Optimizing Energy Efficiency for Minimum Latency Broadcast in Low-Duty-Cycle Sensor Networks

LIJIE XU, Nanjing University of Posts and Telecommunications
GUIHAI CHEN, Nanjing University
JIANNONG CAO, The Hong Kong Polytechnic University
SHAN LIN, Stony Brook University
HAIPENG DAI and XIAOBING WU, Nanjing University
FAN WU, Shanghai Jiao Tong University

Multihop broadcasting in low-duty-cycle Wireless Sensor Networks (WSNs) is a very challenging problem, since every node has its own working schedule. Existing solutions usually use unicast instead of broadcast to forward packets from a node to its neighbors according to their working schedules, which is, however, not energy efficient. In this article, we propose to exploit the broadcast nature of wireless media to further save energy for low-duty-cycle networks, by adopting a novel broadcasting communication model. The key idea is to let some early wake-up nodes postpone their wake-up slots to overhear broadcasting messages from its neighbors. This model utilizes the spatiotemporal locality of broadcast to reduce the total energy consumption, which can be essentially characterized by the total number of broadcasting message transmissions. Based on such model, we aim at minimizing the total number of broadcasting message transmissions of a broadcast for low-duty-cycle WSNs, subject to the constraint that the broadcasting latency is optimal. We prove that it is NP-hard to find the optimal solution, and design an approximation algorithm that can achieve a polylogarithmic approximation ratio. Extensive simulation results show that our algorithm outperforms the traditional solutions in terms of energy efficiency.

Categories and Subject Descriptors: C.2.2 [**Computer-Communication Networks**]: Network Protocols

General Terms: Algorithms, Design

A preliminary version of this article was presented in *Proceedings of the 10th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (IEEE MASS 2013)* [Xu et al. 2013].
This work was partly supported by the State Key Development Program for Basic Research of China (973 Program) (Grant No. 2012CB316201), National Natural Science Foundation of China (Grants No. 61472252, No. 61321491, No. 61133006, No. 61373130, and No. 61422208), NUPTSF (Grant No. NY214169), NSF of Jiangsu Province (Grant No. BK20141319), EU FP7 CROWN project under Grant No. PIRSES-GA-2013-610524, Funding for Central Institutions of China (No. 20620140515), CCF-Tencent Open Fund, ANR/RGC Joint Research Scheme (RGC No. A-PolyU505/12), NSF CNS-1239108, CNS-1218718, and IIS-1231680.
Authors' addresses: L. Xu, School of Computer Science and Technology, Nanjing University of Posts and Telecommunications, 9 Wenyuan Road, Qixia District, Nanjing 210023, China; email: ljxu@njupt.edu.cn; G. Chen (corresponding author), State Key Laboratory for Novel Software Technology, Nanjing University, 163 Xianlin Avenue, Qixia District, Nanjing 210023, China, he is also with Shanghai Key Laboratory of Scalable Computing and Systems, Shanghai Jiao Tong University, 800 Dongchuan Road, Minhang District, Shanghai 200240, China; email: gchen@nju.edu.cn; J. Cao, Internet and Mobile Computing Lab, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong; email: csjcao@comp.polyu.edu.hk; S. Lin, Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY 11794-2350, USA; email: shan.x.lin@stonybrook.edu; H. Dai and X. Wu, State Key Laboratory for Novel Software Technology, Nanjing University, 163 Xianlin Avenue, Qixia District, Nanjing 210023, China; emails: dhpphd2003@gmail.com, wuxb@nju.edu.cn; F. Wu, Shanghai Key Laboratory of Scalable Computing and Systems, Shanghai Jiao Tong University, 800 Dongchuan Road, Minhang District, Shanghai 200240, China; email: fwu@cs.sjtu.edu.cn.

DOI: http://dx.doi.org/10.1145/2753763

Additional Key Words and Phrases: Wireless sensor networks, low duty cycle, multihop broadcast, energy efficient, minimum broadcasting latency

ACM Reference Format:
Lijie Xu, Guihai Chen, Jiannong Cao, Shan Lin, Haipeng Dai, Xiaobing Wu, and Fan Wu. 2015. Optimizing energy efficiency for minimum latency broadcast in low-duty-cycle sensor networks. ACM Trans. Sensor Netw. 11, 4, Article 57 (July 2015), 31 pages.
DOI: http://dx.doi.org/10.1145/2753763

1. INTRODUCTION

Wireless Sensor Networks (WSNs) have been widely used for various applications, such as environmental monitoring [Liu et al. 2013a, 2013b; Li and Liu 2009], scientific exploration [Li et al. 2013], and navigation systems [Wang et al. 2013]. Many of these applications require broadcasting to frequently disseminate system configurations and code updates to the whole network. The total energy consumption and the broadcasting latency are the main performance metrics for evaluation of broadcasting algorithms.

It is important and very challenging to minimize the energy consumption of broadcasting for low-duty-cycle WSNs, in which every sensor node has its own working schedule to wake up periodically to perform sensing and communication tasks. Existing solutions for broadcasting in low-duty-cycle WSNs (such as Guo et al. [2009], Hong et al. [2010], Wang and Liu [2009], Sun et al. [2009], Niu et al. [2013], Su et al. [2009], Jiao et al. [2010], and Li et al. [2011]) usually implement one-hop broadcast with multiple unicasts, which is energy inefficient especially for applications of large message broadcasting, such as *code update*. Actually, the broadcast nature of wireless media offers opportunities to reduce the total number of broadcasting message transmissions, even for duty-cycled networks where every node has its own schedule. To improve the energy efficiency of broadcasting, nodes should adjust their working schedules to maximize the number of receivers for each forwarding message.

Compared with always-awake networks, low-duty-cycle sensor networks usually yield a notable increase on communication latency due to periodic sleeping [Gu and He 2007], and thus latency is always taken as the first consideration for such networks. In this article, we mainly focus on the problem of how to achieve energy-efficient broadcast with minimum latency for low-duty-cycle WSNs. To achieve optimal latency and high energy efficiency of broadcasting, we come up with a novel broadcasting communication model, which fully exploits the spatiotemporal locality of broadcasting to reduce the total number of broadcasting message transmissions. The basic idea is to allow nodes to adjust their wake-up schedules to overhear forwarding messages sent by their neighbors. Some nodes may postpone their wake-up slots to receive the broadcasting message, increasing their latency. But these nodes can be carefully selected so that they are not on latency-critical paths, which indicates their schedule changes do not affect the minimum broadcasting latency. Based on such a broadcasting communication model, we find that the total energy consumption for broadcasting can be essentially characterized by the total number of broadcasting message transmissions, and thus our objective is to design a broadcast with minimum total number of broadcasting message transmissions for low-duty-cycle WSNs, subject to the constraint that the broadcasting latency is optimal, which we call the *Latency-optimal Minimum Energy Broadcast Problem* (LMEB).

The main contributions of this work are as follows:

—To the best of our knowledge, this is the first work that both utilizes the spatiotemporal locality of broadcasting and proposes a solution with a provable approximation

ratio, for energy-efficient broadcast problem with minimum latency constraint in low-duty-cycle WSNs.
—We prove that the LMEB problem is NP-hard. Then, we model the LMEB problem as the *Directed Latency-optimal Group Steiner Tree Problem* (DLGST) by capturing the spatiotemporal characteristic of multihop broadcasting, and propose an efficient solution for this problem.
—Based on the solution to the DLGST problem, we further devise a novel Broadcasting Schedule Construction Algorithm to derive the solution to the LMEB problem, which essentially avoids the redundant transmissions and reduces the collision probability as much as possible.
—We show that the approximation ratio of our solution is $O(\log N \cdot \log d_{max})$, where N and d_{max} denote the number of sensor nodes and the maximum node degree, respectively.
—Extensive simulation results show that our solution makes a significant improvement over the traditional solutions in terms of energy efficiency.

The rest of the article is organized as follows: Section 2 summarizes the related work. Section 3 illustrates the network model and formally states the problem. A Detailed descriptions of our proposed scheme and performance analysis are presented in Section 4, followed by the simulation results and the discussions about practical issues in Sections 6 and 5, respectively. We conclude the article in Section 7.

2. RELATED WORK

The broadcast problem in low-duty-cycle WSNs has received a lot of attention from the research community in the past few years [Guo et al. 2009; Hong et al. 2010; Wang and Liu 2009; Sun et al. 2009; Niu et al. 2013; Su et al. 2009; Jiao et al. 2010; Li et al. 2011; Zhu et al. 2010; Guo et al. 2011; Lai and Ravindran 2010b; Han et al. 2013a, 2013b, 2013c; Cheng et al. 2013; Xu and Chen 2013; Kyasanur et al. 2006].

Guo et al. [2009] proposed Opportunistic Flooding to make probabilistic forwarding decisions at the sender based on the delay distribution of next-hop nodes. Hong et al. [2010] studied the Minimum-Transmission Broadcast Problem in uncoordinated duty-cycled networks and proved its NP-hardness. They proposed a centralized approximation algorithm with a logarithmic approximation ratio and a distributed approximation algorithm with a constant approximation ratio for this problem. Wang and Liu [2009] proposed a broadcasting scheme to achieve the controllable tradeoff between energy and latency by using a dynamic-programming approach. Another solution ADB [Sun et al. 2009], which is designed to be integrated with the receiver-initiated MAC protocol [Sun et al. 2008], was proposed to reduce both redundant transmissions and delivery latency of broadcasting by avoiding collisions and transmissions over poor links. In Niu et al. [2013], the authors investigated the energy-efficient broadcast problem with minimum latency constraint in low-duty-cycle WSNs with unreliable links, and proposed a distributed heuristic solution to tackle this problem. In Han et al. [2013a], the authors studied the duty-cycle-aware Minimum-Energy Multicasting problem in WSNs both for one-to-many multicasting and for all-to-all multicasting. Han et al. [2013c] studied the problem of minimizing the expected total transmission power for reliable data dissemination in duty-cycled WSNs. Due to the NP-hardness of the problem, they designed efficient approximation algorithms with provable performance bounds for it. Cheng et al. [2013] proposed a novel Dynamic Switching-based Reliable Flooding (DSRF) framework, which is designed as an enhancement layer to provide efficient and reliable delivery for a variety of existing flooding tree structures in low duty-cycle WSNs. However, all of these works inefficiently implement one-hop broadcast with multiple unicasts, which do not fully utilize the spatiotemporal locality of broadcasting.

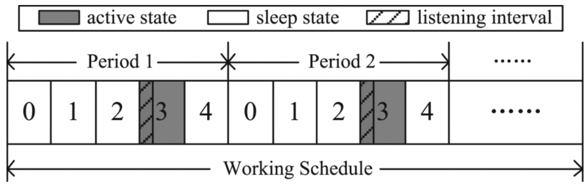

Fig. 1. An example of working schedule with $L = 5$ and $T_s(\cdot) = 3$.

Actually, the broadcast nature of wireless media offers opportunities to reduce the total number of broadcasting message transmissions, even for low-duty-cycle networks.

To achieve higher energy efficiency of broadcasting, a few works that make the best of the spatiotemporal locality of broadcasting were proposed recently. In Guo et al. [2011], the authors considered link correlation and devised a novel flooding scheme to reduce energy consumption of broadcasting by making nodes with high correlation be assigned to a common sender. Lai et al. [2010b] proposed a Hybrid-cast protocol that adopts opportunistic forwarding with delivery deferring to shorten broadcasting latency and transmission number. However, all of these existing solutions are heuristic and fail to provide a provable approximation ratio. Moreover, all of them mainly focus on energy efficiency optimization but do not take latency constraint into account.

3. MOTIVATION

3.1. Network Model and Assumptions

Without loss of generality, we assume that N sensor nodes are uniformly and densely deployed in a circular sensory field with a fixed radius of R and the sink node is located at the center of the sensory field. Each node has the same communication range r_c. Also, it is assumed that time is divided into a number of equal time slots and each time slot is set long enough so that it can accommodate the transmission of the potential large broadcasting message. Each time slot is either in *sleep state*, where each node will turn its radio off, or in *active state*, where each node will keep awake for a short duration of *listening interval* to make the event sensing and channel listening at the beginning.

In our model, we assume all the sensor nodes are operated at low-duty-cycle mode, where each sensor node determines its own working schedule depending on a particular power management protocol (e.g., Cao et al. [2005]) immediately after deployment. For simplicity, we assume the working schedule of each node is periodic and alternates between one *active state* and $L - 1$ *sleep states*. Here, we use $T_s(j)$ to represent the scheduled active time slot in each period of working schedule for any node j. Figure 1 explicitly illustrates an example of the periodic working schedule where $L = 5$ and $T_s(\cdot) = 3$. Further, we use the undirected spatiotemporal topology graph $G = (V, E, W, L)$ to represent the network topology and nodes' working schedules, where V represents the set of N nodes including the sink node v_0 and all sensing nodes $\{v_1, \ldots, v_{N-1}\}$, E represents the set of all communication links, W denotes the set of working schedules for all nodes, and L denotes the schedule period length of each node. We denote by $d(v_i, v_j)$ the point-to-point transmission latency from node v_i to node v_j for any edge $(v_i, v_j) \in E$, and $d(v_i, v_j)$ can be determined as follows:

If $v_i = v_0$,

$$d(v_i, v_j) = \begin{cases} T_s(v_j) - T_s(v_i) + 1, & if\ T_s(v_j) \geq T_s(v_i); \\ T_s(v_j) - T_s(v_i) + L + 1, & otherwise, \end{cases} \tag{1}$$

and if $v_i \neq v_0$,

$$d(v_i, v_j) = \begin{cases} T_s(v_j) - T_s(v_i), & if\ T_s(v_j) > T_s(v_i); \\ T_s(v_j) - T_s(v_i) + L, & otherwise. \end{cases} \tag{2}$$

The same as with most literature for low-duty-cycle WSNs (e.g., Guo et al. [2009], Hong et al. [2010], Wang and Liu [2009], Niu et al. [2013], Su et al. [2009], Jiao et al. [2010], Li et al. [2011], Gu and He [2007], Zhu et al. [2010], Guo et al. [2011], Han et al. [2013a], and Cheng et al. [2013]), we assume *time synchronization is achieved, and each node can transmit its packets at any time, while it can only receive the packets from its neighbors in active states*. Specifically, each node v_i will wake up at the beginning of the *active state* and keep listening for a period of *listening interval*; if any broadcasting packet in which the target receiver ID is v_i is received, it will keep receiving until all packets of the broadcasting message are received and then go to sleep immediately; otherwise, it will go to sleep immediately. If any sender wants to send the broadcasting message to its receiver, it will set a timer to wake up itself at the beginning of the receiver's next *active state* to finish the transmission, and then go to sleep.

Besides this, we also have the following basic assumptions:

(1) Each node cannot do sending and receiving simultaneously.
(2) Each node is aware of the working schedules of all its neighboring nodes within two hops; this can be realized via local information exchange between neighboring nodes initially after the network is deployed.
(3) For simplicity, we do not consider the packet collision problem due to the fact that the low-duty-cycle operation inherently reduces the probability of collision to a great extent, which has been experimentally verified in Wang and Liu [2009].
(4) The working schedules of any node and its neighbors are different from each other. It is usually true for low-duty-cycle WSNs, since we usually improve the network performance (e.g., to minimize average detection delay) by carefully designing the working schedules of all nodes (e.g., Cao et al. [2005]) to make the neighboring nodes rotate the sensory coverage. Further, this assumption will be relaxed in Section 4.5.

3.2. Problem Statement

In traditional solutions for broadcasting, all nodes will receive the broadcasting message at their scheduled wake-up time slots, which could lead to the minimum broadcasting latency but, however, draw much more energy consumption since any one-hop broadcast is actually realized by a number of unicasts. To achieve higher energy efficiency of broadcasting, we come up with a novel broadcasting communication model that is based on the spatiotemporal locality of broadcasting. This model defines two kinds of receivers, that is, *DelayedReceivers* and *InstantReceivers*, for any sender. The sender will send the broadcasting message to each *InstantReceiver*, and also it will send a short beacon packet that only contains the ID of some *InstantReceiver* v_j, say *Beacon*(v_j), to each *DelayedReceiver*. Upon receiving the *Beacon*(v_j) from the sender, any *DelayedReceiver* will go to sleep immediately and defer its message receiving time by setting a timer to wake up itself at the next *active state* of the *InstantReceiver* v_j. Note that, the *DelayedReceiver* can be aware of the working schedule of the *InstantReceiver* v_j due to the assumption that each node is aware of the working schedules of all its neighboring nodes within two hops.

Figure 2 illustrates a simple example for the one-hop broadcast case, where the number labeled within each pair of brackets denotes the scheduled wake-up time slot (e.g., $v_0(3)$ represents $T_s(v_0) = 3$) and the schedule period length L is set as 10.

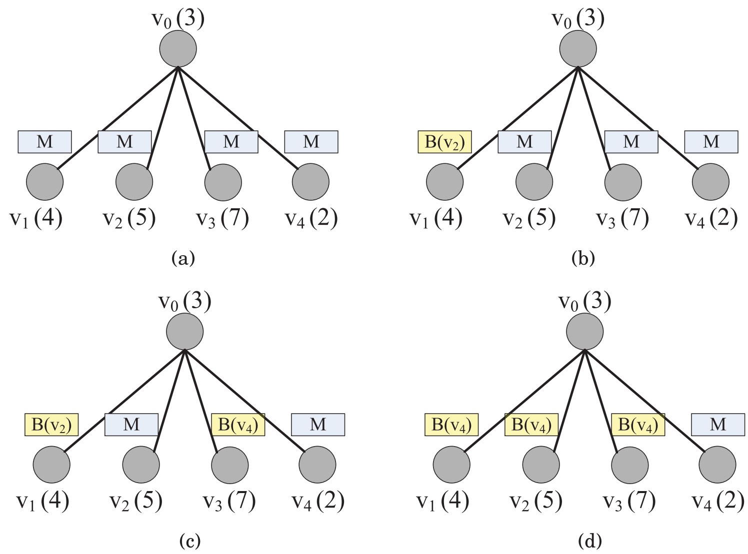

Fig. 2. (a) Broadcast without deferring. (b) Broadcast with one *DelayedReceiver*. (c) Broadcast with two *DelayedReceivers*. (d) The optimal broadcast.

Figure 2(a) shows a traditional solution, in which the sink node v_0 delivers the message to its neighbors one by one to realize the broadcasting (i.e., to set nodes v_1, v_2, v_3, v_4 as the *InstantReceivers*). It requires total energy consumption of $E_{total} = 4 \times k \times e_s^d + 4 \times k \times e_r^d$, where k denotes the number of data packets contained in a broadcasting message, and e_s^d and e_r^d denote the energy consumption when sending and receiving a data packet, respectively. As shown in Figure 2(b), if the sink node v_0 delivers the beacon packet *Beacon*(v_2) to the *DelayedReceiver* v_1 and delivers the broadcasting message to the *InstantReceivers* $\{v_2, v_3, v_4\}$, node v_1 will defer its message receiving time by setting a timer to wake up itself at the next scheduled active time slot of the *InstantReceiver* v_2 (i.e., time slot 5) and the total energy consumption for broadcasting will be $E'_{total} = e_s^b + 3 \times k \times e_s^d + e_r^b + 4 \times k \times e_r^d$, where e_s^b and e_r^b denote the energy consumption when sending and receiving a beacon packet, respectively. As shown in Wang et al. [2006], it is usual that a data packet has a length of 133 bytes and a beacon packet has only a length of 19 bytes, which indicates that $e_s^b + e_r^b$ is far less than e_s^d in practice. Thus, total energy benefit of deferring the message receiving time of any receiver, that is, $\Delta = E_{total} - E'_{total} = k \times e_s^d - (e_s^b + e_r^b)$, must be greater than zero. For applications with a large broadcasting message that contains a large number of data packets (i.e., *code update*), especially, this benefit will be significant as $k \gg 1$. Moreover, we can easily find that based on such a broadcasting communication model, the total energy benefit will increase as the number of *InstantReceivers* decreases, which implies that *total energy consumption for broadcasting can be essentially characterized by total number of broadcasting message transmissions under this model*. Figure 2(c) shows an example of broadcast with two *DelayedReceivers*, that is, the sink node v_0 delivers the beacon packet *Beacon*(v_2) to the *DelayedReceiver* v_1, delivers the beacon packet *Beacon*(v_4) to the *DelayedReceiver* v_3, and delivers the broadcasting message to the *InstantReceivers* $\{v_2, v_4\}$. According to the previous conclusion, we can find that it must have a higher energy efficiency than the case in Figure 2(b). Obviously, the schedule in Figure 2(d) must be the optimal solution, where the sink node v_0 delivers the beacon packet *Beacon*(v_4) to the *DelayedReceivers* $\{v_1, v_2, v_3\}$ and delivers the broadcasting message to the *InstantReceiver* v_4.

According to the previous example, we can find that total energy consumption for broadcasting will benefit from receive deferring. Based on such a broadcasting communication model, we present the definitions of *Forwarding Sequence* and *Broadcasting Schedule* in low-duty-cycle WSNs as follows.

Definition 3.1 (*Forwarding Sequence*). For any forwarder v_i of the broadcasting message, its Forwarding Sequence $S_f(v_i)$ is defined as a sequence of its receivers sorted based on the scheduled wake-up time, namely,

$$S_f(v_i) = <[r_1^1, \ldots, r_1^{k_1}], \underline{r_1}, \ldots, [r_j^1, \ldots, r_j^{k_j}], \underline{r_j}>, \tag{3}$$

where $r_j^k(k = 1, \ldots, k_j)$ and the underlined r_j, respectively, denote the *DelayedReceivers* and *InstantReceivers* of node v_i. Specifically, the forwarder v_i will send the short control packet *Beacon*(r_j) to each *DelayedReceiver* r_j^k and send the broadcasting message to each *InstantReceiver* r_j. Here, [ ] denotes an optional item.

Definition 3.2 (*Broadcasting Schedule*). Given a spatiotemporal topology graph $G = (V, E, W, L)$, the schedule strategy of any node $v_i \in V$, say $M(v_i)$, can be defined as follows:

$$M(v_i) = (\alpha, \beta), \tag{4}$$

where

$$\alpha \in \{0, \ldots, N-1\}, \quad \beta = \begin{cases} S_f(v_i), & \alpha > 0; \\ NULL, & \alpha = 0. \end{cases}$$

In Equation (4), the variable α denotes node v_i's total forwarding number of the broadcasting message, and if v_i is the forwarder (i.e., $M(v_i)\cdot\alpha > 0$), β will denote the Forwarding Sequence $S_f(v_i)$, which represents that once receiving the broadcasting message, node v_i will send the short beacon packet or the broadcasting message to each node in $S_f(v_i)$ in sequence. Obviously, $M(v_i)\cdot\alpha$ must be equal to the number of *InstantReceivers* in $S_f(v_i)$. Here, *NULL* denotes the omitted item and it is obvious that $M(v_i)\cdot\beta = NULL$ for any node v_i with $M(v_i)\cdot\alpha = 0$. Specifically, it must have $M(v_0)\cdot\alpha > 0$ for the sink node v_0.

Here, a broadcasting schedule M in the network can be defined as the set of all nodes' schedule strategies:

$$M = \{M(v_i) | v_i \in V\}, \tag{5}$$

such that $I_\alpha = \{v_i | v_i \in V \text{ and } M(v_i)\cdot\alpha > 0\}$ subjects to

(1) **connectivity**, that is, there must exist a subtree $T = (I_\alpha, E^T)$, where $E^T \subseteq E$ and for any edge $(v_i, v_j) \in E^T$, it must have $v_j \in M(v_i)\cdot\beta$ if v_i is the parent of v_j;
(2) **coverage**, that is, $\bigcup_{v_i \in I_\alpha} M(v_i)\cdot\beta = V - \{v_0\}$;
(3) **nonredundancy**, that is, $M(v_i)\cdot\beta \bigcap M(v_j)\cdot\beta = \emptyset$ for any $v_i, v_j \in I_\alpha$ $(i \neq j)$.

In the preceding definition, note that, we assume each node cannot send the beacon packets until the broadcasting message is received in order to avoid potential simultaneous sending and receiving, as well as to simplify the problem. As stated before, we will utilize total number of broadcasting message transmissions to characterize total energy consumption for broadcasting. Here, we take two broadcasting schedules shown in Figure 3 as an example to illustrate our problem. There is no deferring for each node (i.e., no *DelayedReceiver* but only *InstantReceivers* exist in the network) when adopting Schedule 1, which achieves the minimum broadcasting latency 17 but the maximum number of broadcasting message transmissions 5. For Schedule 2, the

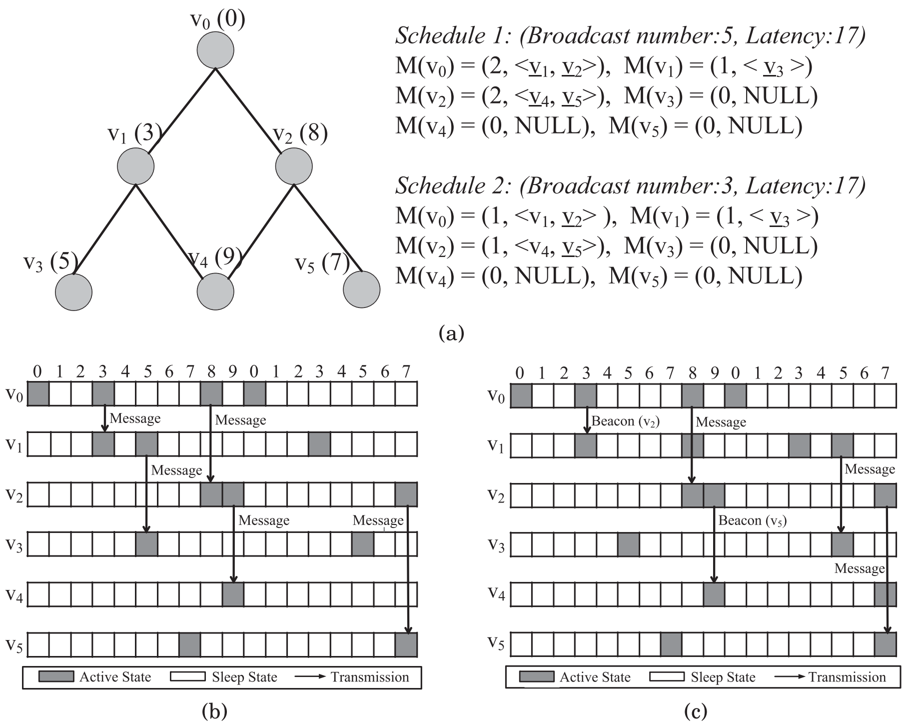

Fig. 3. (a) The original topology graph with two defined broadcasting schedules. (b) Illustration of Schedule 1. (c) Illustration of Schedule 2.

number of broadcasting message transmissions can be reduced to three without increasing the broadcasting latency as nodes v_1 and v_4 defer their receiving time to the scheduled wake-up time slots of v_2 and v_5, respectively. From the preceding example, we can find that there could exist multiple broadcasting schedules that have the same minimum broadcasting latency but different numbers of broadcasting message transmissions. Accordingly, our objective is to address the following LMEB.

PROBLEM 1 (LMEB). *Given an undirected spatiotemporal topology graph $G = (V, E, W, L)$, find an efficient broadcasting schedule M to optimize the total number of broadcasting message transmissions, that is, to minimize $\sum_{i=0}^{N-1} M(v_i) \cdot \alpha$, subject to the constraint that the broadcasting latency is minimized.*

THEOREM 3.3. *The LMEB problem is NP-hard.*

Note that the proofs of all the theorems in this article will be included in th Appendix.

4. APPROXIMATION ALGORITHM

In order to solve the LMEB problem, in this section, we propose an efficient approximation solution.

4.1. Overview

To better capture the spatiotemporal characteristic of multihop broadcasting, we first transform the original topology graph into a directed Spatiotemporal Relationship Graph (SRG). Then, we prove that the LMEB problem on the original topology graph is equivalent to the DLGST on its corresponding SRG, and solve it by adopting a

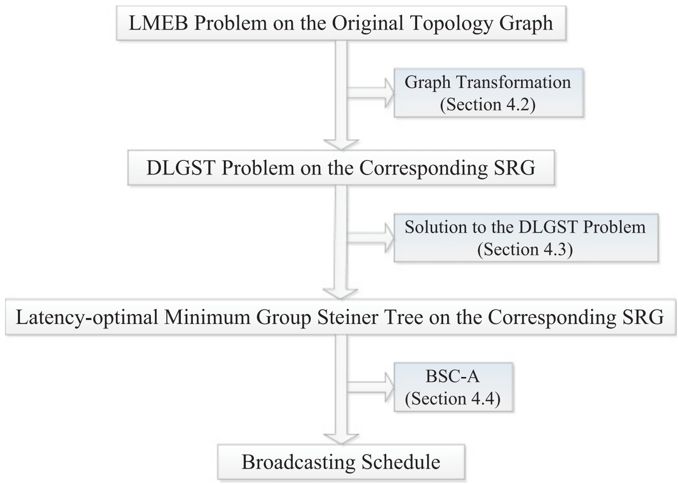

Fig. 4. Overview of solution to the LMEB problem.

deterministic randomized-rounding based approach. Based on the solution to the DLGST problem, finally, we devise a novel BSC-A to derive the solution of the LMEB problem, which essentially avoids the redundant transmissions and reduces the collision probability as much as possible. Figure 4 explicitly illustrates the general process of our solution.

4.2. Graph Transformation

Definition 4.1 (*Coverage Set*). Given a Sender-InstantReceiver pair (v_s, v_r) and a time slot t ($t \in \{0, \ldots, L-1\}$), the coverage set $CS(v_s, v_r, t)$ is defined as follows:

(1) if $t < T_s(v_r)$, $CS(v_s, v_r, t) = \{x | x \in N(v_s) - \{v_0\}$ *and* $T_s(x) \in \{t+1, \ldots, T_s(v_r)\}\}$,
(2) if $t > T_s(v_r)$, $CS(v_s, v_r, t) = \{x | x \in N(v_s) - \{v_0\}$ *and* $T_s(x) \notin \{T_s(v_r)+1, \ldots, t\}\}$,
(3) if $t = T_s(v_r)$, $CS(v_s, v_r, t) = \{x | x \in N(v_s) - \{v_0\}\}$,

in which v_0 denotes the sink node, and $N(v_i)$ denotes the neighbors set of node v_i.

OBSERVATION 1. *Given a spatiotemporal topology graph $G = (V, E, W, L)$ and any edge $(v_s, v_r) \in E$, if it requires that node v_s be the sender (i.e., forwarder) and node v_r be the InstantReceiver of node v_s, then an efficient broadcasting schedule must make sure that when being received by v_r, the broadcasting message also has been received by all the nodes in the coverage set $CS(v_s, v_r, T_c(v_s)) - \{v_r\}$, where $T_c(v_s)$ denotes the time slot that the uncovered node v_s receives the broadcasting message.*

As an example, in Figure 5(a), the sender v_0 is assumed to receive the broadcasting message at its scheduled wake-up time slot, namely, $T_c(v_0) = T_s(v_0) = 3$. If we let node v_3 be the *InstantReceiver* of the sender v_0, according to Observation 1, all the nodes in $CS(v_0, v_3, T_c(v_0)) = \{v_1, v_2, v_3\}$ must be ensured to have been covered at the coverage time of v_3 (i.e., the time slot 7 when the uncovered node v_3 receives the broadcasting message); this is because any schedule that makes the coverage time of v_1 or v_2 be preceded by that of v_3 will never benefit from both broadcasting latency and number of broadcasting message transmissions. In other words, v_1 as well as v_2 must be covered by one of the following three ways:

(1) covered by the sender v_0 at time slot $T_s(v_3)$ as the *DelayedReceiver*;
(2) covered by any other sender before time slot $T_s(v_3)$ as the *DelayedReceiver*; or

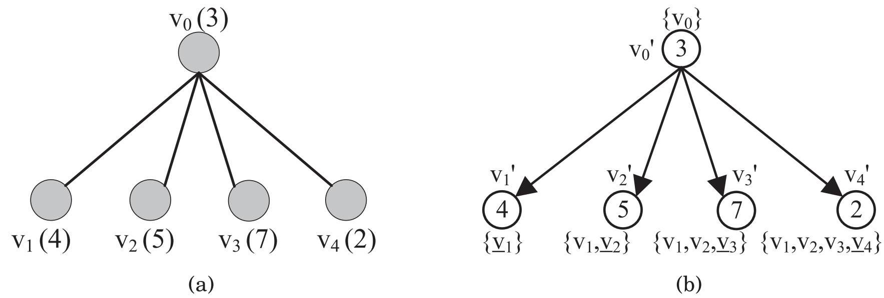

Fig. 5. (a) An example of one-hop topology. (b) Illustration of SRG.

(3) covered by the sender v_0 or any other sender before time slot $T_s(v_3)$ as the *InstantReceiver*.

In order to better exhibit the spatiotemporal characteristic of broadcasting, any one-hop broadcast (e.g., Figure 5(a)) can be characterized by a directed SRG (e.g., Figure 5(b)) where each edge represents one broadcasting message transmission and each vertex represents a coverage set. For any vertex v_i' in SRG, we let $S(v_i')$ denote the coverage set that represented by v_i' and $IR(v_i')$ denote the *InstantReceiver* in $S(v_i')$. Also, we let $T_s(v_i')$ denote the coverage time slot of vertex v_i' and set $T_s(v_i') = T_s(IR(v_i'))$. Specifically, any directed edge (v_i', v_j') in SRG represents one broadcasting message transmission from a sender $s \in S(v_i')$ to the *InstantReceiver* $IR(v_j')$ at time slot $T_s(IR(v_j'))$, and vertex v_j' represents the resulting coverage set $CS(s, IR(v_j'), T_c(s))$ after this transmission where $T_c(s) = T_s(v_i')$. Specifically, we set $S(v_0') = \{v_0\}$ and $T_s(v_0') = T_s(v_0)$ for the root vertex v_0' in SRG. For each directed edge (v_i', v_j') in SRG, we use a *Sender-InstantReceiver* pair, that is, $P(v_i', v_j') = <$*Sender*, *InstantReceiver*$>$, to mark it.

The following *Spatiotemporal Relationship Graph Construction Algorithm* (SRGC-A) will introduce how to efficiently construct a directed SRG $G' = (V', E', W', L)$ from a undirected spatiotemporal topology graph $G = (V, E, W, L)$ in detail: Initially, SRG only contains a root vertex v_0' where $S(v_0') = \{v_0\}$ and $T_s(v_0') = T_s(v_0)$. Starting with considering the sink node v_0 as the sender, we respectively regard each neighbor v_i of the sink as the *InstantReceiver*, then insert a directed edge from the vertex v_0' to the newly added vertex v_{new}' and set $S(v_{new}') = CS(v_0, v_i, T_s(v_0))$, $IR(v_{new}') = v_i$, $P(v_0', v_{new}') = <v_0, v_i>$. For any newly added vertex v_{new}', we in turn select each node $v_i \in S(v_{new}')$ as the sender and select each node $v_j \in N(v_i) - \{v_0\}$ as the *InstantReceiver*, then search all the vertices in SRG to check whether there exists a vertex v' with $S(v') = CS(v_i, v_j, T_s(v_{new}'))$ and $IR(v') = v_j$. If so, we just insert a directed edge from v_{new}' to v' with $P(v_{new}', v') = <v_i, v_j>$; otherwise, we add a new vertex v' as well as the directed edge (v_{new}', v') into SRG and then set $S(v') = CS(v_i, v_j, T_s(v_{new}'))$, $IR(v') = v_j$, $T_s(v') = T_s(v_j)$, $P(v_{new}', v') = <v_i, v_j>$. The preceding process repeats until no new vertex addition to SRG is possible. Algorithm 1 shows the detailed process of SRGC-A.

THEOREM 4.2. *The worst-case time complexity of SRGC-A is $O(N^2 d_{max}^6)$, where d_{max} denotes the maximum node degree in the network.*

It is noteworthy to mention that SRGC-A could be more efficient and offer better time performance guarantee than that shown in Theorem 4.2, if a high-efficient search algorithm, such as the hash-based search algorithm, is adopted in practice.

For convenience of description, as shown in Figure 5(b), the root vertex v_0' in SRG is directed denoted by the coverage set $\{v_0\}$ and any nonroot vertex v_i' in SRG is directly

ALGORITHM 1: Spatiotemporal Relationship Graph Construction

Input: The undirected spatiotemporal topology graph $G = (V, E, W, L)$.
Output: The directed spatiotemporal relationship graph $G' = (V', E', W', L)$.

```
V' = {v'_0};
S(v'_0) = {v_0}; T_s(v'_0) = T_s(v_0); flag(v'_0) = 1;      //v_0 ∈ V is the sink node
while {v'|v' ∈ V' and flag(v') == 1} ≠ ∅ do
    select any vertex v'_new ∈ {v'|v' ∈ V' and flag(v') == 1};
    for each node v_i ∈ S(v'_new) do
        for each node v_j ∈ N(v_i) − {v_0} do
            isfound = 0;
            for each vertex v' ∈ V' do
                if S(v') == CS(v_i, v_j, T_s(v'_new)) and IR(v') == v_j then
                    add a directed edge (v'_new, v') into E';
                    P(v'_new, v') =<v_i, v_j>;
                    isfound = 1; break;
                end
            end
            if isfound == 0 then
                add a new vertex v' into V';
                add a directed edge (v'_new, v') into E';
                S(v') = CS(v_i, v_j, T_s(v'_new)); IR(v') = v_j;
                T_s(v') = T_s(v_j);
                flag(v') = 1;
                P(v'_new, v') =<v_i, v_j>;
            end
        end
    end
    flag(v'_new) = 0;
end
```

denoted by the coverage set $S(v_i')$, in which the underlined node denotes the *InstantReceiver* $IR(v_i')$, and the number labeled within any vertex v_i' represents its coverage time slot $T_s(v_i')$. We can find that SRG well captures the spatiotemporal characteristic of broadcasting and one broadcasting schedule can be implicitly represented by a subtree of SRG that is rooted from the vertex $\{v_0\}$ and consists of vertices that collectively cover all the nodes in the original topology graph. As an example of multihop broadcasting, Figure 6(a) shows the resulting SRG by performing SRGC-A on the original topology graph in Figure 3(a).

Next, we first define the DLGST and then show that our target problem can be transformed into the DLGST problem. Essentially, the DLGST problem is a variant of the classic *Group Steiner Tree Problem* (GST) [Reich and Widmayer 1990]. Given a weighted graph $G = (V, E)$, a root $r \in V$, and a set of groups where each group is defined as a subset $S \subseteq V$, the classic GST problem is to find a minimum weight r-rooted subtree containing at least one vertex from each group.

Definition 4.3 (*Latency of Tree*). Given a spatiotemporal tree $T = (V, E, W, L)$, the latency of T, say $D(T)$, can be defined as follows:

$$D(T) = \max_{v \in V - \{v_0\}} \{D_T(v_0, v)\}, \tag{6}$$

where v_0 denotes the root of the tree, and $D_T(i, j)$ denotes the End-to-End (E2E) latency from vertex i to vertex j on T.

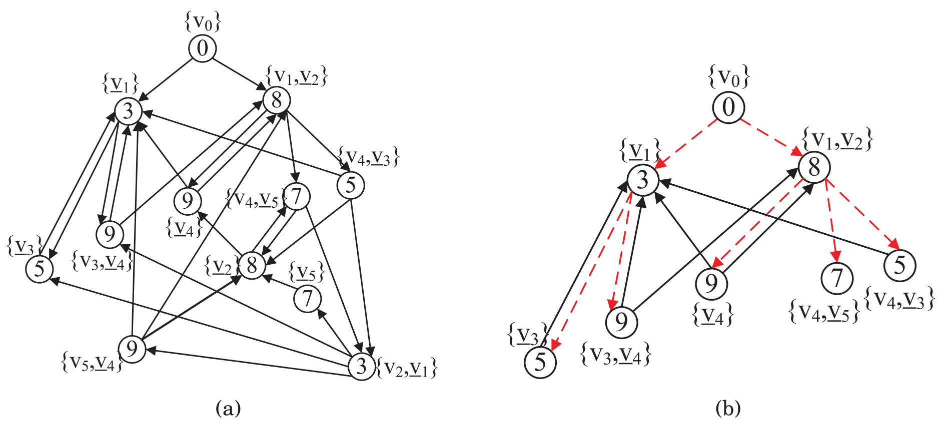

Fig. 6. (a) SRG of the original topology graph in Figure 3(a) (underlined letters denote the *InstantReceivers*). (b) The simplified SRG (dashed edges constitute LGT).

PROBLEM 2 (DLGST). *Given a directed spatiotemporal graph* $G = (V, E, W, L)$ *with weight* $w_e = 1$ *for each directed edge* $e \in E$ *and a family of groups (i.e., subsets of vertices)* $f = \{g_1, g_2, \ldots, g_k\}(g_i \subseteq V)$*, find a directed minimum weight subtree* $T^* = (V^* \subseteq V, E^* \subseteq E, W^* \subseteq W, L)$ *rooted from the root vertex* v_0*, subject to the constraints that*

(1) $V^* \cap g_i \neq \phi$ *for any* $i \in \{1, \ldots, k\}$*;*
(2) $D(T^*)$ *is minimized.*

THEOREM 4.4. *The LMEB problem on the original topology graph is equivalent to the DLGST problem on its corresponding SRG, where the vertices whose coverage sets contains a common sensor node belong to one group.*

4.3. Solution to the DLGST Problem

According to Theorem 4.4, our objective thus turns to solve the DLGST problem on SRG. To this end, we come up with an efficient solution. Overall, we first find a Latency-optimality Guaranteed Tree (LGT) from SRG and approximate our problem as the GST problem on LGT; a deterministic randomized-rounding based algorithm is then proposed to solve this problem.

4.3.1. LGT Construction. Here, we define the Minimum Latency Path Tree (MLPT) in any graph G as a spanning subtree of G where the E2E delay from the root to each vertex is minimal. We can easily have the following conclusion.

THEOREM 4.5. *Under our proposed broadcasting communication model, the minimum broadcasting latency must be equal to* $D(T_{min})$*, where* T_{min} *denotes the MLPT in the original topology graph.*

According to Theorem 4.5, we can further simplify SRG by removing all the vertices whose minimum root-to-vertex latencies are more than $D(T_{min})$ and the associated edges from SRG. This is because our expected subtree of SRG, which represents the latency-optimal broadcasting schedule, will absolutely not include any vertex whose minimum root-to-vertex latency in SRG is more than the optimal broadcasting latency. Thus, our target problem can be further reduced to the DLGST problem on the simplified SRG.

We use $OPT_{GST}(T)$ and $OPT_{DLGST}(G)$ to denote the cost of the optimal solution for the GST on any tree T and that for the DLGST problem on any directed graph G, respectively, and the following conclusion holds.

THEOREM 4.6. *Given a simplified SRG G' where the vertices whose coverage sets contains a common sensor node belongs to a group, we must have*

$$OPT_{GST}(T') \leq h(T') \cdot OPT_{DLGST}(G'), \tag{7}$$

where T' denotes any latency-optimal spanning subtree of G' and $h(T')$ denotes the height of tree T'. Suppose that the parameters R, L, and r_c are fixed, then $h(T')$ must be bounded by a constant.

For any latency-optimal spanning subtree of the simplified SRG, we call it the LGT. According to Theorem 4.6, obviously, we are expected to find a LGT with lower height to achieve a better performance guarantee. Here, we adopt the following approach, which is similar to the *Bellman-Ford Algorithm*, to construct the LGT.

—**Initialization:** Given a simplified SRG G', we let $D_{min}(v'_0, v'_i)$ and $hopcount(v'_0, v'_i)$ denote the minimum E2E latency and the hop count from root v'_0 to vertex v'_i, respectively. Initially, we set $D_{min}(v'_0, v'_0) = hopcount(v'_0, v'_0) = 0$, and set $D_{min}(v'_0, v'_i) = hopcount(v'_0, v'_i) = \infty$ and $p(v'_i) = null$ for any $v'_i \neq v'_0$, where $p(v)$ denotes the parent of vertex v.
—**Iteration:** For each edge (v'_i, v'_j) in G', if $D_{min}(v'_0, v'_i) + d(v'_i, v'_j) < D_{min}(v'_0, v'_j)$, we will update $D_{min}(v'_0, v'_j) = D_{min}(v'_0, v'_i) + d(v'_i, v'_j)$, $hopcount(v'_0, v'_j) = hopcount(v'_0, v'_i) + 1$ and set $p(v'_j) = v'_i$. If $D_{min}(v'_0, v'_i) + d(v'_i, v'_j) = D_{min}(v'_0, v'_j)$, we will check whether $hopcount(v'_0, v'_i) + 1 < hopcount(v'_0, v'_j)$; if so, we update $hopcount(v'_0, v'_j) = hopcount(v'_0, v'_i) + 1$ and set $p(v'_j) = v'_i$. The preceding process is repeated until there is no update in G'.

For the original topology graph with $D(T_{min}) = 17$ (i.e., Figure 3(a)), we can derive the LGT (i.e., Figure 6(b)) by adopting the preceding approach on its SRG (i.e., Figure 6(a)).

4.3.2. Edge Selection on LGT. As seen previously, accordingly, we can approximate our problem as the GST problem on LGT, which has guaranteed the optimality of broadcasting latency. In Garg et al. [1998], the authors proposed an efficient method to address the GST Problem on tree. However, Garg et al. [1998] required that the input tree should be a binary one where each group is a subset of its leaves and groups are pairwise disjoint, and also it only gives a probabilistic solution. Based on the solution in Garg et al. [1998], we devise a deterministic method, which consists of three steps:

(1) *Tree Transformation*

Given a LGT $T' = \{V', E', W', L\}$, we first convert T' into a binary tree in which each group is a subset of its leaves and groups are pairwise disjoint via the following operations:

—For each internal (i.e., nonroot and nonleaf) vertex v'_i in LGT, we insert an edge from vertex v'_i to a newly added vertex v'_{new} sharing the same $S(\cdot)$ and $IR(\cdot)$ with v'_i (i.e., $S(v'_{new}) = S(v'_i)$ and $IR(v'_{new}) = IR(v'_i)$).
—For each leaf vertex v'_i in LGT, if $|S(v'_i)| > 1$, we insert $|S(v'_i)|$ edges from v'_i to $|S(v'_i)|$ newly added vertices sharing the same $S(\cdot)$ and $IR(\cdot)$ with v'_i.
—For each nonleaf vertex v'_i with more than two children, we first add a new vertex v'_{new} sharing the same $S(\cdot)$ and $IR(\cdot)$ with v'_i into LGT; specifically, if v'_i is not the root, we replace v'_i with v'_{new} to be the child of $p(v'_i)$. Then, we delete an edge from v'_i to any

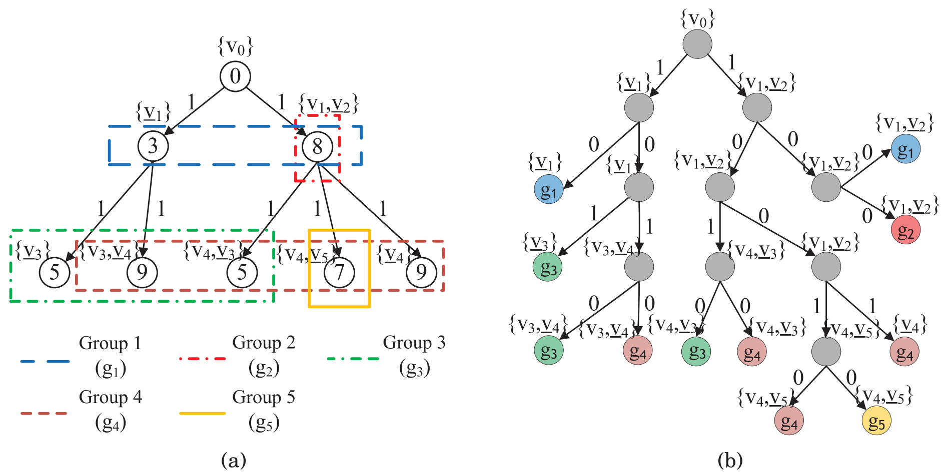

Fig. 7. (a) Illustration of LGT (i.e., T'). (b) Illustration of the transformed binary LGT (i.e., T^*).

child v'_j of v'_i, and insert the edges (v'_{new}, v'_i) and (v'_{new}, v'_j). This process is repeated until a binary tree is fully built.

—We check each edge (v'_i, v'_j) in the preceding binary tree; if $S(v'_i) = S(v'_j)$ and $IR(v'_i) = IR(v'_j)$, we set the weight $w_{(v'_i, v'_j)} = 0$; otherwise, $w_{(v'_i, v'_j)} = 1$.

Here, we partition all the nonroot vertices in LGT T' into $N-1$ groups, that is, $\{g_1, \ldots, g_{N-1}\}$, where any vertex $v' \in V'$ belongs to group g_i if and only if $v_i \in S(v')$ $(i \in \{1, \ldots, N-1\})$. Correspondingly, we also partition all the leaves in the transformed binary LGT into $N-1$ pairwise disjoint groups, which respectively correspond to $\{g_1, \ldots, g_{N-1}\}$. Figure 7 illustrates an example of tree transformation in which the members in one group are marked with the same color. Apparently, we can safely draw the conclusion that *the GST Problem on LGT is equivalent to the minimum weight GST Problem on the transformed binary LGT in which each group g_i ($i \in \{1, \ldots, N-1\}$) is a subset of its leaves and all groups are pairwise disjoint*.

(2) *Randomized Rounding*

Let $T^* = (V^*, E^*)$ be the transformed binary LGT; as shown in Garg et al. [1998], the minimum weight GST Problem on T^* can be formulated as the following 0-1 Integer Programming:

$$
\begin{aligned}
&(IP)\min \sum_{e^* \in E^*} w_{e^*} x_{e^*} \\
&s.t. \sum_{e^* \in \partial S} x_{e^*} \geq 1,\ \forall S \subset V^*\ so\ that\ r \in S \\
&\qquad\quad and\ S \cap g_i = \phi\ for\ some\ i \in \{1, \ldots, N-1\} \\
&\qquad\quad w_{e^*} \in \{0,1\},\ x_{e^*} \in \{0,1\},\ \forall e^* \in E^*,
\end{aligned}
\tag{8}
$$

where r denotes the root vertex of T^*, and ∂S denotes the set of edges with exactly one end point in S.

In the preceding formulation, the binary variable x_{e^*} indicates whether to select the edge e^* or not. Given a group g_i, apparently, it requires that at least one edge with exactly one end point in S should be selected for any vertex set S that separates the root from g_i. Here, x_{e^*} can be relaxed to the range of [0,1] and regarded as the capacity of edge e^*, which implies that any cut that separates the root from all the vertices in a given group has capacity of at least 1. According to the *Max-flow Min-cut Theorem*, the maximum flow from the root to any group must be at least 1. In other words, there must exist a flow whose value is exactly 1 from the root to any group. Thus, we can relax the preceding Integer Programming to the following Linear Programming.

$$\begin{aligned}
&(LP)\min \sum_{(u,v)\in E^*} w_{(u,v)}x_{(u,v)}\\
&s.t. \sum_{u\in g}\sum_{v\in V_g} f_g(v,u)=1\\
&\sum_{(u,v)\in E_g} f_g(u,v)=\sum_{(v,w)\in E_g} f_g(v,w),\ \forall v\in V_g-g-\{r\}\\
&\quad 0\le f_g(u,v)\le x_{(u,v)}\le 1,\ \forall(u,v)\in E_g\\
&\quad w_{(u,v)}\in\{0,1\},\ \forall(u,v)\in E_g\\
&\quad \forall g\in\{g_1,\dots,g_{N-1}\},
\end{aligned} \tag{9}$$

where f_g denotes the flow from the root to group g and $T_g = (V_g, E_g)$ denotes the subtree of T^*, which consists of the paths from the root r to each leaf vertex in group g.

Similar to Garg et al. [1998], we adopt the following approach, which is called the *Randomized-rounding based Edge Selection Algorithm* (RES-A), to make the edge selection.

—We define a *Selected Edge Set*, which is initially set as empty.
—We make the following *random selection operation*: Each edge e^* in T^* is marked with probability of $\frac{x_{e^*}}{x_{p(e^*)}}$, in which x_{e^*} can be figured out from Equation (9) and $p(e^*)$ denotes the parent edge of e^*. For any edge e^* with one end point is the root; specifically, it is marked with probability of x_{e^*}. An edge is added into the *Selected Edge Set* if and only if the edges including itself and all its ancestors are marked.
—We check whether the GST is generated by combining all the edges in the *Selected Edge Set* and the zero-weight edges in T^*; if yes, the edge selection process is terminated; otherwise, we repeat the preceding *random selection operation* until the edge selection is terminated or the *random selection operation* has been repeated for $\lceil \eta \cdot \log(N-1) \cdot \log \max_{1\le i\le N-1} |g_i| \rceil$ rounds, where η is a constant.

The following lemma, which has been proven in Garg et al. [1998], explicitly shows the performance of the aforementioned randomized-rounding based approach.

LEMMA 4.7 [GARG ET AL. 1998]. *For a binary tree in which each group is a subset of its leaves and groups are pairwise disjoint, the probability that its root fails to reach any group g after one round* ***random selection operation*** *is at most about* $1 - \frac{1}{64 \log \max_{1\le i\le N-1} |g_i|}$.

(3) *Edge Compensation and Reduction*

Different from Garg et al. [1998], which only gives a probabilistic solution, we will make sure our solution is deterministic by edge compensation. If the root is not connected to some group g after executing RES-A, specifically, we will establish the minimum weight path from the root to group g and then add the edges on this path that have not been selected by RES-A into the *Selected Edge Set*. Finally, we further reduce the solution on the transformed binary LGT to that on the original LGT by removing all the zero-weight edges from the *Selected Edge Set*.

4.4. Broadcasting Schedule Construction

By adopting the previously mentioned solution, we can approximately obtain the minimum Group Steiner Tree on LGT that consists of the edges in *Selected Edge Set*, called $T^G = (V^G, E^G, W^G, L)$, which implicitly represents a latency-optimal broadcasting schedule that the total number of broadcasting message transmissions is at most $|E^G|$. We can easily find that the broadcasting schedule represented by T^G must

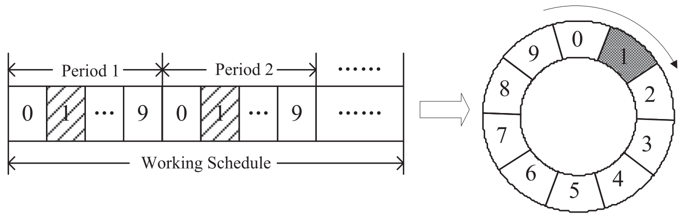

Fig. 8. The wake-up schedule ring with $L = 10$.

satisfy the properties of *connectivity* and *coverage* in Definition 3.2. However, it may not satisfy the property of *nonredundancy* in Definition 3.2; in other words, it could incur redundant transmissions and unnecessary collisions since a randomized approach is adopted in the building of T^G. Next, we will introduce how to transform T^G into the corresponding broadcasting schedule as defined in Definition 3.2, which essentially avoids the redundant transmissions and unnecessary collisions.

For any edge $e_i^G = (v_s^G, v_r^G)$ in E^G, we use $<e_i^G \cdot sender,\ e_i^G \cdot receiver>$ to denote its *Sender-InstantReceiver* pair where $e_i^G \cdot sender \in S(v_s^G)$ and $e_i^G \cdot receiver = IR(v_r^G)$. In Figure 7(a), for example, the edge $(\{v_1, \underline{v_2}\}, \{v_4, \underline{v_3}\})$ is marked with the *Sender-InstantReceiver* pair $<v_1,\ v_3>$. For any sensor node v_i, we use $t_{v_i}^{min}$ and $T_c^{min}(v_i)$ to respectively denote node v_i's *minimum coverage time* and *minimum coverage time slot* (i.e., the time and the corresponding time slot when node v_i is covered for the first time in the schedule T^G). Specifically,

$$t_{v_i}^{\min} = \min_{v \in V^G \ and \ v_i \in S(v)} \left\{ D_{T^G}(r^G, v) \right\}, \tag{10}$$

$$T_c^{\min}(v_i) = T_s\left(\arg \min_{v \in V^G \ and \ v_i \in S(v)} \left\{ D_{T^G}(r^G, v) \right\} \right), \tag{11}$$

where r^G is the root of T^G. For convenience of description, we use a ring to characterize one working schedule period, namely, time slots from 0 to L-1 are distributed in the ring according to the clockwise sequence. Figure 8 shows an example with $L = 10$.

To achieve the transformation from T^G to our target broadcasting schedule, here, we propose a BSC-A that consists of the following two steps.

(1) *Schedule Initialization*

For any node v_i, its schedule strategy $M(v_i)$ can be initially generated from T^G as follows.

—If there does not exist any edge $e_i^G \in E^G$ where $e_i^G \cdot sender = v_i$, we will set $M(v_i) \cdot \alpha = 0$ and $M(v_i) \cdot \beta = NULL$.

—If there exists at least one edge indicating the sender is v_i in E^G, we will mark v_i with the forwarder and $M(v_i) \cdot \beta$ can be built in the following way: For any edge $e_i^G = (v_s^G, v_r^G)$ in E^G where $e_i^G \cdot sender = v_i$, we check whether $S(v_r^G) \subseteq CS(v_i, e_i^G \cdot receiver, T_c^{min}(v_i))$; if yes, we add node $e_i^G \cdot receiver$ into $M(v_i) \cdot \beta$ if it is not in $M(v_i) \cdot \beta$ and mark it with the *InstantReceiver*; otherwise, node v_i' will be added into $M(v_i) \cdot \beta$ if it is not in $M(v_i) \cdot \beta$ and be marked with the *InstantReceiver*, where v_i' is the neighboring node of v_i whose scheduled wake-up time slot is the furthest away from the time slot $T_c^{min}(v_i)$ in the wake-up schedule ring along with the clockwise direction.

—Then, we sort $M(v_i) \cdot \beta$ as $<\underline{\beta_1}, \underline{\beta_2}, \ldots, \underline{\beta_{m(v_i)}}>$ according to the clockwise sequence of their scheduled time slots in the wake-up schedule ring with starting from the time slot $T_c^{min}(v_i)$.

—Finally, we add all the nodes in set $CS(v_i, \beta_{m(v_i)}, T_c^{min}(v_i)) - M(v_i) \cdot \beta$ into $M(v_i) \cdot \beta$ and mark them with the *DelayedReceivers*, and then we reorder $M(v_i) \cdot \beta$ according to the clockwise sequence of their scheduled time slots in the wake-up schedule ring with starting from the time slot $T_c^{min}(v_i)$.

(2) *Schedule Adjustment*

After the previous step, we can get the initial *Forwarding Sequence* of each forwarder. However, the broadcasting schedule based on these initial *Forwarding Sequences* could incur redundant transmissions and unnecessary collisions, since a randomized approach is adopted in the building of T^G. Suppose that we have any three forwarders v_i, v_j, and v_k, of which initial *Forwarding Sequences* can be represented as follows:

$$\begin{aligned} M(v_i) \cdot \beta &= <v_3, \underline{v_5}, v_6, v_7, \underline{v_8}, v_{10}, \underline{v_9}>, \\ M(v_j) \cdot \beta &= <v_2, v_4, \underline{v_6}, v_7, v_{12}, \underline{v_8}>, \\ M(v_k) \cdot \beta &= <\underline{v_5}, v_1, \underline{v_6}, v_{11}, v_8, \underline{v_9}>. \end{aligned} \tag{12}$$

We find that if node v_5 receives the broadcasting message from v_i no later than that from v_k, the transmission from v_k to v_5 will be redundant and thus v_5 can be removed from $M(v_k) \cdot \beta$. In addition to the redundant transmissions, unnecessary collisions could also be inevitable for our derived schedule. For example, the collision would happen when the time v_6 takes to receive the broadcasting message from v_j is the same as that from v_k. If v_6 receives the broadcasting message from v_j no later than that from v_k, actually, we can get an equivalent *Forwarding Sequence* by letting v_1 in $M(v_k) \cdot \beta$ be the *InstantReceiver* and removing v_6 from $M(v_k) \cdot \beta$.

Definition 4.8 (*Remove Back*). Given any *Forwarding Sequence* $M(v_i) \cdot \beta$ that contains node v_j, the operation *Remove Back* v_j in $M(v_i) \cdot \beta$ is defined as follows: (1) If v_j is the *DelayedReceiver*, remove v_j from $M(v_i) \cdot \beta$; (2) If v_j is the *InstantReceiver*, replace v_j with the previous node of v_j in $M(v_i) \cdot \beta$ by the *InstantReceiver* and then remove v_j from it; particularly, if the previous node of v_j is also the *InstantReceiver* or v_j is the first node in $M(v_i) \cdot \beta$, just remove v_j from $M(v_i) \cdot \beta$.

In order to satisfy the property of *nonredundancy* in Definition 3.2, in this step, we propose the following approach to further adjust $M(v_i) \cdot \alpha$ and $M(v_i) \cdot \beta$ values for each node v_i.

—For each nonsink node v_i, we first find the edge $e_i^G = (v_s^G, v_r^G)$ in E^G such that $v_i \in S(v_r^G)$ and $D_{T^G}(r^G, v_r^G) = t_{v_i}^{min}$, and node $e_i^G \cdot$ *sender* is thus selected to be the candidate sender for v_i.
—Then, we check the *Forwarding Sequence* of each forwarder v_j where $v_j \neq e_i^G \cdot$ *sender*; if $v_i \in M(v_j) \cdot \beta$, *Remove Back* v_i in $M(v_j) \cdot \beta$.
—After the previous process, if the new resulting *Forwarding Sequence* of any forwarder v_j is empty, we will set $M(v_j) \cdot \alpha = 0$ and $M(v_j) \cdot \beta = NULL$.
—For each forwarder v_j, finally, we will set $M(v_j) \cdot \alpha$ as the number of *InstantReceivers* in $M(v_j) \cdot \beta$.

Obviously, the aforementioned approach can ensure each sensing node appears in exactly one forwarder's *Forwarding Sequence*, which implies the property of *nonredundancy* in Definition 3.2 must be satisfied. Theorem 4.11 explicitly shows the performance of our solution.

LEMMA 4.9. *Let M^* denote the resulting broadcasting schedule by performing BSC-A on T^G; we can have that*

(1) M^* *must be latency optimal;*

(2) $\sum_{i=0}^{N-1} M^*(v_i) \cdot \alpha \leq |E^G|$.

LEMMA 4.10. $\log \max_{1 \leq i \leq N-1} |g_i| \leq O(\log d_{max})$.

THEOREM 4.11. *When* $\eta \geq 64$*, the approximation ratio of our solution is* $O(\log N \cdot \log d_{max})$*.*

A straightforward observation from Theorem 4.11 is that we can set the parameter η as 64 in our solution to guarantee a polylogarithmic approximation ratio.

4.5. Extension

Note that we assumed the working schedules of neighboring nodes are different from each other, which is commonly seen in low-duty-cycle WSNs. Nevertheless, our solution can also be extended to the generalized case where a few of the neighboring nodes could have the identical wake-up schedule, by simply regarding the set of neighbors having identical wake-up time slot as one *virtual node*. For example, the initial *Forwarding Sequence* of forwarder v_i can be represented as follows.

$$M(v_i) \cdot \beta = <\{v_3, v_5\}, v_6, v_7, \{\underline{v_8}, \underline{v_{10}}, \underline{v_9}\}>, \tag{13}$$

where $\{v_3, v_5\}$ and $\{\underline{v_8}, \underline{v_{10}}, \underline{v_9}\}$ denote two *virtual nodes*, that is, $T_s(v_3) = T_s(v_5)$ and $T_s(v_8) = T_s(v_{10}) = T_s(v_9)$. Here, a *virtual node* is called the *DelayedReceiver* (*InstantReceiver*) if and only if all sensor nodes in this *virtual node* are the *DelayedReceivers* (*InstantReceivers*), and any *InstantReceiver virtual node* in $M(v_i) \cdot \beta$ represents one broadcasting message transmission.

In BSC-A, a *virtual node* is *Removed Back* in the *Forwarding Sequence* of any forwarder v_i if and only if node v_i is not the candidate sender for each node in this *virtual node*. Otherwise, we only need to remove the sensor nodes whose candidate senders are not v_i from the *virtual node*.

5. PRACTICAL ISSUES

In this section, we will discuss the practical issues faced when implementing our proposed solution.

Note that, we make the same assumptions as most of the existing works about broadcast scheduling for low-duty-cycle WSNs; that is, the assumptions made in our article are all commonly used in the existing related works and our solution does NOT bring any additional overhead compared with the existing related works. Actually, these commonly used assumptions will cost much less overhead in practice. For example, we only need to realize a local synchronization between neighboring nodes in our article. In real WSNs, local synchronization can be achieved by using an existing high-efficient MAC-layer time stamping technique, Flooding Time Synchronization Protocol (FTSP) [Maróti et al. 2004], which achieves an accuracy of 2.24us with the cost of exchanging only a few bytes of packets among neighboring nodes every 15 minutes. Since the length of each time slot is usually long enough (at least tens of milliseconds) in practice, the accuracy of 2.24us is sufficient. Also, the assumption that each node is aware of the working schedules of all its neighboring nodes within two hops can be realized by just exchanging the schedules between neighboring nodes twice in the initialization phase of the network. Specifically, each node will initially keep awake and determine its own working schedule according to a particular power management protocol immediately after the deployment, and then exchange the working schedule with its neighbors. Once getting the information about all its neighbors' working schedules, each node will deliver it to each of its neighbors again. In our solution, we will use a binary string to represent the working schedule, for example, to use the binary string <0010> to represent the periodic working schedule with $T_s(\cdot) = 2$ and $L = 4$. In this way, we can find that the exchange cost of the working schedules between neighboring nodes is quite low especially when an efficient string compression scheme is adopted. More importantly, this exchange is only a one-time task during the implementation of our solution. Therefore, this assumption will bring much less overhead in practice.

Upon getting the information about the working schedules of all the neighbors, each node will deliver it to the sink immediately. The sink will derive the spatiotemporal network topology graph based on the collected information from all nodes, then execute our centralized algorithm to obtain the broadcasting schedule and distribute it to each node in the network. This will be done during the initialization phase of the network and is a one-time task. Actually, this is also the commonly used implementation for most of the existing centralized algorithms. Once getting the transmission strategy, each node will put itself into the low-duty-cycle mode according to its own working schedule. Upon receiving a packet, any node v will check its header to see whether it is a broadcast packet; if yes, node v will forward this packet according to its own transmission strategy.

In this article, we assume that our target applications will not experience a notable change on the link qualities, which implies that the topology changes mainly come from the energy depletion of sensing nodes. In practice, some emerging technologies (e.g., Wireless Charging Technology [Dai et al. 2014] and Mobile Robot Technology [Fletcher et al. 2010]) can help us deal with such kind of topology changes. For example, we can set an energy threshold for each node, and any node will transmit an alarm packet to the sink once its residual energy is below this threshold. Upon receiving the alarm packet, the sink will send a mobile charger to the target node and wirelessly recharge it, or send a mobile robot there to replace the target node with a new one that has the same code and configuration as the target node. In this case, we do not need to consider the topology change and the network initialization phase is just a one-time task, which implies the control traffic in the network initialization phase will bring much less overhead compared with the long-term run of the broadcasting applications so that its cost can be approximately neglected.

6. PERFORMANCE EVALUATION

In this section, we evaluate the performance of our solution via simulations.

In our setting, we assume sensor nodes are uniformly distributed in a circular sensory field with a radius of $R = 50$m and the sink node is located at the center of the sensory field. For simplicity, we assume that one period of any node's working schedule contains only one *active state* time slot and we let each sensing node independently and randomly determine its own working schedule. For the sink node v_0, specifically, we set $T_s(v_0) = 0$. Further, we adopt the following classic energy consumption model that is commonly used in much of the existing literature such as Heinzelman et al. [2000]:

$$e_s(l) = l \cdot E_{elec} + l \cdot \varepsilon_{amp} r_c^2, \quad e_r(l) = l \cdot E_{elec}, \tag{14}$$

where $E_{elec} = 50$nJ/bit, $\varepsilon_{amp} = 100$pJ/bit/m^2, l denotes the packet length, and $e_s(l)$ and $e_r(l)$ denote the energy consumed by sending a packet and receiving a packet, respectively. The same as with literature [Wang et al. 2006], we define that each data packet and each beacon packet have a length of 133 bytes and 19 bytes, respectively. Unless otherwise stated, we set $N = 300$, $L = 100$, $r_c = 10$m, and all the results are obtained by averaging over 10 experiments.

Here, we take the following two approaches as the baselines to evaluate the performance of our solution.

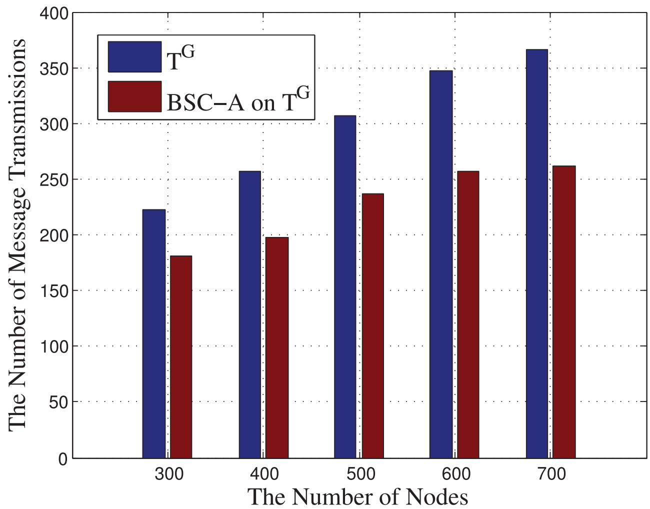

Fig. 9. The number of message transmissions vs. N.

—**MLPT-based approach:** The sink node broadcasts the message directly along with the MLPT of the original topology graph. In this approach, no deferring strategy is employed by each node.

—**Heuristic approach:** Initially, we can find a MLPT from the original topology graph, which essentially represents a broadcasting schedule M^{+}. Given any sensing node v_i and the forwarder v_j where $v_i \in M^{+}(v_j)\cdot\beta$, if v_i is the *InstantReceiver* and also not the last node in $M^{+}(v_j)\cdot\beta$, we will check whether the minimum broadcasting latency still holds when v_i turns to be the *DelayedReceiver* in $M^{+}(v_j)\cdot\beta$; if yes, we will mark v_i with *candidate* and figure out $\Delta_T(v_i)$, which denotes the sum of increased E2E latencies for all nodes if v_i turns to be the *DelayedReceiver*. Then, we will find the *candidate* with the smallest $\Delta_T(\cdot)$ and let it be the *DelayedReceiver*. The previous process will be repeated until no *candidate* can be found. Similar to Section 4.5, this approach can be also extended to the generalized case where a few of the neighboring nodes could have the identical wake-up schedule by simply regarding the set of neighbors having identical wake-up time slot as one *virtual node*.

First, we will evaluate the performance of BSC-A. Let M^{*} denote the solution adopting BSC-A on T^G; Figures 9–11 show that $\sum_{i=0}^{N-1} M^{*}(v_i)\cdot\alpha$ achieves around 10%–30% decrease over the total number of message transmissions for the schedule represented by T^G, which implies the high efficiency of our proposed BSC-A on the reduction of redundant transmissions. It is shown that the vary of parameter N has a more significant affect than L and r_c on the performance of BSC-A. In Figure 9, specifically, we can find that BSC-A will get a much better performance on the reduction of redundant transmissions as the network density increases; this is because the network with higher density will lead to a SRG with a larger number of vertices and edges, which would make the schedule represented by T^G bring more redundant transmissions and provide more opportunities for BSC-A to reduce the number of message transmissions by efficiently avoiding the redundancy of transmissions.

Then, we compare our proposed approximation solution with two baseline solutions in terms of total energy consumption for broadcasting. As shown in Figures 12–14, our solution derives a better performance than both the MLPT-based approach and the heuristic approach under various network configurations; the network density and the

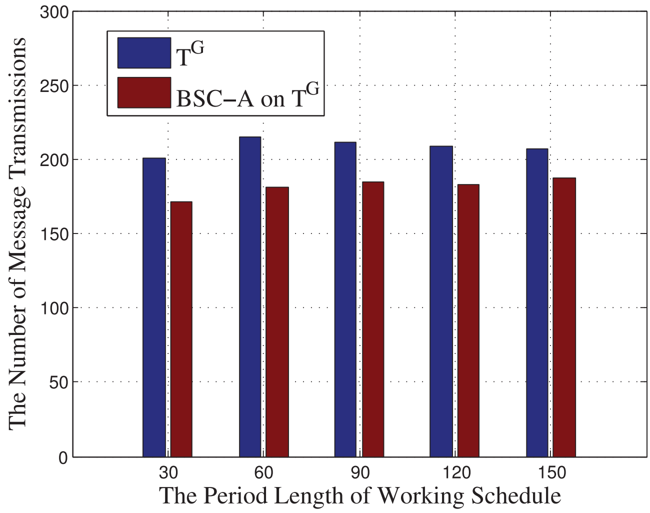

Fig. 10. The number of message transmissions vs. L.

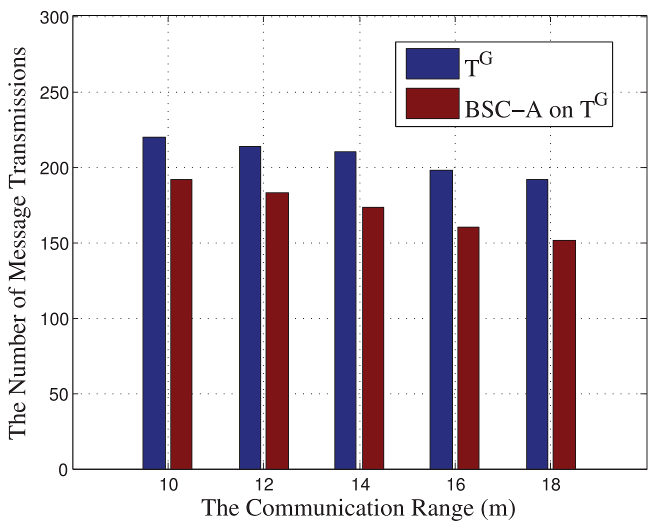

Fig. 11. The number of message transmissions vs. r_c.

communication range of each node affect the performance advantage of our solution to a greater extent compared with the duty cycle. In Figures 12 and 14, we can find that compared with baseline solutions, our solution will get a better performance advantage as the network density or the communication range of each node rises. Figure 13 exhibits the impact of duty cycle on energy efficiency. Specifically, we can find that along with L decreases (i.e., duty cycle increases), all of our solutions and baseline solutions would have a better performance since the decrease of L will make more neighboring nodes have the same working schedule, which would reduce the transmission number of both broadcasting messages and beacon packets. When L is more than 60, however, our solution and baseline solutions will all get a stable performance. Note that our solution exhibits a better performance than the heuristic approach. This is because the considered heuristic approach is based on a structured topology (i.e., MLPT) and the

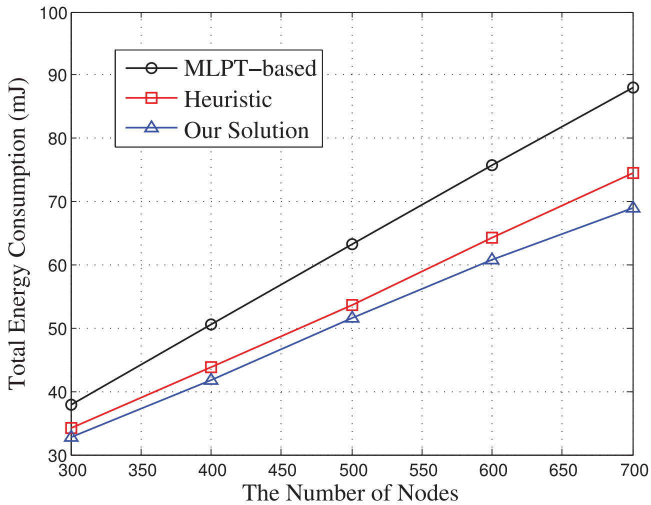

Fig. 12. Total energy consumption vs. N.

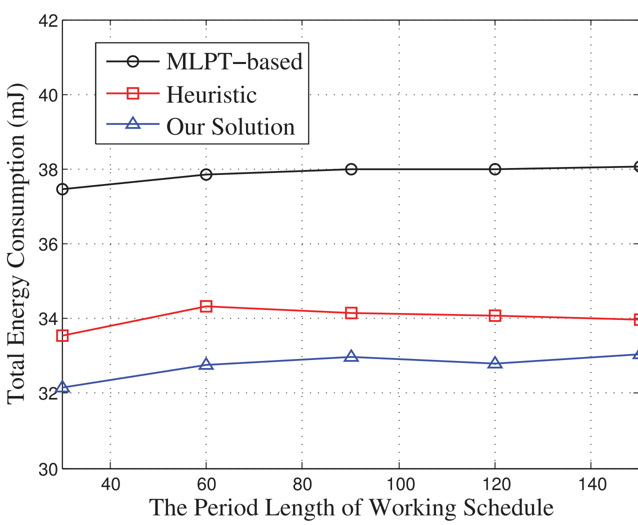

Fig. 13. Total energy consumption vs. L.

result is derived from the greedy updates of broadcast scheduling strategy on MLPT, but our solution is based on an unstructured topology and thus it will have a wider optional range of the feasible solutions than the heuristic approach, on the condition that the broadcasting latency is minimized.

Further, we exhibit the relationship between total energy consumption for broadcasting and the number of packets in each broadcasting message, which is denoted by $|T|$. In many applications for broadcasting, such as *code update*, the broadcasting message is relatively large (i.e., $|T| \gg 1$). As seen from Figure 15, we find that our solution will get a better performance advantage as the broadcasting message is larger, which implies our solution is more suitable for the applications of large message broadcasting.

Note that, the previous experiments are all based on the classic energy consumption model as shown in Equation (14). However, the existing literature [Wang et al. 2006]

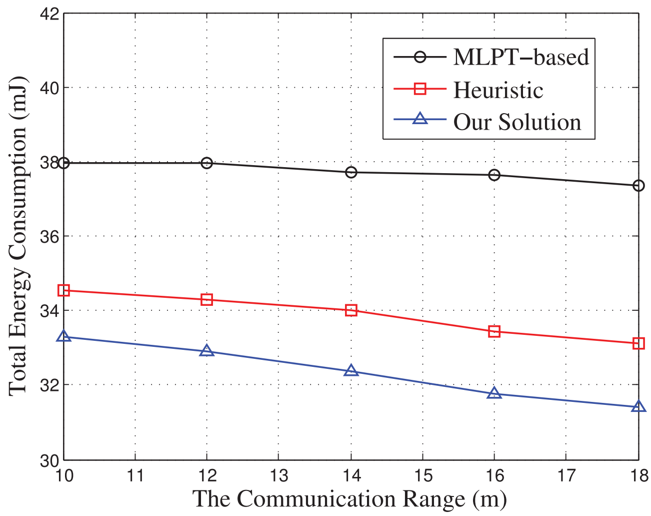

Fig. 14. Total energy consumption vs. r_c.

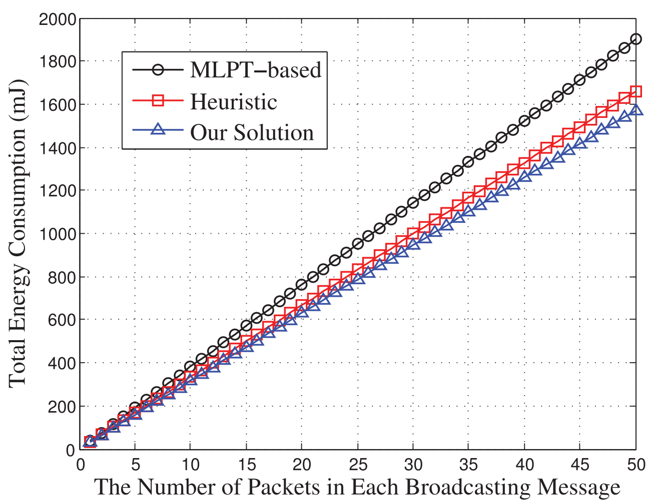

Fig. 15. Total energy consumption vs. $|T|$.

has found that the drain efficiencies of power amplifiers of current sensor node devices are actually always less than 100% and it gives a more realistic energy consumption model as shown in Equation (15). For each sensor device, the drain efficiency of power amplifier denotes the ratio of RF output power to DC input power.

$$e_s(l) = l \cdot E_{elec} + \frac{l \cdot \varepsilon_{amp} r_c^2}{\eta}, \quad e_r(l) = l \cdot E_{elec}, \tag{15}$$

where the parameter η denotes the drain efficiency of power amplifier and its value depends on the specific device. In order to evaluate the adaptivity of our solution to the various energy consumption patterns, we compare our solution with the baseline solutions under the realistic energy consumption models with different η values. Figure 16 shows that our solution will always perform better than the baselines under whatever

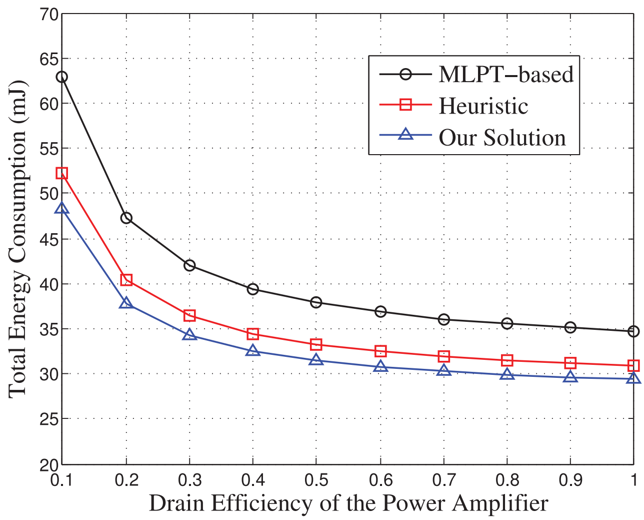

Fig. 16. Total energy consumption vs. η.

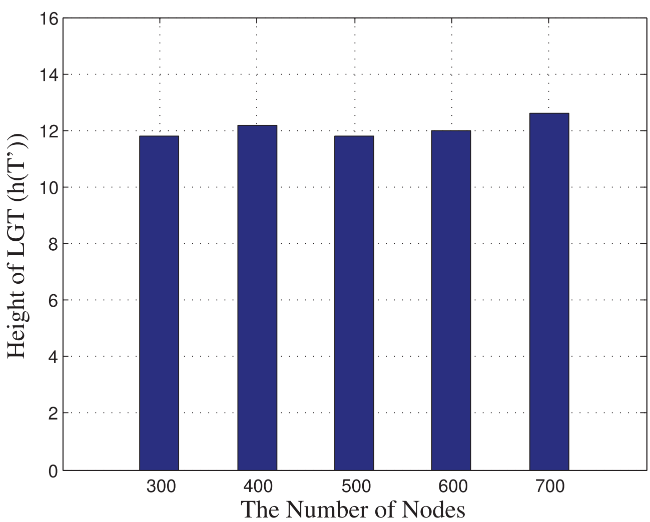

Fig. 17. Height of LGT vs. N.

energy consumption pattern, and the performance advantage will not be degraded no matter how η varies.

Next, we proceed to evaluate the value of $h(T')$ where T' denotes LGT. We respectively consider the following three cases of network configurations: ($L = 100$, $r_c = 10$m), ($N = 300$, $r_c = 10$m), and ($L = 100$, $N = 300$). Figures 17 and 18 exhibit similar results; that is, the value of $h(T')$ almost keeps stable, namely, around 12, no matter how the number of nodes or the length of working schedule varies. As shown in Figure 19, however, $h(T')$ drops as the communication range of each node increases, which means it is only related to the communication range of each node on the condition that R is fixed. Intuitively, this is because the value of $h(T')$ can be approximately considered as the ratio of the latency of MLPT on the original topology graph, which is generally

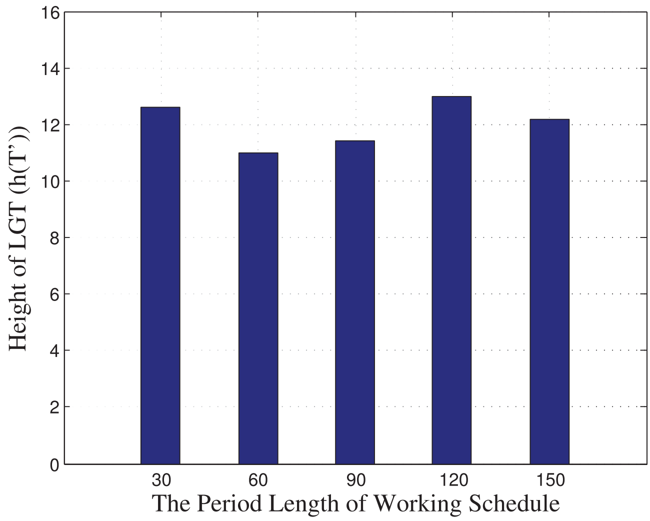

Fig. 18. Height of LGT vs. L.

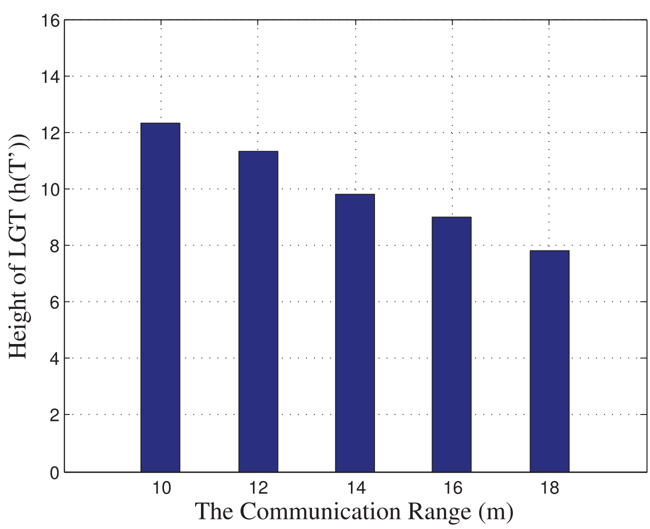

Fig. 19. Height of LGT vs. r_c.

determined by the height of MLPT and average one-hop latency on MLPT, to the average one-hop latency on LGT. Actually, average one-hop latency on MLPT and that on LGT have the same order of magnitude once L is fixed, and the height of MLPT is generally determined by R and r_c. According to the simulation results depicted in Figures 17–19, we can obviously find that the value of $h(T')$ is always a small constant without being related to N under whatever network configuration.

7. CONCLUSIONS

In this article, we consider how to utilize broadcasting spatiotemporal locality to address the broadcast scheduling problem in low-duty-cycle WSNs. We first transform our

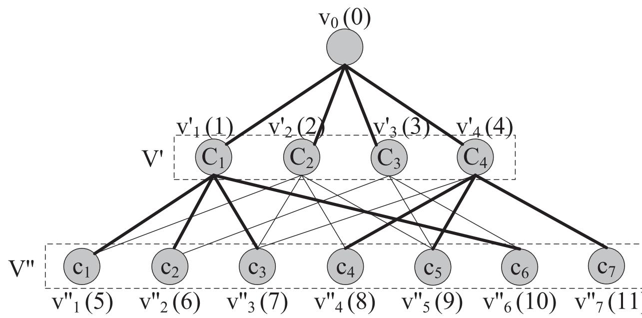

Fig. 20. An instance of the LMEB problem with $L = 12$. Given an optimal broadcasting schedule $M = \{M(v_0) = (1, <v'_1, v'_2, v'_3, \underline{v'_4}>), M(v'_1) = (1, <v''_1, v''_2, v''_3, \underline{v''_6}>), M(v'_4) = (1, <v''_4, v''_5, \underline{v''_7}>), M(v'_2) = M(v'_3) = M(v''_j) = (0, NULL)(j = 1, \ldots, 7)\}$, we can easily get the optimal solution of the *Set Cover Problem*, that is, $\{C_1, C_4\}$.

target problem into the Latency-optimal Group Steiner Tree Problem on the Spatiotemporal Relationship Graph, which is shown to be NP-hard, and then approximately solve this problem by using a deterministic randomized-rounding based method. Also, a novel BSCA is proposed to further avoid the redundant transmissions and reduce the collision probability as much as possible. Finally, the high efficiency of our solution is evaluated through both theoretical analysis and simulations.

APPENDIX

A. PROOF OF THEOREM 3.3

Given any instance of the *Set Cover Problem* with $U = \{c_1, \ldots, c_m\}$ and a collection of sets $\{C_1, \ldots, C_n\}$ where any set $C_i \subseteq U$, we can construct an instance $\widetilde{G} = (\widetilde{V}, \widetilde{E}, \widetilde{W}, \widetilde{L})$ of our LMEB problem in polynomial time by letting (1) $\widetilde{V} = \{v_0\} \bigcup V' \bigcup V''$, where $V' = \{v'_1, \ldots, v'_n\}$ and $V'' = \{v''_1, \ldots, v''_m\}$, corresponding naturally to $\{C_1, \ldots, C_n\}$ and $\{c_1, \ldots, c_m\}$; (2) $\widetilde{E} = (\{v_0\} \times V') \bigcup E'$, where the edge $(v'_i, v''_j) \in E'$ if and only if $v'_i \in V'$, $v''_j \in V''$, and $c_j \in C_i$; (3) $T_s(v_0) = 0$, $T_s(v'_i) = i$ for any $i \in \{1, \ldots, n\}$, $T_s(v''_j) = n + j$ for any $j \in \{1, \ldots, m\}$; (4) $\widetilde{L} = m + n + 1$. We give an example of the instance construction. For the instance of the *Set Cover Problem* with $m = 7$, $n = 4$ and $C_1 = \{c_1, c_2, c_3, c_6\}$, $C_2 = \{c_1, c_3, c_4, c_5\}$, $C_3 = \{c_2, c_5, c_6\}$, $C_4 = \{c_3, c_4, c_5, c_7\}$, we can obtain its corresponding instance of the LMEB problem in Figure 20.

It is easily seen that the minimum broadcasting latency in $\widetilde{G}$ must be $m + n$. In order to guarantee the minimum broadcasting latency, all nodes in V' must receive the message from v_0 since $T_s(v'_i) < T_s(v''_j)$ for any $v'_i \in V'$ and $v''_j \in V''$. In other words, if M is the optimal broadcasting schedule in $\widetilde{G}$, then $M(v_0)$ must be $(1, <v'_1, \ldots, \underline{v'_n}>)$. For any forwarder v'_i in V', obviously, the optimal forwarding strategy must be to defer the receiving time of its neighbors in V'' so that they can simultaneously wake up at time slot $\max_{v \in V''\ and\ (v'_i, v) \in E'}\{T_s(v)\}$ and only one message transmission is needed. Accordingly, we can easily prove that the *Set Cover Problem* can be solved in polynomial time if and only if the latency-optimal minimum energy broadcasting schedule can be found from $\widetilde{G}$ in polynomial time. As this proof is simple and direct, we omit the detailed process for saving space. Thus, the *Set Cover Problem*, which is a well-known NP-hard problem, is polynomial time reducible to the LMEB problem. This indicates that the LMEB problem must be NP-hard.

B. PROOF OF THEOREM 4.2

Seeing from SRGC-A, we find that the vertex search operation actually dominates the whole construction procedure of SRG. In SRGC-A, it results in at most d_{max}^2 SRG vertices for any node in G where d_{max} denotes the maximum node degree in G, thus, there are in total at most $N \cdot d_{max}^2$ vertices in SRG. We assume there are ultimately x vertices in SRG after executing SRGC-A ($x \leq N \cdot d_{max}^2$). As we know, the size of any vertex's coverage set is at most d_{max}, which means any vertex will result in at most d_{max}^2 edges and the total number of edges in SRG is thus at most $x \cdot d_{max}^2$. Actually, we can divide all the edges in SRG into two categories: (1) $x-1$ edges, which are connected to the new added vertices after the search operation; and (2) $x \cdot d_{max}^2 - x + 1$ edges, which are connected to the existing vertices after the search operation. The total search time for the first category is at most $\sum_{m=0}^{x-2} m = \frac{(x-2)(x-1)}{2}$, and that for the second category is at most $(x-1)(x \cdot d_{max}^2 - x + 1)$.

Due to the fact that $x \leq N \cdot d_{max}^2$, the total search time of SRGC-A is therefore at most $\frac{(x-2)(x-1)}{2} + (x-1)(x \cdot d_{max}^2 - x + 1) = O(x^2 d_{max}^2) \leq O(N^2 d_{max}^6)$.

C. PROOF OF THEOREM 4.4

In SRG, we can partition all the nonroot vertices into $N-1$ groups according to the common members involved in their coverage sets. Here, we let G denote the original topology graph and G' denote its corresponding SRG. For any node v_i in G, specifically, any vertex v' in G' belongs to group g_i if and only if $v_i \in S(v')$, where $i \in \{1, \ldots, N-1\}$. Consequently, one broadcasting schedule can be implicitly represented by a subtree of SRG that is rooted from the vertex $\{v_0\}$ and connects at least one vertex in each group of SRG. We can easily find that for any latency-optimal broadcast scheduling M in G, it can be characterized by a corresponding Latency-optimal Group Steiner Tree $T = (V_T, E_T)$ in G', where $\sum_{i=0}^{N-1} M(v_i) \cdot \alpha = |E_T|$. Also, it is easily seen that for any Latency-optimal Group Steiner Tree $T = (V_T, E_T)$ in G', it can be transformed into a latency-optimal broadcast scheduling M, where $\sum_{i=0}^{N-1} M(v_i) \cdot \alpha \leq |E_T|$. This indicates the cost of the optimal solution for the LMEB problem on G must be equal to that for the DLGST problem on G'. Thus, the proof is completed.

D. PROOF OF THEOREM 4.5

Actually, T_{min} can be regarded as a broadcasting schedule without waiting. Assume that node v_b is the leaf on T_{min} of which sink-to-node latency $D_T(v_0, v_b)$ is the maximum overall. Obviously, we cannot find a broadcasting schedule whose latency is less than $D_T(v_0, v_b)$, given that the schedule guarantees v_b is covered. This is because the E2E latency will not benefit from waiting in duty-cycled WSNs, which has been shown in Lai and Ravindran [2010a]. Thus, the optimal broadcasting latency must be $D_T(v_0, v_b)$. As $D_T(v_0, v_b) = D(T_{min})$, the proof is completed.

E. PROOF OF THEOREM 4.6

We denote by $T_{opt} = (V_T, E_T)$ the optimal solution of our target problem. Given any latency-optimal spanning subtree T', it is easy to see that the subtree of T' containing all the vertices in $V_T - \{r\}$ (r denotes the root vertex), say $\widetilde{T} = (\widetilde{V}, \widetilde{E})$, must be a feasible solution. For the worst case in which all the vertices in $V_T - \{r\}$ are just the leaves of T', we have $OPT_{GST}(T') \leq |\widetilde{E}| \leq \sum_{i \in V_T - \{r\}} |Path(r, i)| \leq h(T') \cdot |V_T - \{r\}| = h(T') \cdot |E_T| = h(T') \cdot OPT_{DLGST}(G')$, where $|Path(r, i)|$ denotes the hop count of the path from root r to vertex i. As all nodes are assumed to be uniformly and densely deployed in the sensory field, we can easily find that the latency of MLPT in the original topology graph, that is, the optimal broadcasting latency according to Theorem 4.5, is at most about $\frac{R \cdot L}{r_c}$ time slots where r_c denotes the communication range of each node, and obviously each hop in T' will cost at least one time slot, which implies $h(T')$ must be at most $\xi = \frac{R \cdot L}{r_c}$, a constant independent of N. As shown in our simulation results, indeed, $h(T')$ is always a small value that is far less than ξ in practice.

F. PROOF OF LEMMA 4.9

For any broadcasting schedule M and any sensor node v_i, we let $t_M(v_i)$ denote the time when v_i receives the broadcasting message in M. Obviously, we can have the following observation: *Given two broadcasting schedules* $\{M_1, M_2\}$ *and a sensor node* v_i*, if* $M_1(v_i) \cdot \beta = M_2(v_i) \cdot \beta$ *and* $t_{M_1}(v_i) \leq t_{M_2}(v_i)$*, we must have* $M_1(v_i) \cdot \alpha = M_2(v_i) \cdot \alpha$ *and* $t_{M_1}(v) \leq t_{M_2}(v)$ *for each* $v \in M_1(v_i) \cdot \beta$.

According to the previous observation, we can easily find that $t_{M^*}(v_i) \leq t_{v_i}^{min}$ for each node v_i; this is because (1) in the *Schedule Initialization Step*, we choose $t_{v_j}^{min}$ as the starting forwarding time of any forwarder v_j; and (2) in the *Schedule Adjustment Step*, we choose the sender that lets any node v_i be covered at time $t_{v_i}^{min}$ as the candidate sender of v_i, and also the operation of *Remove Back* would further shorten the *minimum coverage time* of some nodes. As the minimum broadcasting latency must be $\max_{v_i \in V^G}\{t_{v_i}^{min}\}$, we thus have

$$\max_{v_i \in V^G}\{t_{M^*}(v_i)\} = \max_{v_i \in V^G}\left\{t_{v_i}^{min}\right\},$$

which indicates M^* must be latency optimal. In addition, we can find that (1) in the *Schedule Initialization Step*, no operation would bring additional transmissions, and when there exist two or more edges in E^G that have the same *Sender-InstantReceiver* pair, the number of transmissions could be reduced; and (2) in the *Schedule Adjustment Step*, the operation of *Remove Back* would further reduce the redundant transmissions and no additional transmissions will be produced. Thus, we must have

$$\sum_{i=0}^{N-1} M^*(v_i) \cdot \alpha \leq |E^G|.$$

The proof is thus completed.

G. PROOF OF LEMMA 4.10

For any sensor node v_i, according to SRGC-A, we find that any one of its neighbors could generate at most $\widehat{N_{max}} = \sum_{i=1}^{d_{max}} i$ SRG vertices that contain v_i, therefore, the total number of the SRG vertices that contain v_i is at most $|N(v_i)| \cdot \widehat{N_{max}} \leq d_{max} \cdot \sum_{i=1}^{d_{max}} i$, which implies $\max_{1 \leq i \leq N-1} |g_i| \leq \frac{d_{max}^3 + d_{max}^2}{2} \leq d_{max}^3$, so we have

$$\log \max_{1 \leq i \leq N-1} |g_i| \leq \log d_{max}^3 = O(\log d_{max}). \tag{16}$$

The proof is thus completed.

H. PROOF OF THEOREM 4.11

As Equation (9), of which the optimal solution is equal to the expected cost (i.e., the expected number of selected edges) $\mathbf{E}[cost/round]$ after each round of RES-A, is the LP-relaxation of Equation (8) of which the optimal solution is equal to $OPT_{GST}(T^*)$, we must have $\mathbf{E}[cost/round] \leq OPT_{GST}(T^*)$. Also, we observe that for any group g, the number of the added edges in the *Edge Compensation* step after executing RES-A, called N_g, will not exceed $OPT_{GST}(T^*)$ since the minimum weight path from the root to group g in the *Edge Compensation* step must be no longer than the path from the root to g that belongs to the optimal GST on T^*. Further, we use A to denote the event that the root fails to reach any group g after executing RES-A. According to Lemma 4.7, we have

$$\begin{aligned}\mathbf{Pr}[A] &\leq \left(1 - \frac{1}{64 \log \max\limits_{1\leq i\leq N-1} |g_i|}\right)^{\lceil \eta \cdot \log(N-1) \cdot \log \max_{1\leq i\leq N-1} |g_i| \rceil} \\ &\approx \left(1 - \frac{1}{64 \log \max\limits_{1\leq i\leq N-1} |g_i|}\right)^{(64 \log \max_{1\leq i\leq N-1} |g_i|) \cdot \frac{\eta \cdot \log(N-1)}{64}}.\end{aligned} \tag{17}$$

Due to the fact that $\lim_{x\to\infty}(1 - \frac{1}{x})^x = e^{-1}$ and $\log \max_{1\leq i\leq N-1} |g_i| \geq 1$, we can find that if $\eta \geq 64$, then

$$\mathbf{Pr}[A] \leq e^{-\frac{\eta \cdot \log(N-1)}{64}} \leq e^{-\log(N-1)} \leq e^{\log \frac{1}{N-1}} \leq \frac{1}{N-1}. \tag{18}$$

Let $\widetilde{N}$ denote the number of groups that fail to reach the root after executing RES-A, thus, we have $\mathbf{E}[\widetilde{N}] = (N-1)\mathbf{Pr}[A] \leq 1$ since the $N-1$ groups in T^* are disjoint and independent. Let $\Delta = \lceil \eta \cdot \log(N-1) \cdot \log \max_{1\leq i\leq N-1} |g_i| \rceil$; due to Theorem 4.6 and the fact that $OPT_{GST}(T^*) = OPT_{GST}(T')$, the expected cost of the solution T^G, namely, $\mathbf{E}[|E^G|]$, is thus at most

$$\begin{aligned}&\Delta \cdot \mathbf{E}[cost/round] + \mathbf{E}[\widetilde{N}] \cdot N_g \\ &\quad \leq (\Delta + 1) \cdot OPT_{GST}(T^*) \\ &\quad = (\Delta + 1) \cdot OPT_{GST}(T') \\ &\quad \leq h(T') \cdot (\Delta + 1) \cdot OPT_{DLGST}(G') \\ &\quad \leq \xi \cdot (\Delta + 1) \cdot OPT_{DLGST}(G').\end{aligned} \tag{19}$$

For the final solution M^* resulted from BSC-A, according to Lemma 4.9 and Lemma 4.10, we have

$$\mathbf{E}\left[\sum_{i=0}^{N-1} M^*(v_i) \cdot \alpha\right] \leq \mathbf{E}[|E^G|] \leq O(\log N \cdot \log d_{max}) \cdot OPT_{DLGST}(G'). \tag{20}$$

The proof is thus completed.

REFERENCES

Qing Cao, Tarek Abdelzaher, Tian He, and John Stankovic. 2005. Towards optimal sleep scheduling in sensor networks for rare-event detection. In *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN)*. IEEE, 20–27.

Long Cheng, Yu Gu, Tian He, and Jianwei Niu. 2013. Dynamic switching-based reliable flooding in low-duty-cycle wireless sensor networks. In *INFOCOM, 2013 Proceedings IEEE*. IEEE, 1393–1401.

Haipeng Dai, Xiaobing Wu, Guihai Chen, Lijie Xu, and Shan Lin. 2014. Minimizing the number of mobile chargers for large-scale wireless rechargeable sensor networks. *Computer Communications* 46 (2014), 54–65.

Greg Fletcher, Xu Li, Amiya Nayak, and Ivan Stojmenovic. 2010. Randomized robot-assisted relocation of sensors for coverage repair in wireless sensor networks. In *2010 IEEE 72nd Vehicular Technology Conference Fall (VTC 2010-Fall)*. IEEE, 1–5.

Naveen Garg, Goran Konjevod, and R. Ravi. 1998. A polylogarithmic approximation algorithm for the group Steiner tree problem. In *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. Society for Industrial and Applied Mathematics, 253–259.

Yu Gu and Tian He. 2007. Data forwarding in extremely low duty-cycle sensor networks with unreliable communication links. In *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems (SenSys)*. ACM, 321–334.

Shuo Guo, Yu Gu, Bo Jiang, and Tian He. 2009. Opportunistic flooding in low-duty-cycle wireless sensor networks with unreliable links. In *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking (Mobicom)*. ACM, 133–144.

Shuo Guo, Song Min Kim, Ting Zhu, Yu Gu, and Tian He. 2011. Correlated flooding in low-duty-cycle wireless sensor networks. In *Proceedings of the 19th IEEE International Conference on Network Protocols (ICNP)*. IEEE, 383–392.

Kai Han, Yang Liu, and Jun Luo. 2013a. Duty-cycle-aware minimum-energy multicasting in wireless sensor networks. *IEEE/ACM Transactions on Networking (TON)* 21, 3 (2013), 910–923.

Kai Han, Jun Luo, Yang Liu, and Athanasios V. Vasilakos. 2013b. Algorithm design for data communications in duty-cycled wireless sensor networks: A survey. *IEEE Communications Magazine* 51, 7 (2013).

Kai Han, Liu Xiang, Jun Luo, Mingjun Xiao, and Liusheng Huang. 2013c. Energy-efficient reliable data dissemination in duty-cycled wireless sensor networks. In *Proceedings of the 14th ACM International Symposium on Mobile Ad Hoc Networking andComputing*. ACM, 287–292.

Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. 2000. Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences (HICSS)*. IEEE, 1–10.

Jue Hong, Jiannong Cao, Wenzhong Li, Sanglu Lu, and Daoxu Chen. 2010. Minimum-transmission broadcast in uncoordinated duty-cycled wireless ad hoc networks. *IEEE Transactions on Vehicular Technology (TVT)* 59, 1 (2010), 307–318.

Xianlong Jiao, Wei Lou, Junchao Ma, Jiannong Cao, Xiaodong Wang, and Xingming Zhou. 2010. Duty-cycle-aware minimum latency broadcast scheduling in multi-hop wireless networks. In *Proceedings of the 30th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 754–763.

Pradeep Kyasanur, Romit Roy Choudhury, and Indranil Gupta. 2006. Smart gossip: An adaptive gossip-based broadcasting service for sensor networks. In *2006 IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. IEEE, 91–100.

Shouwen Lai and Binoy Ravindran. 2010a. On distributed time-dependent shortest paths over duty-cycled wireless sensor networks. In *Proceedings of INFOCOM*. IEEE, 1–9.

Shouwen Lai and Binoy Ravindran. 2010b. On multihop broadcast over adaptively duty-cycled wireless sensor networks. In *Proceedings of the International Conference on Distributed Computing in Sensor Systems (DCOSS)*. Springer, 158–171.

Mo Li and Yunhao Liu. 2009. Underground coal mine monitoring with wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)* 5, 2 (2009), 10:1–10:29.

Mo Li, Zheng Yang, and Yunhao Liu. 2013. Sea depth measurement with restricted floating sensors. *ACM Transactions on Embedded Computing Systems (TECS)* 13, 1 (2013), 1:1–1:21.

Zhenjiang Li, Junliang Liu, and Shaojie Tang. 2011. Understanding the flooding in low-duty-cycle wireless sensor networks. In *Proceedings of the International Conference on Parallel Processing (ICPP)*. IEEE, 673–682.

Yunhao Liu, Yuan He, Mo Li, Jiliang Wang, Kebin Liu, and Xiangyang Li. 2013a. Does wireless sensor network scale? A measurement study on GreenOrbs. *IEEE Transactions on Parallel and Distributed Systems (TPDS)* 24, 10 (2013), 1983–1993.

Yunhao Liu, Xufei Mao, Yuan He, Kebin Liu, Wei Gong, and Jiliang Wang. 2013b. CitySee: Not only a wireless sensor network. *IEEE Network* (2013), 42–47.

Miklós Maróti, Branislav Kusy, Gyula Simon, and Ákos Lédeczi. 2004. The flooding time synchronization protocol. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*. ACM, 39–49.

Jianwei Niu, Long Cheng, Yu Gu, Junghyun Jun, and Qingquan Zhang. 2013. Minimum-delay and energy-efficient flooding tree in asynchronous low-duty-cycle wireless sensor networks. In *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 1261–1266.

Gabriele Reich and Peter Widmayer. 1990. Beyond Steiner's problem: A VLSI oriented generalization. In *Graph-Theoretic Concepts in Computer Science*. Springer, 196–210.

Lu Su, Bolin Ding, Yong Yang, Tarek F. Abdelzaher, Guohong Cao, and Jennifer C. Hou. 2009. Ocast: Optimal multicast routing protocol for wireless sensor networks. In *Proceedings of the 17th IEEE International Conference on Network Protocols (ICNP)*. IEEE, 151–160.

Yanjun Sun, Omer Gurewitz, Shu Du, Lei Tang, and David B. Johnson. 2009. ADB: An efficient multihop broadcast protocol based on asynchronous duty-cycling in wireless sensor networks. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys)*. ACM, 43–56.

Yanjun Sun, Omer Gurewitz, and David B. Johnson. 2008. RI-MAC: A receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks. In *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems (SenSys)*. ACM, 1–14.

Feng Wang and Jiangchuan Liu. 2009. Duty-cycle-aware broadcast in wireless sensor networks. In *Proceedings of INFOCOM*. IEEE, 468–476.

Jiliang Wang, Zhenjiang Li, Mo Li, Yunhao Liu, and Zheng Yang. 2013. Sensor network navigation without locations. *IEEE Transactions on Parallel and Distributed Systems (TPDS)* 24, 7 (2013), 1436–1446.

Qin Wang, Mark Hempstead, and Woodward Yang. 2006. A realistic power consumption model for wireless sensor network devices. In *Proceedings of the 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks (SECON)*, Vol. 1. IEEE, 286–295.

Lijie Xu, Jiannong Cao, Shan Lin, Haipeng Dai, Xiaobing Wu, and Guihai Chen. 2013. Energy-efficient broadcast scheduling with minimum latency for low-duty-cycle wireless sensor networks. In *2013 IEEE 10th International Conference on Mobile Ad-Hoc and Sensor Systems (MASS)*. IEEE, 163–167.

Lijie Xu and Guihai Chen. 2013. LBcast: Load-balanced broadcast scheduling for low-duty-cycle wireless sensor networks. In *2013 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 7–12.

Ting Zhu, Ziguo Zhong, Tian He, and Zhi-Li Zhang. 2010. Exploring link correlation for efficient flooding in wireless sensor networks. In *Proceedings of NSDI*. USENIX, 49–64.

Received May 2014; revised January 2015; accepted March 2015