

Duty-Cycle-Aware Minimum-Energy Multicasting in Wireless Sensor Networks

Abstract—In duty-cycled wireless sensor networks, the nodes switch between active and dormant states, and each node may determine its active/dormant schedule independently. This complicates the Minimum-Energy Multicasting (MEM) problem, which has been primarily studied in always-active wireless ad-hoc networks. In this paper, we study the duty-cycle-aware MEM problem in wireless sensor networks, and we present a formulation of the Minimum-Energy Multicast Tree Construction and Scheduling (MEMTCS) problem. We prove that the MEMTCS problem is NP-hard, and it is unlikely to have an approximation algorithm with a performance ratio of $(1 - o(1)) \ln \Delta$, where Δ is the maximum node degree in a network. We propose a polynomial-time approximation algorithm for the MEMTCS problem with a performance ratio of $\mathcal{O}(H(\Delta + 1))$, where $H(\cdot)$ is the harmonic number. We also provide a distributed implementation of our algorithm. Finally, we perform extensive simulations and the results demonstrate that our algorithm significantly outperform other known algorithms in terms of both the total energy cost and the transmission redundancy.

I. INTRODUCTION

Wireless Sensor Networks (WSNs) are decentralized wireless networks without any preexisting infrastructures, and the sensor nodes are usually powered by batteries. Since the limited battery lifetime imposes a severe constraint on the network performance, it is imperative to develop energy conservation mechanisms for WSNs. One common approach for energy conservation in WSNs is duty-cycling, in which each node switches between active and dormant states, and the active/dormant schedule can differ from node to node [1]–[7]. Duty-cycling can be easily implemented, and is proved to be an effective way for energy conservation [2]. Therefore, Duty-Cycled Wireless Sensor Networks (DC-WSNs) have often been adopted by various applications [8]–[10].

Multicasting is a crucial component of wireless networking, and it has been applied to WSNs in supporting data dissemination for distributed data management (e.g., [11]). Therefore, designing an energy efficient multicast protocol is of great importance. In an Always-Active Wireless Ad-hoc NETWORK (AA-WANET), the network topology is static, and each forwarding node can cover all its neighboring nodes by only one transmission. Therefore, the main task of the Minimum-Energy Multicasting (MEM) problem in AA-WANETs is to select appropriate forwarding nodes such that a multicast tree with the minimum energy cost can be constructed. This problem was proved to be NP-hard, and some approximation algorithms have been proposed [12]–[16].

In DC-WSNs, however, new challenges arise. More specifically, the network topology is now intermittently connected, and a forwarding node may transmit the same data packet

many times to reach its neighboring nodes. Therefore, to design energy efficient multicasting algorithms in DC-WSNs, not only the forwarding nodes should be selected appropriately to construct a multicast tree, but also the transmissions of each forwarding node need to be scheduled intelligently to cover the receiver nodes and reduce the transmission redundancy. These two problems must be handled holistically so that the total energy cost can be reduced. Thus, the existing solutions for the MEM problem in AA-WANETs are not suitable for DC-WSNs, and we need to design new energy efficient multicasting algorithms to meet the challenges in DC-WSNs.

A. Background and Motivations

The MEM problem in wireless ad-hoc networks has been studied in [12]–[16]. Wieselthier *et al.* [12] studied the minimum power broadcast/multicast routing problems under a scenario where each node can adjust its transmission power continuously, and proposed several greedy heuristics. Wan *et al.* [13] proved that the heuristics proposed by [12] have linear approximation ratios, and provided several approximation algorithms with constant approximation ratios for the MEM problem based on the Minimum Steiner Tree algorithm. Liang [14] considered a scenario in which each wireless node can adjust its transmission power in a discrete fashion, and the communication links are symmetric. He proposed a centralized approximation algorithm with performance ratio $4 \ln |M|$ for building a minimum-energy multicasting tree, where M is the set of destination nodes in a multicast request. Li *et al.* [15] considered a case in which all nodes have a fixed transmission power and the communication links are asymmetric. They converted the minimum-energy multicasting problem to an instance of the Directed Steiner Tree (DST) problem [17], and presented several heuristics. Liang *et al.* [16] studied the minimum-energy all-to-all multicasting problem in wireless ad-hoc networks, which aims at building a shared multicast tree such that the total energy consumption of performing an all-to-all multicast session by this tree is minimized. They proved the NP-completeness of this problem and also proposed several approximation algorithms. However, all the algorithms proposed in [12]–[16] took **the assumption that the network nodes are always-active**. Therefore, they are not directly applicable to DC-WSNs.

Recently, the **data dissemination problems in DC-WSNs** have started to attract attentions from the research community. Wang and Liu [3] tackled the broadcast scheduling problem in DC-WSNs based on the dynamic-programming approach. Guo *et al.* [4] considered the effect of unreliable links on broad-

casting, and proposed an opportunistic forwarding scheme to reduce the broadcast delay and redundancy in DC-WSNs. Hong *et al.* [6] studied the Minimum-Transmission Broadcasting (MTB) problem in DC-WSNs. They adopted a **restricted duty-cycling model**, where only one active time slot exists in a working period of each node. They proposed a centralized algorithm with approximation ratio of $3(\ln \Delta + 1)$ and a distributed algorithm with approximation ratio of 20. Note that under their particular duty-cycling model, any node can determine its optimal transmission schedule in polynomial-time for transmitting a data packet to a set of neighboring nodes. Therefore, **their methods cannot be adapted to our case**. Xiong *et al.* [7] adopted the same duty-cycling model as that in [6], and studied the load balancing problem for data dissemination in DC-WSNs. They proposed several scheduling algorithms for data transmissions such that the maximum workload of the forwarding nodes is minimized. However, they assumed that the routing paths are pre-computed, and their concern is load balancing instead of energy-efficiency.

To the best of our knowledge, the only work which studied the MEM problem in DC-WSNs is [5]. In [5], the authors adopted another **restricted duty-cycling model** in which the active time slots of any node must be consecutive, and proposed two optimal algorithms (“oCast” and “DB-oCast”) for the minimum-energy multicasting problem in DC-WSNs. As the energy cost for receiving data was neglected in [5], the algorithms proposed in [5] actually minimize the transmission redundancy in a multicast session. Furthermore, as oCast and DB-oCast were both based on dynamic programming, they only work for multicast sessions with a very small number of destination nodes. Although translating their problem into the DST problem may lead to approximation algorithms [17], the resulting approximation ratio (depending on the number of multicast destinations) is far from satisfactory.

B. Our Contributions

In this paper, we study the MEM problem in DC-WSNs using a **generic duty-cycling model**, where each wireless node determines its active/dormant schedule without any constraints. We formulate the MEM problem for DC-WSNs and prove its NP-hardness. We propose an approximation algorithm with guaranteed performance ratio, as well as a distributed implementation of our algorithm. The contributions of our work can be summarized as follows:

- 1) We formulate the Minimum-Energy Multicast Tree Construction and Scheduling (MEMTCS) problem and prove its NP-hardness. We also prove that, unless $NP \subseteq DTIME(n^{O(\log \log n)})$, the MEMTCS problem cannot be approximated with a performance ratio of $(1 - o(1)) \ln \Delta$, where Δ is the maximum node degree of the network graph under consideration.
- 2) We propose a polynomial-time approximation algorithm for the MEMTCS problem with an approximation ratio of $12\rho H(\Delta + 1) + 4\rho$, where $H(\cdot)$ is the harmonic number and ρ is the approximation ratio of a given algorithm for the Minimum Steiner Tree (MST) problem.

- 3) We present a distributed implementation of the proposed algorithm, and we conduct extensive simulations to evaluate the performance of our algorithm. The simulation results demonstrate that our algorithm significantly outperform other known algorithms in terms of both the total energy cost and the transmission redundancy.
- 4) To the best of our knowledge, we are the first to present polynomial-time approximation algorithm with provable approximation ratio for the MEM problem in **generic DC-WSNs**. Moreover, as the Minimum-Transmission Multicasting/Broadcasting problems can be seen as special cases of our problem, we also provide the first approximation algorithms with provable approximation ratios for these problems in DC-WSNs under a generic duty-cycling model.

The rest of our paper is organized as follows. In Section II, we introduce the wireless network model and formulate the MEMTCS problem. In Section III, we analyze the complexity of MEMTCS, and we propose an approximation algorithm for it. A distributed implementation of the proposed algorithm is presented in Section IV. In Section V, we evaluate the performance of the proposed algorithm by simulations. Section VI concludes the paper. In order to maintain fluency, we postpone the proofs to the appendix.

II. ASSUMPTIONS AND DEFINITIONS

In this section, we first describe our network model and related parameters, then we present the formulation of the MEMTCS problem that we tackle in this paper.

A. Network Model and Parameters

A wireless sensor network is modeled by an undirected graph $G = (V, E)$, where V is the set of wireless nodes, and E is the set of links. The nodes in V are distributed in a two-dimensional plane and each node is equipped with an omni-directional antenna. We assume that all nodes have the same fixed transmission power, and there exists a link between two nodes if they are within the transmission range of each other. We also assume that each node has a unique ID and knows the IDs of its one-hop neighbors.

We assume that time is divided into equal-length slots, and each time slot is long enough for sending or receiving a data packet. Without loss of generality, we assume that the working schedule of each node is periodic, and the *working period* of any node has K time slots. To save energy, every node switches between active and dormant states, and we denote by $\Gamma(u)$ the set of active time slots in the working period of node u ($\Gamma(u) \subseteq \{1, 2, \dots, K\}$ and $\Gamma(u) \neq \emptyset$). The active/dormant schedule is independently determined by each node without any constraints. Note that our duty-cycling model is similar with the model used in [1], [4], and the duty-cycling models used in [5]–[7] can be considered as special cases of our model. Following a very common setting in DC-WSNs, we assume that **a node can wake up its transceiver to transmit a packet at any time slot, but can only receive a packet when it is active**. We also assume that the time

synchronization is achieved in network, and each node knows the active/dormant schedule of its neighboring nodes. These assumptions are similar to those of [1], [3]–[7].

We denote by e_s the energy cost for sending a data packet by any node, and denote by e_r the energy cost for receiving a data packet. It is well known that a wireless sensor node has different power consumption levels at different working states such as transmitting, receiving, and idle-listening. Let e_{tx} , e_{rx} , and e_{il} be the energy consumptions of radio for transmitting, receiving and idle-listening, respectively. Usually, e_{tx} is larger than e_{rx} , whereas e_{il} is only slightly smaller than e_{rx} . Since a node has to consume at least e_{il} when it is active, we set e_s to be the value of $e_{tx} - e_{il}$, and set e_r to be the value of $e_{rx} - e_{il}$. Clearly, $e_s \geq e_r \geq 0$.

For the convenience of description, we clarify some other notations here. For any node u , we denote by $nb_G(u)$ the set of neighboring nodes of u in G . Suppose that T is an arbitrary tree. We denote by $N(T)$ the set of nodes in T . Denote by $E(T)$ the set of edges in T . Denote by $d^1(T)$ the set of nodes in T with degree one. Denote by $d^+(T)$ the set of nodes in T with degree greater than one. If T is a rooted tree, then we denote by $nl(T)$ the set of non-leaf nodes in T , and denote by $child(u, T)$ the set of child nodes of node u in T . Suppose that r is the root node of T . Clearly, if $r \in d^1(T)$, then $nl(T) = d^+(T) \cup \{r\}$; otherwise $nl(T) = d^+(T)$.

B. The MEMTCS Problem

In a multicast session, there exists a *terminal set* $M \subseteq V$ and a source node $s \in M$, such that the data sent by s should be received by all the nodes in $M - \{s\}$. A multicast tree T is a sub-tree of G which is rooted at s , and each terminal node in M is a tree node of T .

Since all nodes have the same transmission power, the energy cost for sending a data packet using a multicast tree T can be determined by the number of total transmissions of the nodes in $nl(T)$. In DC-WSNs, the time slot at which a node transmits the packet decides which neighbors can receive it. Therefore, the transmission schedule of each forwarding node plays a key role on the energy cost for multicasting. Furthermore, incorrect schedules can even prevent the destination nodes from receiving a data packet.

For example, consider the wireless network shown in Fig. 1(a). Suppose that the terminal set is $\{n_1, n_6, n_7, n_8\}$, and node n_1 is the source node. The set of numbers associated with each node indicates the active time slots in the working period of that node. Fig. 1(b) and (c) are two multicast trees T_1 and T_2 , respectively. If node n_2 in T_1 transmits at time slot 3 or 4, then only one child node (n_6 or n_4) can receive the data. Therefore, a correct transmission schedule of node n_2 for T_1 must be a set of time slots which contains $\{3, 4\}$.

From this observation, we introduce the concept of “feasible schedule”. Actually, finding a feasible schedule of any non-leaf node in a multicast tree equals to finding a “hitting set” [18]. We clarify this by Definition 1 and Definition 2.

Definition 1 (Hitting Set [18]): Given a collection \mathcal{C} of subsets of a finite set \mathcal{F} , a *hitting set* is a subset $\mathcal{F}' \subseteq \mathcal{F}$

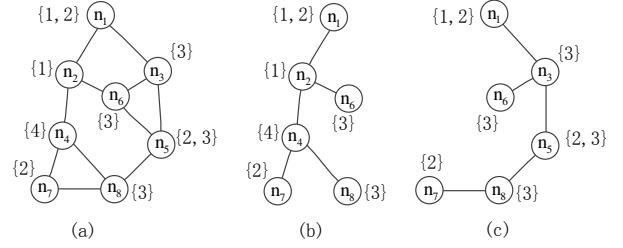


Fig. 1. A DC-WSN graph G (a) and two multicast trees T_1 (b) and T_2 (c).

such that \mathcal{F}' contains at least one element from each subset in \mathcal{C} .

Definition 2 (Feasible Schedule): Given a multicast tree T in G , a function $B : nl(T) \rightarrow 2^{\{1, 2, \dots, K\}}$ is called a *feasible schedule* for T if and only if for any $u \in nl(T)$, $B(u)$ is a hitting set of the collection $\{\Gamma(v) | v \in child(u, T)\}$.

According to Definition 2, the energy cost for sending a data packet using a feasible schedule B on multicast tree T can be written as $\sum_{u \in nl(T)} |B(u)| \cdot e_s$. This is only part of the total energy cost in a multicast session, because the energy consumption for receiving data also needs to be taken into account. In our case, since each forwarding node u in $nl(T)$ knows the duty-cycling schedules of all its neighboring nodes, it can always transmit at the time slots in $B(u)$ such that each node in $child(u, T)$ receives the same data packet only once. Therefore, the energy cost for receiving a data packet using multicast tree T can be written as $(|N(T)| - 1) \cdot e_r$.

Based on the above discussions, we introduce the Minimum-Energy Multicasting Tree Construction and Scheduling (MEMTCS) problem in Definition 3:

Definition 3 (MEMTCS): Given a WSN G , a terminal set $M \subseteq V$, and a source node $s \in M$, the MEMTCS problem seeks a 2-tuple $\langle T_{opt}, B_{opt} \rangle$ in which T_{opt} is a multicast tree rooted at s and B_{opt} is a feasible schedule for T_{opt} , such that the total energy cost

$$\Pi(T_{opt}, B_{opt}) = \sum_{u \in nl(T_{opt})} |B_{opt}(u)| \cdot e_s + (|N(T_{opt})| - 1) \cdot e_r$$

is minimized.

For example, we can find two feasible schedules B_1 and B_2 for T_1 and T_2 in Fig. 1, respectively, such that $B_1(n_1) = \{1\}$, $B_1(n_2) = \{3, 4\}$, $B_1(n_4) = \{2, 3\}$, and $B_2(n_1) = B_2(n_3) = B_2(n_5) = \{3\}$, $B_2(n_8) = \{2\}$. Suppose that $e_s = 10$ and $e_r = 2$. We can get $\Pi(T_1, B_1) = 60$ and $\Pi(T_2, B_2) = 50$. Actually, $\langle T_2, B_2 \rangle$ is the optimal solution for n_1 to send data to $\{n_6, n_7, n_8\}$ in Fig. 1.

III. SOLVING THE MEMTCS PROBLEM

We first briefly evaluate the hardness of MEMTCS. We prove it is NP-hard by a reduction from the Minimum Hitting Set (MHS) problem [18], and we claim this in Theorem 1:

Theorem 1: The MEMTCS problem is NP-hard.

The MHS problem was proved to be equivalent to the Minimum Set Cover (MSC) problem [18], [19]. Moreover, Fiege [20] has proved that, unless NP has quasi-polynomial time

algorithms, there does not exist a polynomial-time algorithm for the MSC problem with performance ratio of $(1 - o(1)) \ln n$, where n is the size of the MSC problem. Therefore, with the proof of Theorem 1, we can easily get:

Corollary 1: Unless $NP \subseteq DTIME(n^{O(\log \log n)})$, there does not exist a polynomial-time approximation algorithm with performance ratio of $(1 - o(1)) \ln \Delta$ for the MEMTCS problem, where Δ is the maximum node degree of network graph under consideration.

Next, we propose an approximation algorithm for the MEMTCS problem. We first provide a brief overview of our algorithm in Section III-A, then describe our methods in details in Section III-B–III-E.

A. An Overview of the Proposed Algorithm

Our approximation algorithm consists of several steps. Firstly, we use a graph transformation method to transform the original network graph G into an *extended graph* \tilde{G} where the possible transmitting time slots of the nodes in G are represented as *satellite nodes*, and the nodes in \tilde{G} are connected in a particular way to facilitate the design of our approximation algorithm. Secondly, we propose the concept of *Minimum Satellite Bridge* (MSB) in \tilde{G} as well as an algorithm for finding an approximation MSB. The MSB is actually a special tree in \tilde{G} whose nodes can cover all the nodes in M . Finally, we map the approximation for MSB to a multicast tree in G and a feasible schedule for the multicast tree, which serves as an approximate solution to the MEMTCS problem.

To find the approximation ratio of our algorithm, we propose another concept *Minimum Isotropic Scattering Tree* (MIST), which is a special multicast tree T_I in G spanning the nodes in M . We prove that T_I serves as a quantitative “bridge” between the number of nodes in a MSB and $\langle T_{\text{opt}}, B_{\text{opt}} \rangle$. As a result, we obtain the approximation ratio of our algorithm.

B. Graph Transformation

The first step of our approach is to transform the original network graph into an *extended graph*. We introduce the concept of the extended graph in Definition 4:

Definition 4 (Extended Graph): The *extended graph* of G is an undirected graph $\tilde{G} = (\tilde{V}, \tilde{E})$, where \tilde{V} is the set of nodes and \tilde{E} is the set of edges. The nodes and edges in \tilde{G} are created by the following steps:

- (i) Initially, $\tilde{V} = V$ and $\tilde{E} = \emptyset$;
- (ii) For each node $u \in V$ and each time slot $i \in \bigcup_{v \in \text{nb}_G(u)} \Gamma(v)$, create a new node $\lambda(u, i)$ in \tilde{V} . The node $\lambda(u, i)$ is called a *satellite node* of u on slot i , and u is called a *nuclear node* of $\lambda(u, i)$. The set of all satellite nodes of u is denoted by $\Psi(u)$;
- (iii) For each node $u \in V$, create an undirected edge between each pair of nodes in $\Psi(u) \cup \{u\}$. In other words, the sub-graph induced by $\Psi(u) \cup \{u\}$ is a complete graph;
- (iv) For each edge $(u, v) \in E$, each time slot $i \in \Gamma(v)$ and each time slot $j \in \Gamma(u)$, create three undirected edges $(\lambda(u, i), \lambda(v, j))$, $(\lambda(u, i), v)$ and $(u, \lambda(v, j))$.

From Definition 4, we can see that \tilde{V} can be partitioned into two disjoint subsets: V and V_S , where V is the set of all nuclear nodes, and V_S is the set of all satellite nodes. A nuclear node may have multiple satellite nodes, but any satellite node only has one nuclear node. An example of the extended graph is shown in Fig.2.

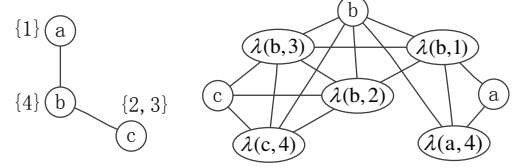


Fig. 2. A DC-WSN graph G (left) and its extended graph \tilde{G} (right).

According to the construction rules in Definition 4, we can obtain some useful properties of \tilde{G} , as described in Lemma 1 and Lemma 2. Their proofs are omitted due to page limits.

Lemma 1: There are at most $(K + 1)|V|$ nodes and $\binom{K + 1}{2} |V| + 3K^2 |E|$ edges in \tilde{G} .

Lemma 2: Any two nuclear nodes cannot be adjacent in \tilde{G} , and any satellite node in V_S can be adjacent to at most $\Delta + 1$ nuclear nodes in \tilde{G} .

C. Minimum Satellite Bridge

We first introduce the concept of *Minimum Satellite Bridge* (MSB) in Definition 5, and then we propose an approximate algorithm for finding a MSB. As we will see later, finding a MSB is an important building block of our algorithm for solving the MEMTCS problem.

Definition 5 (Minimum Satellite Bridge): Given $\tilde{G} = (\tilde{V}, \tilde{E})$ and the terminal set $M \subseteq V$, a *Satellite Bridge* SB is a sub-tree of \tilde{G} which satisfies:

- 1) The nodes in SB are all satellite nodes;
- 2) Each node in the terminal set M is adjacent to at least one node in SB .

The *Minimum Satellite Bridge* (MSB) SB^* is an SB with a minimum number of nodes.

Next, we propose an approximation algorithm with a performance ratio of $O(H(\Delta + 1))$ for finding a MSB, as shown in Algorithm 1. This algorithm consists of two stages. The first stage is lines 1-6, and the second stage is line 7. In the first stage, we use a greedy set cover algorithm to find a small satellite node set C that can cover all the nodes in M . In each loop, we first find a node v which has the maximum number of adjacent nodes in the uncovered node set UC (line 3). Then we add v into C (line 4) and update UC (line 5). In the second stage (line 7), an approximate Steiner tree algorithm is applied upon C .

Since each satellite node can cover at most $\Delta + 1$ nodes in M (according to Lemma 2), the first stage of Algorithm 1 has an approximation ratio of $H(\Delta + 1)$ [21]. The approximation ratio of the second stage is a constant ρ . By taking advantage of the special structure of the extended graph, we can get:

Algorithm 1 Finding an approximate MSB

Input: The extended graph \tilde{G} and the terminal set M .

Output: An approximate MSB \overline{SB} .

- 1: $C = \emptyset, UC = M$;
 - 2: **while** $UC \neq \emptyset$ **do**
 - 3: $v = \arg \max_{u \in V_S - C} |nb_{\tilde{G}}(u) \cap UC|$;
 - 4: $C = C \cup \{v\}$;
 - 5: $UC = UC - (nb_{\tilde{G}}(v) \cap UC)$;
 - 6: **end while**
 - 7: Let \tilde{G}_S be the sub-graph of \tilde{G} induced by V_S . Assign each edge in \tilde{G}_S a weight of 1. Compute an approximate minimum Steiner tree ST in \tilde{G}_S which connects the nodes in C ;
 - 8: **return** $\overline{SB} = ST$.
-

Theorem 2: The approximation ratio of Algorithm 1 is $3\rho H(\Delta + 1) + \rho$, where $H(\cdot)$ is the harmonic number and ρ is the approximation ratio of the algorithm used in Algorithm 1 for finding a minimum Steiner tree.

D. Minimum Isotropic Scattering Tree

Now we show that an MSB can be mapped to a special multicast tree in G called the *Minimum Isotropic Scattering Tree* (MIST) and a feasible schedule for the internal nodes in the MIST. Although MSB does not produce an optimal solution to the MEMTCS problem, we show that it leads to an approximation for MEMTCS, and we use MIST as a medium to derive the approximation ratio. We first introduce the concept of MIST in Definition 6:

Definition 6 (Minimum Isotropic Scattering Tree): For any tree T in G and any node $u \in d^+(T)$, we denote by $\Upsilon(u, T)$ a minimum hitting set of the collection $\{\Gamma(v) | v \in nb_T(u)\}$, and define $\Xi(T) = \sum_{u \in d^+(T)} |\Upsilon(u, T)|$. The *Minimum Isotropic Scattering Tree* (MIST) T_I is a tree in G such that T_I spans M and that $\Xi(T_I)$ is minimized.

We claim that $\Xi(T_I)$ actually equals to the number of nodes in a minimum satellite bridge, which is proved by Lemma 3, Lemma 4, and Theorem 3. Lemma 3 claims that the number of nodes in a MSB is no more than $\Xi(T_I)$, whereas Lemma 4 implies that $\Xi(T_I)$ is no more than the number of nodes in a MSB. With these two lemmas, Theorem 3 can be readily proved.

Lemma 3: $|N(SB^*)| \leq \Xi(T_I)$.

Lemma 4: Any satellite bridge SB can be mapped to a 2-tuple $\langle R, F \rangle$, where R is a tree in G spanning the nodes in M , and F is a function that satisfies:

- 1) For any $u \in d^+(R)$, $F(u)$ is a hitting set of the collection $\{\Gamma(v) | v \in nb_R(u)\}$;
- 2) $\sum_{u \in d^+(R)} |F(u)| \leq |N(SB)|$.

Theorem 3: $\Xi(T_I) = |N(SB^*)|$

The proofs take advantage of the special structure of the extended graph. In addition, the proof of Lemma 4 actually provides a method of mapping any satellite bridge to a multicast tree and a feasible schedule for the internal nodes in

the multicast tree. This mapping can be roughly described as follows. According to the construction rules of the extended graph, the satellite nodes in a satellite bridge can be mapped to the nuclear nodes that they belong to, as well as the transmitting time slots on these nuclear nodes. Furthermore, we can find a multicast tree spanning these nuclear nodes and the destination nodes in M , and the internal nodes in the multicast tree are all the mapped nuclear nodes. According to the special node-connecting method of the extended graph, the mapped transmitting time slots of any internal node in the multicast tree can cover all its neighboring nodes in the tree.

E. Approximation Algorithm for MEMTCS

Based on the methods introduced by the previous sections, we propose our algorithm for the MEMTCS problem, as shown in Algorithm 2.

Algorithm 2 Approximation for MEMTCS

Input: A DC-WSN G , a terminal set M , and a source node $s \in M$.

Output: A multicast tree \bar{T} and a feasible schedule \bar{B} .

- 1: Construct the extended graph $\tilde{G} = (\tilde{V}, \tilde{E})$ of G ;
 - 2: Use Algorithm 1 to compute an approximate minimum satellite bridge \overline{SB} ;
 - 3: Use the method in the proof of Lemma 4 to map \overline{SB} to a 2-tuple $\langle \hat{T}, \hat{F} \rangle$. Let \bar{T} be the rooted tree got by designating s as the root of \hat{T} ;
 - 4: **for** each node $u \in nl(\bar{T})$ **do**
 - 5: **if** $u \in d^+(\bar{T})$ **then**
 - 6: $\bar{B}(u) = \hat{F}(u)$;
 - 7: **else**
 - 8: Let v be u 's child node in \bar{T} . Find an arbitrary
 - 9: $i \in \Gamma(v)$ and let $\bar{B}(u) = \{i\}$;
 - 10: **end if**
 - 11: **end for**
 - 12: **return** $\langle \bar{T}, \bar{B} \rangle$.
-

The output of Algorithm 2 is a 2-tuple $\langle \bar{T}, \bar{B} \rangle$. From Lemma 4, It is easy to know that \bar{T} is a multicast tree spanning the nodes in M and \bar{B} is a feasible schedule for \bar{T} . Next, we prove the approximation ratio of Algorithm 2 by Lemma 5, Lemma 6 and Theorem 4. Lemma 5 is actually based on a special property of the MHS problem, i.e, if we add a subset in an instance of the MHS problem, then the cardinality of the result minimum hitting set will increase at most 1. Using Lemma 5, Lemma 6 finds out a quantitative relationship between $\Xi(T_I)$ and $\langle T_{\text{opt}}, B_{\text{opt}} \rangle$, which is used in the proof of Theorem 4.

Lemma 5: For any node $u \in d^+(T_{\text{opt}})$, we have

$$|\Upsilon(u, T_{\text{opt}})| \leq |B_{\text{opt}}(u)| + 1.$$

Lemma 6: $\langle T_{\text{opt}}, B_{\text{opt}} \rangle$ is related to T_I by

$$\sum_{u \in nl(T_{\text{opt}})} |B_{\text{opt}}(u)| \geq \Xi(T_I) - |d^+(T_{\text{opt}})| + 1.$$

Theorem 4: The approximation ratio of Algorithm 2 is $12\rho H(\Delta + 1) + 4\rho$.

The dominating running time of Algorithm 2 is the time on constructing \overline{SB} in line 2, using Algorithm 1. Lines 1-6 in Algorithm 1 can be implemented in $\mathcal{O}(|\tilde{V}|^2)$ time. If we use the 2-approximation algorithm proposed in [22] to compute an approximate minimum Steiner tree in line 7, the resulting time complexity is $\mathcal{O}(|\tilde{V}| \log |\tilde{V}| + |\tilde{E}|)$. Given K as a predefined constant, the time complexity of Algorithm 2 is $\mathcal{O}(|V|^2 + |E|)$, and the approximation ratio is $24H(\Delta + 1) + 8$.

IV. DISTRIBUTED IMPLEMENTATION

In this section, we provide a distributed implementation of Algorithm 2 for the MEMTCS problem. Note that the main operation of Algorithm 2 is line 2, in which Algorithm 1 is called to compute an approximate minimum satellite bridge. Therefore, we first propose the distributed implementation of Algorithm 1.

As we have described in Section III-C, Algorithm 1 consists of two stages: the first stage is lines 1-6, in which a greedy strategy is used to find a small satellite node set C covering the nodes in M , and the second stage is line 7, in which an approximate minimum Steiner tree is computed. The first stage of Algorithm 1 can be decentralized in a way similar with the distributed dominating set algorithm in [23], which is shown in Algorithm 3.

Algorithm 3 Distributed implementation of the first stage of Algorithm 1

- 1: Each white node u with non-empty $rn timer(u)$ broadcasts a message (“election”, $|rn timer(u)|$, $u.ID$);
 - 2: Each red node checks all the “election” messages it receives, and finds a node v whose value of $|rn timer(v)|$ is the maximum (break tie by choosing the node with largest ID). Then it sends v a message “you win”;
 - 3: If a white node u receives “you win” messages from all nodes in $rn timer(u)$, then it colors itself blue, and broadcasts a message “I am dominator”;
 - 4: If a red node receives an “I am dominator” message, then it colors itself green, and broadcasts a message “I am dominated”;
 - 5: If a white node u receives a “I am dominated” message from a neighboring node w , then it deletes w from $rn timer(u)$.
-

In Algorithm 3, each node in \tilde{G} is colored red, green, white, or blue. The red nodes are the nodes in M which are not covered yet. The green nodes are the nodes in M which are already covered by some blue nodes. The blue nodes are the nodes which are selected into the resulting node set C , and the white nodes are the nodes which are not selected. Initially, each node in M is colored red, and all other nodes are colored white. Besides, each white node u owns a set $rn timer(u)$, which is initialized to be the set of IDs of all u ’s red neighbors.

It can be seen that Algorithm 3 is a faithful implementation of the greedy strategy in lines 1-6 of Algorithm 1, so the approximation ratio of Algorithm 3 is $H(\Delta + 1)$. There are at most $|M|$ rounds in Algorithm 3, because it repeats until no

red nodes exist, and at least one red node turns green in each round. Therefore, the message complexity of Algorithm 3 is $\mathcal{O}(|M| \cdot |\tilde{V}|) = \mathcal{O}(|M| \cdot |V|)$.

The second stage of Algorithm 1 can be decentralized by using a distributed Steiner tree algorithm in the literature [24]–[26]. If we adopt the 2-approximation distributed algorithm proposed in [24], then the message complexity is $\mathcal{O}(|M| \cdot |V|)$ and the time complexity is $\mathcal{O}(|M| \cdot D)$, where D is the diameter of G .

The distributed implementation of Algorithm 2 (except line 2) is trivial: an arbitrary spanning tree of the sub-graph induced by $N(\overline{SB})$ needs to be found in line 3. To accomplish this, the distributed *Depth-First Search* (DFS) algorithm proposed by Makki *et al.* [27] can be applied. The time complexity and message complexity of the distributed DFS algorithm are both $\mathcal{O}(|V|)$.

Based on these discussions, we can get Corollary 2:

Corollary 2: There exists a distributed algorithm for MEMTCS. It has an approximation ratio of $24H(\Delta + 1) + 8$. The time complexity and message complexity of the distributed algorithm are $\mathcal{O}(D \cdot |V|)$ and $\mathcal{O}(|M| \cdot |V|)$, respectively, where D is the diameter of G .

Though our distributed implementation for Algorithm 2 is based on the extended graph \tilde{G} , it can be easily adapted to run on G . According to the construction method of \tilde{G} , any satellite node in V_S can be seen as a local “pseudo node” administrated by its nuclear node in V . Therefore, each nuclear node can send messages for its satellite nodes, and do the computation that its satellite nodes need to do. Let $\tilde{\mathcal{A}}$ be the distributed algorithm running on \tilde{G} , and let \mathcal{A} be the adapted version of $\tilde{\mathcal{A}}$ running on G . If several satellite nodes of the same nuclear node u send their messages simultaneously to other nuclear nodes in $\tilde{\mathcal{A}}$, then u in \mathcal{A} can send these messages at different time slots. Note that any nuclear node has at most K satellite nodes and K is a predefined constant. Therefore, if we define a constant $\delta (\delta > K)$, then any nuclear node u in \mathcal{A} can receive messages for all its satellite nodes after waiting δ time, and then u can do the computation for its satellite nodes based on the messages it receives. The message passing processes between the satellite nodes administrated by the same nuclear node in $\tilde{\mathcal{A}}$ become local computations of that nuclear node in \mathcal{A} , and no new messages are generated in \mathcal{A} . Therefore, \mathcal{A} has the same time and message complexity as $\tilde{\mathcal{A}}$.

We should point out that our distributed algorithm requires all nodes to be active (i.e., no duty-cycling). This is not a major problem, as the distributed algorithm is meant to construct and schedule the minimum energy multicast tree and is thus needed only upon changes in the network topology.

V. PERFORMANCE EVALUATION

In this section we evaluate the performance of our algorithm via simulations. The simulations focus on the effect of various network conditions on the performance of various multicasting algorithms. In the simulations, we deploy wireless nodes randomly in a $1000m \times 1000m$ square, and the transmission range of each node is set to 300 meters. Each node randomly

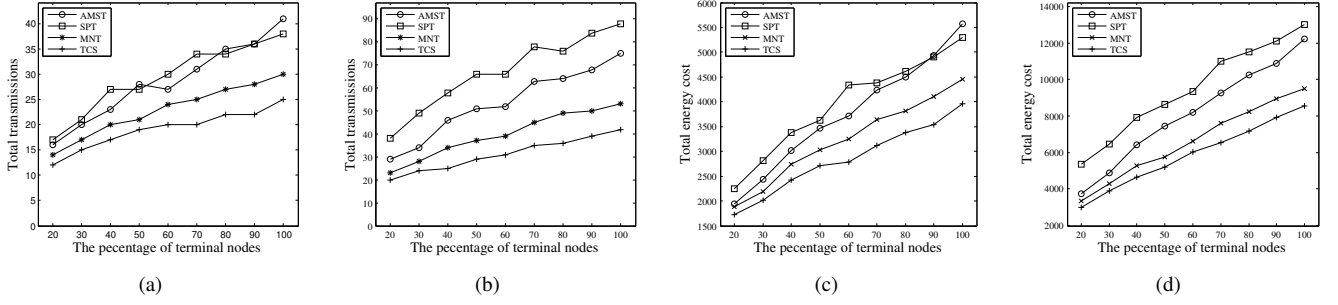


Fig. 3. Performance evaluation of various multicasting algorithms. The percentage of terminal nodes scales from 20% to 100%. (a) $|V|=100$, (b) $|V|=300$, (c) $|V|=100$, (d) $|V|=300$.

picks some time slots in the working period as its active time slots. Without loss of generality, the energy cost for sending and receiving a data packet by any node is set to 100 and 15, respectively.

To the best of our knowledge, there is no polynomial-time minimum-energy multicasting algorithms designed for DC-WSNs under a generic duty-cycling model. So we have to compare our algorithms with several traditional multicasting algorithms, including the *Shortest Path Tree* (SPT) algorithm, the *Approximate Minimum Steiner Tree* (AMST) algorithm, and the minimal data overhead tree (the MNT algorithm) proposed by [28], [29]. The SPT algorithm computes shortest paths from the source node to the receiver nodes, and aggregates these shortest paths to construct a multicast tree. The AMST algorithm computes an approximate minimum Steiner tree spanning all the nodes in the terminal set M . Here, we adopt the AMST algorithm provided by Kou *et al.* [30], which was also used by Liang *et al.* [16] to solve the minimum-energy all-to-all multicasting problem in always-active wireless ad-hoc networks. The MNT algorithm was designed for reducing the multicast redundancy in static wireless ad-hoc networks. The work in [28] and [29] has proved that MNT can reduce the redundant transmissions in a multicast session more effectively than other heuristics.¹

To use the multicast trees constructed by the SPT, AMST and MNT algorithms in a DC-WSN environment, we need to find a transmission schedule for each forwarding node in the multicast trees. Obviously, the most energy-efficient transmission schedule for any forwarding node u in a multicast tree T is the minimum hitting set of the collection $\{\Gamma(v) | v \in \text{child}(u, T)\}$. However, finding a minimum hitting set is a NP-hard problem. Therefore, we use a greedy hitting set algorithm [31] to find the transmission schedules of the forwarding nodes. Since the greedy algorithm is essentially the best-possible polynomial-time approximation algorithm for the minimum hitting set problem (unless $NP \subseteq DTIME(n^{O(\log \log n)})$), each forwarding node in the multicast trees generated by the SPT, AMST and MNT algorithms has the best-possible energy-efficient transmission schedule.

In the simulations, we compare our algorithms with the traditional ones using two metrics including the total energy cost and the transmission redundancy. Note that minimizing the transmission redundancy is an important optimization objective both in multicasting and in broadcasting [3]–[6]. However, to the best of our knowledge, there is no other work that provides polynomial-time algorithms with guaranteed approximation ratios for minimizing the transmission redundancy of multicasting/broadcasting in DC-WSNs under a generic duty-cycling model.

In Fig. 3, we compare Algorithm 2 (denoted by TCS) with SPT, AMST and MNT. The length of working period is set to 20, and the percentage of terminal nodes scales from 20% to 100% with an increment of 10%. The algorithm proposed in [30] is adopted in Algorithm 2 for computing an approximate minimum Steiner tree. Fig. 3(a) and (b) plot the number of total transmissions in a multicast session. The number of network nodes is set to 100 and 300, respectively. We can see that MNT outperforms SPT and AMST greatly, because the multicast tree generated by MNT has less forwarding nodes (non-leaf nodes) than the other multicast trees [28]. We also see that TCS significantly outperform the other algorithms, and the transmission redundancy is reduced by about 20% as the percentage of terminal nodes approaches 90%. The reason is that, since the traditional SPT, AMST and MNT algorithms generate multicast trees regardless of the duty cycles of the wireless nodes, they can't optimize the transmission schedules of the forwarding nodes in a global manner. Therefore, although the transmission schedules are optimized locally in the SPT, MST and MNT algorithms using a best-possible optimization algorithm (the greedy algorithm), their transmission redundancies are still high. On the contrary, the TCS algorithm builds the multicast tree and finds the transmission schedules of the forwarding nodes in a holistic manner by taking advantage of the special structure of the extended graph, so the transmission redundancy is reduced more effectively than the traditional algorithms.

Fig. 3(c) and (d) plot the total energy costs of different multicasting algorithms. Again, the numbers of network nodes in Fig.3(c) and (d) are set to 100 and 300, respectively. We can see that the TCS algorithm still outperforms the other algorithms. The explanation is that, compared with the

¹We refrain from comparing with the algorithm proposed in [14] and oCast [5], as they both have a much higher complexity than ours: $O(|M|^2 \cdot |V|^2)$ for the former and exponential in $|M|$ for the latter.

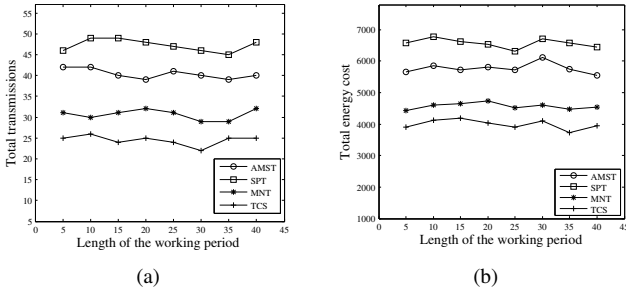


Fig. 4. Performance evaluation of various multicasting algorithms under scenarios with different lengths of the working period. The length of the working period scales from 5 to 40. The number of nodes in V is 150 and the number of nodes in M is 100.

other algorithms, the TCS algorithm significantly reduces the number of total transmissions, and generates a multicast tree with moderate number of nodes. As a result, the total energy cost of the TCS algorithm is lower than the other algorithms when both the energy cost for sending data and the energy cost for receiving data are considered. We also notice that the TCS algorithm performs better when the percentage of destination nodes increases. This is because that more redundant transmissions are reduced by TCS when the multicast tree grows.

In Fig. 4 we study how the length of the working period impacts the performance of our algorithms. The number of network nodes is set to 150, and the number of terminal nodes is set to 100. The length of the working period scales from 5 to 40 with an increment of 5. Fig. 4(a) and (b) plot the number of total transmissions and the total energy cost for multicasting, respectively. It is evident that our algorithm still outperform the other algorithms when the length of the working period changes. Meanwhile, we also see that, for all algorithms compared in Fig. 4(a)-(b), both the transmission redundancy and the total energy cost do not vary much. An explanation is that, since the active time slots of any node are randomly selected from the working period, the number of common active time slots of any two different nodes does not vary much when the length of the working period increases. Therefore, the performance of all these algorithms are rather independent of the length of the working period under our generic duty-cycling model.

VI. CONCLUSION

In this paper, we have studied the Minimum-Energy Multicasting (MEM) problem in Duty-Cycled Wireless Sensor Networks (DC-WSNs). We formalized the Minimum-Energy Multicasting Tree Construction and Scheduling (MEMTCS) problem, and proved its NP-hardness. A lower bound on the approximation ratio of any polynomial-time algorithm for the MEMTCS problem was given in our work. We presented an approximation algorithm with guaranteed approximation ratio for the MEMTCS problem, and proposed a distributed implementation of our algorithm. The simulation results demonstrate that our algorithm outperform other related algorithms in terms of both the total energy cost and the transmission redundancy.

APPENDIX

PROOFS FOR THE MEMTCS PROBLEM

Proof of Theorem 1: Given an instance $(\mathcal{C}, \mathcal{F})$ of the minimum hitting set problem, we create a wireless network graph G by the following method:

Let the elements in \mathcal{F} be $\{f_1, f_2, \dots, f_p\}$, and let the subsets in \mathcal{C} be C_1, C_2, \dots, C_q . For each $C_j (1 \leq j \leq q)$, create a node v_j in G , and let $\Gamma(v_j) = \{i | (1 \leq i \leq p) \wedge (f_i \in C_j)\}$. Create a node x in G , and connect x to each $v_j (1 \leq j \leq q)$.

Let x be the source node and $\{x\} \cup \{v_j | 1 \leq j \leq q\}$ be the terminal set M in multicasting. Let $e_s = 1$ and $e_r = 0$. It is easy to prove that \mathcal{F} has a hitting set for \mathcal{C} of size at most k if and only if G has a multicast tree T and a feasible schedule B for T such that $\Pi(T, B) \leq k$. Therefore, the MEMTCS problem is NP-hard. \square

Proof of Theorem 2: Each node $u \in V_S$ may cover a set $nb_{\tilde{G}}(u) \cap M$, whose cardinality is bounded by $\Delta + 1$ (see Lemma 2). Therefore, lines 1-6 of Algorithm 1 intrinsically represent a greedy algorithm for finding a minimum set cover [21]. Suppose that C^* is a smallest set of satellite nodes covering the nodes in M , we have:

$$|C| \leq H(\Delta + 1)|C^*| \quad (1)$$

According to Definition 5, we know that the nodes in SB^* also cover the nodes in M . Hence

$$|C^*| \leq |N(SB^*)| \quad (2)$$

Let ST^* be a minimum Steiner tree in \tilde{G}_s which connects the nodes in C . We have:

$$|E(\overline{SB})| \leq \rho |E(ST^*)| \quad (3)$$

For any node $c \in C - N(SB^*)$, there must exist a node $m \in M$ adjacent to c . Since SB^* is a satellite bridge, m must be adjacent to a certain node t in SB^* . For convenience, we assume that $c \notin \Psi(m)$ and $t \notin \Psi(m)$ (otherwise the theorem can be proved in a similar way). According to Definition 4(iv), we know that there must exist two nodes $c', t' \in \Psi(m)$ such that c' is adjacent to c , and t' is adjacent to t . Since c' and t' are both satellite nodes of m , if $c' \neq t'$, then c' and t' must be adjacent, according to Definition 4(iii). Therefore, there exists a path from c to t in \tilde{G}_s whose length is no more than 3, as shown in Fig. 5. In other words, there must exist a tree in \tilde{G}_s whose node set contains $N(SB^*) \cup C$, and it has at most

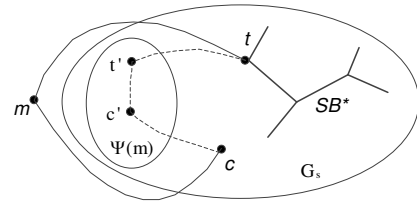


Fig. 5. Connecting the node c in C to SB^*

$|N(SB^*)| + 3|C| - 1$ edges. Since ST^* is a minimum Steiner tree that connects the nodes in C , it follows that:

$$|E(ST^*)| \leq |N(SB^*)| + 3|C| - 1 \quad (4)$$

Finally, with equation (1)–(4), we can get:

$$\begin{aligned}
|N(\overline{SB})| &= |E(\overline{SB})| + 1 \leq \rho|E(ST^*)| + 1 \\
&\leq \rho(|N(SB^*)| + 3|C| - 1) + 1 \\
&\leq \rho(|N(SB^*)| + 3H(\Delta + 1)|C^*| - 1) + 1 \\
&\leq (3\rho H(\Delta + 1) + \rho)|N(SB^*)| - (\rho - 1) \\
&\leq (3\rho H(\Delta + 1) + \rho)|N(SB^*)|,
\end{aligned}$$

hence the approximation ratio $3\rho H(\Delta + 1) + \rho$. \square

Proof of Lemma 3: Let S_1 be the set $\{\lambda(u, i) | u \in d^+(T_I) \wedge i \in \Upsilon(u, T_I)\}$. Clearly, $|S_1| = \Xi(T_I)$. Furthermore, we have the following statements:

- 1) According to Definition 4(iii), we know that for any node $u \in d^+(T_I)$ and any $i_1, i_2 \in \Gamma(u), i_1 \neq i_2$, the edge $(\lambda(u, i_1), \lambda(u, i_2))$ is in \tilde{E} .
- 2) For any two neighboring nodes v_1 and v_2 in $d^+(T_I)$, there must exist $j_1 \in \Upsilon(v_1, T_I) \cap \Gamma(v_2)$ and $j_2 \in \Upsilon(v_2, T_I) \cap \Gamma(v_1)$. According to Definition 4(iv), the edge $(\lambda(v_1, j_1), \lambda(v_2, j_2))$ is in \tilde{E} .
- 3) For any node $m \in M$, if $m \in d^+(T_I)$, then for any $i \in \Upsilon(m, T_I)$, $\lambda(m, i)$ is adjacent to m , according to Definition 4(iii). If $m \in d^1(T_I)$, then there must exist a node $m_1 \in d^+(T_I)$ adjacent to m . Since $\Upsilon(m_1, T_I)$ is a minimum hitting set of the collection $\{\Gamma(v) | v \in nb_{T_I}(m_1)\}$, there must exist $k_1 \in \Upsilon(m_1, T_I) \cap \Gamma(m)$. Therefore, $\lambda(m_1, k_1) \in S_1$. According to Definition 4(iv), we have $(\lambda(m_1, k_1), m) \in \tilde{E}$.

From 1) and 2) it is easy to know that the sub-graph induced by S_1 in \tilde{G} is connected. From 3) we know that every node in M is adjacent to a node in S_1 . Let SB_1 be an arbitrary spanning tree of the sub-graph induced by S_1 . Then SB_1 must be a satellite bridge. Since SB^* is a minimum satellite bridge, we have $|N(SB^*)| \leq |N(SB_1)| = |S_1| = \Xi(T_I)$. \square

Proof of Lemma 4: Suppose that the nodes in SB belong to q nuclear nodes. Therefore, $N(SB)$ can be partitioned into q mutually disjoint subsets: A_1, A_2, \dots, A_q , such that all the nodes in A_i have the same nuclear node a_i , and $a_i \neq a_j$ for any $i \neq j, 1 \leq i, j \leq q$. Let A be the set $\{a_1, a_2, \dots, a_q\}$. According to Definition 5, for any node $u \in M - A$, we can find a node $u' \in N(SB)$ adjacent to u . So we can connect u to SB by adding the edge (u, u') . When all the nodes in $M - A$ are connected to SB , we get a new tree SB' .

Then we consider the graph $G' = (V', E')$ whose node set V' is $M \cup A$ and edge set E' is $E'_1 \cup E'_2$, where $E'_1 = \{(a_i, a_j) | \exists(u, v) \in SB' \wedge u \in A_i \wedge v \in A_j \wedge (1 \leq i \neq j \leq q)\}$ and $E'_2 = \{(a_i, v) | \exists(u, v) \in SB' \wedge u \in A_i \wedge v \in (M - A) \wedge (1 \leq i \leq q)\}$. From Definition 4(iv), it is easy to know that G' is a connected sub-graph of G , and all nodes in $M - A$ are degree-one nodes in G' .

Let R be an arbitrary spanning tree of G' . Clearly, we have $M \subseteq N(R)$ and $d^+(R) \subseteq A$. Let $F(a_i) = \{l | \exists \lambda(a_i, l) \in A_i\}$ for $1 \leq i \leq q$. Clearly, $|F(a_i)| = |A_i|$. For $a_i \in A$ and $w \in nb_R(a_i)$, we have:

- 1) If $(a_i, w) \in E'_1$, then there must exist $a_j \in A, u \in A_i$ and $v \in A_j$ such that $w = a_j, (u, v) \in SB'$ and $i \neq j$.

According to Definition 4(iv), there must exist a time slot $l_1 \in \Gamma(a_j)$ such that $u = \lambda(a_i, l_1)$. Note that l_1 is also in $F(a_i)$. Therefore, $F(a_i) \cap \Gamma(w) = F(a_i) \cap \Gamma(a_j) \neq \emptyset$.

- 2) If $(a_i, w) \in E'_2$, then there must exist $u \in A_i$ adjacent to w . By reasoning similar to 1), we also have $F(a_i) \cap \Gamma(w) \neq \emptyset$.

From 1) and 2), we know that any $F(a_i), 1 \leq i \leq q$ is a hitting set of the collection $\{\Gamma(v) | v \in nb_R(a_i)\}$. Since $d^+(R) \subseteq A$, we have $\sum_{u \in d^+(R)} |F(u)| \leq \sum_{u \in A} |F(u)| = \sum_{1 \leq i \leq q} |F(a_i)| = \sum_{1 \leq i \leq q} |A_i| = |N(SB)|$. \square

Proof of Theorem 3: From Lemma 4 we know that we can find a tree R^* in G spanning the nodes in M and a function F^* such that:

- 1) For any $u \in d^+(R^*)$, $F^*(u)$ is a hitting set of the collection $\{\Gamma(v) | v \in nb_{R^*}(u)\}$;
- 2) $\sum_{u \in d^+(R^*)} |F^*(u)| \leq |N(SB^*)|$.

Clearly, we also have $\sum_{u \in d^+(R^*)} |\Upsilon(u, R^*)| \leq \sum_{u \in d^+(R^*)} |F^*(u)|$, and $\Xi(T_I) \leq \sum_{u \in d^+(R^*)} |\Upsilon(u, R^*)|$. Therefore, $\Xi(T_I) \leq \sum_{u \in d^+(R^*)} |F^*(u)| \leq |N(SB^*)|$. From Lemma 3 we know $|N(SB^*)| \leq \Xi(T_I)$. Therefore, $\Xi(T_I) = |N(SB^*)|$. \square

Proof of Lemma 5: From Definition 3 we know that $B_{\text{opt}}(u)$ must be a minimum hitting set of the collection $\{\Gamma(v) | v \in \text{child}(u, T_{\text{opt}})\}$. If u is the root of T_{opt} , $\text{child}(u, T_{\text{opt}}) = nb_{T_{\text{opt}}}(u)$. Therefore, we have $|\Upsilon(u, T_{\text{opt}})| = |B_{\text{opt}}(u)| \leq |B_{\text{opt}}(u)| + 1$.

Otherwise, let u' be the parent node of u in T_{opt} . Find an arbitrary $j \in \Gamma(u')$, then $B_{\text{opt}}(u) \cup \{j\}$ must be a hitting set of the collection $\{\Gamma(v) | v \in nb_{T_{\text{opt}}}(u)\}$. So we still have $|\Upsilon(u, T_{\text{opt}})| \leq |B_{\text{opt}}(u) \cup \{j\}| \leq |B_{\text{opt}}(u)| + 1$. \square

Proof of Lemma 6: If $s \in d^1(T_{\text{opt}})$, then $|B_{\text{opt}}(s)| = 1$. With Lemma 5, we have:

$$\begin{aligned}
\sum_{u \in nl(T_{\text{opt}})} |B_{\text{opt}}(u)| &= |B_{\text{opt}}(s)| + \sum_{u \in d^+(T_{\text{opt}})} |B_{\text{opt}}(u)| \\
&\geq 1 + \sum_{u \in d^+(T_{\text{opt}})} (|\Upsilon(u, T_{\text{opt}})| - 1) \\
&= \Xi(T_{\text{opt}}) - |d^+(T_{\text{opt}})| + 1
\end{aligned}$$

If $s \in d^+(T_{\text{opt}})$, then $|B_{\text{opt}}(s)| = |\Upsilon(s, T_{\text{opt}})|$. Therefore, we still have:

$$\begin{aligned}
\sum_{u \in nl(T_{\text{opt}})} |B_{\text{opt}}(u)| &= |B_{\text{opt}}(s)| + \sum_{u \in d^+(T_{\text{opt}}) \setminus \{s\}} |B_{\text{opt}}(u)| \\
&\geq |\Upsilon(s, T_{\text{opt}})| + \sum_{u \in d^+(T_{\text{opt}}) \setminus \{s\}} (|\Upsilon(u, T_{\text{opt}})| - 1) \\
&= \sum_{u \in d^+(T_{\text{opt}})} |\Upsilon(u, T_{\text{opt}})| - (|d^+(T_{\text{opt}})| - 1) \\
&= \Xi(T_{\text{opt}}) - |d^+(T_{\text{opt}})| + 1
\end{aligned}$$

Now $\sum_{u \in nl(T_{\text{opt}})} |B_{\text{opt}}(u)| \geq \Xi(T_I) - |d^+(T_{\text{opt}})| + 1$ follows from the fact that $\Xi(T_{\text{opt}}) \geq \Xi(T_I)$. \square

Proof of Theorem 4: Using Lemma 6, we have:

$$\begin{aligned}
& \Pi(T_{\text{opt}}, B_{\text{opt}}) \\
&= \sum_{u \in nl(T_{\text{opt}})} |B_{\text{opt}}(u)| \cdot e_s + (|N(T_{\text{opt}})| - 1) \cdot e_r \\
&\geq (\Xi(T_I) - |d^+(T_{\text{opt}})| + 1) \cdot e_s + (|N(T_{\text{opt}})| - 1) \cdot e_r \\
&= (\Xi(T_I) + 1) \cdot e_s + (|N(T_{\text{opt}})| - 1) \cdot e_r - |d^+(T_{\text{opt}})| \cdot e_s
\end{aligned}$$

Since each node in $d^+(T_{\text{opt}})$ must transmit at least once, we have $|d^+(T_{\text{opt}})| \cdot e_s \leq \Pi(T_{\text{opt}}, B_{\text{opt}})$, and hence: $(\Xi(T_I) + 1) \cdot e_s + (|N(T_{\text{opt}})| - 1) \cdot e_r \leq 2\Pi(T_{\text{opt}}, B_{\text{opt}})$.

Let $\alpha = 3\rho H(\Delta + 1) + \rho$. With Lemma 4, Theorem 2 and Theorem 3, we have:

$$\begin{aligned}
\sum_{u \in d^+(\bar{T})} |\bar{B}(u)| &= \sum_{u \in d^+(\hat{T})} |\hat{F}(u)| \leq |N(\bar{SB})| \\
&\leq \alpha |N(SB^*)| = \alpha \Xi(T_I)
\end{aligned}$$

Besides, we have some evident inequalities: $e_r \leq e_s$, $|d^1(\bar{T})| \leq |M| \leq |N(T_{\text{opt}})|$, and $|\bar{B}(u)| \geq 1$ (for any $u \in d^+(\bar{T})$). Using these inequalities, we can get:

$$\begin{aligned}
& \Pi(\bar{T}, \bar{B}) \\
&= \sum_{u \in nl(\bar{T})} |\bar{B}(u)| \cdot e_s + (|N(\bar{T})| - 1) \cdot e_r \\
&= \sum_{u \in d^+(\bar{T}) \cup \{s\}} |\bar{B}(u)| \cdot e_s + (|d^+(\bar{T})| + |d^1(\bar{T})| - 1) \cdot e_r \\
&\leq \left(\sum_{u \in d^+(\bar{T})} |\bar{B}(u)| + 1 \right) \cdot e_s + \sum_{u \in d^+(\bar{T})} e_s + (|M| - 1) \cdot e_r \\
&\leq \left(2 \sum_{u \in d^+(\bar{T})} |\bar{B}(u)| + 1 \right) \cdot e_s + (|N(T_{\text{opt}})| - 1) \cdot e_r \\
&\leq (2\alpha \cdot \Xi(T_I) + 1) \cdot e_s + (|N(T_{\text{opt}})| - 1) \cdot e_r \\
&\leq 2\alpha \cdot ((\Xi(T_I) + 1) \cdot e_s + (|N(T_{\text{opt}})| - 1) \cdot e_r) \\
&\leq 4\alpha \cdot \Pi(T_{\text{opt}}, B_{\text{opt}}) \\
&= (12\rho H(\Delta + 1) + 4\rho) \Pi(T_{\text{opt}}, B_{\text{opt}}),
\end{aligned}$$

hence the approximation ratio $12\rho H(\Delta + 1) + 4\rho$. \square

REFERENCES

- [1] Y. Gu and T. He, "Data forwarding in extremely low duty-cycle sensor networks with unreliable communication links," in *Proc. ACM SenSys*, 2007, pp. 321–334.
- [2] G. Anastasi, M. Conti, M. D. Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: A survey," *Ad Hoc Networks*, vol. 7, no. 3, pp. 537–568, 2009.
- [3] F. Wang and J. Liu, "Duty-cycle-aware broadcast in wireless sensor networks," in *Proc. IEEE INFOCOM*, 2009, pp. 468–476.
- [4] S. Guo, Y. Gu, B. Jiang, and T. He, "Opportunistic flooding in low-duty-cycle wireless sensor networks with unreliable links," in *Proc. ACM MobiCom*, 2009, pp. 133–144.
- [5] L. Su, B. Ding, Y. Yang, T. F. Abdelzaher, G. Cao, and J. C. Hou, "ocast: Optimal multicast routing protocol for wireless sensor networks," in *Proc. IEEE ICNP*, 2009, pp. 151–160.
- [6] J. Hong, J. Cao, W. Li, S. Lu, and D. Chen, "Minimum-transmission broadcast in uncoordinated duty-cycled wireless ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 59, no. 1, pp. 307–318, 2010.
- [7] S. Xiong, J. Li, M. Li, J. Wang, and Y. Liu, "Multiple task scheduling for low-duty-cycled wireless sensor networks," in *Proc. IEEE INFOCOM*, 2011, pp. 1323–1331.
- [8] C. Gui and P. Mohapatra, "Power conservation and quality of surveillance in target tracking sensor networks," in *Proc. ACM MobiCom*, 2004, pp. 129–143.
- [9] T. He, P. Vicaire, T. Yan, Q. Cao, G. Zhou, L. Gu, L. Luo, R. Stoleru, J. A. Stankovic, and T. F. Abdelzaher, "Achieving long-term surveillance in vigilnet," in *Proc. IEEE INFOCOM*, 2006, pp. 1–12.
- [10] L. Mo, Y. He, Y. Liu, J. Zhao, S. Tang, X.-Y. Li, and G. Dai, "Canopy closure estimates with greenorbs: sustainable sensing in the forest," in *Proc. ACM SenSys*, 2009, pp. 99–112.
- [11] G. Xing, M. Li, H. Luo, and X. Jia, "Dynamic multiresolution data dissemination in wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 8, pp. 1205–1220, 2009.
- [12] J. Wieselthier, G. Nguyen, and A. Ephremides, "On the construction of energy-efficient broadcast and multicast trees in wireless networks," in *Proc. IEEE INFOCOM*, 2000, pp. 585–594.
- [13] P.-J. Wan, G. Calinescu, and C.-W. Yi, "Minimum-power multicast routing in static ad hoc wireless networks," *IEEE/ACM Trans. Netw.*, vol. 12, no. 3, pp. 507–514, 2004.
- [14] W. Liang, "Approximate minimum-energy multicasting in wireless ad hoc networks," *IEEE Trans. Mobile Comput.*, vol. 5, no. 4, pp. 377–387, 2006.
- [15] D. Li, Q. Liu, X. Hu, and X. Jia, "Energy efficient multicast routing in ad hoc wireless networks," *Computer Communications*, vol. 30, no. 18, pp. 3746–3756, 2007.
- [16] W. Liang, R. Brent, Y. Xu, and Q. Wang, "Minimum-energy all-to-all multicasting in wireless ad hoc networks," *IEEE Trans. Wireless Commun.*, vol. 8, no. 11, pp. 5490–5499, 2009.
- [17] M. Charikar, C. Chekuri, T.-y. Cheung, Z. Dai, A. Goel, S. Guha, and M. Li, "Approximation algorithms for directed steiner problems," in *Proc. ACM-SIAM SODA*, 1998, pp. 192–200.
- [18] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York: W. H. Freeman, 1979.
- [19] G. Ausiello, A. D'Atri, and M. Protasi, "Structure preserving reductions among convex optimization problems," *Journal of Computer and System Sciences*, vol. 21, no. 1, pp. 136–153, 1980.
- [20] U. Feige, "A threshold of $\ln n$ for approximating set cover," *J. ACM*, vol. 45, pp. 634–652, July 1998.
- [21] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms, Second Edition*. MIT Press and McGraw-Hill, 2001.
- [22] K. Mehlhorn, "A faster approximation algorithm for the steiner problem in graphs," *Information Processing Letters*, vol. 27, no. 3, pp. 125–128, 1988.
- [23] B. Liang and Z. J. Haas, "Virtual backbone generation and maintenance in ad hoc network mobility management," in *Proc. IEEE INFOCOM*, 2000, pp. 1293–1302.
- [24] F. Bauer and A. Varma, "Distributed algorithms for multicast path setup in data networks," *IEEE/ACM Trans. Netw.*, vol. 4, no. 2, pp. 181–191, 1996.
- [25] P. Chalermsook and J. Fakcharoenphol, "Simple distributed algorithms for approximating minimum steiner trees," in *Proc. COCOON*, 2005, pp. 380–389.
- [26] R. B. Muhammad, "Distributed steiner tree algorithm and its application in ad-hoc wireless networks," in *Proc. ICWN*, 2006, pp. 173–178.
- [27] S. A. M. Makki and G. Havas, "Distributed algorithms for depth-first search," *Information Processing Letters*, vol. 60, no. 1, pp. 7–12, 1996.
- [28] P. M. Ruiz and A. F. Gomez-Skarmeta, "Approximating optimal multicast trees in wireless multihop networks," in *Proc. ISCC*, 2005, pp. 686–691.
- [29] P. M. Ruiz and I. Stojmenovic, "Cost-efficient multicast routing in ad hoc and sensor networks," in *Handbook on Approximation Algorithms and Metaheuristics*, T. F. Gonzalez, Ed. Chapman & Hall/CRC, 2007, pp. 65–1.
- [30] L. Kou, G. Markowsky, and L. Berman, "A fast algorithm for steiner trees," *Acta Informatica*, vol. 15, pp. 141–145, 1981.
- [31] D. S. Johnson, "Approximation algorithms for combinatorial problems," in *Proc. ACM STOC*, 1973, pp. 38–49.