



Towards energy-fairness for broadcast scheduling with minimum delay in low-duty-cycle sensor networks



Lijie Xu^{a,b,*}, Xiaojun Zhu^{b,c}, Haipeng Dai^b, Xiaobing Wu^b, Guihai Chen^{b,d}

^a School of Computer Science and Technology, Nanjing University of Posts and Telecommunications, Nanjing 210023, China

^b State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China

^c College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China

^d Shanghai Key Laboratory of Scalable Computing and Systems, Shanghai Jiao Tong University, Shanghai, China

ARTICLE INFO

Article history:

Received 5 December 2013

Revised 22 January 2015

Accepted 8 September 2015

Available online 14 September 2015

Keywords:

Wireless sensor networks

Low-duty-cycle

Broadcast scheduling

Load balancing

Energy efficient

ABSTRACT

Broadcast scheduling for low-duty-cycle wireless sensor networks (WSNs) has been extensively studied recently. However, existing solutions mainly focused on optimizing delay and (or) total energy consumption without considering load distribution among nodes. Due to limited energy supply for sensor nodes, heavily loaded sensors often run out of energy quickly, reducing the lifetime of the whole network. In this paper, we target at minimizing the maximum transmission load of a broadcast schedule for low-duty-cycle WSNs, subject to the constraint that each node should have the minimum end-to-end delay under the broadcast schedule. We prove that it is NP-hard to find the optimal schedule. Then, we devise a *Load-Balanced Parents Assignment Algorithm* (LBPA-A) that achieves λ -approximation ratio, where λ denotes the maximum number of neighbors that are scheduled to wake up at the same time and is typically a small number in low-duty-cycle WSNs. Further, we introduce how to solve this problem in a distributed manner. Our simulation results reveal that compared with the traditional solutions, our proposed LBPA-A and distributed solution both exhibit much better average performance in terms of energy-fairness, total energy consumption and delivery ratio.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

As an important fundamental function, multi-hop broadcasting in wireless sensor networks (WSNs) has been extensively studied in the past few years. Many applications in WSNs, e.g., environmental monitoring, medical care system and scientific exploration, require that the sink node should disseminate the system configurations to all sensor nodes in a timely and energy-efficient manner. In practice, it has been verified [1] that idle listening is the major source of energy waste in WSNs. As shown in [2], the commonly adopted ChipCon CC2420 radio draws 18.8 mA at receiving mode or idle listening mode, and draws 17.4 mA at sending mode, which implies that idle listening actually consumes approximately the same amount of power as in receiving and sending mode. In order to significantly reduce the energy waste caused by idle listening, sensor nodes are often put in a low-duty-cycle mode where every sensor

node has its own working schedule to alternate periods of work with sleep.

How to optimize the energy efficiency of broadcasting in low-duty-cycle WSNs has been well-investigated by many existing works, in which the proposed solutions can achieve high energy efficiency in terms of total energy consumption. However, it may not be appropriate to take total energy consumption as the main metric to characterize energy efficiency in sensor networks. As we know, it is typically hard to replace or recharge batteries for sensor nodes as many WSNs are deployed in a tough environment that human beings are not easy to access to. This fact implies it is more important for sensor networks to take energy-fairness as the first concern, since unbalanced load could make nodes with heavy workload deplete their energy much faster so that the network is disabled earlier, e.g., the monitoring field cannot be fully covered, or the network becomes disconnected. Currently, most of the existing works mainly pay attention to load-balanced *data collection* in low-duty-cycle WSNs. However, very few of them consider load balancing of *data dissemination*, which is an important function in sensor networks. In low-duty-cycle WSNs, one-hop broadcasting is usually implemented by multiple unicasts. Once the broadcasting schedule is established, it is normally performed for a rather long duration and is unlikely

* Corresponding author. Tel.: +8613611588773.

E-mail addresses: ljxu83@gmail.com, huoye3333@163.com (L. Xu), xzhu@nuaa.edu.cn (X. Zhu), dhpphd2003@gmail.com (H. Dai), wuxb@nju.edu.cn (X. Wu), gchen@nju.edu.cn (G. Chen).

to be updated frequently due to high update overhead. Therefore, a broadcasting schedule without careful design may lead to highly unbalanced broadcasting load among sensor nodes. As the number of broadcasting increases, this unbalancing will be further intensified. For low-duty-cycle WSNs where load-balanced data collection are adopted, designing a broadcasting schedule to guarantee energy fairness among sensor nodes is an important issue.

Compared with always-awake networks, low-duty-cycle sensor networks usually yield a notable increase on communication delay due to the periodic sleeping, and thus delay is always taken as the first consideration in such networks. In many broadcasting applications without cooperation requirement such as *configuration dissemination*, each node is expected to receive the broadcasting message as soon as possible to update the configuration so that the new system requirement can be satisfied in a short period of time. In other words, the end-to-end (E2E) delay from the sink node to each sensing node is desired to be minimized for broadcasting in such applications. For example, GreenOrbs is a consistently operating sensor network system deployed on Tianmu mountain of China for the aim of forest monitoring. It periodically collects various sensory data including temperature, humidity, illumination, and carbon dioxide titer. The collected information is utilized to support various significant applications, such as forest surveillance, forestry observation and research, fire risk evaluation, and succor in the wild. Occasionally, we need to change the sampling period or other parameters of each node. In this case, the sink node needs to broadcast the updated configuration in the network as soon as possible so that the new system requirement can be satisfied as soon as possible. As for another example, alarm detection system is also a type of widely used applications for WSNs. Upon detecting that a parameter (e.g., temperature, and humidity) is above or below some threshold, the sensor node will report it to the sink quickly so that a prompt action can be taken. For this kind of applications, we sometimes need to change the system requirement (e.g., to change the alarm threshold), and the sink needs to broadcast the message attached with the updated alarm threshold in the network as soon as possible so that the chance of false positive or false negative can be reduced as much as possible. To this end, we identify the minimum E2E delay as a firm requirement, and thus our optimization goal becomes balancing the energy consumption among nodes while guaranteeing the minimum E2E delay.

In this paper, we focus on the *Load-Balanced Minimum E2E-delay Broadcast Scheduling Problem* (LB-MEBS), namely how to minimize the maximum transmission load of a broadcasting schedule for low-duty-cycle WSNs, subject to the constraint that each node should have the minimum end-to-end delay under the broadcasting schedule.

The main contributions of this work are as follows:

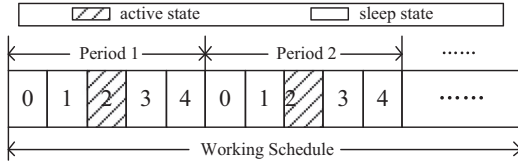
- To the best of our knowledge, we are the first to investigate the load-balanced minimum delay broadcast scheduling problem for low-duty-cycle WSNs. We transform our problem into the equivalent *Load-Balanced Parents Assignment Problem* (LBPA), and prove its NP-hardness.
- We propose the *Load-Balanced Parents Assignment Algorithm* (LBPA-A) to tackle the LBPA problem, and show that LBPA-A can achieve λ -approximation, where λ denotes the maximum number of neighbors that are scheduled to wake up at the same time and is typically a small number in low-duty-cycle WSNs.
- We also propose an efficient distributed solution, i.e., DLBPA-A, to solve our problem. The message complexity of this algorithm is $O(N^2 + N \cdot d_{max}^2)$ where N and d_{max} denote the number of nodes and the maximum node degree in the network, respectively.
- Our simulation results reveal that compared with the traditional solutions, LBPA-A and DLBPA-A both exhibit much better average performance in terms of energy-fairness, total energy consumption and delivery ratio.

The rest of the paper is organized as follows: [Section 2](#) summarizes the related work. [Section 3](#) illustrates the network model and formulates the problem. [Section 4](#) analyzes the problem hardness. Detailed description and analysis of our proposed algorithm are presented in [Section 5](#). [Section 6](#) introduces an efficient distributed solution, followed by the discussion about practical issues and the simulation results in [Section 7](#) and [Section 8](#). [Section 9](#) concludes our findings.

2. Related work

In recent years, a number of works that focus on energy-efficient broadcast scheduling problem in low-duty-cycle WSNs have been proposed [3–13]. Hong et al. [4] 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. In [6], the authors considered link correlation and devised a novel flooding scheme to reduce energy consumption of broadcasting by letting nodes with high correlation be assigned to a common sender. Xu et al. [7] utilized the broadcasting spatiotemporal locality to address the latency-optimal minimum energy broadcast problem in low-duty-cycle WSNs. In [10], 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. [11] 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. [12] 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. In [13], 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. However, all of these works mainly focus on the minimization of total energy consumption rather than load balancing. Recently, Glossy broadcasting scheme [14] has gained much attraction amongst researchers, as it uses constructive interference for flooding and thus eliminates any need to establish a schedule for broadcasting. However, it mainly focuses on the latency and reliability of broadcasting but not the load balance problem.

Currently, load balancing for applications of data collection in sensor networks has also been extensively investigated by many works [15–22]. Wu et al. [15] proposed a novel nonuniform node distribution strategy to achieve nearly balanced energy depletion in the network. Jurdak et al. [16] proposed a cross-layer framework to balance the load in sensor networks via greedy local decisions. Xiong et al. [17] studied the multiple task scheduling problem for low-duty-cycle WSNs. They presented several efficient scheduling algorithms to achieve load balancing among sensor nodes in both spatial and temporal dimensions. In [18], the authors developed a delay-constrained data aggregation scheme for duty cycle sensor networks to balance the nodal lifetime of all nodes. Besides, some works such as [19] also consider employing a mobile sink or mobile relay to collect data so as to balance loads among sensor nodes. In [20], the authors investigated the problem of controlling node sleep intervals so as to achieve the min-max energy fairness in asynchronous duty-cycling sensor networks, the proposed algorithm is self-adjustable to the traffic load variance and is able to serve as a unified framework for a variety of asynchronous duty-cycling MAC protocols. However, to the best of our knowledge, none of the existing works consider load balancing of broadcasting applications for sensor networks under low-duty-cycle

Fig. 1. An example of working schedule with $L = 5$.

operation, which is an important issue for those networks with load balanced data collection.

3. Network model and problem statement

3.1. Network model and assumptions

We assume that time is divided into a number of equal time slots and each time slot is set long enough to accommodate at least one packet transmission. The network is a multi-hop network, and for energy conservation, it is assumed to be put in a *low-duty-cycle mode*. In such mode, each sensor node alternates between *sleep state* and *active state* according to its predefined working schedule. Such working schedule depends on a particular power management protocol, e.g., [23]. When staying in *sleep state*, any node will turn all its function modules off except a timer to wake itself up, and in *active state*, it will turn on its radio and keep awake. An active node is able to sense an event, transmit a packet or receive a packet. We assume that working schedule is periodic and all sensor nodes have the same period length of working schedule. For simplicity and without loss of generality, we suppose that one period of any node's working schedule consists of L time slots where only one time slot is in *active state* and the others are in *sleep states*. Here, the working schedule of any node j is denoted by the set $T_s(j) = \{i | i \in \{0, \dots, L-1\} \text{ and } \tau_j^i = 1\}$, in which $\tau_j^i = 1$ denotes the time slot i for each schedule period of node j is in *active state*. Fig. 1 illustrates an example of the periodic working schedule $\{2\}$, where $L = 5$ and the node only keeps awake at time slot 2 for each period of working schedule.

In this paper, 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 $\{v_1, v_2, \dots, v_{N-1}\}$, 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. Note that, our solution is also applicable to the general case where one period of any node's working schedule contains one or more than one *active state* time slot.

As the same with most of the literature for low-duty-cycle WSNs (e.g., [3–5,7,10–13,24–27]), we assume that (1) time synchronization is achieved; (2) each node is aware of the working schedules of all its neighboring nodes; and (3) each node can transmit its packets at any time but can only receive packets from its neighbors in *active states*. Specifically, if any sender wants to send a 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. For simplicity, we do not consider the packet collision problem in our model, and this practical issue and its efficient solution will be discussed in Section 8. Besides, we make a practical *power consumption assumption*, that is, idle listening consumes approximately the same amount of power as in receiving and sending mode, which has been shown in [2].

In this paper, we also assume that at any time the network is processing at most one request, i.e., two successive broadcast requests of the sink do never overlap in the network. This assumption is always true in practice since the broadcasting originated from the sink (e.g., configuration dissemination) is usually a relatively infrequent event in real sensor network systems, which implies the time dura-

Table 1

Summary of the primary notations.

Symbol	Meaning
$T_s(v)$	The set of active time slots of node v
d_{ij}	Point-to-point transmission delay from node v_i to v_j
$\text{delay}(u, v)$	Minimum end-to-end delay from node u to node v
$\text{Load}(v)$	Transmission load of node v
$\text{deg}(v)$	Degree of node v
$\text{CPS}(v)$	Candidate parents set of node v
$\text{CCS}(v)$	Candidate children set of node v
$\text{Load}_E(v)$	Transmission load of node v in the edge set E
$\text{deg}_E(v)$	Degree of node v in the edge set E
$\text{CPS}_E(v)$	Candidate parents set of node v in the edge set E
$\text{CCS}_E(v)$	Candidate children set of node v in the edge set E
$S(v, t)$	The set of message receivers in $\text{CCS}(v)$ if node v sends a message at time slot t
$\text{TR}(v, t)$	The transmission of node v at time slot t
$\text{RD}_E(v, t)$	The redundancy degree of the transmission $\text{TR}(v, t)$ in the edge set E
$\text{TS}(v)$	Transmission strategy set of node v
$\text{LNS}(v)$	Local nodes set of node v

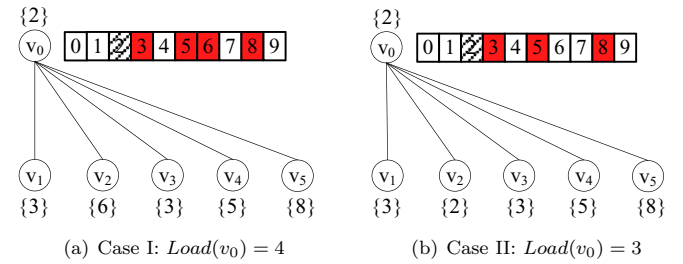


Fig. 2. Illustration of transmission load.

tion between any two successive broadcast requests is usually long enough (much longer than the minimum broadcast latency) for real broadcast applications. Specifically, the sink in real broadcast applications will usually not send a request (i.e., configuration update command) until the last sent request is received by all the nodes in the network and the new updated configuration at each node is running for a duration (i.e., the new updated system requirement is satisfied for a duration). Therefore, we can consider broadcast request one by one without overlapping, in other words, we only need to focus on one broadcast request in the paper.

A summary of the primary notations in this paper is given in Table 1.

3.2. Problem statement

Definition 1 (Transmission load). Suppose node v_i needs to send a message to all nodes in a subset $N(v_i)$ of its neighbors. We define its **transmission load** $\text{Load}(v_i)$ as the number of additional slots that v_i must be awake to finish the transmission, where an additional slot of v_i is a slot originally not scheduled to be awake. That is,

$$\text{Load}(v_i) = \left| \bigcup_{v_j \in N(v_i)} T_s(v_j) - T_s(v_i) \right| \quad (1)$$

where $T_s(v)$ denotes the set of active time slots of node v .

Here, we use an example, as shown in Fig. 2 with $L = 10$, to illustrate the above definition. In Fig. 2(a), suppose that the sender v_0 wants to broadcast a packet to all the receivers in $\{v_1, \dots, v_5\}$. We can find that none of the receivers has the same working schedule with the sender, and sender v_0 has to additionally wake up and transmit the packet at time slots 3, 5, 6 and 8 so as to finish the one-hop broadcasting, so the transmission load $\text{Load}(v_0) = |\{3, 5, 6, 8\} - \{2\}| = 4$ in this case. In Fig. 2(b), node v_2 has the same working schedule $\{2\}$ with the sender v_0 , which implies v_0 will transmit the packet to v_2

Table 2
Summary of the defined problems.

Abbreviation	Problem full name	Position	Hardness
LB-MEBS	Load-Balanced Minimum E2E-delay Broadcast Scheduling Problem	Problem 1, Section 3.2	NP-hard
LBPA	Load-Balanced Parents Assignment Problem	Problem 2, Section 4	NP-hard
SLBS	Spatiotemporal Load-Balanced Semi-matching Problem	Problem 3, Section 4	NP-hard
DBS	Degree-Balanced Semi-matching Problem	Problem 4, Section 5.3	P

at its scheduled time slot and thus the transmission energy from v_0 to v_2 can be approximately neglected since in practice, idle listening mode has nearly the same energy consumption rate with the sending mode [2]. Accordingly, in Fig. 2(b), the broadcasting energy consumption of sender v_0 mainly comes from the transmissions at 3 non-scheduled additional time slots (i.e., time slots 3, 5 and 8), so the transmission load $Load(v_0) = |\{2, 3, 5, 8\} - \{2\}| = 3$.

In practice, idle listening mode has nearly the same energy consumption rate with the receiving mode, which implies the receiving energy consumption for all nodes can be approximately neglected since in our model, each node only receives the packet at its scheduled time slot. In this paper, therefore, we will consider *transmission load* as the main metric of energy consumption.

In many broadcasting applications without cooperation requirement such as *configuration dissemination*, each node is expected to receive the broadcasting message to update the configuration as soon as possible. Thus, it is important to minimize the E2E delay of the broadcasting for such applications in low-duty-cycle WSNs. Here, we regard load balancing as the main consideration of energy efficiency, and in practice, the sink node is usually not energy-critical as it can be supplied with energy continuously, so it is unnecessary to consider the energy consumption of the sink node in our problem. Thus, our objective is to address the following *Load-Balanced Minimum E2E-delay Broadcast Scheduling Problem* (LB-MEBS) in low-duty-cycle WSNs.

Problem 1 (LB-MEBS). *Given a spatiotemporal topology graph $G=(V, E, W, L)$, find a minimum E2E-delay spanning tree (i.e., Minimum Delay Path Tree (MDPT) where the E2E delay from the sink v_0 to each sensing node is minimal) T^* rooted at the sink node v_0 so that the maximum transmission load of non-sink sensing nodes on T^* is minimized.*

In order to facilitate the reading, a summary of all the problems defined in this paper is given in Table 2.

3.3. Solution overview

In the following sections, we will prove the NP-hardness of our target LB-MEBS problem and propose an approximation algorithm to solve this problem. Specifically, we first transform LB-MEBS problem into the equivalent *Load-Balanced Parents Assignment Problem* (LBPA) and show that it is NP-hard. Then, LBPA problem can further be transformed into several independent subproblems, and we can find that each subproblem is actually a generalization of the well-known *Degree-Balanced Semi-matching Problem* (DBS) and no existing solution was proposed for this problem. Based on the state-of-the-art solution to DBS problem, we propose an efficient greedy-like solution to solve each subproblem. Actually, the performance of our solution can be well-guaranteed as it benefits from the high efficiency of the state-of-the-art solution to DBS problem. Also, we come up with a fully distributed solution for our target problem. The basic idea is to let each node determine an initial transmission strategy set and then iteratively make a local competition at each node by exchanging a few of control packets between local nodes. If any node has the locally maximum transmission load at some round of the local competition, it will be the competition winner and drop one redundant transmission from its transmission strategy set, and this process will be repeated until there does not exist any redundant transmission for

each node. Intuitively, our distributed solution would perform well as it adopts a locally greedy strategy to reduce the maximum transmission load in any local area. Finally, we will verify the high-efficiency of our proposed solutions by both theoretical analysis and extensive simulations.

4. Problem hardness analysis

In this section, we will analyze the hardness of our target problem. First, we construct a *Minimum Delay Path Graph* (MDPG) from G which can make sure that the minimum E2E-delay constraint is satisfied. Then, we transform LB-MEBS problem into the equivalent *Load-Balanced Parents Assignment Problem* (LBPA) on MDPG, which is independent of the minimum E2E delay constraint. Finally, we analyze the hardness of LBPA problem.

Here, we use $delay(v_0, v_i)$ to denote the minimum E2E delay from sink v_0 to any node v_i and we initially set $delay(v_0, v_0) = 0$, and $delay(v_0, v_i) = \infty$ for any $v_i \neq v_0$. For any sensing node v_i , we use $CPS(v_i)$ and $CCS(v_i)$ to denote its *Candidate Parents Set* and *Candidate Children Set* respectively, where $CPS(v_i)$ and $CCS(v_i)$ are both initially set as null. For any node v_i , we define $T_s(v_i) = \{t_i\}$. Further, we assume the sink node v_0 starts the broadcast operation at its scheduled time slot t_0 and we denote by d_{ij} the point-to-point transmission delay for any edge $(v_i, v_j) \in E$, and d_{ij} can be determined as follows:

$$\begin{aligned} & \text{If } v_i = v_0, \\ d_{ij} &= \begin{cases} t_j - t_i + 1, & \text{if } t_j \geq t_i; \\ t_j - t_i + L + 1, & \text{otherwise} \end{cases} \end{aligned} \quad (2)$$

and if $v_i \neq v_0$,

$$d_{ij} = \begin{cases} t_j - t_i, & \text{if } t_j > t_i; \\ t_j - t_i + L, & \text{otherwise} \end{cases} \quad (3)$$

Next, we will first present the following *Minimum Delay Path Graph Construction Algorithm* (MDPGC-A), which is similar to Bellman-Ford Algorithm, to construct the *Minimum Delay Path Graph* (MDPG): Initially, each node will keep awake and determines its own working schedule depending on a particular power management protocol immediately after deployment, and then exchange the working schedule with its neighbors. Afterwards, the sink v_0 starts to broadcast a control packet containing its ID and $delay(v_0, v_0)$. Upon receiving the broadcast control packet from any neighboring node v_i , sensing node v_j will figure out d_{ij} according to Eqs. (2) and (3), and compares the current stored $delay(v_0, v_j)$ with $delay(v_0, v_i) + d_{ij}$.

- If $delay(v_0, v_i) + d_{ij} < delay(v_0, v_j)$, node v_j will update the current $delay(v_0, v_j)$ as $delay(v_0, v_i) + d_{ij}$, then send a feedback packet to its neighbors to make node v_i add v_j into the set $CCS(v_i)$ and each node $v'_i \in CPS(v_j)$ remove v_j from the set $CCS(v'_i)$ if $CPS(v_j) \neq \emptyset$, and next it will update $CPS(v_j)$ as $\{v_i\}$ and broadcast a new control packet containing its ID and the updated $delay(v_0, v_j)$;
- If $delay(v_0, v_i) + d_{ij} = delay(v_0, v_j)$, node v_j will just add v_i into the set $CPS(v_j)$ and reply a feedback packet to v_i to make node v_i add v_j into the set $CCS(v_i)$;
- If $delay(v_0, v_i) + d_{ij} > delay(v_0, v_j)$, nothing is performed by node v_j .

The above process continues until no new control/feedback packet is received for each node.

Obviously, a MDPT can be obtained if any sensing node v_i in MDPG selects any node in $CPS(v_i)$ as its parent. Our target problem, i.e., LBMEBS problem, is thus equivalent to the following *Load-Balanced Parents Assignment Problem* (LBPA) which is independent of the minimum E2E delay constraint.

Problem 2 (LBPA). Given a spatiotemporal topology graph $G = (V, E, W, L)$ and its corresponding MDPG, for each non-sink sensing node v_i in MDPG, choose a node in $CPS(v_i)$ to be its parent so that the maximum transmission load of non-sink sensing nodes on the resulting MDPT is minimized.

Next, we will analyze the hardness of LBPA problem. First, we come up with the *Spatiotemporal Load-Balanced Semi-matching Problem* (SLBS) and prove that it is NP-hard. Then, we show the NP-hardness of LBPA problem by reduction from SLBS problem.

Definition 2 (Semi-matching). Given a spatiotemporal bipartite graph $G = (S \cup R, E, W, L)$ where S is the set of senders, R is the set of receivers and $E \subseteq S \times R$, W denotes the set of working schedules for all nodes in $S \cup R$, and L denotes the schedule period length of each node, a semi-matching is defined as a set M of edges with $M \subseteq E$ such that each node in R is incident with exactly one edge in M .

Problem 3 (SLBS). Given a spatiotemporal bipartite graph $G = (S \cup R, E, W, L)$ where S is the set of senders, R is the set of receivers and $E \subseteq S \times R$, W denotes the set of working schedules for all nodes in $S \cup R$, and L denotes the schedule period length of each node, find a semi-matching M in G such that $\max_{s \in S} \{Load_M(s)\}$ is minimized, where $Load_M(s)$ denotes the **transmission load** of the sender s in M .

Lemma 4.1. SLBS problem is NP-hard.

Proof. We refer to the decision version of SLBS problem as DP_{SLBS} : Given a spatiotemporal bipartite graph $G = (S \cup R, E, W, L)$ and a non-negative integer k , is there a semi-matching so that the maximum transmission load of the senders in S is at most k ? We will prove that DP_{SLBS} is NP-complete by reduction from the well-known NP-hard *Monotone 3SAT* problem [28], where we are given a set U of boolean variables and a collection C of clauses over U such that each clause $c \in C$ contains either only negated variables or only un-negated variables and $|c| = 3$. The question is whether there is a satisfying truth assignment for C .

Given any instance of *Monotone 3SAT* with variables $x_1, \dots, x_n$ and clauses $c_1, \dots, c_m$, construct the bipartite graph $G = (S \cup R, E, W, 3)$ where $S = \{x_1, \dots, x_n\}$ and $R = \{c_1, \dots, c_m\}$, corresponding naturally to the set of variables and clauses. There is an edge $(x_p, c_q) \in E$ if and only if either $x_p \in c_q$ or $\bar{x}_p \in c_q$. The set W is as follows. (1) $T_s(x) = \{0\}$ for all $x \in S$; (2) $T_s(c_p) = \{1\}$ if clause c_p consists of only un-negated variables; (3) $T_s(c_p) = \{2\}$ if clause c_p consists of only negated variables. The non-negative integer k is set as 1. It is easy to see that this reduction can be done in polynomial time. We give an example of the reduction. For the instance of the *Monotone 3SAT* problem

$$(x_1 \vee x_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3) \wedge (x_2 \vee x_4 \vee x_5) \wedge (\bar{x}_2 \vee \bar{x}_3 \vee \bar{x}_5)$$

which consists of four clauses and $U = \{x_1, \dots, x_5\}$, we obtain its corresponding instance of DP_{SLBS} in Fig. 3.

We claim that the *Monotone 3SAT* problem is satisfiable if and only if there exists a semi-matching in the constructed bipartite graph G with a maximum transmission load of 1.

(i) Suppose an instance of *Monotone 3SAT* problem is satisfiable and let $t: U \rightarrow \{true, false\}$ be a satisfying truth assignment. We show that a desired semi-matching can be derived from this truth assignment. For node $x \in S$ such that $t(x) = true$, we let it be parent of all neighbors whose active time slot is 1, and for node $x \in S$ such that $t(x) = false$, we let it be parent of all neighbors whose active time

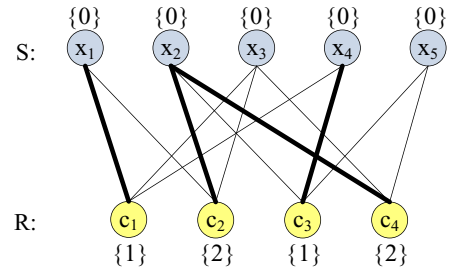


Fig. 3. An instance of DP_{SLBS} with $L = 3$. Given any semi-matching with the maximum transmission load of 1, e.g., the set of bold edges, we can easily get a truth assignment $t(\{x_1, x_2, x_3, x_4, x_5\}) = \{true, false, false, true, false\}$.

slot is 2. In cases where a single node in R is assigned multiple parents, we arbitrarily keep one of them and delete the others. Since each clause has at least one literal evaluating to *true* so that it has at least one parent, the edges between parents and children cover all nodes in R , forming a legitimate semi-matching. It is easy to see that this semi-matching must have a maximum transmission load of 1.

(ii) Suppose M is a semi-matching for the constructed DP_{SLBS} instance with a maximum transmission load of 1. We can obtain a satisfying truth assignment t for the *Monotone 3SAT* problem. Due to maximum transmission load of 1, the children in M of any node $x \in S$ have the same active time slot, either 1 or 2. Therefore, we let $t(x) = true$ if the active time slot of the children of node x is 1, and let $t(x) = false$ if either x does not have child or its children have active time slot 2. One can check that this truth assignment satisfies the *Monotone 3SAT* problem.

It is obvious that DP_{SLBS} is in NP since we can non-deterministically select a solution and check its transmission load in polynomial time. Therefore, DP_{SLBS} is NP-complete, and the proof is completed. $\square$

Theorem 4.1. LBPA problem is NP-hard.

Proof. Given any instance $G = (S \cup R, E, W, L)$ of SLBS problem where $S = \{s_1, \dots, s_n\}$, $R = \{r_1, \dots, r_m\}$ and suppose that $T_s(v) = \{t_v\}$ for any $v \in S \cup R$, we can construct an instance $G' = (V', E', W', L + 1)$ of LBPA problem in polynomial-time as follows. First, we let $V' = \{v_0\} \cup V_s \cup V_r$ where $V_s = \{s_1, \dots, s_n\}$, $V_r = \{r_1, \dots, r_m\}$ and v_0 denotes the sink node, and $E' = (\{v_0\} \times V_s) \cup E$. Then, for any node $r_q \in V_r$, if there exists at least one node s_p in V_s so that $T_s(s_p) = T_s(r_q)$, we will update V_r by removing node r_q from V_r , and update E' by removing all the edges that are attached to r_q from E' . Next, we set W' by letting 1) $T_s(v_0) = \{0\}$; 2) $T_s(s_p) = \{0\}$ for any node $s_p \in V_s$; 3) $T_s(r_q) = \{t_{r_q} + 1\}$ for any node $r_q \in V_r$.

By such a construction, we can easily prove that the optimal spatiotemporal load-balanced semi-matching on G can be found in polynomial-time if and only if the optimal MDPT, in which the maximum transmission load of non-sink sensing nodes is minimized, can be found from G' in polynomial-time. As this proof is simple and obvious, we omit the detailed process for saving space. Accordingly, SLBS problem is polynomial-time reducible to LBPA problem. According to Lemma 4.1, the proof is now completed. $\square$

5. Centralized solution

To solve our target problem, in this section, we will propose an efficient solution, i.e., *Load-Balanced Parents Assignment Algorithm* (LBPA-A), which mainly consists of (1) Component Construction; (2) Component Simplification; (3) Degree-balanced Semi-matching; (4) Redundant Transmissions Reduction.

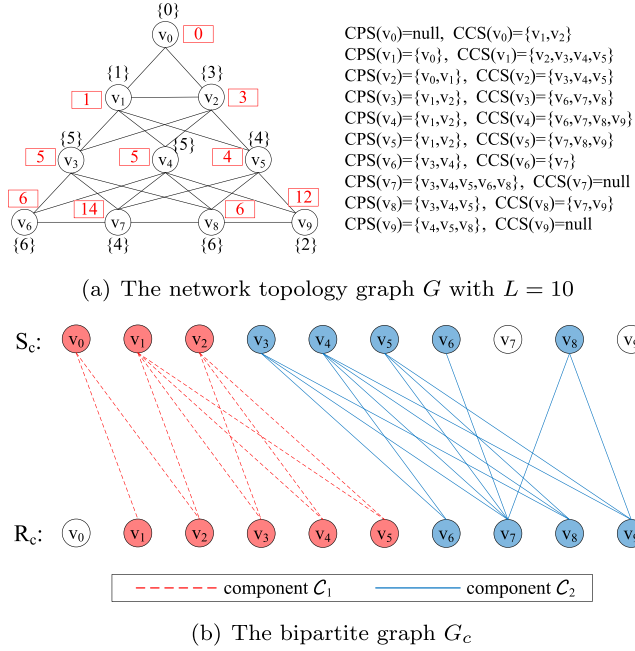


Fig. 4. An example of component construction.

5.1. Component construction

Definition 3 (Component). Given a spatiotemporal topology graph $G = (V, E, W, L)$ and its MDPG, we can define an undirected bipartite graph $G_c = (S_c \cup R_c, E_c, W, L)$ to characterize the minimum E2E-delay broadcasting in low-duty-cycle WSNs, where $S_c = V$ is the set of potential senders, $R_c = V$ is the set of potential receivers and $E_c \subseteq S_c \times R_c$. For any $s \in S_c$ and any $r \in R_c$, specifically, E_c contains an edge that connecting s and r if and only if $s \in CPS(r)$. In this paper, a **component** is defined as a maximal connected bipartite subgraph of the bipartite graph G_c . Here, a subgraph of G_c is called maximal connected if and only if it is connected and any other edge in E_c that does not belong to this subgraph is not incident with it.

Based on the information about candidate parents for all nodes, we can easily construct the bipartite graph G_c according to its definition mentioned above, and then derive all the *components* from G_c by the following steps:

- Initially, we mark all the edges in E_c with *unselected*.
- We assume that a *component* C initially only contains one edge that is arbitrarily selected from the *unselected* edges in E_c , and then we extend the *component* C by iteratively merging C with the edges incident to C , until no edge incident to C is found in E_c . Next, we mark all the edges in the resulting C with *selected*.
- We repeat Step 2 until all the edges in E_c are marked with *selected*, and finally all the *components* $\{C_1, C_2, \dots, C_m\}$ in G_c can be found.

Fig. 4 explicitly illustrates an example of component construction. Given a network topology graph G with $L = 10$ as shown in Fig. 4(a), we can figure out the minimum E2E delay from the sink to each node in G , and obtain the *Candidate Parents Sets* for all nodes according to the aforementioned MDPG construction algorithm. As shown in Fig. 4(b), accordingly, we can easily construct the corresponding bipartite graph G_c which consists of two *components* $\{C_1, C_2\}$.

Observation 1. Components are independent of each other, that is, the parents assignment in one component will NOT affect that in any other component in terms of both delay-optimality and energy-fairness for all nodes.

According to **Observation 1**, it is obvious that LBPA problem can thus be transformed into several independent subproblems, i.e., SLBS problem on each *component*. Specifically, our objective is thus to solve the following independent subproblem: for any component $C_i = (S_i \cup R_i, E_i, W_i, L)$ where $S_i \subseteq S_c$, $R_i \subseteq R_c$, $E_i \subseteq E_c$ and $W_i \subseteq W$, find a *semi-matching*, namely assign **exactly one** parent from the senders in S_i to each receiver in R_i , so that the maximum transmission load of the senders in S_i is minimized.

5.2. Component simplification

As stated before, we do not consider the energy consumption of the sink node in our problem due to the fact that the sink node can be always supplied with energy continuously in practice. For any neighbor of the sink node, we thus select the sink node as its parent. Correspondingly, we can simplify any component $C_i = (S_i \cup R_i, E_i, W_i, L)$ by deleting all the edges that are incident to the nodes in $N_0 \cap R_i$ from C_i , where N_0 denotes the set of the sink node's neighbors. As shown in Fig. 4, the sink v_0 has two neighbors $\{v_1, v_2\}$, we thus let v_0 be the parent of both v_1 and v_2 , and remove the edges (v_0, v_1) , (v_0, v_2) and (v_1, v_2) from the *component* C_1 to get a *simplified component* of C_1 .

Due to our *power consumption assumption*, actually, the *transmission load* of any sender s_i^p will not involve the transmission from s_i^p to any of its receivers that have the same working schedule with s_i^p . Therefore, we can further simplify any *component* $C_i = (S_i \cup R_i, E_i, W_i, L)$ as follows: For any receiver $r_i^q \in R_i$, we check whether there exist one or multiple senders in $CPS(r_i^q)$ which have the same working schedule with r_i^q . If yes, we will let any one of them be the parent of r_i^q and then delete all the edges that are incident to r_i^q from C_i . For example, node v_7 in Fig. 4(b) has one sender v_5 in $CPS(v_7)$ which has the same working schedule of v_7 , i.e., $T_s(v_5) = T_s(v_7) = \{4\}$. Thus, we can assign v_5 as the parent of v_7 and remove the edges (v_3, v_7) , (v_4, v_7) , (v_5, v_7) , (v_6, v_7) and (v_8, v_7) from the *component* C_2 to get a *simplified component* of C_2 .

Finally, we update $CCS(s_i^p)$ for each sender s_i^p in each *simplified component* C_i , and by the above simplification, our objective will accordingly turn to be how to solve the SLBS problem on each *simplified component*.

5.3. Degree-balanced Semi-matching

Before solving the SLBS problem, we first review the well-known *Degree-Balanced Semi-matching Problem* (DBS), which has been well-studied by many existing works [29,30].

Problem 4 (DBS [29]). Given a bipartite graph $G = (S \cup R, E)$ where S is the set of senders, R is the set of receivers and $E \subseteq S \times R$, find a *semi-matching* M in G such that $\max_{s \in S} \{deg_M(s)\}$ is minimized, where $deg_M(s)$ denotes the degree of the sender s in M , i.e., the number of edges in M that are incident with s .

Obviously, our proposed SLBS problem is a generalization of the DBS problem. Specially, if the duty cycle in the network is so low that all the neighbors of any node have different working schedules, then $Load_M(s)$ must be equal to $deg_M(s)$ for any semi-matching M and thus the SLBS problem will turn to be the DBS problem. As we know, it has been shown that the DBS problem is equivalent to the *optimal semi-matching problem* which is solvable in polynomial-time [29], and the state-of-the-art solution [30] of the DBS problem provides an efficient algorithm which runs in $O(\sqrt{|S \cup R|} \cdot |E| \cdot \log|S \cup R|)$ time. For convenience of description, we call the solution proposed in [30] the *Degree-Balanced Semi-matching Algorithm* (DBS-A).

Here, we will address the SLBS problem on any *simplified component* C_i by directly performing DBS-A on C_i and analyze its performance. Let M_i^q denote the solution (i.e., the resulting semi-matching) when performing DBS-A on the *simplified component* C_i , and $OPT(C_i)$

denote the maximum transmission load of senders in the optimal solution of our target problem, i.e., the SLBS problem on C_i . Also, we denote by $S(s_i^p, t)$ the set of message receivers in $CCS(s_i^p)$ if the sender s_i^p transmits a message at time slot t where $t \in \bigcup_{j \in CCS(s_i^p)} T_s(j)$, namely

$$S(s_i^p, t) = \{r | r \in CCS(s_i^p) \text{ and } T_s(r) = \{t\}\}. \quad (4)$$

We have the following conclusion.

Lemma 5.1. *Given any simplified component $C_i = (S_i \cup R_i, E_i, W_i, L)$, we have*

$$\max_{s \in S_i} \{Load_{M_i^d}(s)\} \leq \lambda_i \cdot OPT(C_i)$$

where λ_i denotes the maximum number of the receivers that are scheduled to wake up at the same time and have at least one common candidate parent in C_i , namely $\lambda_i = \max_{s_i^p \in S_i \text{ and } t \in \bigcup_{j \in CCS(s_i^p)} T_s(j)} \{|S(s_i^p, t)|\}$.

Proof. Let the semi-matching M_i^l denote the optimal solution of the SLBS problem on C_i and define $s_d^{max} \in \arg \max_{s \in S_i} \{deg_{M_i^l}(s)\}$. We can obviously find that

$$\begin{aligned} OPT(C_i) &= \max_{s \in S_i} \{Load_{M_i^l}(s)\} \\ &\geq Load_{M_i^l}(s_d^{max}) \end{aligned} \quad (5)$$

and thus

$$\begin{aligned} \lambda_i \cdot OPT(C_i) &\geq \lambda_i \cdot Load_{M_i^l}(s_d^{max}) \\ &\geq deg_{M_i^l}(s_d^{max}) \\ &= \max_{s \in S_i} \{deg_{M_i^l}(s)\}. \end{aligned} \quad (6)$$

Since M_i^d is the semi-matching where the maximum degree of the senders in C_i is minimized, then

$$\max_{s \in S_i} \{deg_{M_i^d}(s)\} \leq \max_{s \in S_i} \{deg_{M_i^l}(s)\}. \quad (7)$$

Also, due to the fact that $Load_{M_i^d}(s) \leq deg_{M_i^d}(s)$ for any $s \in S_i$, we further define $s_l^{max} \in \arg \max_{s \in S_i} \{Load_{M_i^d}(s)\}$ and must have

$$\begin{aligned} \max_{s \in S_i} \{Load_{M_i^d}(s)\} &= Load_{M_i^d}(s_l^{max}) \\ &\leq deg_{M_i^d}(s_l^{max}) \\ &\leq \max_{s \in S_i} \{deg_{M_i^d}(s)\}. \end{aligned} \quad (8)$$

According to Eqs. (6)–(8), we can derive that

$$\max_{s \in S_i} \{Load_{M_i^d}(s)\} \leq \lambda_i \cdot OPT(C_i). \quad (9)$$

Thus, the proof is completed. $\square$

Theorem 5.1. *For the LBPA problem, our solution that applies DBS-A to each simplified component achieves λ -approximation, where λ denotes the maximum number of neighbors that are scheduled to wake up at the same time.*

Proof. Suppose there are m independent simplified components $\{C_1, \dots, C_m\}$ in the problem. Let the found schedules by DBS-A be $M_1, \dots, M_m$ respectively, and let the corresponding optimal schedules be $M_1^*, \dots, M_m^*$. Due to Lemma 5.1, for all i , we have

$$Load(M_i) \leq \lambda_i Load(M_i^*)$$

where $Load(M)$ is the maximum load of all senders under schedule M . It is then straightforward to see that

$$\max_i Load(M_i) \leq \lambda \max_i Load(M_i^*)$$

since $\lambda \geq \lambda_i$ for all i . Note that the term $\max_i Load(M_i)$ is the load of our solution and the term $\max_i Load(M_i^*)$ is the load of the optimal solution. $\square$

In low-duty-cycle WSNs, we usually improve the network performance (e.g., to minimize average detection delay) by carefully designing the working schedules of all nodes (e.g., [23]) to make the neighboring nodes rotate the sensory coverage, which implies very few of the neighboring nodes will have the same working schedule in practice. Moreover, for a specific network performance requirement, it is usual that each node will correspondingly adjust (e.g., decrease) its duty cycle as the number of nodes N varies (e.g., increase), which means λ is actually independent of N and only depends on the given network performance requirement. Thus, λ is typically a small number in low-duty-cycle WSNs.

5.4. Redundant transmissions reduction

As shown in Theorem 5.1, the DBS-A can actually provide the performance guarantee in the worst case. However, we can find that some redundant transmissions would exist in the solution that adopting DBS-A. As a simple example, Fig. 5(a) shows a simplified component C_i , and suppose that the set of the bold edges is the solution that adopting DBS-A on C_i . Obviously, the transmission from s_i^1 to r_i^1 is redundant since the receiver r_i^1 can also be covered by the transmission from s_i^2 to r_i^3 . Likewise, the transmission from s_i^2 to r_i^4 is redundant since the receiver r_i^4 can also be covered by the transmission from s_i^3 to r_i^5 . Thus, we can further make an improvement by removing the redundant edges (s_i^1, r_i^1) and (s_i^2, r_i^4) from our solution, which will reduce the maximum transmission load and the total transmission load from 2 and 6 to 1 and 4 respectively. In other words, the average case performance of our solution could be further improved by efficiently reducing the redundancy of transmissions. In this subsection, accordingly, we will devise an efficient heuristic approach to reduce the redundant transmissions in the solution adopting DBS-A. By doing so, the performance of our solution can be further improved in terms of both energy-fairness and total energy consumption.

In order to better characterize the redundancy of transmissions, we utilize the *Transmission Coverage Graph* (TCG) to represent our solution (i.e., the semi-matching M_i^d) that adopting DBS-A on any simplified component C_i , and M_i^d can be easily transformed into its corresponding TCG, say T_i^d , as follows. Initially, T_i^d consists of all the edges in M_i^d , and then any edge (s_i^p, r) $\notin M_i^d$ in C_i will be added into T_i^d if and only if there exists at least one edge (s_i^p, r_i^q) in M_i^d where $T_s(r_i^q) = T_s(r)$. Obviously, our solution M_i^d is equivalent to T_i^d since in M_i^d , any transmission from s_i^p to r_i^q will also cover all the nodes with working schedule $T_s(r_i^q)$ in $CCS(s_i^p)$ due to the spatiotemporal nature of broadcasting. For the solution which consists of the bold edges in Fig. 5(a), we thus use the equivalent TCG which consists of the bold edges in Fig. 5(b) to represent it.

Here, let $Load_{T_i^d}(s_i^p)$ denote the transmission load of the sender s_i^p in T_i^d , $deg_{T_i^d}(r_i^q)$ denote the degree of the receiver r_i^q in T_i^d , and $CCS_{T_i^d}(s_i^p)$ denote the *Candidate Children Set* of the sender s_i^p in T_i^d , i.e., $CCS_{T_i^d}(s_i^p) = \{r | r \in CCS(s_i^p) \text{ and } (s_i^p, r) \in T_i^d\}$. Also, we let $TR(s_i^p, t)$ ($t \in \bigcup_{j \in CCS_{T_i^d}(s_i^p)} T_s(j)$) denote one transmission of the sender s_i^p at time slot t , which will cover all the nodes in $S(s_i^p, t)$, and $E_i(s_i^p, t)$ denote the set of edges that represents the transmission $TR(s_i^p, t)$, i.e., $E_i(s_i^p, t) = \{(s_i^p, r) | r \in S(s_i^p, t)\}$. For any transmission $TR(s_i^p, t)$ in our solution T_i^d , we will use the *Redundancy Degree* (RD) to characterize its redundancy level, which is defined as follows:

Definition 4 (Redundancy Degree (RD)). Given any simplified component $C_i = (S_i \cup R_i, E_i, W_i, L)$ and our solution T_i^d , the Redundancy Degree (RD) of any transmission $TR(s_i^p, t)$ in T_i^d , saying $RD_{T_i^d}(s_i^p, t)$, is

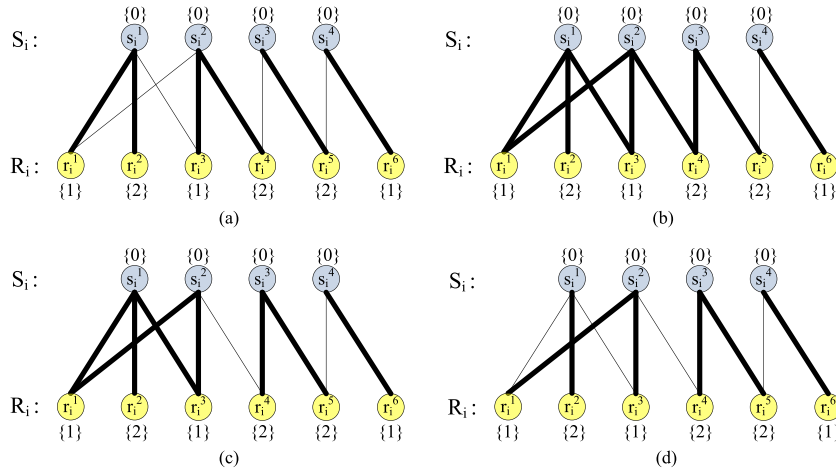


Fig. 5. An example of redundant transmissions reduction.

defined as

$$RD_{T_i^d}(s_i^p, t) = \min_{r_i^q \in S(s_i^p, t)} \{deg_{T_i^d}(r_i^q)\} \quad (10)$$

where $s_i^p \in S_i$ and $t \in \bigcup_{j \in CCS_{T_i^d}(s_i^p)} T_s(j)$.

For any transmission $TR(s_i^p, t)$ in T_i^d , it is called a *redundant transmission* if and only if $RD_{T_i^d}(s_i^p, t) > 1$. If $RD_{T_i^d}(s_i^p, t) = 1$, specially, it is called a *necessary transmission* since there must exist at least one receiver that can only be covered by the transmission $TR(s_i^p, t)$ in T_i^d .

In order to further improve the energy-fairness of our solution T_i^d for any *simplified component* C_i , we will propose a greedy algorithm, which consists of the following two steps, to efficiently reduce the redundant transmissions in T_i^d :

Transmission Removal Step: First, we find out $S_{T_i^d}^{max}$ which denotes the set of senders with the maximum transmission load in T_i^d , i.e., $S_{T_i^d}^{max} = \arg \max_{s_i^p \in S_i} \{Load_{T_i^d}(s_i^p)\}$. For each transmission $TR(s_i^p, t)$ where $s_i^p \in S_{T_i^d}^{max}$ and $t \in \bigcup_{j \in CCS_{T_i^d}(s_i^p)} T_s(j)$, if it is a *redundant transmission*, we will define a metric $\Delta(s_i^p, t)$ as follows:

$$\Delta(s_i^p, t) = \sum_{s \in S_i - \{s_i^p\} \text{ and } t \in \bigcup_{j \in CCS_{T_i^d}(s)} T_s(j)} \frac{I(s, s_i^p, t) \cdot Load_{T_i^d}(s)}{RD_{T_i^d}(s, t)} \quad (11)$$

where

$$I(s, s_i^p, t) = \begin{cases} 1, & \text{if } RD_{T_i^d}(s, t) \neq RD_{T_i^d - E_i(s_i^p, t)}(s, t); \\ 0, & \text{otherwise} \end{cases} \quad (12)$$

From all the *redundant transmissions* of the senders in $S_{T_i^d}^{max}$, we will first choose the one with the smallest $\Delta(s_i^p, t)$ ¹, say $TR(s^*, t^*)$, and then update T_i^d as $T_i^d - E_i(s^*, t^*)$. Next, we repeat this step until there is no *redundant transmission* for all the senders in $S_{T_i^d}^{max}$.

Transmission Replacement Step: When there is no *redundant transmission* for all the senders in $S_{T_i^d}^{max}$, we will check each *necessary transmission* $TR(s_i^p, t)$ of the senders in $S_{T_i^d}^{max}$ that whether all the nodes in the set $S_{deg=1}(s_i^p, t) = \{r | r \in S(s_i^p, t) \text{ and } deg_{T_i^d}(r) = 1\}$ can be covered by the transmissions out of T_i^d while improving

¹ Specially, if there exist multiple redundant transmissions with the smallest $\Delta(s_i^p, t)$, we arbitrarily choose one from them.

the energy-fairness. Specifically, we let $E_i^r(s_i^p, t) = \bigcup_c E_i(s, t)$ where $c = \{s \in S_i - \{s_i^p\}, t \notin \bigcup_{j \in CCS_{T_i^d}(s)} T_s(j) \text{ and } Load_{T_i^d}(s) \leq Load_{T_i^d}(s_i^p) - 2\}$, and $TR(s_i^p, t)$ is called the *replaceable necessary transmission* if and only if $S_{deg=1}(s_i^p, t) \subseteq E_i^r(s_i^p, t)$. We check that whether there exists any *replaceable necessary transmission* for all the senders in $S_{T_i^d}^{max}$. If no, the algorithm is terminated; otherwise, we will arbitrarily choose one from them, say $TR(s^*, t^*)$, then update T_i^d as $(T_i^d - E_i(s^*, t^*)) \cup E_i^r(s^*, t^*)$ and go back to the *Transmission Removal Step*.

In general, the basic idea of our greedy algorithm mentioned above is to greedily reduce the maximum transmission load in T_i^d by each time either removing one *redundant transmission* of the sender with the maximum transmission load, or replacing one of its *necessary transmissions* with the out-of- T_i^d transmissions from the other senders so that $\max_{s \in S_i} \{Load_{T_i^d}(s)\}$ or $|S_{T_i^d}^{max}|$ is reduced by 1. For the sender with the maximum transmission load, the removal of one *redundant transmission* could influence (i.e., decrease) the RD values of the transmissions from the other senders, and when the RD value of one transmission decreases to 1, it will be no longer redundant. Here, we employ the comparison metric $\Delta(s_i^p, t)$ to characterize the influence of the removal of the *redundant transmission* $TR(s_i^p, t)$ on the RD values of the transmissions from the other senders, which is related to three factors: the number of influenced transmissions that characterized by $I(s, s_i^p, t)$, the transmission load of any influenced transmission $TR(s, t)$ (i.e., $Load_{T_i^d}(s)$) and the RD value of any influenced transmission $TR(s, t)$ (i.e., $RD_{T_i^d}(s, t)$). In order to achieve better energy-fairness, intuitively, we prefer to each time remove the one which will bring smaller number of influenced transmissions, smaller $Load_{T_i^d}(s)$ and larger $RD_{T_i^d}(s, t)$ among all the *redundant transmissions* of the senders in $S_{T_i^d}^{max}$.

Fig. 5 shows a simple example of our greedy algorithm on a *simplified component* C_i , where the set of bold edges in Fig. 5(a) and that in Fig. 5(b) represent M_i^d and T_i^d , respectively. In T_i^d , we can find that $S_{T_i^d}^{max} = \{s_1^1, s_2^2\}$ and the transmission $TR(s_2^2, 2)$ has the smallest $\Delta(s_i^p, t)$ (i.e. $\Delta(s_2^2, 2) = 0$) among all the *redundant transmissions* of the senders in $S_{T_i^d}^{max}$. As shown in Fig. 5(c), thus, the redundant edge (s_2^2, r_4^4) will be removed from T_i^d . Next, we can further find that $S_{T_i^d}^{max} = \{s_1^1\}$, and as shown in Fig. 5(d), we will remove the redundant edges (s_1^1, r_1^1) and (s_1^1, r_3^3) from T_i^d since the transmission $TR(s_1^1, 1)$ is currently the only *redundant transmission* of the senders in T_i^d . In Fig. 5(d), it can be seen that the maximum transmission load has

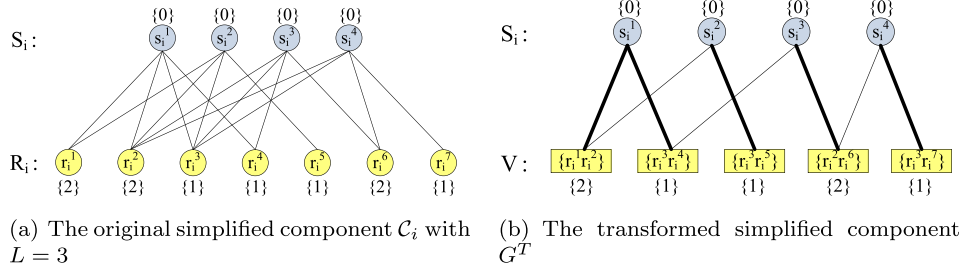


Fig. 6. An example of component transformation.

been decreased from 2 to 1 and $S_{T_i^d}^{max} = \{s_i^1, s_i^2, s_i^3, s_i^4\}$, our greedy algorithm will thus be terminated since neither any *redundant transmission* nor any *replaceable necessary transmission* can be found among all the transmissions of the senders in $S_{T_i^d}^{max}$. Intuitively, the greedy strategy with the comparison metric $\Delta(s_i^p, t)$ could have a better performance than that with the comparison metric $RD_{T_i^d}(s_i^p, t)$, since the latter does not consider the influence of the removed *redundant transmission* on any other *redundant transmission* in T_i^d . In this example, if we utilize the greedy strategy that each time removes the transmission with the largest $RD_{T_i^d}(s_i^p, t)$, instead of the transmission with the smallest $\Delta(s_i^p, t)$, from T_i^d among all the *redundant transmissions* of the senders in $S_{T_i^d}^{max}$, the transmission $TR(s_i^2, 1)$ (i.e., the redundant edges (s_i^2, r_i^1) and (s_i^2, r_i^3)) would be first removed from T_i^d which would cause the termination of the greedy algorithm without decreasing the maximum transmission load. This is because the removal of the transmission $TR(s_i^2, 1)$ would affect the transmission $TR(s_i^1, 1)$ and make it no longer redundant.

By greedily removing or replacing the transmission of the sender with the maximum transmission load, we improve the performance of our solution in terms of energy-fairness. However, we can find that there could still exist redundant transmissions for the senders without the maximum transmission load after running our greedy algorithm, which suggests that the performance of our solution could be further improved in terms of total energy consumption. Specifically, our problem can be described as follows. Given a time slot t where $t \in \bigcup_{r \in R_i} T_s(r)$ and a set of senders $S_i(t) = \{s | s \in S_i \text{ and } t \in \bigcup_{j \in CCS(s_i^p)} T_s(j)\}$, how to select a minimum subset of $S_i(t)$, say $S_i^*(t)$, so that $\bigcup_{s \in S_i^*(t)} S(s, t) = \{r | r \in R_i \text{ and } T_s(r) = \{t\}\}$. Obviously, this is a typical *Set Cover Problem* which is NP-hard and can be well-solved with the state-of-the-art solution. In our solution, finally, we will remove all the edges in $\bigcup_{s \in S_i(t) - S_i^*(t)} E_i(s, t)$ from T_i^d for each $t \in \bigcup_{r \in R_i} T_s(r)$.

Accordingly, we can find that our solution adopting DBS-A on C_i can be further improved in terms of both energy-fairness and total energy consumption by the aforementioned approach.

5.5. Improvement

The SLBS problem has been shown as NP-hard by Lemma 4.1. However, we can find that it is solvable in polynomial-time for some special instances of the *simplified component*, which indicates that the average performance of LBPA-A could be further improved by first checking whether each *simplified component* belongs to this kind of special instances. The following theorem illustrates the **sufficient condition** that the SLBS problem is solvable in polynomial-time.

Theorem 5.2. Given a simplified component $C_i = (S_i \cup R_i, E_i, W_i, L)$, if $S(s_i^p, t) \not\subseteq \bigcup_{s \in S_i - \{s_i^p\}} f(s, t)$ for each $s_i^p \in S_i$ and each $t \in \bigcup_{j \in CCS(s_i^p)} T_s(j)$, then there must exist an efficient algorithm so that the SLBS problem on

C_i can be solved in polynomial-time. Specifically,

$$f(s, t) = \begin{cases} S(s, t), & \text{if } S(s, t) \neq S(s_i^p, t); \\ \text{null}, & \text{otherwise} \end{cases} \quad (13)$$

Proof. By capturing the spatiotemporal characteristic of broadcasting, we can make a **component transformation** for any *simplified component* $C_i = (S_i \cup R_i, E_i, W_i, L)$ as follows. First, we define a bipartite graph $G^T = (S_i \cup V, E^T)$ where V and E^T are both initially set as null. For each $s_i^p \in S_i$ and each $t \in \bigcup_{j \in CCS(s_i^p)} T_s(j)$ in C_i , we will check that whether there exists a vertex v that represents the set $S(s_i^p, t)$ in V . If yes, we will add an edge (s_i^p, v) into E^T ; otherwise, we will add the vertex v that represents the set $S(s_i^p, t)$ into V and then add an edge (s_i^p, v) into E^T .

Obviously, the above *component transformation* can be done in polynomial-time. Actually, the *transformed simplified component* G^T , in which one transmission is represented by just one edge, explicitly exhibits the spatiotemporal characteristic of broadcasting. For any sender $s_i^p \in S_i$ and any $t \in \bigcup_{j \in CCS(s_i^p)} T_s(j)$ in C_i , $S(s_i^p, t) \not\subseteq \bigcup_{s \in S_i - \{s_i^p\}} f(s, t)$ implies that in the corresponding G^T , the vertex $S(s_i^p, t)$ in V must contain at least one receiver which does not belong to any other vertex in V . Therefore, if $S(s_i^p, t) \not\subseteq \bigcup_{s \in S_i - \{s_i^p\}} f(s, t)$ for each $s_i^p \in S_i$ and each $t \in \bigcup_{j \in CCS(s_i^p)} T_s(j)$ in some C_i , our target problem, i.e., how to find a SLBS in such a C_i where each receiver should be covered, can be equivalent to that how to find a DBS in its corresponding *transformed simplified component* G^T . Accordingly, we can solve the SLBS problem on such a C_i in polynomial-time by directly performing DBS-A on its corresponding *transformed simplified component* G^T . $\square$

Fig. 6 illustrates a simple example of the *component transformation*. For the *simplified component* C_i (i.e., Fig. 6(a)), we can find that it has satisfied the *sufficient condition* in Theorem 5.2. Therefore, we can solve the SLBS problem on this C_i in polynomial-time by adopting DBS-A on its corresponding *transformed simplified component* G^T (i.e., Fig. 6(b)). According to the solution adopting DBS-A on G^T which is represented by the set of bold edges in Fig. 6(b), we can get our final broadcasting schedule as $\{TR(s_i^1, 1), TR(s_i^1, 2), TR(s_i^2, 1), TR(s_i^3, 2), TR(s_i^4, 1)\}$.

6. Distributed implementation

Different from the aforementioned centralized solution which requires the global topology information, in this section, an efficient distributed approach, which we call the *Distributed Load-Balanced Parents Assignment Algorithm* (DLBPA-A), will be proposed to solve our target problem.

6.1. Algorithm description

Definition 5 (Local node). Given a spatiotemporal topology graph $G = (V, E, W, L)$ and any two nodes $v_i, v_j \in V$, node v_i is called the

local node of node v_j if and only if $CCS(v_i) \cap CCS(v_j) \neq \emptyset$ and there exists at least one node $v_k \in CCS(v_i) \cap CCS(v_j)$ so that $T_s(v_k) \neq T_s(r)$ for each $r \in CPS(v_k)$.

Here, we come up with an efficient distributed algorithm DLBPA-A, which mainly consists of *Initialization Phase* and *Competition Phase*.

Overall, in *Initialization Phase*, each node will figure out its initial transmission load and send it to all of its local nodes. In *Competition Phase*, each node will compete with its local nodes, and the *competition winner* which has the local maximum transmission load will make the *redundancy reduction decision*. This process will be repeated until no redundancy is found for each node.

Initialization Phase: First, we will perform the MDPGC-A as stated in Section 3.1 in a distributed way, accordingly, each node v_i will get its $CPS(v_i)$ and $CCS(v_i)$, and the *Transmission Strategy* of each node v_i , say $TS(v_i)$, will be initially set as $\{TR(v_i, t) | t \in \bigcup_{j \in CCS(v_i)} T_s(j)\}$. If $CCS(v_i) = \emptyset$, in particular, $TS(v_i)$ will be initially set as *null*. Then, each sensing node v_i will make the following decision:

- if $v_0 \in CPS(v_i)$, v_i will choose v_0 as its parent, then mark itself with *assigned node* and send a control packet containing the assigned parent ID v_0 to all nodes in $CPS(v_i)$;
- if $v_0 \notin CPS(v_i)$ and there exists a nonempty set $S' = \{j | j \in CPS(v_i) \text{ and } T_s(j) = T_s(v_i)\}$, v_i will arbitrarily choose one node in S' as its parent, then mark itself with *assigned node* and send a control packet containing the assigned parent ID to all nodes in $CPS(v_i)$;
- if $v_0 \notin CPS(v_i)$ and there does not exist a nonempty set $S' = \{j | j \in CPS(v_i) \text{ and } T_s(j) = T_s(v_i)\}$, v_i will mark itself with *unassigned node*, then set its degree $deg(v_i) = |CPS(v_i)|$ and send a control packet containing $CPS(v_i)$ and $deg(v_i)$ to all nodes in $CPS(v_i)$.

Afterwards, each node v_i will wait until the control packets from all nodes in $CCS(v_i)$ are received. For each $t \in \bigcup_{j \in CCS(v_i)} T_s(j)$, if the nodes in $S(v_i, t)$ are all the *assigned nodes* and none of their assigned parent IDs is v_i , then v_i will remove the transmission $TR(v_i, t)$ from $TS(v_i)$. Next, v_i will update $CCS(v_i)$ by removing all the *assigned nodes* from $CCS(v_i)$. Based on the received control packets from all the *unassigned nodes* in $CCS(v_i)$, v_i will first figure out its transmission load $Load(v_i) = |\bigcup_{j \in CCS(v_i)} T_s(j)|$ and find its *local nodes set* $LNS(v_i) = \bigcup_{j \in CCS(v_i)} CPS(j) - \{v_i\}$, and then send a competition packet containing $Load(v_i)$ to all the nodes in $LNS(v_i)$ by selecting the nodes in $CCS(v_i)$ as the forwarders. Specially, if $CCS(v_i) = \emptyset$, then $Load(v_i) = 0$ and no competition packet will be sent by v_i .

Competition Phase: Upon receiving the competition packets from all the nodes in $LNS(v_i)$, v_i will figure out its *local maximum-load nodes set* $L_{max}(v_i) = \arg \max_{j \in LNS(v_i) \cup \{v_i\}} Load(j)$. If v_i is the *competition winner* (i.e., $v_i \in L_{max}(v_i)$ and also v_i has the smallest ID number among all the nodes in $L_{max}(v_i)$), it will first figure out the Redundancy Degree (RD) $RD(v_i, t) = \min_{j \in S(v_i, t)} deg(j)$ for each transmission $TR(v_i, t) (t \in \bigcup_{j \in CCS(v_i)} T_s(j))$ and count the number of *redundant transmissions* $|TS_r(v_i)|$ where the *redundant transmissions set* $TS_r(v_i) = \{TR(v_i, t) | TR(v_i, t) \in TS(v_i) \text{ and } t \notin T_s(v_i) \text{ and } RD(v_i, t) > 1\}$, and then make the following *redundancy reduction decision*:

- if $|TS_r(v_i)| > Load(v_i) - \max_{j \in LNS(v_i)} Load(j) + 1$, v_i will first update $TS(v_i) = TS(v_i) - TS_r^*(v_i)$ where $TS_r^*(v_i)$ denotes the set of *redundant transmissions* with the largest $Load(v_i) - \max_{j \in LNS(v_i)} Load(j) + 1$ RD values in $TS_r(v_i)$ and update $Load(v_i) = \max_{j \in LNS(v_i)} Load(j) - 1$, then send a competition packet containing $Load(v_i)$ and $T(v_i)$ to all the nodes in $CCS(v_i)$ where $T(v_i) = \{t | TR(v_i, t) \in TS_r^*(v_i)\}$;
- if $|TS_r(v_i)| \leq Load(v_i) - \max_{j \in LNS(v_i)} Load(j) + 1$, v_i will first update $TS(v_i) = TS(v_i) - TS_r(v_i)$ and set $Load(v_i) = 0$, then send a competition packet containing $Load(v_i)$ and $T(v_i)$ to all the nodes in $CCS(v_i)$ where $T(v_i) = \{t | TR(v_i, t) \in TS_r(v_i)\}$.

Upon receiving this competition packet, any $v_j \in CCS(v_i)$ will 1) check whether $Load(v_i)$ in this competition packet is 0. If yes, it will add v_i into a set $CPS_0(v_j)$ which is initially set as *null*; 2) check whether $T_s(v_j) \subseteq T(v_i)$. If no, v_j will just forward a new competition packet containing $Load(v_i)$ to all the nodes in $CPS(v_j) - CPS_0(v_j)$; if yes, v_j will update $deg(v_j) = deg(v_j) - 1$ and then forward a new competition packet containing $Load(v_i)$ and $deg(v_j)$ to all the nodes in $CPS(v_j) - CPS_0(v_j)$. For any node v_i with non-zero transmission load, when receiving all the competition packets attached with the *transmission load* of any $v_j \in LNS(v_i)$ from all nodes in $CCS(v_i) \cap CCS(v_j)$, it will first update the current $Load(v_i)$ and update $deg(k)$ for each $k \in CCS(v_i) \cap CCS(v_j)$, then check whether it is the *competition winner*. If no, it will just keep waiting; if yes, it will recompute the RD value of each transmission, make the aforementioned *redundancy reduction decision* and then keep waiting. The above process is iteratively performed until each of the nodes becomes the *terminated node* and any node v_i is called the *terminated node* if and only if either $Load(v_i) = 0$, or $Load(j) = 0$ for each $j \in LNS(v_i)$.

Finally, any node v_i will go into the low-duty-cycle mode according to the working schedule immediately after it becomes the *terminated node*, and once receiving a broadcasting message, v_i will forward it according to the *Transmission Strategy* $TS(v_i)$.

6.2. Example

Fig. 7 illustrates an example of our proposed DLBPA-A on the component as shown in Fig. 7(a). Initially, each node in S_i will set its *Transmissions Set* as follows (i.e., Fig. 7(b)):

$$TS(s_i^1) = \{TR(s_i^1, 1), TR(s_i^1, 2)\};$$

$$TS(s_i^2) = \{TR(s_i^2, 1), TR(s_i^2, 2)\};$$

$$TS(s_i^3) = \{TR(s_i^3, 2)\}; \quad TS(s_i^4) = \{TR(s_i^4, 1), TR(s_i^4, 2)\}.$$

then each node $r_i^q \in R_i$ will identify itself as the *unassigned node* and send a control packet containing $CPS(r_i^q)$ and $deg(r_i^q)$ to all nodes in $CPS(r_i^q)$. Accordingly, each node in S_i will respectively get its *transmission load* and *local nodes set* as follows:

$$Load(s_i^1) = 2, \quad LNS(s_i^1) = \{s_i^2\};$$

$$Load(s_i^2) = 2, \quad LNS(s_i^2) = \{s_i^1, s_i^3\};$$

$$Load(s_i^3) = 1, \quad LNS(s_i^3) = \{s_i^2, s_i^4\};$$

$$Load(s_i^4) = 2, \quad LNS(s_i^4) = \{s_i^3\}.$$

Next, each node $s_i^p \in S_i$ will send a competition packet containing $Load(s_i^p)$ to all the nodes in $LNS(s_i^p)$, and thus s_i^1 will become the *competition winner* since $s_i^1 \in L_{max}(s_i^1) = \{s_i^1, s_i^2\}$ and s_i^1 has the smaller ID number than s_i^2 . Likewise, s_i^4 will also become the *competition winner* since it has a larger transmission load than s_i^3 (i.e., Fig. 7(c)).

Afterwards, s_i^1 and s_i^4 will respectively find that $TS_r(s_i^1) = \{TR(s_i^1, 1)\}$ and $TS_r(s_i^4) = \{TR(s_i^4, 2)\}$, and according to our *redundancy reduction decision*, s_i^1 will update $TS(s_i^1) = \{TR(s_i^1, 2)\}$, set $Load(s_i^1) = 0$ and send a competition packet containing $Load(s_i^1)$ and $T(s_i^1)$ to all the nodes in $CCS(s_i^1)$. Similarly, s_i^4 will update $TS(s_i^4) = \{TR(s_i^4, 1)\}$, set $Load(s_i^4) = 0$ and send a competition packet containing $Load(s_i^4)$ and $T(s_i^4)$ to all the nodes in $CCS(s_i^4)$, then s_i^1 and s_i^4 will both identify themselves as *terminated nodes*, and 1) r_i^1 will update $deg(r_i^1) = 1$ and then send a new competition packet containing $Load(s_i^1)$ and $deg(r_i^1)$ to s_i^2 ; 2) r_i^3 will update $deg(r_i^3) = 1$ and then send a new competition packet containing $Load(s_i^1)$ and $deg(r_i^3)$ to s_i^2 ; 3) r_i^5 will update $deg(r_i^5) = 1$ and then send a new competition packet containing $Load(s_i^4)$ and $deg(r_i^5)$ to s_i^3 . Next, s_i^2 will become the *competition winner* (i.e., Fig. 7(d)) and according to our *redundancy reduction decision*, s_i^2 will first update $TS(s_i^2) = \{TR(s_i^2, 1)\}$, set

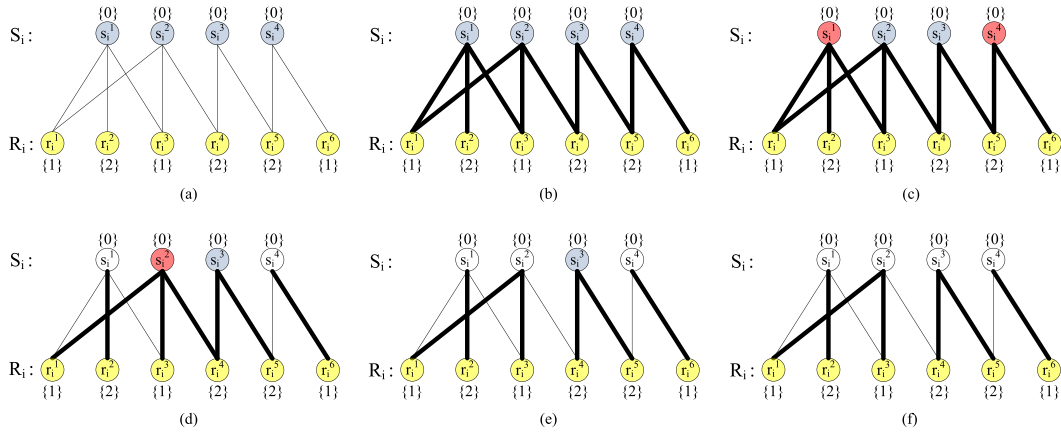


Fig. 7. An example of distributed solution (red nodes denote *competition winners* and white nodes denote *terminated nodes*). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

$Load(s_i^2) = 0$ and send a competition packet containing $Load(s_i^2)$ and $T(s_i^2)$ to all the nodes in $CCS(s_i^2)$, and then identify itself as *terminated node* (i.e., Fig. 7(e)).

Finally, r_i^4 will update $deg(r_i^4) = 1$ and then send a new competition packet containing $Load(s_i^2)$ and $deg(r_i^4)$ to s_i^3 , and s_i^3 will identify itself as *terminated node* since $Load(j) = 0$ for each $j \in LNS(s_i^3)$, therefore, we can get the final broadcasting schedule (i.e., Fig. 7(f)) as follows:

$$\begin{aligned} TS(s_i^1) &= \{TR(s_i^1, 2)\}; & TS(s_i^2) &= \{TR(s_i^2, 1)\}; \\ TS(s_i^3) &= \{TR(s_i^3, 2)\}; & TS(s_i^4) &= \{TR(s_i^4, 1)\}. \end{aligned}$$

6.3. Message complexity

In *Initialization Phase*, it is obvious that the MDPGC-A will cost $O(N^2)$ transmissions where N denotes the number of nodes in the network. After the MDPGC-A, each sensing node v_i will send one control packet to all nodes in $CPS(v_i)$ and some of nodes will then respectively send one competition packet. Upon receiving a competition packet from any node in $CPS(v_i)$, any *unassigned node* v_i will forward this competition packet to all nodes in $CPS(v_i)$. As we know, $|CPS(v_i)| \leq d_{max}$ where d_{max} denotes the maximum node degree in the network. Let $COST_{init}$ denote the number of transmissions in *Initialization Phase*, and we can have that $COST_{init} \leq O(N^2) + N + N + N \cdot d_{max} = O(N^2)$.

In *Competition Phase*, once any node v_i is determined as the *competition winner*, it will send one competition packet to all nodes in $CCS(v_i)$ and each node $v_j \in CCS(v_i)$ will then send at most one competition packet to all nodes in $CPS(v_j) - CPS_0(v_j)$. As $|CCS(v_i)| \leq d_{max}$, the event that any node is determined as the *competition winner* will result in at most $1 + d_{max}$ transmissions. Let $COST_{comp}$ denote the number of transmissions in *Competition Phase*, and it is obvious that any node v_i will determine itself as the *competition winner* for at most $|\bigcup_{j \in CCS(v_i)} T_s(j)|$ times. We can thus have that $COST_{comp} \leq (1 + d_{max}) \cdot |\bigcup_{j \in CCS(v_i)} T_s(j)| \cdot N \leq (1 + d_{max}) \cdot |CCS(v_i)| \cdot N \leq (1 + d_{max}) \cdot d_{max} \cdot N = O(N \cdot d_{max}^2)$.

Accordingly, the total message complexity of our proposed DLBPA-A is $COST_{init} + COST_{comp} \leq O(N^2) + O(N \cdot d_{max}^2) = O(N^2 + N \cdot d_{max}^2)$.

7. Practical issues

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

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 paper 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 paper. In real WSNs, local synchronization can be achieved by using an existing high-efficient MAC-layer time stamping technique FTSP (Flooding Time Synchronization Protocol) [31], 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 can be realized by only exchanging the schedules between neighboring nodes immediately after the deployment. In our solution, we will use a binary string to represent the working schedule, e.g., to use the binary string $<0010>$ to represent the periodic working schedule $\{2\}$ with $L = 4$. 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.

In our centralized solution, each node will initially keep awake immediately after the deployment and the sink will derive the network topology according to some existing solution. Based on the network topology, the sink will execute our centralized algorithm to obtain the broadcasting schedule and then distribute it to all nodes in the network, and 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. For our distributed solution, we will first install our proposed distributed algorithm on each node, then each node will execute our distributed algorithm as described in Section 6.1 immediately after the deployment. For both solutions, once getting the transmission strategy, each node will put itself into the low-duty-cycle mode according to its own working schedule (the working schedule can be determined independently by itself or can rely on a particular power management protocol). 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 paper, we do not need to require specific sink location. Note that our objective is to minimize the maximum transmission load of a broadcast schedule for low-duty-cycle WSNs, subject to the constraint that each node should have the minimum end-to-end delay under the

broadcast schedule. We can easily find that the change of sink location will only influence the minimum end-to-end delay for each node, but will absolutely not influence the performance (e.g., approximation ratio) of our solution in terms of energy fairness.

In this paper, we assume that our target applications will not experience a notable change on the link qualities, which implies the topology changes mainly come from the energy depletion of sensing nodes. In practice, some emerging technologies (e.g., Wireless Charging Technology [32] and Mobile Robot Technology [33].) 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.

8. Performance evaluation

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

8.1. Experiment setup

In our setting, we consider that all sensor nodes are uniformly distributed in a 100 m×100 m sensory field with the sink node located at the center of the field. For simplicity and without loss of generality, we assume that one period of any node's working schedule contains only one *active state* time slot, and the *disk communication model* is adopted, i.e., any node A can successfully deliver a packet to any node B if and only if node B is located within the communication range of node A . Actually, our solution can still be available and well-performed for a *general communication model* where the communication link does not totally depend on the distance. Here, we assume that all nodes have the same communication range R_c and we set $R_c = 10$ m. Further, we let each sensing node randomly determine its own working schedule and for the sink node v_0 , we set $T_s(v_0) = \{0\}$. Unless otherwise stated, we set $N=800$, $L=50$ (i.e., duty cycle=2%), and all the results are obtained by averaging over 50 experiments.

As stated in Section 3.1, we assume the interarrival time between any two successive broadcast requests from the sink is more than the minimum broadcast latency, which implies we do not need to define specific broadcasting message traffic in our performance evaluation, since this practical assumption can make sure that the collision between any two requests must not occur under whatever broadcasting message traffic. Likewise, we also do not need to consider the control traffic, because the control traffic only occurs at the construction stage of broadcasting schedule, which means it is only a one-time task. Obviously, the control traffic, which only occurs at the initialization phase of the deployed network, will not influence the broadcasting message traffic in practice. As the control packets are short and their transmissions only occur at the initialization phase of the deployed network, the overhead induced by our proposed solutions are so small compared with the whole energy consumption that can be neglected. Thus we can only take into account the transmission load when computing energy consumption.

Here, we compare our solutions with two traditional solutions. One is the well-known *Bellman-Ford Algorithm*, which is commonly used to find the shortest path tree. Another one is the *Random Parent Selection Algorithm* (RPS-A), in which each sensing node v_i will

randomly select one node in $CPS(v_i)$ as its parent after performing MDPGC-A.

8.2. Performance of energy fairness

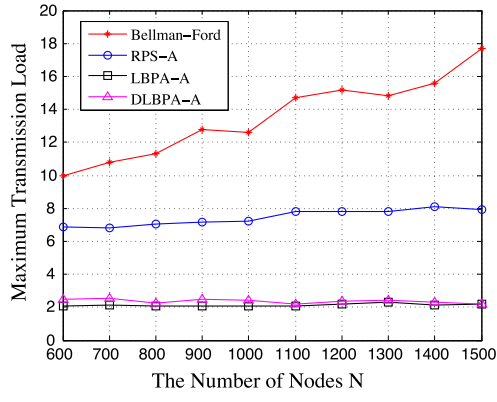
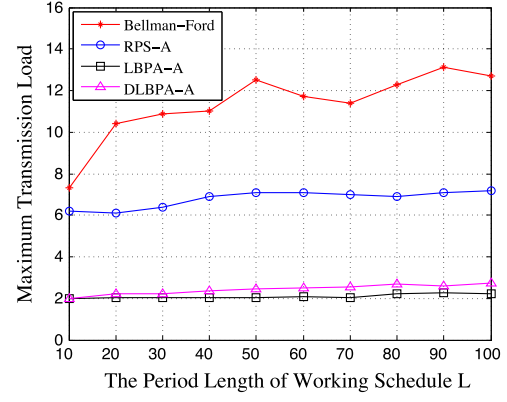
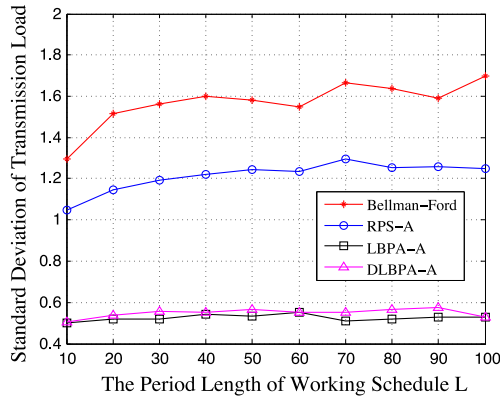
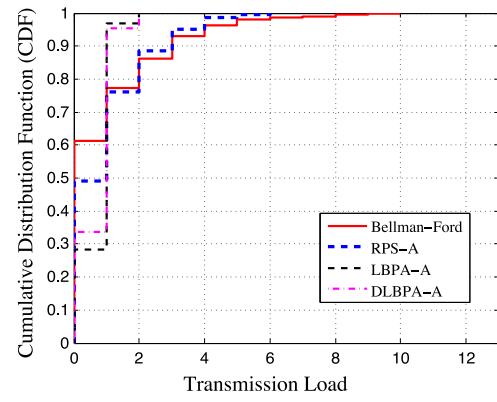
First, we will show the performance of energy fairness. As shown in [1], idle listening mode has nearly the same energy consumption rate with the receiving mode for each real sensor node in practice, which implies any node will consume nearly a constant energy amount in an active time slot no matter whether any packet is received in this slot. In other words, we do not need to consider the receiving energy consumption for all nodes (i.e., the receiving energy consumption for all nodes can be approximately neglected), because in our model, each node only receives the packet at its scheduled active time slot and the receiving of packets does not bring any additional energy consumption. In addition, we assume that our target networks have commonly adopted the load-balanced data collection protocols and also assume that each node has the same initial energy immediately after the deployment, therefore, we only need to consider transmission load as the main metric of energy consumption for broadcasting and similar transmission loads can absolutely ensure energy fairness.

Fig. 8 (a) and (b) compare our solutions with the traditional solutions in terms of the maximum transmission load, and we find that our proposed LBPA-A and DLBPA-A can always get much better performance than *Bellman-Ford Algorithm* and RPS-A no matter how N or L (i.e., duty cycle) varies. As shown in both Fig. 8(a) and Fig. 8(b), *Bellman-Ford Algorithm* exhibits the worst performance since in *Bellman-Ford Algorithm*, each node v_i will only select the first coming node in $CPS(v_i)$ as its parent which would make some nodes undertake much more load. In addition, LBPA-A exhibits a slightly better performance than DLBPA-A. This is because compared with DLBPA-A, LBPA-A utilizes the global information to greedily reduce the redundancy and adopts a better comparison metric considering the influence of the removal of the redundant transmission on the RD values of the transmissions from the other senders. Also, we can find that under the assumption of uniform distribution and *disk communication model*, LBPA-A and DLBPA-A can always get a stable maximum transmission load, i.e., around 2–3, which is near-optimal as the optimal solution must be at least 1.

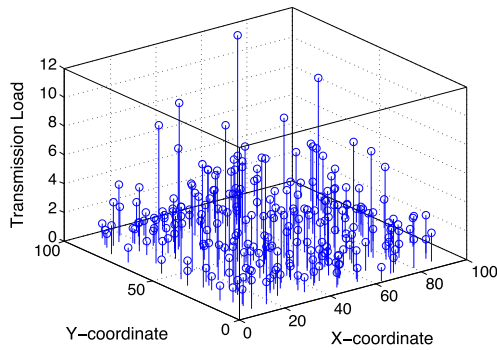
Furthermore, as shown in Fig. 8(c), LBPA-A and DLBPA-A can always achieve a much lower *Standard Deviation* of the transmission load for all sensing nodes, and Fig. 8(d) exhibits a representative CDF of transmission load for these four solutions. Fig. 9 explicitly illustrates the distribution of transmission loads. We can find that LBPA-A and DLBPA-A exhibit a much better load balancing between sensor nodes. Accordingly, we can conclude that compared with the traditional *Bellman-Ford Algorithm* and RPS-A, our proposed LBPA-A and DLBPA-A can obtain a better energy fairness.

8.3. Performance of total energy consumption

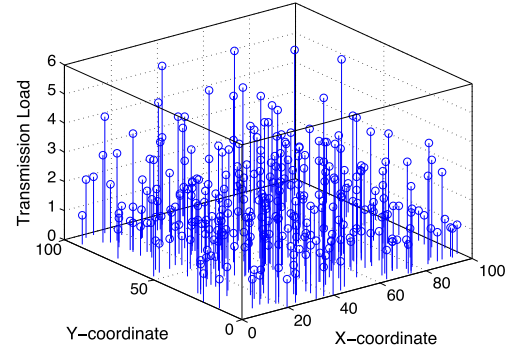
As the efficient schemes for redundant transmissions reduction are adopted, LBPA-A and DLBPA-A perform much better in terms of the total transmission load than *Bellman-Ford Algorithm* and RPS-A, which can be seen in Fig. 10(a) and (b). Here, the total transmission load denotes the transmission load cumulated on all nodes. In Fig. 10(a) and Fig. 10(b), we can find that DLBPA-A performs slightly better than LBPA-A, and compared with *Bellman-Ford Algorithm* and RPS-A, the performance advantage of LBPA-A and DLBPA-A will be enlarged as the number of nodes N increases or the period length of working schedule L decreases (i.e., the duty cycle increases). This is because the increase of N or the decrease of L will make more neighboring nodes have the same working schedule, which would bring more redundant transmissions and provide more opportunities for

(a) maximum transmission load vs. N (b) maximum transmission load vs. L (c) standard deviation vs. L 

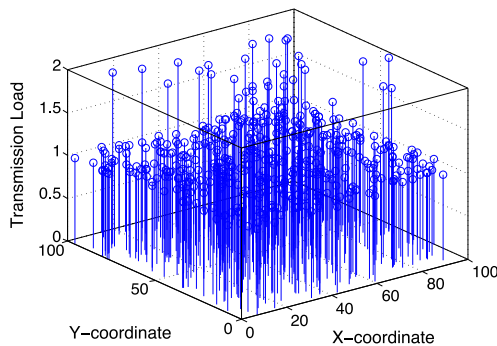
(d) CDF of transmission load

Fig. 8. Performance of energy fairness.

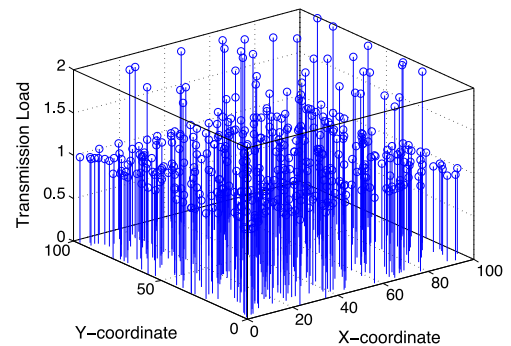
(a) Bellman-Ford



(b) RPS-A



(c) LBPA-A



(d) DLBPA-A

Fig. 9. Illustration of Transmission Load Distribution ($N=600$ and $L=50$).

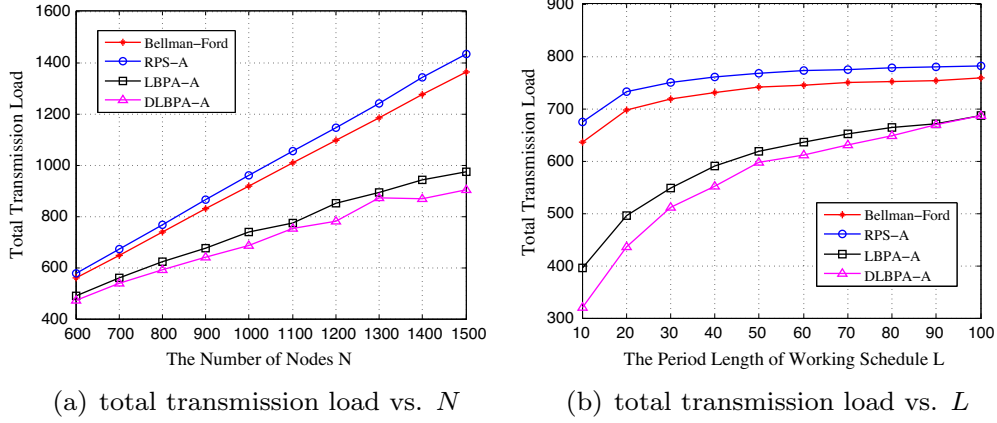
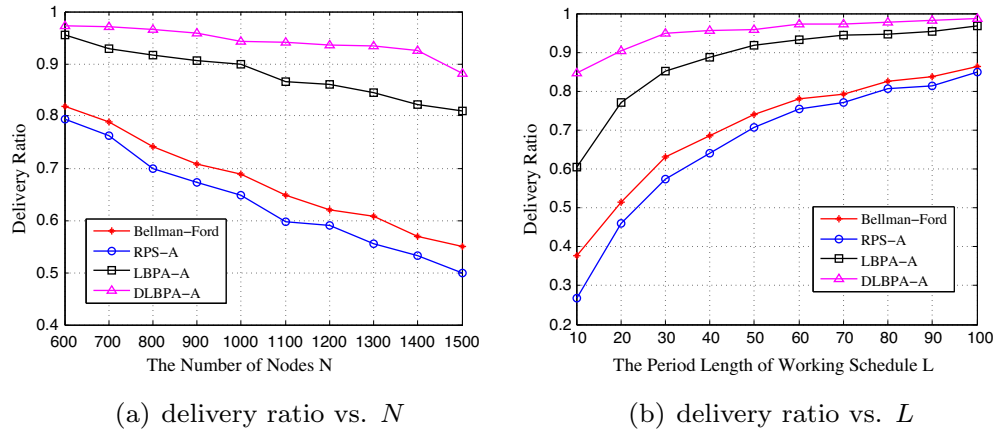


Fig. 10. Performance of total energy consumption.

Fig. 11. Performance of delivery ratio when $|T| = 1$.

LBPA-A and DLBPA-A to improve total energy consumption by efficiently reduce the redundancy of transmissions.

8.4. Performance of delivery ratio

In the above sections, we do not consider the packet collision problem for simplicity. Here, we will mainly investigate the impact of packet collision problem on delivery ratio for our solutions in low-duty-cycle WSNs and propose a simple and efficient approach to further improve the delivery ratio in practice. First, we assume each time slot can only accommodate one packet transmission, i.e., $|T| = 1$ where $|T|$ denotes the largest number of packet transmissions that each time slot can accommodate. Here, we define that a collision will happen if and only if some node receives the same packet from multiple different senders at the same time and once a collision happens at some node, the node will fail to receive this packet. Fig. 11(a) and (b) show the delivery ratio of our solutions and the traditional solutions when $|T| = 1$. We can find that no matter how N or L (i.e., duty cycle) varies, LBPA-A and DLBPA-A can always achieve a much higher delivery ratio than Bellman-Ford Algorithm and RPS-A, and DLBPA-A performs better than LBPA-A. The delivery ratio of our solutions will become higher as N decreases or L increases (i.e., duty cycle decreases). This can be explained by the fact that the decrease of the number of nodes or the decrease of duty cycle will make less neighboring nodes have the same working schedule, which would reduce the collision chance in the network. As shown in Fig. 11(a), LBPA-A and DLBPA-A can always keep a delivery ratio of over 80% when N varies from

600 to 1500 and a delivery ratio of over 90% can be achieved when the network scale is moderate (i.e., $N \leq 1000$). Specifically, the delivery ratio of LBPA-A and DLBPA-A can respectively achieve 95.56% and 97.29% when $N = 600$. As in Fig. 11(b), LBPA-A and DLBPA-A can achieve a delivery ratio of over 90% when $L \geq 50$ (i.e., duty cycle $\leq 2\%$). Specifically, the delivery ratio of LBPA-A and DLBPA-A can respectively achieve 96.85% and 98.7% when $L = 100$ (i.e., duty cycle = 1%). We thus conclude that for moderate-scale low-duty-cycle WSNs, our solutions can always achieve a considerable delivery ratio when $|T| = 1$.

In practice, each time slot is always set as long enough (i.e., $|T| > 1$), because in the applications of data collection, each forwarder is usually expected to receive as more data packets as possible from its children at one time slot to make the data aggregation in order to achieve high energy efficiency. Here, we utilize a simple and efficient *random transmission strategy* to further improve the delivery ratio of our solutions. Specifically, we will divide each time slot into $|T|$ equal time units so that each time unit can just accommodate one packet transmission, and if any node is scheduled to transmit a broadcasting packet at some time slot, it will randomly select a time unit of this time slot to send the packet. By this way, the packet collision chance can be further reduced greatly. Fig. 12 exhibits the relationship between $|T|$ and delivery ratio for all four solutions when *random transmission strategy* is adopted. We can find that the delivery ratio increases as $|T|$ increases for all four solutions and our solutions always exhibit better performance than the traditional solutions regardless of $|T|$, and the delivery ratio of our solutions will be

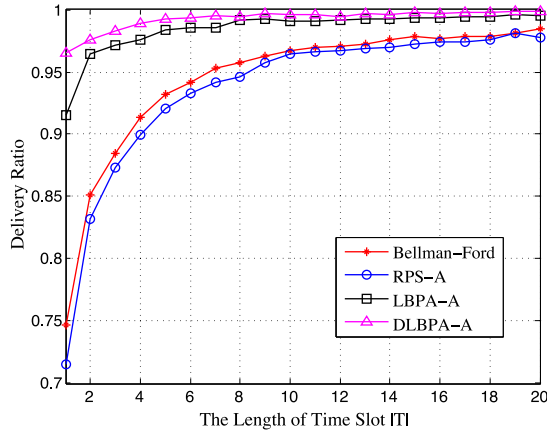


Fig. 12. Relationship between delivery ratio and $|T|$.

significantly improved especially when $|T|$ changes from 1 to 2. In Fig. 12 where $N = 800$, $L = 50$ and $R_c = 10$ m, it can be seen that when $|T| > 1$, both of LBPA-A and DLBPA-A always achieve a delivery ratio of over 95%, and a delivery ratio of over 99% can be achieved when $|T| \geq 8$ for LBPA-A and $|T| \geq 5$ for DLBPA-A, which indicates that for low-duty-cycle WSNs with *random transmission strategy*, the packet collision problem can thus be approximately neglected as the delivery ratio usually approaches to 100% in practice.

As the interarrival time between any two successive broadcast requests is always far more than the minimum broadcast latency in real WSNs, the delivery ratio of our proposed solutions will thus absolutely not be influenced under whatever broadcasting message traffic. Intuitively, the topology change will influence the delivery ratio of broadcasting, however, it is worthless to consider the influence of topology change on the delivery ratio. This is because we can periodically re-execute our algorithm to update the broadcasting schedule in our solutions according to the history record about the frequency of topology change and thus the influence of topology change on the delivery ratio is so less that it can be neglected in practice. Moreover, it does not make any sense to characterize the relationship between topology change and delivery ratio, since we can easily find that some local link or node change could increase, decrease or unchange the delivery ratio and there does not exist a notable correlation between topology change and delivery ratio.

9. Conclusion

In this paper, we mainly focus on the *Load-Balanced Minimum Delay Broadcast Scheduling Problem* (LB-MDBS) for low-duty-cycle WSNs. We first transform our LB-MDBS problem into the equivalent *Load-Balanced Parents Assignment Problem* (LBPA), and prove its NP-hardness. Then, we address this problem by proposing the *Load-Balanced Parents Assignment Algorithm* (LBPA-A) which achieves λ -approximation, where λ denotes the maximum number of neighbors that are scheduled to wake up at the same time and is typically a small number in low-duty-cycle WSNs. Also, we present an efficient distributed solution. Finally, the high-efficiency of our solutions has been evaluated by simulations in terms of energy fairness, total energy consumption and delivery ratio.

Acknowledgment

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 (Grant

No. 61502251, 61472252, 61133006, 61321491) and NUPTSF (Grant No. NY214169); X. Zhu was supported by the National Natural Science Foundation of China (Grant No. 61502232) and China Postdoctoral Science Foundation (Grant No. 2015M570445); H. Dai was supported by the National Natural Science Foundation of China (Grant No. 61502229); X. Wu was supported by the National Natural Science Foundation of China (Grant No. 61373130), NSF of Jiangsu Province (Grant No. BK20141319), EU FP7 CROWN project under grant number PIRSES-GA-2013-610524 and Funding for Central Institutions of China (Grant No. 20620140515).

References

- [1] M. Stemm, R.H. Katz, Measuring and reducing energy consumption of network interfaces in hand-held devices, *IEICE Trans. Commun.* E80-B (8) (1997) pp.1125–1131.
- [2] Texas Instruments. 2.4 GHz IEEE 802.15.4 / ZigBee-Ready RF Transceiver (Rev. C) (2013). Available at <http://www.ti.com/lit/ds/symlink/cc2420.pdf> (accessed 05.12.13).
- [3] F. Wang, J. Liu, On reliable broadcast in low duty-cycle wireless sensor networks, *IEEE Trans. Mobile Comput.* 11 (5) (2012) 767–779.
- [4] J. Hong, J. Cao, W. Li, S. Lu, D. Chen, Minimum-transmission broadcast in uncoordinated duty-cycled wireless ad hoc networks, *IEEE Trans. Veh. Technol.* 59 (1) (2010) pp.307–318.
- [5] S. Guo, S. Kim, T. Zhu, Y. Gu, T. He, Correlated flooding in low-duty-cycle wireless sensor networks, in: *Netw. Protocols (ICNP)*, 2011.
- [6] Y. Sun, O. Gurewitz, S. Du, L. Tang, D.B. Johnson, ADB: an efficient multihop broadcast protocol based on asynchronous duty-cycling in wireless sensor networks, in: *ACM SenSys*, 2009, pp. 43–56.
- [7] L. Xu, J. Cao, S. Lin, H. Dai, X. Wu, G. Chen, Energy-efficient broadcast scheduling with minimum latency for low-duty-cycle wireless sensor networks, in: *Proceedings of IEEE MASS'13*, 2013.
- [8] S. Lai, B. Ravindran, On Multihop Broadcast over Adaptively Duty-Cycled Wireless Sensor Networks, in: *DCOSS'10*, 2010.
- [9] T. Zhu, Z. Zhong, T. He, Z. Zhang, Exploring link correlation for efficient flooding in wireless sensor networks, in: *NSDI*, 2010.
- [10] K. Han, Y. Liu, J. Luo, Duty-cycle-aware minimum-energy multicasting in wireless sensor networks, *IEEE/ACM Trans. Netw. (TON)* 21 (3) (2013) 910–923.
- [11] K. Han, J. Xiang, J. Luo, et al., Energy-efficient reliable data dissemination in duty-cycled wireless sensor networks, in: *Proceedings of the fourteenth ACM international symposium on Mobile ad hoc networking and computing*, ACM, 2013, pp. 287–292.
- [12] L. Cheng, Y. Gu, T. He, et al., Dynamic switching-based reliable flooding in low-duty-cycle wireless sensor networks, in: *Proceedings of IEEE INFOCOM*, 2013, pp. 1393–1401.
- [13] J. Niu, L. Cheng, Y. Gu, J.H. Jun, Q. Zhang, Minimum-Delay and Energy-Efficient Flooding Tree in Asynchronous Low-Duty-Cycle Wireless Sensor Networks, in: *Proceedings of IEEE WCNC*, 2013.
- [14] F. Ferarri, M. Zimmerling, L. Thiele, O. Saukh, Efficient Network Flooding and Time Synchronization with Glossy, in: *Proceedings of ACM IPSN'11*, 2011.
- [15] X. Wu, G. Chen, S.K. Das, Avoiding energy holes in wireless sensor networks with nonuniform node distribution, *IEEE Transactions on Parallel and Distributed Systems (TPDS)* 19 (5) (2008).
- [16] R. Jurdak, P. Baldi, C.V. Lopes, Adaptive Low Power Listening for Wireless Sensor Networks, *IEEE Trans. Mobile Comput.* 6 (8) (2007) 988–1004.
- [17] S. Xiong, J. Li, M. Li, J. Wang, Y. Liu, Multiple task scheduling for low-duty-cycled wireless sensor networks, in: *Proceedings of IEEE INFOCOM'11*, 2011, pp. 1323–1331.
- [18] Z. Li, Y. Peng, D. Qiao, W. Zhang, LBA: Lifetime balanced data aggregation in low duty cycle sensor networks, in: *Proceedings of IEEE INFOCOM'12*, 2012, pp. 1844–1852.
- [19] L. Mai, L. Shanguan, J. Du, Z. Li, M. Li, Load Balanced Rendezvous Data Collection in Mobile Wireless Sensor Networks, in: *Proceedings of IEEE MASS'11*, 2011.
- [20] Z. Li, M. Li, Y. Liu, Towards energy-fairness in asynchronous duty-cycling sensor networks, *ACM Trans. Sensor Netw. (TOSN)* 10 (3) (2014) 38.
- [21] L. Xu, G. Chen, X. Yin, P. Yang, B. Yang, LORP: A Load-Balancing Based Optimal Routing Protocol for Sensor Networks with Bottlenecks, in: *Proceedings of IEEE WCNC'09*, 2009.
- [22] X. Zhu, X. Wu, G. Chen, An exact algorithm for maximum lifetime data gathering tree without aggregation in wireless sensor networks, *Wirel. Netw.* 21 (1) (2015) 281–295.
- [23] Q. Cao, T. Abdelzaher, T. He, J. Stankovic, Towards Optimal Sleep Scheduling in Sensor Networks for Rare Event Detection, in: *IPSN'05*, 2005.
- [24] Y. Gu, T. He, Data forwarding in extremely low duty-cycle sensor networks with unreliable communication links, in: *Proceedings ACM SenSys'07*, 2007, pp. 321–334.
- [25] S. Guo, Y. Gu, B. Jiang, T. He, Opportunistic Flooding in Low-Duty-Cycle Wireless Sensor Networks with Unreliable Links, in: *MobiCom'09*, 2009.
- [26] X. Jiao, W. Lou, J. Ma, J. Cao, X. Wang, X. Zhou, Duty-cycle-aware minimum latency broadcast scheduling in multi-hop wireless networks, in: *ICDCS*, 2010.
- [27] L. Su, B. Ding, Y. Yang, T.F. Abdelzaher, G. Cao, J. Hou, ocast: Optimal multicast routing protocol for wireless sensor networks, in: *ICNP*, 2009, pp. 151–160.

- [28] M.R. Garey, D.S. Johnson, *Computers and Intractability: A guide to the theory of NP-completeness*, W.H. Freeman and Co., 1979.
- [29] N.J. Harvey, R.E. Ladner, L. Lovasz, T. Tamir, Semi-matchings for bipartite graphs and load balancing, in: In WADS, 2003.
- [30] J. Fakcharoenphol, B. Laekhanukit, D. Nanongkai, Faster algorithms for semi-matching problems, in: In IICALP, 2010.
- [31] M. Marti, B. Kusy, G. Simon, A. Ldeczi, The Flooding Time Synchronization Protocol, in: *Proceedings of ACM SenSys'04*, 2004.
- [32] H. Dai, X. Wu, G. Chen, L. Xu, S. Lin, Minimizing the number of mobile chargers for large-scale wireless rechargeable sensor networks, *Comput. Commun.* 46 (2014) 54–65.
- [33] G. Fletcher, X. Li, A. Nayak, et al., Randomized robot-assisted relocation of sensors for coverage repair in wireless sensor networks, in: *Vehicular Technology Conference Fall (VTC 2010-Fall)*, 2010 IEEE 72nd. IEEE, 2010, pp. 1–5.