begin the above process. Note that several subtrees may be generated in this step. Finally, when A_P is empty, all the schedule nodes in A_S and the primary nodes in A_P are included in these subtrees.

Next, we will introduce the method to merge these subtrees.

Let TR denote the set of roots of these subtrees and the source node s is also included in TR. The objective of this process is to merge all the subtrees into the tree which is rooted at s, *i.e.*, T_s. Note that T_s may include only one node at the beginning. Delete s from TR. The merging process mainly works as follows.

Case 1. If there is a node m in T_s and one of its schedule nodes, *i.e.*, a_m, that can cover most root nodes in TR, then create an edge from m to a_m, and connect these root nodes in TR to

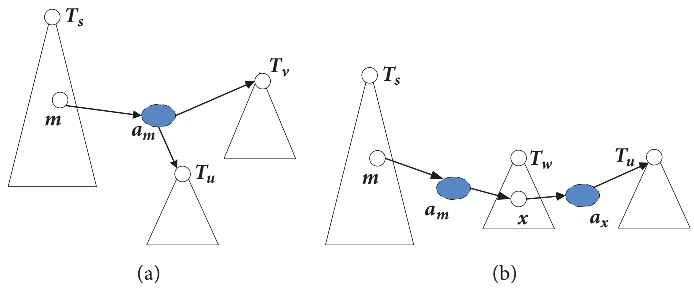


Figure 3: The example of the merging process. Figure 3 is reproduced from Chen et al. (2018) (under the Creative Commons Attribution License/public domain).

schedule node a_m, which is shown in Figure 3(a). Delete these root nodes from TR.

Case 2. Assume u is the root node with the smallest level in TR. There must exists a neighbor of u (*i.e.*, $x \in NB(u)$) in the original graph and the primary node x is not in T_s. But it can be reached by a primary node in T_s, *i.e.*, m, just as the example shown in Figure 3(b). Let the schedule node of m which can reach x be a_m, *i.e.*, $x \in R(a_m)$. Create an edge from m to a_m and an edge from a_m to x. After that, let the schedule node of x that can reach u be a_x, then create an edge from x to a_x and an edge from a_x to u. Let the subtree contains x be T_w; remove node x from T_w. Delete u from TR.

The above process repeats until all the subtrees are added to T_s either by Case 1 or Case 2. And then the pseudo broadcast tree is constructed completely. The correctness of this merging process can be verified by Theorem 10.

Lemma 8. *Let l denote the level of node x in the auxiliary graph. If x is a primary node, then l is even, and if x is a schedule node, then l is odd.*

Proof. First, we prove that if x is a primary node, then l is even. It can be easily verified that there exists a level of schedule nodes between two contiguous level of primary nodes when conducting the breadth-first search initiated from the source node s. Since the level of the source node s is 0, *i.e.*, $L(s) = 0$ at the beginning, then all the schedule nodes of s must be level 1, and then the primary nodes reached by these schedule nodes have the level 2. Actually, let the hop-distance from a primary node u to node s in the original network be k, then we can find the level of the primary node u in the auxiliary graph must be $2k$, which is even.

Second, since the schedule nodes can only be reached by its corresponding primary nodes, whose level is even, then the level of the schedule nodes must be odd.

Therefore, in the generated breadth-first tree, the level of the primary nodes is even and the level of the schedule nodes is odd. □

Lemma 9. *For any node v in the subtree rooted at node u, i.e., T_u, we have $L(v) \geq L(u) - 2$.*

Proof. First, we prove that, for any primary node v in the subtree rooted at node u, we have $L(v) \geq L(u) - 2$. According to the construction of subtree, we can find that the root node u must be a primary node in A_P. Since one of its schedule nodes can reach a node whose level is less than u, *i.e.*, w, then we can have $L(w) = L(u) - 2$. Assume there exist a primary node, *i.e.*, w, that $L(w) < L(u) - 2$. Then there must exist a schedule node a_x in A_S that covers primary node w. According to Lemma 4, then we can have $L(w) - 1 \leq L(a_x) \leq L(w) + 1$, that is, $L(a_x) < L(u) - 3$. The level of the primary node of the schedule node a_x, *i.e.*, x, can be calculated as $L(x) = L(a_x) + 1$, which is less than $L(u) - 2$. Since a_x is in A_S, then we can also have x is in A_P. However, when subtree T_u is constructed, primary node u has the smallest level, then we have $L(x) \geq L(u)$, which is a contradiction. Therefore, the level of all the primary nodes is no less than $L(u) - 2$.

Second, since the level of the schedule nodes are larger than level of its primary nodes, thus, the level of all schedule nodes is also no less than $L(u) - 2$.

Therefore, for any node v in the subtree rooted at node u, we have $L(v) \geq L(u) - 2$. The lemma is proved. □

Theorem 10. *For any subtree rooted at node u, it can be merged into T_s.*

Proof. First, according to the merging process, if there exists a schedule node a_x, whose primary node is included in T_s and can reach node u, then the subtree rooted at node u will be merged into T_s as in Case 1.

Second, we will prove that if there does not exist such a schedule node, then there must exists a primary node m as in the Case 2. Let the set of the primary nodes whose level is less than the one of u and can reach node u by some of their schedule nodes be $P_{up}(u)$. Then, there must exist a primary node in T_s, *i.e.*, m, whose level is less than the one of $P_{up}(u)$, and can reach some node in $P_{up}(u)$, *i.e.*, x, by some of its schedule nodes. According to Lemma 9, the level of the primary nodes in $P_{up}(u)$ is $L(u) - 2$. Obviously, we can have $L(m) = L(u) - 4$. Let the subtree contain node m be T_w. According to Lemma 9, then we can have $L(w) \leq L(m) + 2 \leq L(u) - 2$. Since the primary node with the smallest level is u, then the subtree T_w and node m must have been included in T_s, which means the primary node m is existed.

Therefore, the subtrees except T_s can be added to the primary subtree T_s either by Case 1 or Case 2. The theorem is proved. □

Input: The duty-cycled network G and the source node s;
Output: The broadcast schedule $\mathcal{S}$;
(1) Constructing the auxiliary graph $\mathcal{G}$ as in Section 4.1 by integrating nodes' active time slot into the network;
(2) Exploiting Algorithm 2 for the MSNC problem to obtain a minimum number of schedule nodes, *i.e.*, A_S, to cover all the primary nodes in the auxiliary graph $\mathcal{G}$;
(3) Calculating a set of subtrees as in Section 4.3 with the scheduled nodes in A_S to connect all the primary nodes in $\mathcal{G}$;
(4) Merging all the subtrees into a pseudo broadcast tree which contains the scheduling information of all nodes, *i.e.*, T_s;
(5) Transforming the pseudo broadcast tree T_s to a broadcast schedule, *i.e.*, $\mathcal{S} = \{T, \mathcal{S}(T)\}$;
(6) **return** The broadcast schedule $\mathcal{S}$;

Algorithm 3: Approximate Minimum-Transmission Broadcasting.

4.4. Calculating the Broadcast Schedule. Actually, the pseudo broadcast tree constructed in the above subsection not only includes the information of the broadcast tree but also the scheduling information of nonleaf nodes. It can be transformed to the broadcast schedule (including a broadcast tree T and the transmitting schedules of nonleaf nodes) easily.

First, the broadcast tree T can be obtained by removing all the schedule nodes and create an edge from its father to all of its child nodes in the generated pseudo broadcast tree T_s.

Second, the transmitting schedules of nonleaf nodes in T, *i.e.*, $\mathcal{S}(T)$ can be obtained according to the schedule nodes in the pseudo broadcast tree. That is, for each schedule node a_u in T_s, a transmitting schedule for node $a_u.p$, *i.e.*, $[a_u.p, a_u.t + \chi * |\mathcal{W}|, ch(a_u)]$ is added in $\mathcal{S}(T)$, where $ch(a_u)$ denotes the set of children of schedule node a_u in T_s, and χ is used for collisions avoiding [32, 33] and data freshness guaranteed during broadcasting, which can be just set as the level of node u in the new broadcast tree T.

Now, the complete Approximate Minimum-Transmission Broadcasting (*i.e.*, AMTB) algorithm for MTBDCA problem is introduced and the detailed procedure is shown in Algorithm 3.

4.5. Performance Analysis of AMTB Algorithm. The correctness of AMTB is proved in Theorem 11, where the proof can be found in the conference version [23].

Theorem 11. *The broadcast schedule generated by AMTB is complete and correct.*

Theorem 12 gives the lower bound of MTBDCA problem. The approximation ratio of AMTB is proved in Theorem 13 and the proof of this theorem is available in the conference version [23].

Theorem 12. *The lower bound on the number of transmissions of any optimal broadcast schedule for MTBDCA is at least* $|A_S|/\ln(\Delta+1)$, *where* $|A_S|$ *denote the size of the obtained set of schedule nodes by Algorithm 1.*

Proof. Assume OPT is the size for any optimal schedule for MTBDCA problem. To finish the broadcasting process, all nodes except the source need receive the messages from some intermediate nodes. It means all the primary nodes in the auxiliary graph except the primary node of the source need to be covered at least once, then we can get $OPT \geq |A_S^{opt}|$, where $|A_S^{opt}|$ denote the size of any optimal schedule for MSNC problem. According to the above analysis, $|A_S| \leq |A_S^{opt}| \cdot \ln(\Delta+1)$. Thence, we can get $OPT \geq |A_S^{opt}| \geq |A_S|/\ln(\Delta+1)$. The theorem is proved. □

Theorem 13. *The approximation ratio of AMTB is at most* $\ln(\Delta+1)$.

Lemma 14. *The time complexity of Algorithm 2 is at most* $O(n^2)$.

Proof. Since there are at most $(1+|\mathcal{W}|) * n$ schedule nodes in the auxiliary graph according to Lemma 8, then step (4) in Algorithm 2 takes at most $O((1+|\mathcal{W}|) * n) = O(n)$ time ($|\mathcal{W}|$ is often a constant). And there are at most $n-1$ primary nodes need to be covered, then the step (3) to (8) takes at most $O(n^2)$ time. Thus, the time complexity of Algorithm 2 is $O(n^2)$. □

Theorem 15. *The time complexity of AMTB is* $O(n^2)$.

Proof. First, according to Lemma 8, it takes at most $O(1+\Delta)\times n\times|\mathcal{W}|$ time to construct the auxiliary graph. Since Δ and $|\mathcal{W}|$ are often constant, then Algorithm 1 takes at most $O(n)$ time.

Second, according to Lemma 14, it needs at most $O(n^2)$ time to return a set of minimum schedule nodes for MSNC problem.

Third, to construct the pseudo broadcast tree, several subtrees are first generated and then these subtrees are merged into a single tree. Since there are at most $O(n)$ root nodes, and each subtree takes at most $O(n)$ time, then it takes at most $O(n^2)$ time to generate the subtrees. To merge these subtrees, one need to choose a subtree to merged into T_s at each step, which takes at most $O(n)$ time, and there are at most $O(n)$ subtrees. Thus, it takes at most $O(n^2)$ time to merge the subtrees.

Finally, the broadcast schedule can be obtained by searching the broadcast tree, which takes at most $O(n)$ time.

Therefore, the time complexity of AMTB is $O(n+n^2+n^2+n) = O(n^2)$. □

4.6. Discussion of Algorithms for All-to-All Broadcasting Problem. In the above subsections, the Minimum-Transmission Broadcasting problem with only one source node is investigated. Despite the one-to-all broadcasting problem, the all-to-all broadcasting problem is also very important in wireless networks [34], which is aimed at distributing the messages from all the nodes to all the other nodes. Similar as in [34], we assume the size of messages is unbounded. Then the above AMTB algorithm can still give an upper bound for the all-to-all MTBDCA problem. It mainly works as follows. First, all the messages are aggregated to a center node with the data aggregation technique in [17]. Second, the aggregated messages can be delivered to all the other nodes with the AMTB algorithm. The proposed algorithm can achieve an approximation ratio of $\ln(\Delta + 1) + \Delta$, which is shown in Theorem 16. As for the more efficient algorithms for all-to-all MTBDCA problem, it will be investigated in our future work.

Theorem 16. *The approximation ratio of the proposed algorithm for all-to-all MTBDCA problem is* $\ln(\Delta + 1) + \Delta$.

Proof. Let OPT_a and OPT_1 be the number of transmissions of the optimal solution for the all-to-all and one-to-all MTBDCA problem, respectively. In the above algorithm for all-to-all MTBDCA problem, the data aggregation process takes at most $n - 1$ transmissions, and the broadcasting process takes at most $\ln(\Delta + 1) \times OPT_1$ transmissions according to Theorem 13. Then the total number of transmissions is $n - 1 + \ln(\Delta + 1) \times OPT_1$. In addition, each node covers at most δ neighbors in the AMTB algorithm, then we can have $OPT_1 \times \Delta \leq n$. Since OPT_a must be larger than OPT_1, then we can have $(n-1+(\Delta+1)\times OPT_1)/OPT_a \leq (OPT_1 \times \Delta + \ln(\Delta + 1)\times OPT_1)/OPT_1 \leq \ln(\Delta+1)+\Delta$. The theorem is proved. □

5. Simulation Results

In this section, we will evaluate the performance of AMTB through extensive simulations. In the experiments, AMTB is mainly compared with the following algorithms.

First, the existing algorithms designed for MTB problem in duty-cycled networks, including the classical one, *i.e.*, CSCA [18] and the recently proposed one, *i.e.*, BRMS [21], are both evaluated to demonstrate that the AMTB algorithm can benefit from scheduling with nodes' all active time slots.

Second, two baseline algorithms, *i.e.*, CDS-based (Connected Dominate Set) algorithm and SPT-based (Shortest Path Tree) algorithm, are also implemented. In such two algorithms, a CDS-based and SPT-based broadcast tree is constructed firstly. Then, to reduce the number of transmissions, we calculate the transmitting schedules of all forwarding nodes in the broadcast tree optimally by enumerating all the possible results.

In the simulations, we mainly focus on the performance of the number of transmissions of each algorithm under various network topologies. Similar as in [30], the Networkx [35] tool was used to test more complex networks and generate different network topologies, where the number of sensor nodes is set from 100 to 400. The duty cycle of each node is set from 10% to 35% and the nodes' working plan is generated randomly to satisfy a wide range of configurations. In all the simulation results, each plotted point represents the average of 100 executions.

5.1. Impact of $|W|$. First, the performance of each algorithm under different length of working cycle (*i.e.*, $|\mathscr{W}|$) and number of nodes (*i.e.*, N) is evaluated. In this group of experiments, the duty cycle is set 20% and the average number of neighbors of each node is set 5.

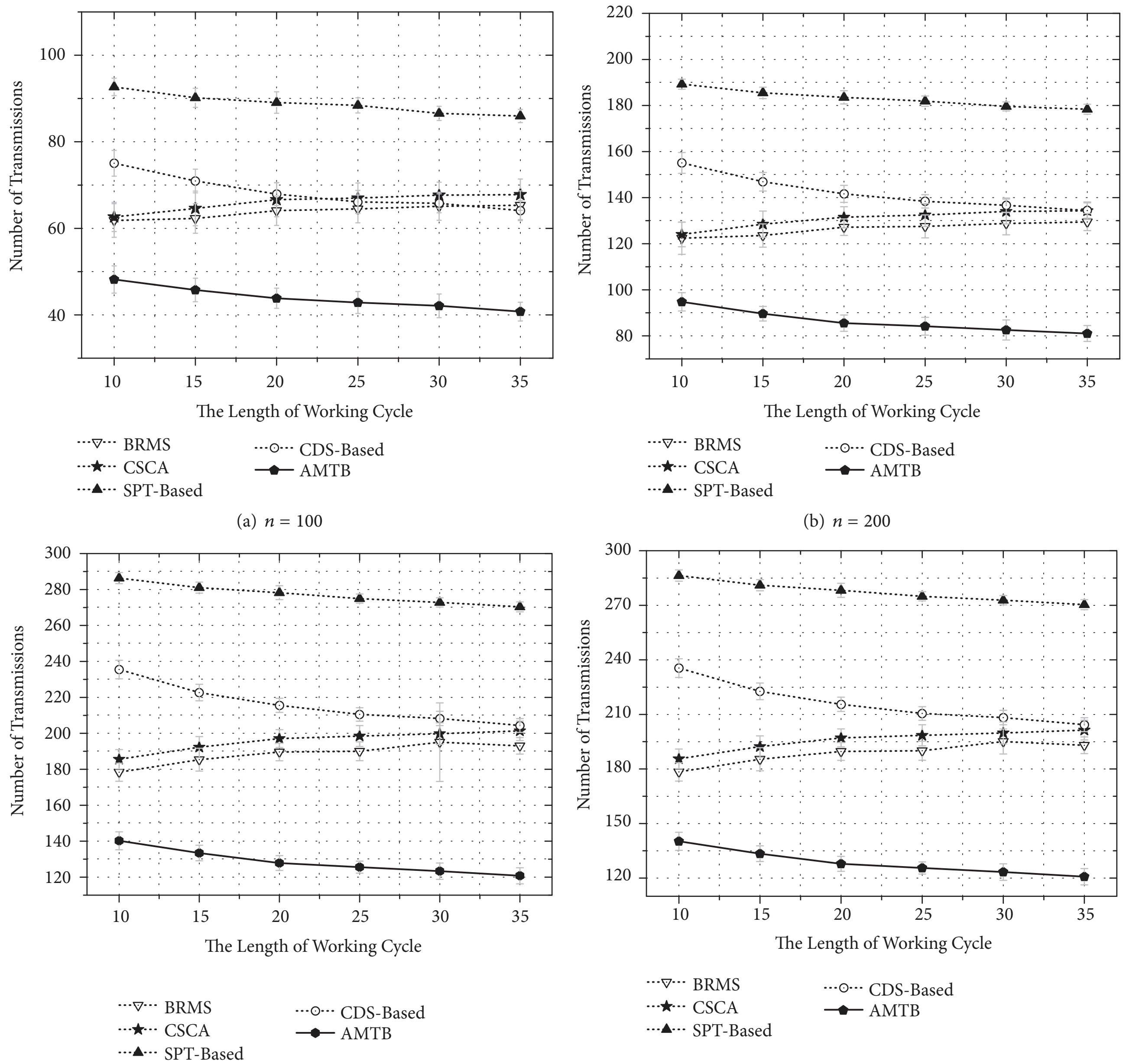
Figure 4 shows the number of transmissions of each algorithm when we vary the length of working cycle under different network size. The number of nodes in Figures 4(a), 4(b), 4(c), and 4(d) is set as 100, 200, 300, and 400, respectively. In all scenarios, the number of transmissions of AMTB is much less than the one of other algorithms. Compared to the recently proposed BRMS algorithm, the number of transmissions of AMTB is decreased by 50% on average. This is mainly due to AMTB can exploit the whole active time slots among all neighbors for scheduling to reduce the number of transmissions. Additionally, although the CDS-based and SPT-based methods consider multiple active time slots during scheduling, they perform even worse than BRMS. This demonstrates that the performance can be still be worse even exploiting the optimal scheduling method if the broadcasting tree is not constructed well. Thus one must choose the forwarding nodes carefully when constructing the broadcasting tree.

Another finding is that when the length of working cycle is increased and the duty cycle remains unchanged, the number of transmissions of BRMS and CSCA method is even more, while the one of the algorithms exploiting multiple active slots for scheduling, *i.e.*, AMTB, SPT-based, and CDS-based method, is reduced. This is due to CSCA and BRMS methods only exploit nodes' first active time slot for scheduling. However, the number of neighbors who shares the same active time slot may be reduced when $|\mathscr{W}|$ is increased. Actually, the number of active time slots in nodes' working plan is increased, and AMTB, SPT-based and CDS-based method can benefit from the increased active time slots to reduce the number of transmissions.

5.2. Impact of Duty Cycle. In this simulation, the performance of each algorithm under different duty cycle is compared. The length of nodes' working cycle is equal to 20.

Figure 5 shows the number of transmissions of each algorithm under different duty cycle. The number of nodes is set from 100 to 400, and the results are shown in Figures 5(a), 5(b), 5(c), and 5(d). As we can see, AMTB perform much better than all the other methods in all scenarios, which demonstrates its efficiency for MTBDCA problem. The SPT-based method still performs much worse than the others. This demonstrates that the SPT-based method may be not suitable for MTB problem in duty-cycled WSNs.

Another finding is that both the number of transmissions of the existing methods (BRMS and CSCA) and the one of AMTB is improved when we increase the duty cycle. This is because the probability of the first active time slot of neighbors being the same is increased when the duty cycle is increased (which means the size of node's working plan is increased). Note that, if we further increase the duty cycle, the number of transmissions of each method will still be decreased until the duty cycle is increased to 1. In this case, the MTBDCA problem is equal to the MTB problem in the traditional WSNs, and the number of transmissions of each method will be steadily.



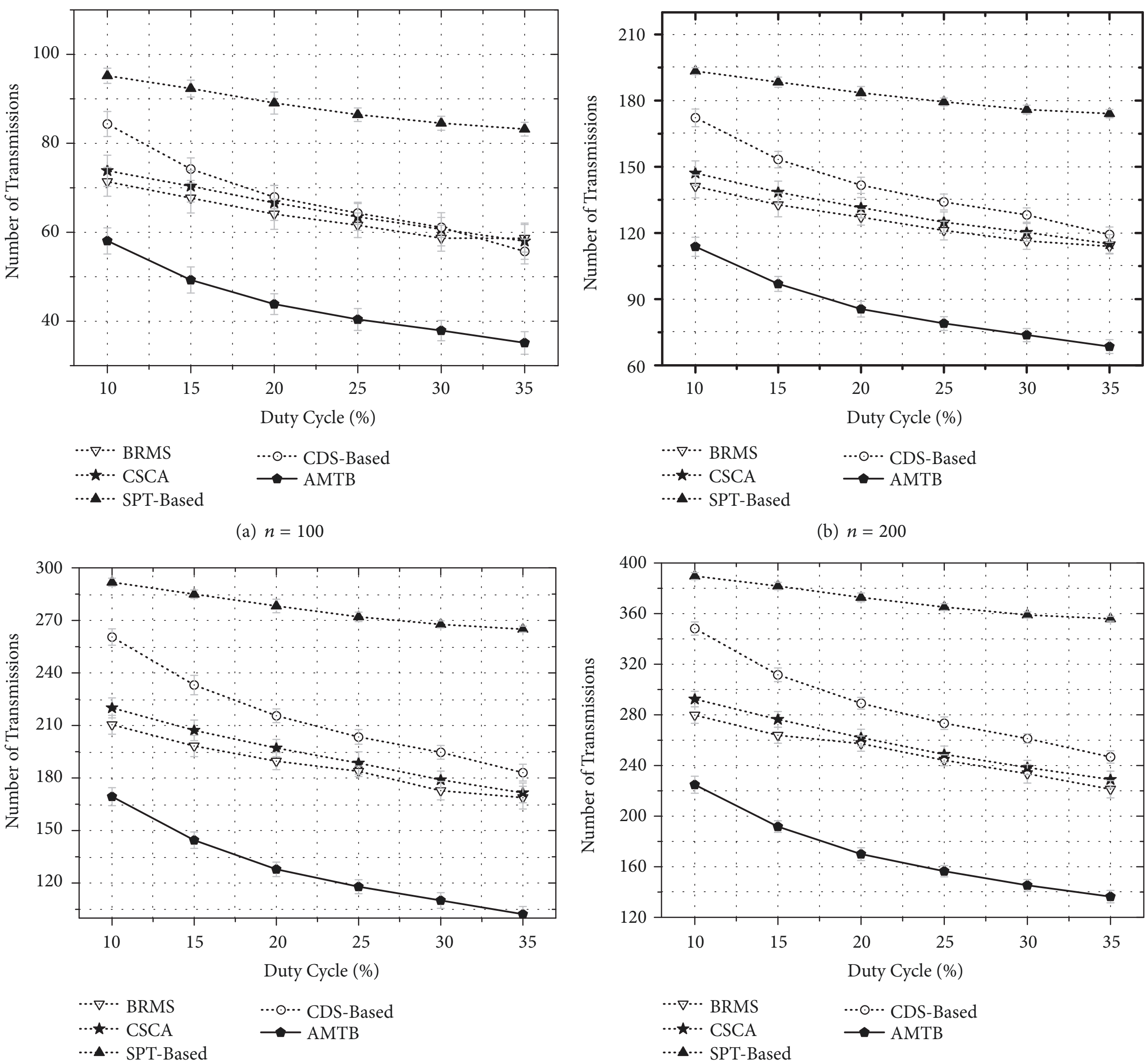
(a) n = 100 (b) n = 200 (c) n = 300 (d) n = 400

Figure 4: The number of transmissions when duty cycle is set 20%. Figure 4 is reproduced from Chen et al. (2018) (under the Creative Commons Attribution License/public domain).

5.3. Impact of Neighbors. Finally, we exploit Networkx to investigate the relationship between the number of transmissions of each method and the average number of neighbors, *i.e.,* ψ. To compare, the number of nodes is set 200 and the average number of neighbors is set 5 and 10 in two group of experiments.

Figure 6 presents the number of transmissions of the above algorithms under different length of working cycle, where the duty cycle is set 20%. Firstly, as we can see in Figures 6(a) and 6(b), the number of transmissions of each method is decreased notably if we increase the average number of neighbors in the network from 5 to 10. Although the total number of nodes in the network is unchanged, the nodes sharing the same active time slot is increased when the number of neighbors is increased. As a result, the required number of transmissions is reduced. Compared to other methods, the proposed AMTB algorithm can reduce the number of transmissions by 40%, which demonstrates the efficiency of the proposed method. Similar to Figure 4, the number of transmissions of BRMS and CSCA are both increased when the length of working cycle is increased since they only exploit nodes' first active time slot for scheduling.

Figure 7 compares the number of transmissions of each method when we set $n = 200$ and $\psi = 10$ and change



(a) $n = 100$

(b) $n = 200$

(c) $n = 300$

(d) $n = 400$

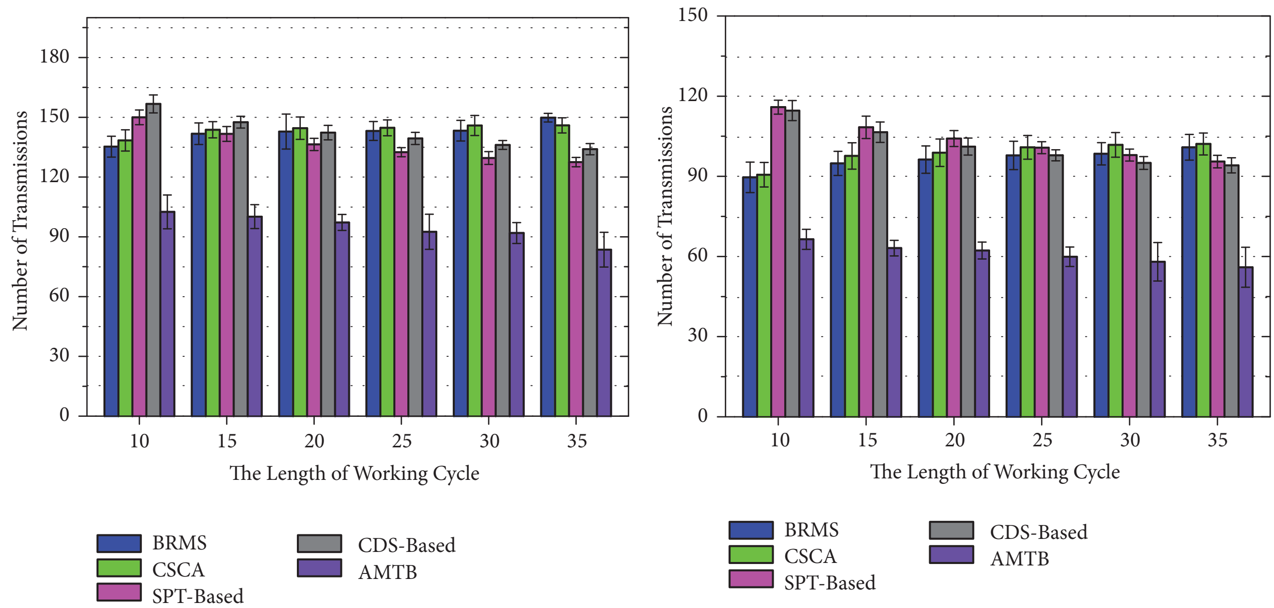
Figure 5: The number of transmissions when $|W|$ is set 20.

the duty cycle. In this case, the trend of each method is similar as in Figure 5, *i.e.*, the number of transmissions of each method is decreased when the duty cycle is increased. And the number of transmissions of AMTB is much less than the one of other methods. Comparing Figure 7(a) with Figure 7(b), the number of transmissions of each method is decreased notably (almost 50%) when the average number of neighbors is increased. This demonstrates that the number of transmissions is also highly related to the average degree in the network.

6. Conclusion and Future Work

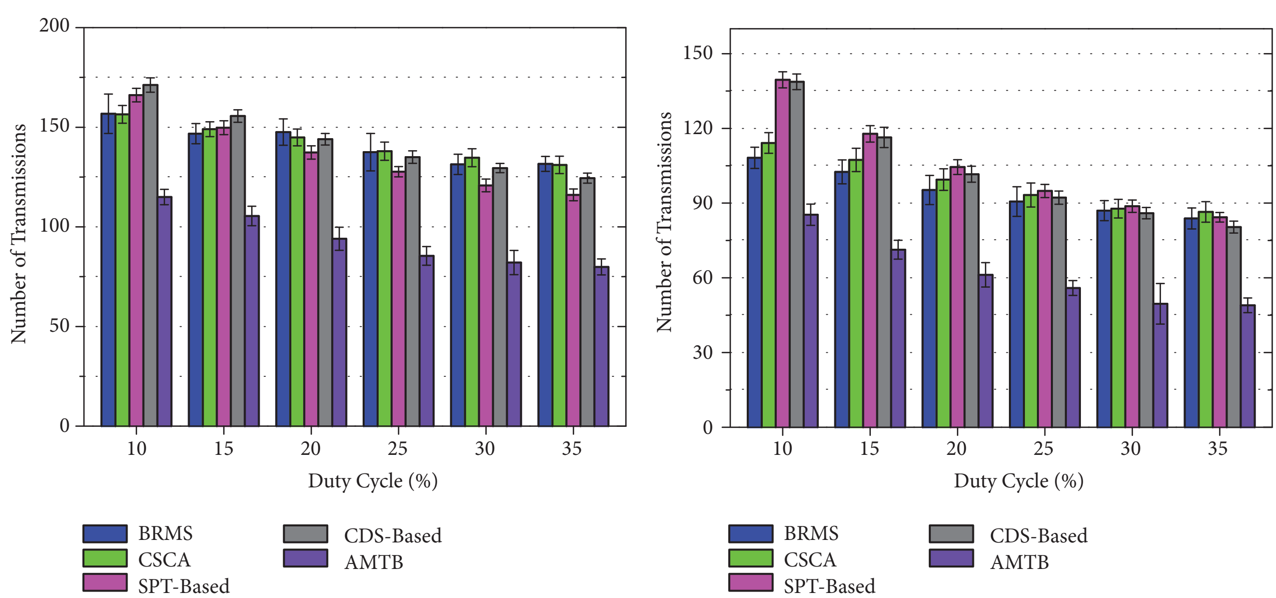
In this paper, the MTBDCA problem in duty-cycled wireless networks is investigated. It is proved to be NP-hard and $o(\ln \Delta)$-inapproximable, where Δ denotes the maximum degree in the network. An auxiliary graph and the minimum schedule node covering problem is proposed to exploit nodes' all active time slots for scheduling. Based on this, a $\ln(\Delta + 1)$-approximation algorithm is proposed for MTBDCA. The efficiency of the proposed algorithm is demonstrated by extensive simulations.

As for the future work, we will further investigate the all-to-all MTBDCA problem, including providing a more efficient algorithm for the all-to-all MTBDCA problem and studying the tight approximation ratio of the proposed methods. The simulations will also be conducted to demonstrate the efficiency of the proposed algorithms.



(a) $\psi = 5$ (b) $\psi = 10$

FIGURE 6: Performance analysis under different $|\mathscr{W}|$.



(a) $\psi = 5$ (b) $\psi = 10$

FIGURE 7: Performance analysis under different *Duty Cycle*.

Data Availability

The simulation data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

We thank T. Le and Z. Cai for useful contributions in the conference version [23]. This work is partly supported by the Opening Project of Guangdong Province Key Laboratory of Cyber-Physical System under Grant no. 2016B030301008, the National Natural Science Foundation of China (NSFC) under Grants nos. 61802071, 61502116, and 61502110, the China

Scholarship Council under Grant no. 201808440012, and the Natural Science Foundation of Guangdong under Grants nos. 2018A030310541, 2015B010104005, and 2017A010101017.

References

[1] S. Cheng, Z. Cai, J. Li, and H. Gao, "Extracting kernel dataset from big sensory data in wireless sensor networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 4, pp. 813–827, 2017.

[2] J. Yu, S. Wan, X. Cheng, and D. Yu, "Coverage contribution area based k-coverage for wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 9, pp. 8510–8523, 2017.

[3] J. Li, S. Cheng, Z. Cai et al., "Approximate holistic aggregation in wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 13, no. 2, pp. 1–24, 2017.

[4] Q. Chen, S. Cheng, H. Gao, J. Li, and Z. Cai, "Energy-efficient algorithm for multicasting in duty-cycled sensor networks," *Sensors*, vol. 15, no. 12, pp. 31224–31243, 2015.

[5] S. Cheng, Z. Cai, J. Li et al., "Curve query processing in wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 11, pp. 5198–5209, 2015.

[6] Z. He, Z. Cai, S. Cheng, and X. Wang, "Approximate aggregation for tracking quantiles and range countings in wireless sensor networks," *Theoretical Computer Science*, vol. 607, no. 3, pp. 381–390, 2015.

[7] I. Chlamtac and S. Kutten, "Tree-based broadcasting in multihop radio networks," *IEEE Transactions on Computers*, vol. C-36, no. 10, pp. 1209–1223, 1987.

[8] W. Lou and J. Wu, "A cluster-based backbone infrastructure for broadcasting in MANETs," in *Proceedings of the International Parallel and Distributed Processing Symposium, IPDPS '03*, 2003.

[9] B. Das and V. Bharghavan, "Routing in ad-hoc networks using minimum connected dominating sets," in *Proceedings of the IEEE International Conference on Communications (ICC '97)*, pp. 376–380, June 1997 (Basque).

[10] J. Wu and H. Li, "On calculating connected dominating set for efficient routing in ad hoc wireless networks," in *Proceedings of the 3rd International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM '99)*, pp. 7–14, 1999.

[11] W. Lou and J. Wu, "On reducing broadcast redundancy in ad hoc wireless networks," *IEEE Transactions on Mobile Computing*, vol. 1, no. 2, pp. 111–123, 2002.

[12] J. Wu and W. Lou, "Forward-node-set-based broadcast in clustered mobile ad hoc networks," *Wireless Communications and Mobile Computing*, vol. 3, no. 2, pp. 155–173, 2003.

[13] J. Wu, "An enhanced approach to determine a small forward node set based on multipoint relays," in *Proceedings of the Vehicular Technology Conference*, pp. 2774–2777, 2003.

[14] C. Adjih, P. Jacquet, and L. Viennot, "Computing connected dominated sets with multipoint relays," *Ad Hoc and Sensor Wireless Networks*, vol. 1, no. 1, pp. 27–39, 2005.

[15] J. Wu, W. Lou, and F. Dai, "Extended multipoint relays to determine connected dominating sets in MANETs," *IEEE Transactions on Computers*, vol. 55, no. 3, pp. 334–347, 2006.

[16] Q. Chen, H. Gao, S. Cheng et al., "Centralized and distributed delay-bounded scheduling algorithms for multicast in duty-cycled wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 25, no. 6, pp. 3573–3586, 2017.

[17] Q. Chen, H. Gao, S. Cheng et al., "Distributed non-structure based data aggregation for duty-cycle wireless sensor networks," in *Proceedings of the IEEE INFOCOM*, pp. 145–153, 2017.

[18] J. Hong, J. Cao, W. Li, S. Lu, and D. Chen, "Minimum-transmission broadcast in uncoordinated duty-cycled wireless Ad Hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 1, pp. 307–318, 2010.

[19] T. L. Duc, D. T. Le, H. Choo et al., "On minimizing the broadcast redundancy in duty-cycled wireless sensor networks," in *Proceedings of the ACM International Conference on Ubiquitous Information Management and Communication*, 2013.

[20] T. Le-Duc, D. Le, H. Choo et al., "Level-based approach for minimum-transmission broadcast in duty-cycled wireless sensor networks," *Pervasive and Mobile Computing*, vol. 27, pp. 116–132, 2016.

[21] T. Le-Duc and V. Zalyubovskiy, "Towards broadcast redundancy minimization in duty-cycled wireless sensor networks," *International Journal of Communication Systems*, vol. 30, no. 6, pp. 1–12, 2017.

[22] L. Cheng, J. Niu, Y. Gu, C. Luo, and T. He, "Achieving efficient reliable flooding in low-duty-cycle wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 24, no. 6, pp. 3676–3689, 2016.

[23] Q. Chen, T. Le, L. Cheng et al., "Approximate minimum-transmission broadcasting in duty-cycled WSNs," in *Proceedings of the WASA*, pp. 40–52, 2018.

[24] P.-J. Wan, K. M. Alzoubi, and O. Frieder, "Distributed construction of connected dominating set in wireless Ad Hoc networks," *Mobile Networks and Applications*, vol. 9, no. 2, pp. 141–149, 2004.

[25] P.-J. Wan, L. Wang, and F. Yao, "Two-phased approximation algorithms for minimum CDS in wireless ad hoc networks," in *Proceedings of the 28th International Conference on Distributed Computing Systems, ICDCS '08*, pp. 337–344, July 2008.

[26] T. Shi, S. Cheng, J. Li et al., "Constructing connected dominating sets in battery-free networks," in *Proceedings of the IEEE INFOCOM*, pp. 1–9, May 2017.

[27] J. Yu, N. Wang, and G. Wang, "Constructing minimum extended weakly-connected dominating sets for clustering in ad hoc networks," *Journal of Parallel and Distributed Computing*, vol. 72, no. 1, pp. 35–47, 2012.

[28] J. Yu, X. Ning, Y. Sun et al., "Constructing a self-stabilizing CDS with bounded diameter in wireless networks under SINR," in *Proceedings of the IEEE INFOCOM*, May 2017.

[29] K. Han, Y. Liu, and J. Luo, "Duty-cycle-aware minimum-energy multicasting in wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 21, no. 3, pp. 910–923, 2013.

[30] Q. Chen, H. Gao, Y. Li et al., "Edge-based beaconing schedule in duty-cycled multihop wireless networks," in *Proceedings of the INFOCOM*, 2017.

[31] U. Feige, "A threshold of ln n for approximating set cover," *Journal of the ACM*, vol. 45, no. 4, pp. 634–652, 1998.

[32] J. Yu, B. Huang, X. Cheng et al., "Shortest link scheduling algorithms in wireless networks under SINR," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 3, pp. 2643–2657, 2017.

[33] X. Zhang, J. Yu, W. Li, X. Cheng, D. Yu, and F. Zhao, "Localized algorithms for yao graph-based spanner construction in wireless networks under SINR," *IEEE/ACM Transactions on Networking*, vol. 25, no. 4, pp. 2459–2472, 2017.

[34] X. Jiao, W. Lou, J. Ma, J. Cao, X. Wang, and X. Zhou, “Minimum latency broadcast scheduling in duty-cycled multihop wireless networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 1, pp. 110–117, 2012.

[35] Networkx, 2012, http://networkx.lanl.gov.