IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received October 6, 2019, accepted October 22, 2019, date of publication October 25, 2019, date of current version November 14, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2949569

DBDC: A Distributed Bus-Based Data Collection Mechanism for Maximizing Throughput and Lifetime in WSNs

CHIH-YUNG CHANG[1], (Member, IEEE), CHUNG-CHIH LIN[1], CUIJUAN SHANG[2], I-HSIUNG CHANG[1], AND DIPTENDU SINHA ROY[3]

[1]Department of Computer Science and Information Engineering, Tamkang University, New Taipei City 25137, Taiwan
[2]School of Computer and Information Engineering, Chuzhou University, Chuzhou 239000, China
[3]Department of Computer Science and Engineering, National Institute of Technology Meghalaya, Shillong 793003, India

Corresponding author: Chih-Yung Chang (cychang@mail.tku.edu.tw)

This work was supported by the Ministry of Science and Technology, Taiwan under Grant MOST 106-2221-E-032-015-MY3 and Grant MOST 108-2622-E-032-002-CC3.

ABSTRACT Data collection is one of the most important research topics in WSNs. In literature, many studies have proposed centralized solutions to cope with the data collection problem. However, most of them considered controllable mobile sink which is controlled by an algorithm to determine its speed, path, stop locations as well as the performed task. In fact, the uncontrollable mobile sink can be also applied to collect data from a given set of deployed sensors. A number of studies assumed that the sink is fixed and all sensors transmit their data to the sink. However, it leads to the problems of unbalanced workload and network disconnection. Some other studies scheduled the controllable mobile sink. However, the algorithms developed by adopting the controllable mobile sink cannot be applied to the scenarios where the uncontrollable mobile sink is adopted. The main reason is that the stops and arrival time of the uncontrollable mobile sink are unknown. In addition, the problems including the high hardware cost and energy limitation of the controllable mobile sink are still needed to be overcome. This paper proposes a distributed data collection mechanism, called Distributed Bus-based Data Collection (*DBDC*) algorithm, which considers the bus as mobile sink aiming to maximize the amount of collected data and the network lifetime of wireless sensor networks. Applying the proposed *DBDC*, each sensor negotiates with its neighbors based on a bidding procedure such that the sensor that buffers more data can obtain more sharing slots instead of increasing its power level. To prolong the network lifetime, the sensor with higher remaining energy can enlarge its transmission power, aiming to release more sharing slots to cooperatively help the neighbor that buffers more data. Experimental study reveals that the proposed *DBDC* algorithm outperforms related works in terms of throughput, network lifetime and fairness.

INDEX TERMS Data collection, wireless sensor network, slot scheduling, power adjusting, fairness.

I. INTRODUCTION

Wireless sensor networks (WSNs) are composed of many sensor nodes each of which supports functions including sensing, data processing, and communication. The WSNs are used in many applications, including agriculture, country boundaries, battled surveillance, machine health monitoring as well as environmental monitoring [1], [2]. With the high demand for smart life, WSNs have become a key infrastructure of the Internet of Things [3].

The associate editor coordinating the review of this manuscript and approving it for publication was Yasar Amin.

Data collection is one of the most important research topics in WSNs. In literature, a large number of studies [4]–[7] have discussed the data collection issue. Since sensors are battery powered, energy conservation has been an important issue which has been widely discussed in the past few years. Basically, these works can be classified into two categories: fixed sink and mobile sink.

In the category of data collection using fixed sink node, many studies [8]–[11] discussed how to construct a topology for each sensor transmitting its data to the fixed sink in a multi-hop manner. In this way, the sensor closer to the fixed sink will consume more energy for relaying data received from other sensors, leading to unbalanced energy of sensors or even network disconnection.

To avoid the problem of unbalanced workload and network disconnection, many studies considered mobile sink to visit some sensors [12]–[15] or even all sensors [16]–[18]. There are two types of research works in this area, depending on whether the mobile sink is controllable. The controllable mobile sink is generally a special purpose mobile device which is controlled by an algorithm to determine its speed, path, stop locations as well as operations designed in the performing task. There have been many studies in the past using the controllable mobile sink. Most of them aimed to choose proper anchor nodes from static sensors and construct the shortest path for the mobile sink. However, problems such as high hardware cost and energy limitation are still needed to be overcome.

Another type of research works relies on the inaccessible mobile sink to collect data from sensors. These uncontrollable mobile sinks can be animals, climbers, buses or bicycles which are embedded with a data collection equipment. Since the stops and arrival time of the mobile sink is unknown [19], the duration of contacts between mobile sink and static sensor cannot be controlled. This challenge constrains the data amount transmitted from the static sensor to the mobile sink and hence the buffers of static sensors might be overflowed. Another challenge is that the communication range of mobile sink is overlapped with more than one static sensors, raising the contention and collision problems.

This paper considers the wireless sensor network where a number of static sensors have been randomly deployed along the road aiming to collect data such as the number of vehicles, individual vehicle speed, traffic flow, noise, PM 2.5 or the number of large trucks. This paper considers to use buses or official vehicles which are uncontrolled in terms of stops and arrival time. Since there are regular bus services, the bus embedded with the data collectors can play the role of mobile sink and periodically collect data from static sensors when the communication range of the bus is overlapped with that of static sensors.

This paper aims to develop a distributed data collection mechanism which uses bus for collecting data from roadside static sensors. The main goal of the developed mechanism is to maximize the amount of collected data while prolonging the network lifetime of wireless sensor networks. The proposed *DBDC* algorithm allows each sensor to locally negotiate with neighbors and schedule its time slots based on the size of buffered data, remaining energy and the number of time slots contacted with the bus. The proposed *DBDC* fully utilizes the time and energy resources, aiming to achieve both goals of maximal throughput and maximal network lifetime of the static wireless sensor networks.

The key contributions of the proposed *DBDC* are itemized as follows:

(1) **Achieving higher throughput.** The sensors having more buffered data can achieve higher throughputs by applying the novel policies proposed in *DBDC*. The first policy is to allocate sharing time slots based on the ratio of buffered data size between neighbors. The second policy is to borrow sharing time slots from neighbors by executing the proposed bidding procedure. The third policy is to enlarge the transmission power for increasing the data transmission rate. Compared with the existing work [20], the throughput can be significantly improved.

(2) **Prolonging network lifetime of the WSNs.** The sensors with lower remaining energy adopt lower power level to transmit data to the bus for prolonging the network lifetime. In case that the low energy sensors buffered more data, they will be allocated more sharing time slots by executing the bidding procedure, instead of enlarging their transmission power. More, the neighbors with higher remaining energy can also enlarge the transmission power. This can increase the data transmission rate and hence release some sharing time slots which can be borrowed by the low energy neighbor. Compared with existing work [20], the network lifetime of the WSNs can be significantly improved.

(3) **Improving the fairness of resource allocation.** By applying the distributed bidding procedure designed in *DBDC*, sensors having more buffered data and fewer connection time can have more transmission time slots. In addition, the sensor with lower remaining energy can save more energy, as compared with the neighboring sensor with higher remaining energy. Hence the fairness in terms of time and energy resources can be maintained. Compared with existing work [20], the fairness can be significantly improved.

The remaining part of this paper is organized as follows. Section II presents the related work of this study. Section III introduces the network environment and problem statement. Section IV gives the detailed description of the proposed *DBDC* algorithm. Section V presents the simulation results. Finally, a conclusion of the proposed algorithms and future work are drawn.

II. RELATED WORK

In recent years, a large number of data collection mechanisms which adopted mobile sink to collect data from static sensors have been proposed. These studies can be classified into two categories: controllable mobile sink and uncontrollable mobile sink. The following reviews these related studies.

A. NO-DATA-FORWARDING USING CONTROLLABLE MOBILE SINK

In this category, a mobile sink is used to visit all sensors such that sensors can directly transmit data to the mobile sink. Ma and Yang [21] proposed a path construction algorithm which established a path passing through all sensor nodes. Since all sensor nodes can directly transmit data to the mobile sink without any forwarding load, their energies are balanced. Though the proposed algorithm tried to reduce the path length, the delay increases with the sizes of network and monitoring region. Sugihara and Gupta [22] developed an algorithm aiming at constructing a shortest path passing through all sensor nodes for minimizing the delay of data collection. Somasundara *et al.* [23] proposed a path construction algorithm which aims to allow the mobile sink visiting each static sensor for collecting data before the buffer of any sensor overflowed. They formulated the path construction problem as an integer linear programming problem. Though the above-mentioned studies used a mobile sink to visit each sensor to cope with the problem of unbalanced work load, the long path length leads to the problems of long propagation delay and limited sensor buffer which are needed to be overcome.

B. PARTIAL-DATA-FORWARDING USING CONTROLLABLE MOBILE SINK

The concept of partial data forwarding is to select some appropriate sensors from all static sensor nodes as the data collection points (CPs) which will be visited by the mobile sink. This is an alternative which reduces the path length of mobile sink and hence improves the time delay of data collection. To collect all data from the whole sensor network, all the other sensors should transmit their own data to the collection points.

Zhao and Yang [7] proposed a centralized method which iteratively selects one best node to play the CP based on a shortest path tree. In addition, a distributed method was proposed for constructing a set of shortest path trees. All sensor nodes transmitted their own data to its tree root along the tree topology and the mobile sink only visits the tree roots for data collection. The developed selection method of CPs considered the energy consumption for forwarding loads and reduced the total delay for data collection. Similarly, study [13] proposed a path construction mechanism, called WRP, which considered different parameter for selecting CPs based on a tree structure. The constructed path aims to prolong the lifetime of static sensor network under the constraint of limited path length. Almi'ani *et al.* [24] proposed a path construction algorithm for mobile sink to visit some selected CPs. The algorithm firstly partitions the network into several clusters. A cluster head is selected to collect the data from sensors in the same cluster. Then the algorithm constructed a path passing through all cluster heads. Hence the mobile can collect all data along the path.

In the above mentioned studies, the mobile sink can collect data and then get back to the base station within a certain time period. However, the selected CPs will consume more energy than the other sensors, leading to the problem of unbalanced work load. In addition, the challenges such as the hostile outdoor environments and high hardware costs are still needed to be overcome.

C. UNCONTROLLABLE MOBILE SENSOR

Some other studies considered the uncontrollable mobile sensors such as people, buses, bicycles or animals to collect data from sensor networks. Study [20] proposed a Mobi-Cluster algorithm, which is a rendezvous-based solution for data collection. The mobile sinks can be mounted upon city buses that repeatedly follow a predefined trajectory with a periodic schedule. The rendezvous nodes are in close proximity with the mobile sink trajectory. However, the data transmission from the rendezvous nodes to the bus did not consider the shared slot scheduling. Study [25] considered the animals or vehicles as the mobile sinks for collecting data from static sensors. Although there is no energy consumption problem, it requires long time for collecting data from all static sensors since the mobile sinks are totally uncontrollable. Some other studies considered the uncontrollable but predictable mobile sink to collect data from static sensor networks. Wu *et al.* [26] proposed a method which used uncontrollable mobile sink moving along the predefined path to collect data from one-hop sensors periodically. However, the stops and arrival time of the uncontrollable mobile sink are unknown, leading to the difficulty for scheduling the data transmission. In [27], a learning-based technique has been proposed to predict the arrival time probability and thus the duty cycle of sensors can be arranged based on the predictions of next arrival time. Since the overlapping area of sensors can cause contention and collision problem, the transmission scheduling is an important factor which impacts the throughput and network lifetime. A good transmission scheduling is still required to improve the performance of network throughput and lifetime. Study [28] proposed a comprehensive data gathering scheme based on graphing technique, aiming to optimize the energy consumption of all sensor nodes hierarchically. However, the slot utilization and fairness issues were not taken into consideration.

Though aforementioned studies proposed many algorithms using uncontrollable mobile sink to collect data from sensor networks. However, most studies concerned the predicted arrival time of the mobile sink but did not consider the issues of throughput and energy balancing issues of the static sensors. This paper proposed a *DBDC* algorithm using predictable bus to collect data from sensors which are deployed around the roadside. The proposed *DBDC* is a distributed transmission scheduling mechanism aiming to maximize the volume of collected data from static sensors while prolonging the lifetime of static sensor networks.

III. NETWORK ENVIRONMENT AND PROBLEM FORMULATION

This section initially introduces the network environment and assumptions of this work. Then, the problem formulation of the investigated issue is presented.

A. NETWORK ENVIRONMENT

Consider a given WSN W, which comprises a set of n static sensors $S = \{s_1, s_2, \ldots, s_n\}$ randomly deployed along a road L. Assume that each sensor is aware of its own unique ID, location, remaining energy and buffer capacity. Each sensor generates new sensing data in the buffer with a certain constant rate. A bus b will periodically move along the road L at a constant velocity v and its communication range is

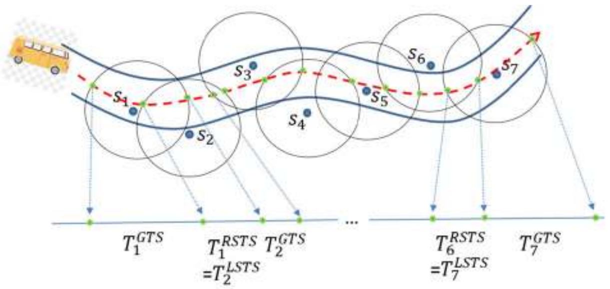


FIGURE 1. **The scenario considered in this paper.**

overlapped with each sensor in S. This work considers the bus as mobile sink which has been embedded with a receiver for collecting data from sensors in S.

B. PROBLEM FORMULATION

This paper proposes a Distributed Bus-based Data Collection (*DBDC*) algorithm which aims at transmitting maximal amount of data from sensor nodes in S to bus b while the network lifetime of W can be prolonged. Let notation l denote the length of the road that is covered by S. Let T_c denote the time period of one round which is the time duration that bus b contacts with the WSN. That is, the bus can receive data from the sensors during T_c. It is obvious that $T_c = l/v$. Let T_c be equally partitioned into q time slots. Each slot duration is τ. The T_c can be presented as $T_c = [t_1, t_2 \ldots, t_q]$ and we have

$$T_c = q\tau \quad \text{and } \tau = T_c/q = l/(v * q)$$

Let notations e_0 and e_i denote the initial energy and remaining energy of each sensor s_i, respectively. Let D_i denote the data volume collected by sensor s_i and $D = \{D_1, \ldots, D_n\}$. Assume that each sensor has u different levels of transmission power.

Let T_i denote the time period that bus b falls into the communication range of sensor s_i. The period T_i consists of two subperiods: *GTS*(Guaranteed Time Slot) and *STS*(Shared Time Slot). The *GTS* refers to the time period that bus b only falls in the communication range of sensor s_i. On the contrary, the *STS* refers to the time period that bus b falls into the common communication range of sensor s_i and its neighboring sensors. The *STS* can be further partitioned into two parts: *Left STS* and *Right STS*, which denote the time slots shared with left or right neighbors, respectively. Fig. 1 depicts the considered scenario where seven sensors deployed in the road.

Let T_i^{GTS} and T_i^{STS} denote the time period of *GTS* and *STS* of sensor s_i, respectively. Let $t_{i,j}$ denote the j–th slots of time period T_i. Let $t_{i,j}^{GTS}$ and $t_{i,j}^{STS}$ denote the j-th time slots of T_i^{GTS} and T_i^{STS} of sensor s_i, respectively. We have,

$$T_i = T_i^{GTS} + T_i^{STS}$$

Let Boolean variables $\lambda_{i,j}^{GTS}$ and $\lambda_{i,j}^{STS}$ denote whether or not sensor s_i transmits data to bus in slots $t_{i,j}^{GTS}$ and $t_{i,j}^{STS}$, respectively, in the final schedule of sensor s_i. That is,

$$\lambda_{i,j}^{GTS} = \begin{cases} 1 & s_i \text{ transmits data to bus } b \text{ in } t_{i,j}^{GTS} \\ 0 & otherwise \end{cases}$$

$$\lambda_{i,j}^{STS} = \begin{cases} 1 & s_i \text{ transmits data to bus } b \text{ in } t_{i,j}^{STS} \\ 0 & otherwise \end{cases}$$

The total volume of transmitted data, denoted by $D_i^{transmitted}$, from sensor s_i to bus can be measured by

$$D_i^{transmitted} = \sum_{j=1}^{T_i^{GTS}} \lambda_{i,j}^{GTS} * r_{i,j} + \sum_{j=1}^{T_i^{STS}} \lambda_{i,j}^{STS} * r_{i,j}$$

The remaining data in the buffer of sensor s_i is obviously

$$D_i - D_i^{transmitted}$$

The main goal of this paper aims to completely transmit data from each sensor s_i to the bus. The following Exp. (1) reflects the main goal of this paper.

First Objective:

$$Min(\sum_{i=1}^{n}(D_i - D_i^{transmitted})) \tag{1}$$

In addition to achieving the primary goal that transmitting all data from sensors to the bus, another important goal is to prolong the newtwork lifetime of the given wireless sensor network. This means that we aim to maximize the lifetime of the sensor with minimal lifetime. The following calculates the energy consumption and lifetime of each sensor s_i. Let $d_{i,j}$ denotes the distance between sensor s_i and bus in the $t_{i,j}$. Let $p_{i,j}$ denotes the power level that sensor s_i uses to transmit data in $t_{i,j}$, and $r_{i,j}$ denote the transmission rate corresponding to $p_{i,j}$. Let $e_{i,j}^c$ denotes the energy consumption of sensor s_i for transmitting one bit from itself to the bus in $t_{i,j}$. It is obvious that the value of $e_{i,j}^c$ is highly related to values of $d_{i,j}$ and $p_{i,j}$. Let $E_{i,j}^c$ denotes energy consumption of sensor s_i in the j-th slots. That is,

$$E_{i,j}^c = (\lambda_{i,j}^{GTS} + \lambda_{i,j}^{STS}) * r_{i,j} * e_{i,j}^c$$

Let E_i^{GTS} and E_i^{STS} denote the energy consumptions during the time periods T_i^{GTS} and T_i^{STS} of sensor s_i, respectively. That is

$$E_i^{GTS} = \sum_{j=1}^{T_i^{GTS}} E_{i,j}^c$$

$$E_i^{STS} = \sum_{j=1}^{T_i^{STS}} E_{i,j}^c$$

The total energy consumption, denoted by E_i, of sensor s_i in each round is

$$E_i = E_i^{GTS} + E_i^{STS}$$

When the first goal that all data can be completely transmitted from sensors to the bus is achieved as shown in Exp. (1), the second goal of this paper aims to maximize the remaining energy of the sensor with minimal remaining energy. Let E denote the full battery energy of each sensor and $\tilde{E}_i$ denote the remaining energy of sensor s_i. Exp. (2) reflects the second goal of this paper.

Second Objective:

$$Max((\underset{s_i \in S}{Min}\, \tilde{E}_i = E - E_i)) \tag{2}$$

Some constraints which should be satisfied are given below when developing an algorithm for achieving goals given in Exps. (1) and (2). Let $p_{i,j}^{min}$ denotes the minimum power level used by sensor s_i to transmit data in $t_{i,j}$. To satisfy the *power constraint*, the value of $p_{i,j}^{min}$ should guarantee that the SNR at the bus side should be better than the predefined value of SNR threshold. Assume that each sensor uses the same power level to transmit data in one slot and let $p_{i,j}^{mds}$ denote the minimum detectable signal (MDS) power level used by sensor s_i to transmit data in $t_{i,j}$. Let $A_{i,j}$ and $A_{i,j}^{mds}$ denote the effective area of $p_{i,j}$ and $p_{i,j}^{mds}$, respectively. Let w denote the wave length of radio frequency.

1) POWER CONSTRAINT

$$p_{i,j}^{min} \geq Min\left(SNR_{th} = \frac{p_{i,j}}{p_{i,j}^{mds}} = \frac{d_{i,j}^2 * w^2}{A_{i,j} * A_{i,j}^{mds}}\right)$$
$$\text{for any } s_i, s_j \in S \tag{3}$$

The second constraint, called ***Valid Sharing Constraint***, asks that the number of used sharing slots by each sensor s_i should not be larger than the length of T_i^{STS}.

2) MAXIMAL SHARING CONSTRAINT

$$\sum_{j=1}^{T_i^{STS}} \lambda_{i,j}^{STS} \leq T_i^{STS} \tag{4}$$

This paper aims to transmit all data from each sensor to the bus and prolong the network lifetime. To achieve the goal given in Exp. (1) of Section 3, the proposed *STS Slots Bidding Phase* aims to maximize the utilizations of *STS* and *GTS* slots. In addition, the proposed *STS Slots and Power Level Adjustment Phase* further adjusts the allocation of sensor's *STS* slots and the power level to increase the throughput. Furthermore, the proposed mechanism initially adopts the lowest power level aiming to achieve the second goal given in Exp. (2) of Section 3. During the execution of the proposed *DBDC* algorithm, the constraints given in Section 3, including power constraint and neighbor sharing constraint are taken into consideration.

IV. THE PROPOSED *DBDC* ALGORITHM

This section presents the proposed Distributed Bus-based Data Collection (*DBDC*) algorithm, which aims to maximize the throughput and network lifetime. The algorithm mainly consists of four phases. In the first phase, each sensor will be aware of the bus arrival and calculate the contact time period between the bus and itself. Each sensor also calculates its initial power level. In the second phase, each sensor aims to determine the minimal working slots of the *GTS* such that all

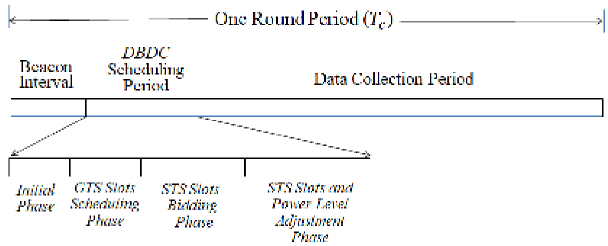


FIGURE 2. Each round T_c of ***DBDC*** consists of beacon interval, ***DBDC*** scheduling peroid and data collection period.

of its data can be transmitted to the bus. In the third phase, neighboring sensors can further bid time slots of *STS*, if they still have remaining data in their own buffer. Finally, in the last phase, each sensor further adjusts its power level, aiming to maximize the network throughput and network lifetime. The last phase will be applied by sensor s_i only if sensor s_i still has remaining data in buffer though all *GTS* and *STS* slots have been used by s_i. Fig. 2 depicts that each round T_c consists of the beacon interval, *DBDC* scheduling period and the data collection period. In the following, we present the details of each phase.

A. INITIAL PHASE

In this phase, each sensor will be aware of the bus arrival and calculate the contact time period between the bus and itself. Each sensor also calculates its initial power level. The bus will periodically broadcast its arrival and speed using a beacon message. The first sensor that detects the bus will wake up all the sensors. Then each sensor listens to the beacon, calculates contact time, including *GTS*, *LSTS* and *RSTS*, between the bus and itself. After that, each sensor exchanges the information of *GTS*, *LSTS* and *RSTS* with neighbors.

Let T_i^{GTS}, T_i^{LSTS} and T_i^{RSTS} denote the time periods of *GTS*, *Left STS* and *Right STS* of sensor s_i, respectively. Let Boolean variables $\lambda_{i,j}^{GTS}$, $\lambda_{i,j}^{LSTS}$ and $\lambda_{i,j}^{RSTS}$ denote whether or not sensor s_i transmits data to bus in slots $t_{i,j}^{GTS}$, $t_{i,j}^{LSTS}$ and $t_{i,j}^{RSTS}$, respectively, in the final schedule of sensor s_i. Let T_i^{GTS}, T_i^{LSTS} and T_i^{RSTS} be expressed by $T_i^{GTS} = [t_{i,1}^{GTS}, t_{i,\rho_i^{GTS}}^{GTS}]$, $T_i^{LSTS} = [t_{i,1}^{LSTS}, t_{i,\rho_i^{LSTS}}^{LSTS}]$ and $T_i^{RSTS} = [t_{i,1}^{RSTS}, t_{i,\rho_i^{RSTS}}^{RSTS}]$, where ρ_i^{GTS}, ρ_i^{LSTS} and ρ_i^{RSTS} denote the number of slots in T_i^{GTS}, T_i^{LSTS} and T_i^{RSTS}, respectively. Let p_k denote the transmission power level of each sensor and $P = \{p_1, \ldots, p_m\}$ denote the set of m possible power levels. Let $p_{i,j}^{threshold}$ denote the threshold of transmission power level of sensor s_i, which guarantees that the transmitted data can be safely received by bus in the j-th slot. Let $t_{i,j}^{GTS}$, $t_{i,j}^{LSTS}$ and $t_{i,j}^{RSTS}$ denote the j-th time slots of T_i^{GTS}, T_i^{LSTS} and T_i^{RSTS} of sensor s_i, respectively. Let $d_{i,j}^{GTS}$ denote the distance between the bus and sensor s_i in slot $t_{i,j}^{GTS}$. The value of $p_{i,j}^{threshold}$ can be determined by

$$p_{i,j}^{threshold} = p_{i,j}^{mds} * \frac{\left(d_{i,j}^{GTS}\right)^2 * w^2}{A_{i,j} * A_{i,j}^{mds}},$$

where $A_{i,j}$ and $A_{i,j}^{mds}$ present effective areas of $p_{i,j}$ and $p_{i,j}^{mds}$, respectively.

In this phase, each sensor s_i will determine its initial power $p_{i,j}^{init}$ for each slot $t_{i,j}$, where

$$p_{i,j}^{init} = \underset{p_k \in P}{Min}\left(p_k - p_{i,j}^{threshold}\right) \geq 0$$

Herein, we notice that the bus is moving during T_i, which causes that the distance between the bus and sensor s_i changes with time. This indicates that the value of $p_{i,j}^{threshold}$ will be changed in each $t_{i,j} \in T_i$.

B. GTS SLOTS SCHEDULING PHASE

In this phase, each sensor s_i aims to determine the minimal number of *GTS* slots for data transmission if its data can be completely received by bus. In case that sensor s_i can completely transmit its data to the bus and there are remaining slots in *GTS*, sensor s_i will stay in sleeping state in the remaining *GTS* slots and all slots in *STS* for conserving its energy. The release of *STS* slots can help neighbor increase its throughput or decrease its energy consumption. The details of *STS* scheduling will be presented in the next Phase. Let $r_{i,j}^{init}$ denote the transmission rate of slot $t_{i,j}^{GTS}$ if sensor s_i adopts $p_{i,j}^{init}$ as its transmission power. Let $D_i^{transmitted}$ denote the total amount of data which can be transmitted from s_i to bus b under the initial transmission power. We have

$$D_i^{transmitted} = \sum_{j=1}^{\rho_i^{GTS}} r_{i,j}^{init} \tag{5}$$

Recall that D_i denote the total data stored in buffer of sensor s_i. Let α_i be a Boolean variable representing whether or not the data of sensor s_i can be completely received by bus during *GTS*. That is,

$$\alpha_i = \begin{cases} 1 & D_i^{transmitted} \geq D_i \\ 0 & otherwise \end{cases}$$

The following proposes distributed scheduling strategy for allocating the *GTS* slots.

1) $\alpha_i = 1$

In this case, sensor s_i adopting the initial power can completely transmit its data to the bus. Therefore, sensor s_i would try to release some *GTS* slots for energy conservation. Let notation $\tilde{T}_i^{GTS}$ denote the time period of T_i^{GTS}, which will be allocated for data transmission for s_i. Recall that ρ_i^{GTS} denote the number of total slots in T_i^{GTS}. Let $\tilde{\rho}_i^{GTS}$ denote the number of slots in $\tilde{T}_i^{GTS}$. The value of $\tilde{\rho}_i^{GTS}$ will be the minimal value that satisfies Exp. (6).

$$\sum_{j=1}^{\tilde{\rho}_i^{GTS}-1} r_{i,j}^{init} < D_i \leq \sum_{j=1}^{\tilde{\rho}_i^{GTS}} r_{i,j}^{init} \tag{6}$$

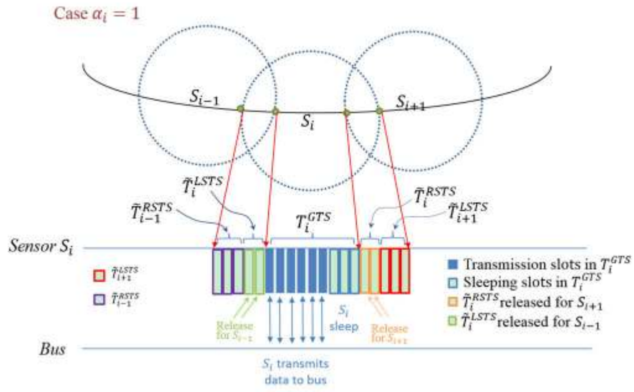


FIGURE 3. An example of case $\alpha_i = 1$.

As a result, sensor s_i has the following schedule.

$$\lambda_{i,j}^{GTS} = \begin{cases} 1 & if\ t_{i,j}^{GTS} \in \tilde{T}_i^{GTS} = [t_{i,1}^{GTS}, t_{i,\tilde{\rho}_i^{GTS}}^{GTS}] \\ 0 & otherwise \end{cases}$$

Fig. 3 depicts an example of case $\alpha_i = 1$. As shown in Fig. 3, sensor s_i is able to completely transmit its data to the bus during $\tilde{T}_i^{GTS}$ which are marked with solid blue color. Therefore, sensor s_i releases $\tilde{T}_i^{LSTS}$ and $\tilde{T}_i^{RSTS}$ which are marked with hollow green and hollow orange colors, to neighbors s_{i-1} and s_{i+1}, respectively.

2) $\alpha_i = 0$

In this case, sensor s_i can not tramsmit all data to the bus even if all slots in T_i^{GTS} have been allocated for data transmission. This also indicates that sensor s_i expects to have more slots in *STS* for transmitting the remaining data.

C. STS SLOTS BIDDING PHASE

In this phase, each sensor s_i executes the proposed bidding procedure for obtaining appreciate *STS* slots, aiming to transmit its remaining data, in case of $\alpha_i = 0$. For the purpose of energy conservation, sensor s_i will not consider to enlarge its initial power level. The remaining data only can be transmitted by bidding the *STS* slots from neighbors.

Similar to the definition of $\tilde{\rho}_i^{GTS}$, let $\tilde{\rho}_i^{LSTS}$ and $\tilde{\rho}_i^{RSTS}$ denote the numbers of slots which will be allocated to s_i in T_i^{LSTS} and T_i^{RSTS}, respectively. Since slots in *STS* are shared by s_i and some of its neighbors, sensor s_i evaluates the values of $\tilde{\rho}_i^{LSTS}$ and $\tilde{\rho}_i^{RSTS}$ according to the neighboring information. Let τ_i denote the weight of sensor s_i, whcih is determined based on its resource requirement, including remaindering ernergy and the size of buffered data. Recall that $\tilde{E}_i$ denotes the remaining energy of sensor s_i. We have,

$$\tau_i = \lambda \tilde{E}_i + (1 - \lambda) D_i.$$

where λ is a weight coefficient representing the important ratio between energy and data size. Then sensor s_i exchanges τ_i with neighbors and calculates its own *LSTS* and *RSTS* based on the following Equs. (7) and (8), respectively.

$$\tilde{\rho}_i^{LSTS} = \frac{\tau_i}{\tau_{i-1} + \tau_i} \tag{7}$$

$$\tilde{\rho}_i^{RSTS} = \frac{\tau_i}{\tau_i + \tau_{i+1}} \tag{8}$$

Similar to the definition of $\tilde{T}_i^{GTS}$, let $\tilde{T}_i^{LSTS}$ and $\tilde{T}_i^{RSTS}$, expressed by the forms $\tilde{T}_i^{LSTS} = [t_{i,1}^{LSTS}, t_{i,\tilde{\rho}_i^{LSTS}}^{LSTS}]$ and $\tilde{T}_i^{RSTS} = [t_{i,1}^{RSTS}, t_{i,\tilde{\rho}_i^{RSTS}}^{RSTS}]$, represent the allocated time periods to sensor s_i in *LSTS* and *RSTS*, respectively. Similar to the definition of $d_{i,j}^{GTS}$, let $d_{i,j}^{LSTS}$ and $d_{i,j}^{RSTS}$ denote the distances between the bus and sensor s_i in slots $t_{i,j}^{LSTS}$ and $t_{i,j}^{RSTS}$, respectively. In case that all slots of T_i^{GTS}, T_i^{LSTS} and T_i^{RSTS} have been allocated, the total data received by bus b can be presented by Equ. (9).

$$D_i^{transmitted} = \sum_{j=1}^{\tilde{\rho}_i^{GTS}} r_{i,j}^{init} + \sum_{j=1}^{\tilde{\rho}_i^{LSTS}} r_{i,j}^{init} + \sum_{j=1}^{\tilde{\rho}_i^{RSTS}} r_{i,j}^{init} \tag{9}$$

Let β_i be a Boolean variable representing whether or not the data of sensor s_i can be completely received during T_i. We have

$$\beta_i = \begin{cases} 1 & D_i^{transmitted} \geq D_i \\ 0 & otherwise \end{cases}$$

The following presents the bidding strategy for scheduling the *STS* slots.

1) $\beta_i = 1$

This implies that all data of s_i can be received by bus. For the purpose of energy conservation, sensor s_i can stay in sleeping mode, instead of communication mode, in some slots of *STS*. The following further calculates the number of *STS* slots which should be scheduled for data transmission. For receving all data from sensor s_i to bus, the solution of $(\tilde{\rho}_i^{LSTS}, \tilde{\rho}_i^{RSTS})$ should satisfy the following condition.

$$\sum_{j=1}^{\rho_i^{GTS}} r_{i,j}^{init} + \sum_{j=1}^{\tilde{\rho}_i^{LSTS}} r_{i,j}^{init} + \sum_{j=1}^{\tilde{\rho}_i^{RSTS}} r_{i,j}^{init} \geq D_i \tag{10}$$

To obtain the possible solution pair of $(\tilde{\rho}_i^{LSTS}, \tilde{\rho}_i^{RSTS})$, the following bidding operations OP1 or OP2 will be applied.

Bidding OP 1: Reduce T_i^{LSTS} by one slot
Bidding OP 2: Reduce T_i^{RSTS} by one slot

In case that Bidding OP1 is applied, condition CD 1 should be satisfied. To minimize the energy consumption of sensor s_i, OP1 can be repeated applied until condition CD 1 can not be satisfied. Finally, the minimal values of $\tilde{\rho}_i^{LSTS}$ and $\tilde{\rho}_i^{RSTS}$ can be derived.

CD 1:

$$\sum_{j=1}^{\rho_i^{GTS}} r_{i,j}^{init} + \sum_{j=1}^{\tilde{\rho}_i^{LSTS}-1} r_{i,j}^{init} + \sum_{j=1}^{\tilde{\rho}_i^{RSTS}} r_{i,j}^{init} < D_i \text{ and}$$
$$D_i \leq \sum_{j=1}^{\rho_i^{GTS}} r_{i,j}^{init} + \sum_{j=1}^{\tilde{\rho}_i^{LSTS}} r_{i,j}^{init} + \sum_{j=1}^{\tilde{\rho}_i^{RSTS}} r_{i,j}^{init}$$

On the other hand, sensor s_i may also apply Bidding OP 2 to minimize the energy consumption of sensor s_i. However, the following condition CD 2 should be satisfied.

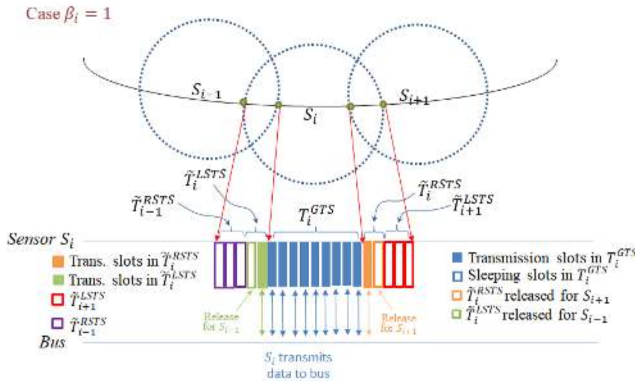


FIGURE 4. An example of case $\beta_i = 1$.

CD 2:

$$\sum_{j=1}^{\rho_i^{GTS}} r_{i,j}^{init} + \sum_{j=1}^{\tilde{\rho}_i^{LSTS}} r_{i,j}^{init} + \sum_{j=1}^{\tilde{\rho}_i^{RSTS}-1} r_{i,j}^{init} < D_i \text{ and}$$
$$D_i \leq \sum_{j=1}^{\rho_i^{GTS}} r_{i,j}^{init} + \sum_{j=1}^{\tilde{\rho}_i^{LSTS}} r_{i,j}^{init} + \sum_{j=1}^{\tilde{\rho}_i^{RSTS}} r_{i,j}^{init}$$

Similarly, the Bidding OP2 can be repeatedly applied until condition CD 2 is not satisfied and hence the minimal values of $\tilde{\rho}_i^{LSTS}$ and $\tilde{\rho}_i^{RSTS}$ can be derived.

Since sensor s_i has obtained the values of $\tilde{\rho}_i^{LSTS}$ and $\tilde{\rho}_i^{RSTS}$, the scheduling results of T_i^{LSTS} and T_i^{RSTS} can be simply achieved, which can significantly reduce the energy consumption of sensor s_i. The following presents the schedule of T_i^{LSTS} and T_i^{RSTS}. Sensor s_i will schedule its *STS* according to the following results.

$$\lambda_{i,j}^{GTS} = 1, \quad \text{for all } t_{i,j}^{GTS} \in \left[t_{i,1}^{GTS}, t_{i,\rho_i^{GTS}}^{GTS}\right] \tag{11}$$

$$\lambda_{i,j}^{LSTS} = \begin{cases} 1 & if\ t_{i,j}^{LSTS} \in [t_{i,1}^{LSTS}, t_{i,\tilde{\rho}_i^{LSTS}}^{LSTS}] \\ 0 & otherwise \end{cases} \tag{12}$$

$$\lambda_{i,j}^{RSTS} = \begin{cases} 1 & if\ t_{i,j}^{RSTS} \in [t_{i,1}^{RSTS}, t_{i,\tilde{\rho}_i^{RSTS}}^{RSTS}] \\ 0 & otherwise \end{cases} \tag{13}$$

Fig. 4 depicts an example of case $\beta_i = 1$. As shown in Fig. 4, sensor s_i is able to completely transmit its data to the bus during T_i^{GTS}, $\tilde{T}_i^{LSTS}$ and $\tilde{T}_i^{RSTS}$. The used T_i^{GTS} slots are marked with blue color while the used $\tilde{T}_i^{LSTS}$ and $\tilde{T}_i^{RSTS}$ are marked with solid green and orange colors, respectively. Hence the unused slots, which are marked with hollow green and hollow orange colors in $\tilde{T}_i^{LSTS}$ and $\tilde{T}_i^{RSTS}$, can be released to neighbors s_{i-1} and s_{i+1}, respectively.

2) $\beta_i = 0$

In this case, sensor s_i applies the initial power but can not transmit all data from itself to the bus even if it utilizes all slots in T_i^{GTS}, $\tilde{T}_i^{LSTS}$ and $\tilde{T}_i^{RSTS}$. For achieving the primary goal of transmitting all data from s_i to the bus, sensor s_i will perform the next phase.

D. STS SLOTS AND POWER LEVEL ADJUSTMENT PHASE

In this phase, each sensor s_i further adjusts its *STS* slots or power level to increase the throughput or lifetime. Sensor s_i which cannot completely transmit its data to the bus will perform the operations designed in this phase. That is, sensor s_i satisfies the condition $\beta_i = 0$. This phase aims to improve both the throughput and network lifetime by adjusting the resources of *STS* slots and remaining energy of sensor s_i. The following presents the desinging idea behind *DBDC*. Let s_j be one neighbor of sensor s_i. There are two ways for sensor s_i to transmit its remaining data and hence improve its throughput. One is to apply the bidding policy, trying to borrow some *STS* slots from neighbor s_j. The other is to increase its own power level, which also increases the transmission rate of s_i.

Let $\tilde{\omega}_i^{LSTS}$ and $\tilde{\omega}_i^{RSTS}$ denote the bidding slots released by right and left neighbors which satisifing the condition $\beta_i = 1$ in the last phase, respectively. Recall that $\tilde{\rho}_i^{LSTS}$ and $\tilde{\rho}_i^{RSTS}$ denote the number of allocated slots of s_i in T_i^{LSTS} and T_i^{RSTS}, respectively. We have

$$\tilde{\rho}_i^{LSTS} = \frac{\tau_i}{\tau_{i-1} + \tau_i} + \tilde{\omega}_i^{LSTS}$$
$$\tilde{\rho}_i^{RSTS} = \frac{\tau_i}{\tau_i + \tau_{i+1}} + \tilde{\omega}_i^{RSTS}$$

Since sensor s_i has borrowed all free slots of *STS* from neighbors, it will apply the procedure of the last phase to check if it can completely transmit all data to bus. In case of $\beta_i = 1$, it implies that sensor s_i obtains its final schedule as shown in Equs. (11)-(13)

On the contrary, if sensor s_i applies the initial power but still can not transmit all data from itself to the bus even if it utilizes all slots in *GTS* and all possible borrowed slots in *STS*, it will adopt energy adjusting policy. To balance the remaining energy among neighbors, sensor s_i can either enlarge its own energy power level or ask neighbors to enlarge their power levels.

Let γ_i be a Boolean variable representing whether or not the remaining energy of sensor s_i is larger than that of neighbor s_k. We have

$$\gamma_i = \begin{cases} 1 & \tilde{E}_i \geq \tilde{E}_k \\ 0 & otherwise \end{cases}$$

The following presents the energy adjusting strategy for scheduling the *STS* slots of sensor s_i.

1) $\gamma_i = 1$

In this case, the remaining energy of sensor s_i is more than that of sensor s_k. To balance the lifetimes of sensors s_i and s_k and transmit all data of s_i to the bus, sensor s_i will enlarge its own energy power level. To obtain the possible solution pair of ($\tilde{\rho}_i^{LSTS}$, $\tilde{\rho}_i^{RSTS}$), the following operation OP3 can be applied. When applying OP 3, sensors s_i should increase its own transmission power level firstly in *GTS* slots. In case that all *GTS* slots have been increased power but the buffered data are still not empty, then *LSTS* and *RSTS* slots would be applied the power adjustment operation.

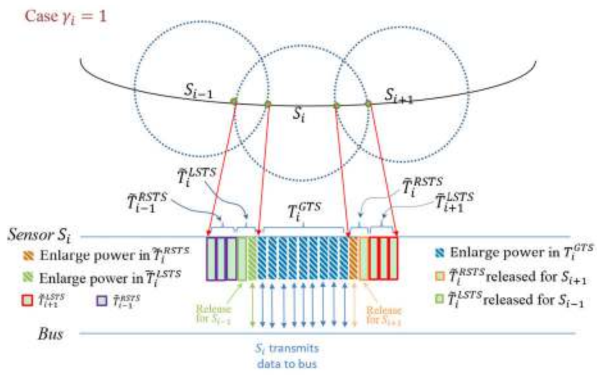


FIGURE 5. An example of case $\gamma_i = 1$.

OP 3: Increase power level $p_{i,j}$ at one slot in an order of *GTS*, *LSTS*, *RSTS*.

In case that OP 3 is applied, condition CD 3 and $\gamma_i = 1$ should be satisfied. To minimize the energy consumption of sensor s_i, OP 3 can be repeated applied until either condition CD 3 is satisfied or condition $\gamma_i = 1$ is not satisfied. Finally, the minimal power level of each slot in ρ_i^{GTS}, $\tilde{\rho}_i^{LSTS}$ and $\tilde{\rho}_i^{RSTS}$ can be derived.

CD 3:

$$\sum_{j=1}^{\rho_i^{GTS}} r_{i,j} + \sum_{j=1}^{\tilde{\rho}_i^{LSTS}} r_{i,j} + \sum_{j=1}^{\tilde{\rho}_i^{RSTS}} r_{i,j} \geq D_i$$

Fig. 5 depicts an example of case $\gamma_i = 1$. As shown in Fig. 5, sensor s_i increases its power level to completely transmit its data to the bus during T_i^{GTS}, $\tilde{T}_i^{LSTS}$ and $\tilde{T}_i^{RSTS}$. The used T_i^{GTS}, $\tilde{T}_i^{LSTS}$ and $\tilde{T}_i^{RSTS}$ slots are marked with shadow blue, shadow green and shadow orange colors, respectively. Since s_i enlarges the transmission power in these slots, the increase of transmission rate helps complete its data transmission earlier and hence the unused slots, which are marked with hollow green and hollow orange colors in $\tilde{T}_i^{LSTS}$ and $\tilde{T}_i^{RSTS}$, can be released to neighbors s_{i-1} and s_{i+1}, respectively.

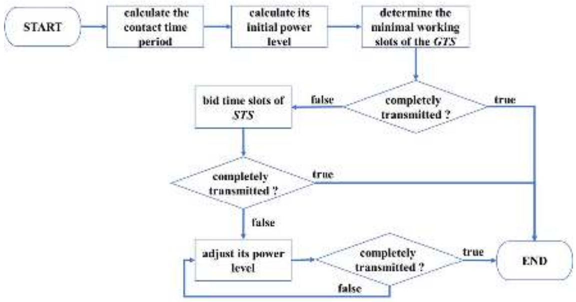
2) $\gamma_i = 0$

In this case, the remaining energy of sensor s_i is less than that of sensor s_k. To balance the lifetime of sensors s_i and s_k, sensor s_k will enlarge its own energy power level to release more *STS* slots for neighboring sensor s_i. After evaluation, sensor s_k informs s_i its *STS* usage and power level assignment such that sensor s_i can further borrow some *STS* slots form s_k and update its schedule ($\tilde{\rho}_i^{LSTS}$, $\tilde{\rho}_i^{RSTS}$).

For obtaining the possible solutions of ($\tilde{\rho}_k^{LSTS}$, $\tilde{\rho}_k^{RSTS}$), the following operations OP 4 or OP 5 can be applied. If senosr s_k is left neighbor of s_i, operation OP 4 is applied. On the contrary, operation OP 5 is applied.

OP 4: Increase power level $p_{k,j}$ at some slots (in an order of *GTS*, *LSTS*, *RSTS*) until one slot in T_k^{RSTS} can be released to T_i^{LSTS}.

OP 5: Increase power level $p_{k,j}$ at some slots (in an order of *GTS*, *LSTS*, *RSTS*) until one slot in T_k^{LSTS} can be released to T_i^{RSTS}.



(a) The workflow of each sensor for the proposed *DBDC*.

Function: ***Initial Phase***

/* determine initial power of each sensor */
$\lambda_{i,j}^{GTS}=\lambda_{i,j}^{LSTS}=\lambda_{i,j}^{RSTS}=0;$
$p_{i,j}^{threshold}=p_{i,j}^{mds}*\frac{\left(d_{i,j}^{GTS}\right)^2*w^2}{A_{i,j}*A_{i,j}^{mds}};$
$p_{i,j}^{init}=p_{i,j}^{threshold};$

(b) Initial Phase Function.

Function: ***GTS Slots Allocation Phase***

/* check whether or not the buffered data can be completely received during GTS */
$T_i^{GTS}=[t_{i,1}^{GTS},t_{i,\rho_i^{GTS}}^{GTS}];$
$D_i^{transmitted}=\sum_{j=1}^{\rho_i^{GTS}}r_{i,j}^{init};$
If $(D_i^{transmitted}<D_i)$
    Goto **Next Phase**;
Release one slot until $(D_i^{transmitted}<D_i)$ is satisfied
If $(t_{i,j}^{GTS}\in[t_{i,1}^{GTS},t_{i,\tilde{\rho}_i^{GTS}}^{GTS}])$? $(\lambda_{i,j}^{GTS}=1)$: $(\lambda_{i,j}^{GTS}=0)$;

(c) *GTS* Slots Allocation Phase Function.

Function: ***STS Slots Bidding Phase***

/* check whether the buffered data can be completely received during STS */
$T_i^{GTS}=[t_{i,1}^{GTS},t_{i,\rho_i^{GTS}}^{GTS}];$
$T_i^{LSTS}=[t_{i,1}^{LSTS},t_{i,\rho_i^{LSTS}}^{LSTS}];$
$T_i^{RSTS}=[t_{i,1}^{RSTS},t_{i,\rho_i^{RSTS}}^{RSTS}];$
$\tau_i=\lambda\tilde{E}_i+(1-\lambda)D_i,$
$\tilde{\rho}_i^{LSTS}=\frac{\tau_i}{\tau_{i-1}+\tau_i};$
$\tilde{\rho}_i^{RSTS}=\frac{\tau_i}{\tau_i+\tau_{i+1}}$
If $(D_i^{transmitted}<D_i)$
    Goto **Next Phase** ;
Release one slot until $(D_i^{transmitted}<D_i)$ is satisfied
Do Executing Bidding OP1 While (CD1 == true);
Do Executing Bidding OP2 While (CD2 == true);
$\lambda_{i,j}^{GTS}=1$
If $(t_{i,j}^{LSTS}\in[t_{i,1}^{LSTS},t_{i,\tilde{\rho}_i^{LSTS}}^{LSTS}])?(\lambda_{i,j}^{LSTS}=1):(\lambda_{i,j}^{LSTS}=0);$
If $(t_{i,j}^{RSTS}\in[t_{i,1}^{RSTS},t_{i,\tilde{\rho}_i^{RSTS}}^{RSTS}])?(\lambda_{i,j}^{RSTS}=1):(\lambda_{i,j}^{RSTS}=0);$

(d) *STS* Slots Bidding Phase Function.

Function: ***STS Slots and Power Level Adjustment Phase***

/* check whether or not the buffered data can be completely received while adopting polices for adjusting power level and biddng slots during GTS and STS */
$\tilde{\rho}_i^{LSTS}=\frac{\tau_i}{\tau_i+\tau_{i+1}}+\tilde{\omega}_i^{LSTS}$
$\tilde{\rho}_i^{RSTS}=\frac{\tau_i}{\tau_i+\tau_{i+1}}+\tilde{\omega}_i^{RSTS}$
If $(D_i^{transmitted}\geq D_i)$
    Exit;
If $(e_i\geq e_k)$? $(\gamma_i=1)$: $(\gamma_i=0)$;
Do /* $\gamma_i=1$ state block (1) */
    Do Executing Bidding OP3 While (CD3 == true));
    Do Executing Bidding OP4 While (CD4== true));
While(γ_i == 1);
Do /* $\gamma_i=0$ state block (2) */
    Do Executing Bidding OP5 While (CD5 == true));
    Do Executing Bidding OP6 While (CD6== true));
    Executing Bidding OP6;
While (γ_i == 0);
$\lambda_{i,j}^{GTS}=1$
If $(t_{i,j}^{RSTS}\in[t_{i,1}^{RSTS},t_{i,\tilde{\rho}_i^{RSTS}}^{RSTS}])$? $(\lambda_{i,j}^{RSTS}=1)$: $(\lambda_{i,j}^{RSTS}=0)$;
$\lambda_{k,j}^{GTS}=1$
If $(t_{k,j}^{LSTS}\in[t_{k,1}^{LSTS},t_{k,\tilde{\rho}_k^{LSTS}}^{LSTS}])$? $(\lambda_{k,j}^{LSTS}=1)$: $(\lambda_{k,j}^{LSTS}=0)$;
If $(t_{k,j}^{RSTS}\in[t_{k,1}^{RSTS},t_{k,\tilde{\rho}_k^{RSTS}}^{RSTS}])$? $(\lambda_{k,j}^{RSTS}=1)$: $(\lambda_{k,j}^{RSTS}=0)$;

(e) STS Slots and Power Level Adjustment Phase Function.

Algorithm: ***Distributed Bus-based Data Collection*** **(*DBDC*)**
Inputs:
A set of n static sensors $S=\{s_1,s_2,\ldots s_i,\ldots s_n\}$. The bus is labeled with b.
Output:
The energy consumption E_i and scheduling results $\lambda_{i,j}^{GTS}$, $\lambda_{i,j}^{LSTS}$ and $\lambda_{i,j}^{RSTS}$

/* determine initial power level */
call **Function *Initial Phase();***
/* evaluate whether or not data volume is enough during GTS slots allocation phase*/
call **Function *GTS Slots Allocation Phase();***
/* evaluate whether or not data volume is enough during STS slots biddding phase*/
call **Function *STS Slots Bidding Phase();***
/* evaluate whether data volume is enough after polices of adjusting power level and biding slots */
call **Function *STS Slots and Power Level Adjustment Phase()***

(f) The *DBDC* algorithm.

FIGURE 6. **The *DBDC* algorithm.**

In case that OP 4 is applied, condition CD 4 and $\gamma_i = 0$ should be satisfied. To maximize the network lifetime and minimize the energy consumption of sensor s_i, OP 4 can be repeated applied until either CD 4 is satisfied or $\gamma_i = 0$ is not satisfied. Finally, the minimal values of $\tilde{\rho}_i^{LSTS}$ and $\tilde{\rho}_i^{RSTS}$ can be derived.

CD 4:

$$\sum_{j=1}^{\rho_i^{GTS}} r_{i,j} + \sum_{j=1}^{\tilde{\rho}_i^{LSTS+1}} r_{i,j} + \sum_{j=1}^{\tilde{\rho}_i^{RSTS}} r_{i,j} \geq D_i$$

On the other hand, we may also apply OP 5 to minimize the energy consumption of sensor s_i. However, the following condition CD 5 should be satisfied.

CD 5:

$$\sum_{j=1}^{\rho_i^{GTS}} r_{i,j} + \sum_{j=1}^{\tilde{\rho}_i^{LSTS}} r_{i,j} + \sum_{j=1}^{\tilde{\rho}_i^{RSTS+1}} r_{i,j} \geq D_i$$

Herein, we notic that the the operations designed for cases $\gamma_i = 1$ and $\gamma_i = 0$ might be applied alternatively until all data of sensor s_i can be completely transmitted to bus. Finally, the schedules of sensor s_i can be obtained as follows.

$$\lambda_{i,j}^{GTS} = 1, \quad \text{for all } t_{i,j}^{GTS} \in \left[t_{i,1}^{GTS}, t_{i,\rho_i^{GTS}}^{GTS}\right]$$

$$\lambda_{i,j}^{LSTS} = \begin{cases} 1 & \text{if } t_{i,j}^{LSTS} \in [t_{i,1}^{LSTS}, t_{i,\tilde{\rho}_i^{LSTS}}^{LSTS}] \\ 0 & \text{otherwise} \end{cases}$$

$$\lambda_{i,j}^{RSTS} = \begin{cases} 1 & \text{if } t_{i,j}^{RSTS} \in [t_{i,1}^{RSTS}, t_{i,\tilde{\rho}_i^{RSTS}}^{RSTS}] \\ 0 & \text{otherwise} \end{cases}$$

Similarly, the schedules of sensor s_k can be obtained as follows.

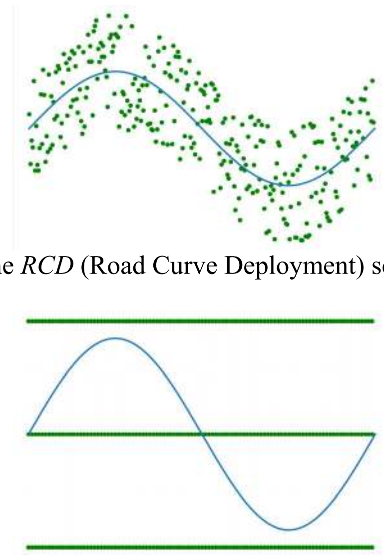
$$\lambda_{k,j}^{GTS} = 1, \quad \text{for all } t_{k,j}^{GTS} \in \left[t_{k,1}^{GTS}, t_{k,\rho_k^{GTS}}^{GTS}\right]$$

$$\lambda_{k,j}^{LSTS} = \begin{cases} 1 & \text{if } t_{k,j}^{LSTS} \in [t_{k,1}^{LSTS}, t_{k,\tilde{\rho}_k^{LSTS}}^{LSTS}] \\ 0 & \text{otherwise} \end{cases}$$

$$\lambda_{k,j}^{RSTS} = \begin{cases} 1 & \text{if } t_{k,j}^{RSTS} \in [t_{k,1}^{RSTS}, t_{k,\tilde{\rho}_k^{RSTS}}^{RSTS}] \\ 0 & \text{otherwise} \end{cases}$$

By applying the operations designed in ***STS and Power Level Adjustment Phase***, each sensor s_i tries its best to enlarge its power level if its remaining energy is larger than that of neighbor s_k. On the contrary, if the remaining energy of s_i is smaller than that of s_k, sensor s_i asks neighbor s_k to release *STS* slots to s_i by enlarging power level of s_k at some slots. As a result, the main purpose of maximizing the size of transmitted data of each sensor s_i can be achieved while the network lifetime between neighboring sensors also can be balanced. Hence the second purpose of prolonging the network lifetime can be also achieved in the proposed *DBDC*.

The following Fig. 6 summarizes the *DBDC* algorithm. Initially, each sensor calculates the contact time period between the bus and itself. Each sensor also calculates its initial power level. In the second phase, each sensor aims to determine the minimal working slots of the *GTS* such that all of its data can be transmitted to the bus. In the third phase, neighboring sensors can further bid time slots of *STS*, if they still have remaining data in their own buffer. Finally, in the last phase, each sensor further adjusts its power level, aiming to maximize the network throughput and network lifetime.



(a) The *RCD* (Road Curve Deployment) scenario.

(b) The *SLD* (Straight Line Deployment) scenario.

FIGURE 7. Two scenarios considered in the experiments.

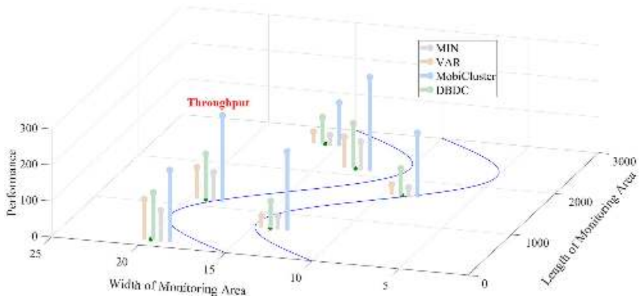
V. PERFORMANCE EVALUATION

This section studies the performance improvements of the proposed Distributed Bus-based Data Collection (*DBDC*) mechanism against the other Minimum Power Level Transmission(*MIN*), Maximum Power Level Transmssion(*MAX*), Variable Power Level Transmission(*VAR*), and *MobiCluster* proposed in [16] mechanisms. The *MIN* mechanism transmits data using minimum detectable signal (*MDS*) power level while the *MAX* mechanism transmits data using maximum power level. In addition, the *VAR* transmits data using power level which is randomly determined between minimum and maximum power levels. In *MobiCluster* algorithm, an adequate number of nodes are selected as rendezvous nodes which are responsible for relaying data of other sensors to the bus. The performances of the four compared algorithms are evaluated in terms of throughput, network lifetime, data loss rate, slot utilization, transmission completion rate as well as fairness index of transmission data. Two senarios are applied in the experiments. As shown in Fig. 7(a), the first scenario, called Road Curve Deployment (*RCD*), randomly deploys sensors marked with green ink along the road. As shown in Fig. 7(b), the second scenario, called Straight Line Deployment (*SLD*), regularly deploys sensors along straight lines. The MATLAB simulator is used as the simulation tool.

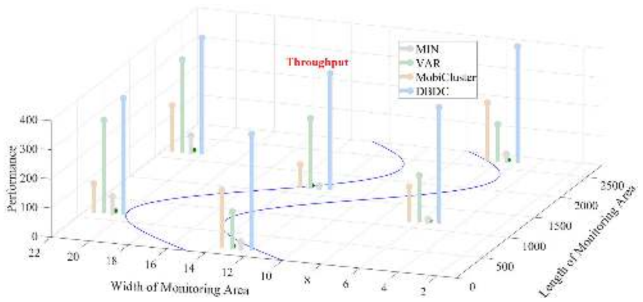
The following illustrates the arranged simulation environment. As shown in Table 1, the area size is 3000m* 30 m. The wireless sensor network is connected and the number

TABLE 1. The simulation settings.

Node deployment	Random
Given Region	3000 m * 30m
The number of sensor nodes	100 – 300
Sensor node transmittion Range	20 – 40 (m)
Sensing data volume	200 – 400 (units)
Transmission power level	1-8
Data Rate	4 – 400 (units)
Energy	10000 – 20000(units)



(a) Throughputs of selected 6 sensors in *RCD* scenario.

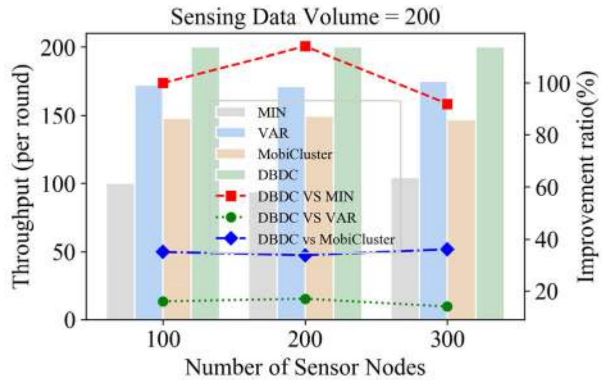


(b) Throughputs of selected 6 sensors in *SLD* scenario.

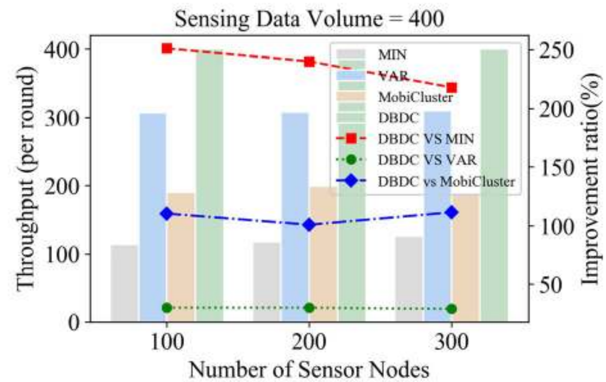
FIGURE 8. Performance snapshots of selected 6 sensor nodes. Four algorithms are compared in terms of throughput in *RCD* and *SLD* scenarios.

of deployed sensors is ranging from 100 to 300. The initial energy of each sensor node is ranging from 10000 to 20000 units. The sensing and communication ranges of each sensor node are set at 20m and 40m, respectively. Each sensor node periodically generates data packets and sends them to the bus in each round. Each node is assumed to be aware of its own and the bus's locations.

Fig. 8 depicts the snapshot of throughputs of selected six sensors by applying the four compared algorithms. The number of sensor nodes is 300. The experiment randomly selects six sensor nodes (mark with green color) and then observe their throughputs. As shown in Fig. 8(a), the *DBDC* and *MAX* have higher throughputs than *MIN* in all cases in the *RCD* scenario. This occurs because that sensors applying *DBDC* reserve *STS* slots to the neighboring sensors that have larger data size. Another reason is that sensors applying *DBDC* dynamically enlarge their power levels when the *GTS* and *STS* are exhausted. The throughputs of *VAR* are high in two sensors but are low in the other four sensors. This occurs because that the data can be transmitted in a high data rate



(a)Sensing data volume=200 in *RCD* scenario.

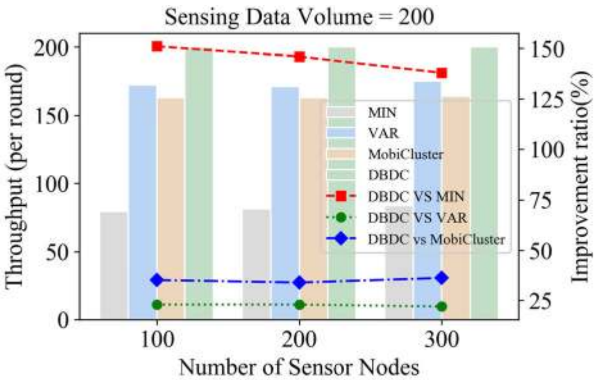


(b)Sensing data volume=400 in scenario *RCD.*

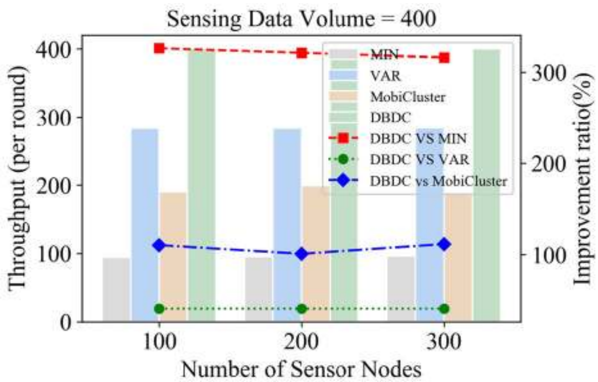
FIGURE 9. The comparisons of four algorithms in terms of the throughput in scenario *RCD*.

by the two nodes closed to the road. Fig. 8(b) depicts similar results. The *MIN* has poor performance because that sensors use minimal power level. When the offered data volume is large, the slots in *GTS* and *STS* are not enough for transmitting all data, resulting in low throughput.

Fig. 9 generally compares the throughputs of the four algorithms in scenario *RCD*. The number of deployed sensors varies ranging from 100 to 300. Figs. 9(a) and 9(b) offer the data volumes 200 units and 400 units, respectively, in each round. As shown in Fig. 9(a), the proposed *DBDC* achieves better performance than *VAR*, *MIN* and *MobiCluster* in all cases. The improvements of *DBDC*, as compared with *MIN*, *VAR and MobiCluster*, vary ranging from 92% to 114%, from 14% to 17% and from 33% to 36%, respectively. This occurs because of several reasons. First, each sensor applying *DBDC* initially allocates *GTS* slots to transmit data and reserves *STS* slots to its neighbors. This helps neighboring sensors transmit more sensing data when their *GTS* are not enough. Second, each sensor applying *DBDC* further enlarges its power level when the allocated *STS* is still not enough. This helps sensors increase their throughputs. As a result, the throughput of *DBDC* is higher than those of *VAR*, *MIN* and *MobiCluster*. Fig. 9(b) depicts similar performance result that *DBDC* outperforms *MobiCluster* in term of throughput.



(a) Sensing data volume=200 in *SLD* scenario.

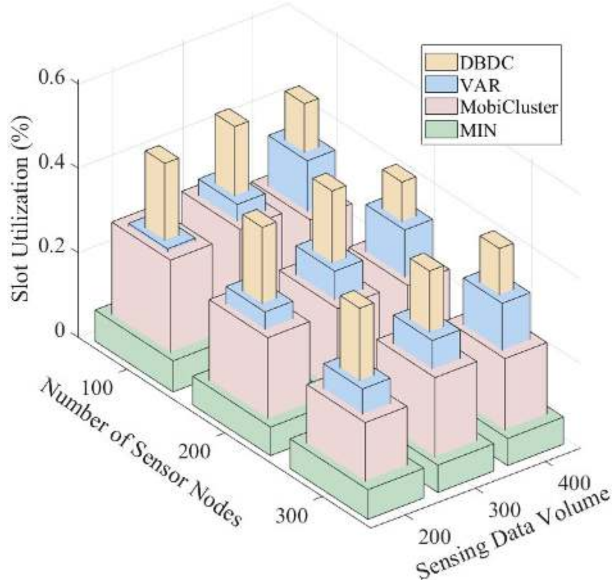


(b)Sensing data volume=400 in *SLD* scenario.

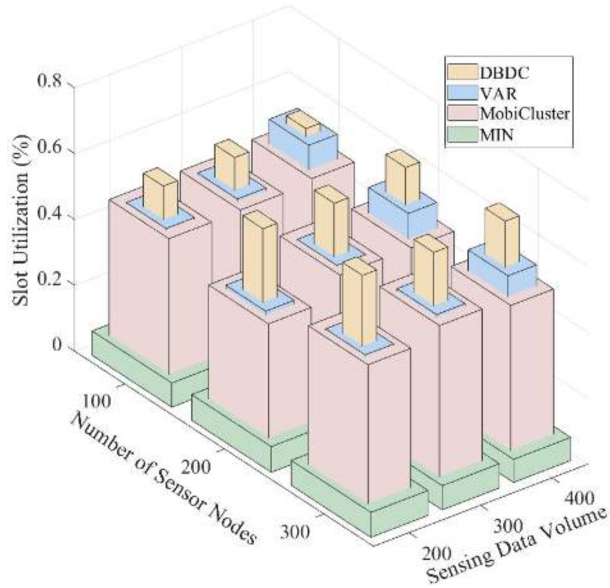
FIGURE 10. The comparisons of four algorithms in terms of the throughput in *SLD* scenario.

The performance improvements of *DBDC*, as compared with *MIN*, *VAR and MobiCluster*, vary ranging from 218% to 251%, from 29% to 30% and from 101% to 112%, respectively.

Fig. 10 further compares the throughputs of four algorithms in scenario *SLD*. The number of deployed sensors varies ranging from 100 to 300. Figs. 10(a) and 10(b) offer the data volumes 200 units and 400 units, respectively, in each round. The proposed *DBDC* has better performance than *MIN, VAR* and *MobiCluster* in all cases. The improvements of *DBDC*, as compared with *MIN, VAR* and *MobiCluster*, vary ranging from 138% to 151%, from 22% to 23% and from 33% to 36%, respectively, as shown in Fig. 10(a). The performance improvements of *DBDC*, as compared with *MIN, VAR* and *MobiCluster*, vary ranging 316% to 326%, from 40% to 41%, and from 101% to 112%, respectively, as shown in Fig. 10(b). Herein, we notice that the improvements in Fig. 10(a) is smaller than that in Fig. 10(b). This occurs because that the offered data volume in Fig. 10(a) is smaller than that in Fig. 10(b). In case of small data volume, some sensors can successfully transmit their data in *GTS* and *STS*, even though they apply *MobiCluster*. However, when the offered traffics grow, *GTS* and *STS* of *MobiCluster* are not enough for use, resulting in low throughputs.



(a) Comparison of slot utilization in *RCD* scenario



(b)Comparison of slot utilization in *SLD* scenario.

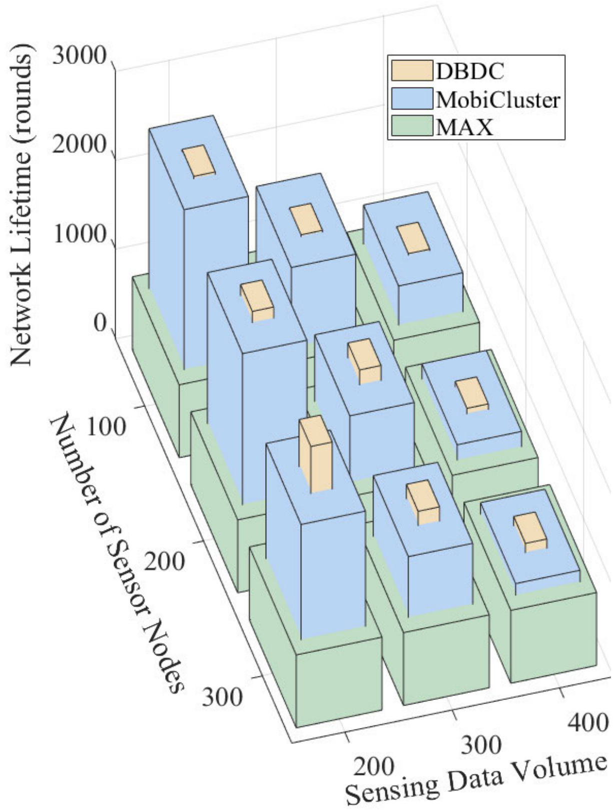
FIGURE 11. The comparisons of four algorithms in terms of slot utilization in *RCD* and *SLD* scenarios.

Fig. 11 compares the performances of *DBDC*, *VAR*, *MobiCluster* and *MAX* in terms of slot utilization. The sensing data volume varies ranging from 200 units to 400 units in each round, and the number of deployed sensors varies ranging from 100 to 300. Figs. 11(a) and 11(b) apply the *RCD* and *SLD* scenarios, respectively. The slot utilization indicates the ratio of the used slots to all allocated slots. In comparison, the proposed *DBDC* achieves better performance than *MobiCluster* in all cases, as shown in both Figs. 11(a) and 11(b). This occurs because that the proposed *DBDC* calculates the number of slots to be used and tries to transmit all sensing data using lower power levels to prolong network lifetime. In *GTS Slots Allocation Phase*, the *DBDC* only allocates *GTS* slots and gives *STS* slots to the neighbors with larger data volume. In *STS Slots Allocation Phase*, the *DBDC* utilizes the *GTS* and *STS* slots under the policy that the data can be completely transmitted by adopting as low as possible power level. As a result, the *DBDC* achieves high performance in terms of slot utilization. On the contrary, the *MAX* always adopts maximal power level and therefore the data can be transmitted in a high data rate. As a result, the slot utilization of *MAX* is low. Besides, the *MobiCluster* and VAR don't pay attention to the scheduling of shared slots, leading to lower slot utilization, as compared with *DBDC*.

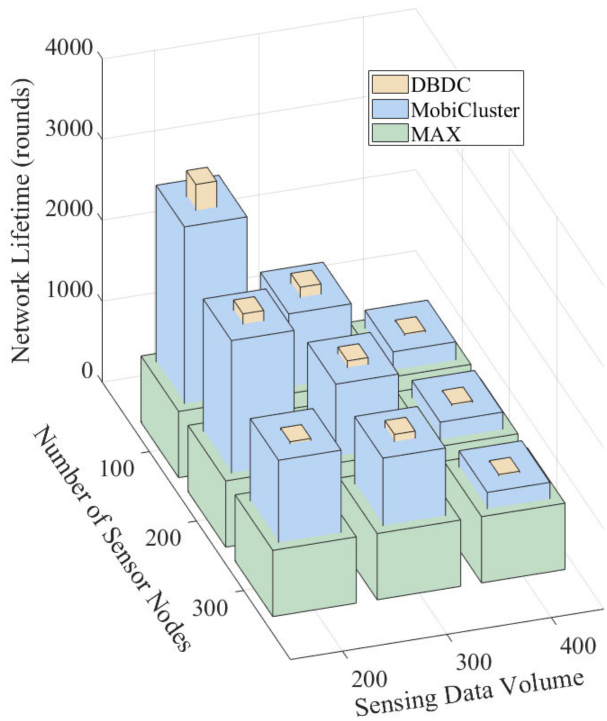
Fig. 12 compares the performances of *DBDC*, *MAX* and *MobiCluster* in terms of network lifetime. The sensing data volume varies ranging from 200 units to 400 units, and the number of deployed sensors varies ranging from 100 to 300. Figs. 12(a) and 12(b) apply the *RCD* and *SLD* scenarios, respectively. In general, the network lifetime increases with the number of sensors and decreases with the data volumes. This occurs because that more sensor nodes share the limited communication time and hence each sensor can obtain fewer slots for data transmission, increasing the remaining energy of each sensor. In comparison, the proposed *DBDC* achieves better performance than *MobiCluster* in all cases, as shown in Figs. 12(a) and 12(b). This occurs because that the proposed *DBDC* transmits data in the proper power level, prolonging the network lifetime. On the contrary, the *MAX* transmits data using the highest transmission power level, leading to shorter network lifetime. In additions, the *MobiCluster* transmits data in a multi-hops manner, leading to shorter network lifetime.

In the proposed *DBDC*, each sensor executing the bidding procedure need to exchange control messages with neighbors. In fact, the neighbors only need to exchange messages two times. In the first time, the exchanged message includes sensor ID, remaining energy, power level as well as the remaining resource of *STS* and *GTS*. In the second time, the exchange message includes the decision of the applied power level and the scheduled result. Consequently, the control overheads are very small, as compared with the data transmitted to the bus. Table 2 further gives the impact of control overheads on the network lifetime. As shown in Table 2, the *DBDC* represents the network lifetime which does not consider the control overheads while the *DBDC-H* represents the network lifetime which considers the control overheads. The network lifetime of *DBDC-H* only is reduced with a small value (1% in average), as compared with that of *DBDC*.

Fig. 13 generally compares the data loss ratios of the three algorithms in scenario *RCD*. The number of deployed sensors varies ranging from 100 to 300. Figs. 13(a) and 13(b) offer the data volumes 200 units and 400 units, respectively, in each round. In general, the data loss ratio increases with the number of sensors and the data volumes. This occurs because that each sensor has fewer opportunities to transmit data if there are more sensors sharing the limited slots. In comparison, the proposed *DBDC* has better performance than *MIN*, *VAR* and *MobiCluster* in all cases. The data loss



(a) Comparison of network lifetime in *RCD* scenario.



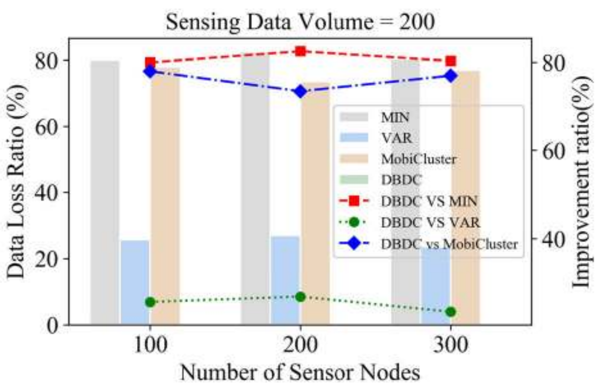
(b) Comparison of network lifetime in *SLD* scenario.

FIGURE 12. The comparisons of three algorithms in terms of the network lifetime of the WSNs in *RCD* and *SLD* scenarios.

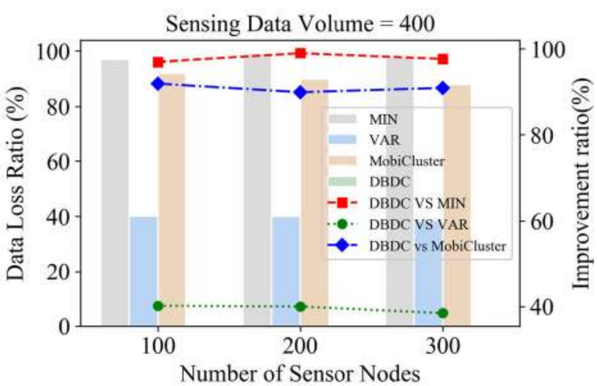
ratio of *DBDC*, as compared with *MIN*, *VAR* and *MobiCluster*, has an improvement ranging from 80% to 83%, from

TABLE 2. **The performance between *DBDC* and *DBDC-H* in term of network lifetime in *RCD* scenario.**

Network Lifetime (rounds)	Data Volume	Number of Sensor Nodes		
		100	200	300
DBDC	200	2631	2631	2631
	300	1724	1723	1652
	400	1262	1041	1041
DBDC-H	200	2630	2630	2630
	300	1723	1722	1651
	400	1261	1040	1040



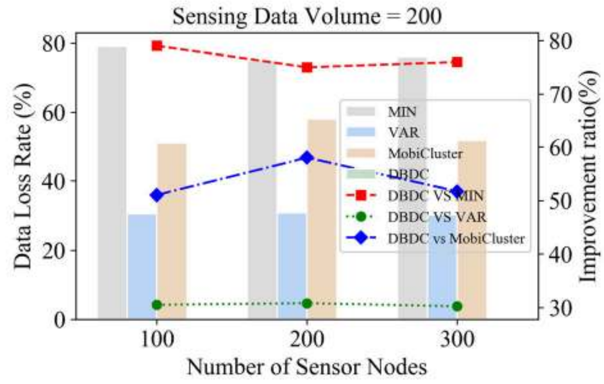
(a) Sensing data volume=200 in *RCD* scenario,



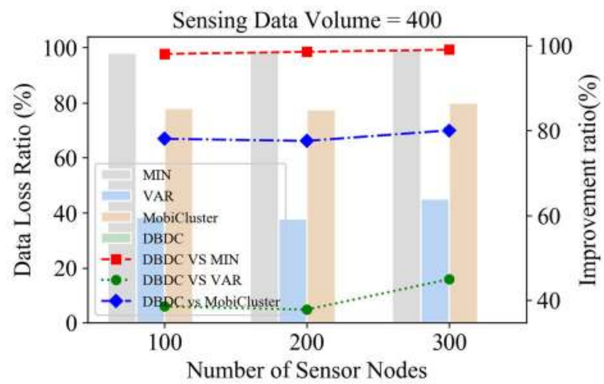
(b) Sensing data volume=400 in scenario RCD

FIGURE 13. **The comparisons of four algorithms in terms of the data loss ratio in *RCD* scenario.**

23% to 27% and from 73% to 78%, respectively, as shown in Fig. 13(a). This occurs because of several reasons. First, the *DBDC* initially allocates *GTS* to transmit data and reserves *STS* to its neighbor in *GTS Slots Allocation Phase*. This helps neighboring sensors transmit more sensing data in the allocated *STS* slots. Second, the *DBDC* further enlarges power level in *Scheduling and Power Adjusting Phase* when the allocated *STS* is still not enough. This helps sensors increase their throughputs. As a result, the data loss ratio of *DBDC* is smaller than those of *MIN*,*VAR* and *MobiCluster*. Fig. 13(b) depicts similar performance result that *DBDC* outperforms *MIN*, *VAR* and *MobiCluster* in term of data loss ratio.



(a) Sensing data volume=200 in scenario *SLD*.



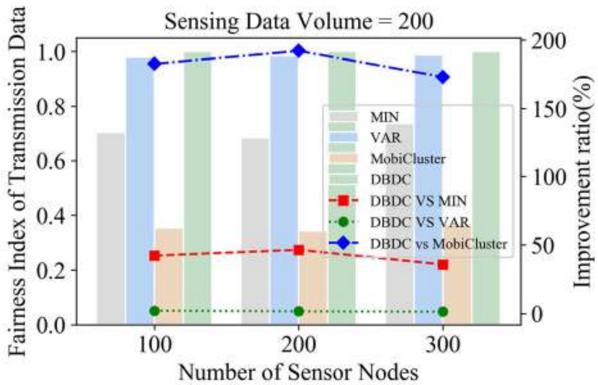
(b)Sensing data volume=400 in scenario *SLD*.

FIGURE 14. **The comparisons of four algorithms in terms of the data loss ratio in *SLD* scenario.**

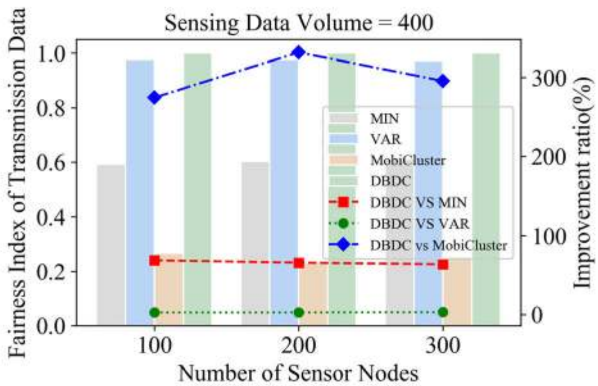
The performance of *DBDC*, as compared with *MIN*, *VAR* and *MobiCluster*, has an improvement ranging from 97% to 99%, from 38% to 41% and from 90% to 92%, respectively.

Fig. 14 further compares the data loss ratios of three algorithms in scenario *SLD*. The number of deployed sensors varies ranging from 100 to 300. Figs. 14(a) and 14(b) offer the data volumes 200 units and 400 units, respectively, in each round. The proposed *DBDC* has better performance than *MIN*, *VAR* and *MobiCluster* in all cases. The performance of *DBDC*, as compared with *MIN*, *VAR* and *MobiCluster*, has improvements ranging from 75% to 79%, from 30% to 31% and from 51% to 58%, respectively, as shown in Fig. 14(a). Moreover, the performance of *DBDC*, as compared with *MIN*, *VAR* and *MobiCluster*, has improvements ranging from 98% to 99%, from 37% to 45% and from 77% to 80%, respectively, as shown in Fig. 14(b). The improvements in Fig. 14(b) is larger than that in Fig. 14(a). This occurs because that the offered data volume in Fig. 14(b) is larger than that in Fig. 14(a). In case of small data volume, some sensors can successfully transmit their data in *GTS* and *STS*, even though they apply *MobiCluster*. However, when the offered traffics grow, the *GTS* and *STS* of *MobiCluster* are not enough, leading to high data loss ratio.

Fig. 15 generally compares the fairness indices of the three algorithms in scenario *RCD*. The number of deployed sensors

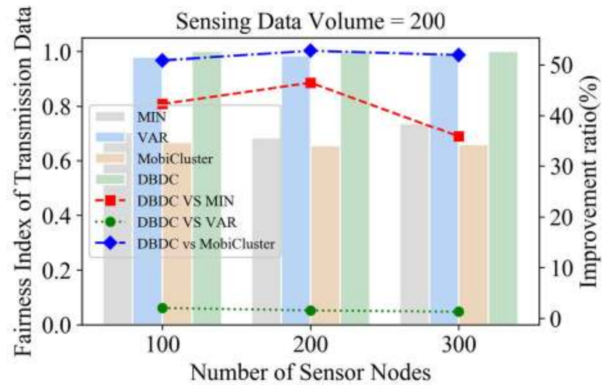


(a) Sensing data volume=200 in *RCD* scenario.

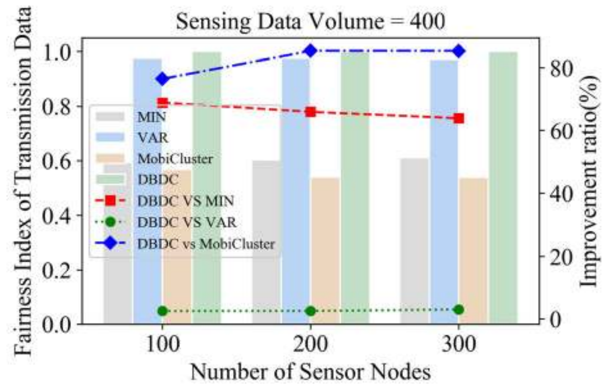


(b) Sensing data volume=400 in *RCD* scenario.

FIGURE 15. The comparisons of four algorithms in terms of fairness index on data transmission in *RCD* scenario.



(a) Sensing data volume=200 in *SLD* scenario.



(b) Sensing data volume=400 in *SLD* scenario.

FIGURE 16. The comparisons of four algorithms in terms of fairness index of data transmission in *SLD* scenario.

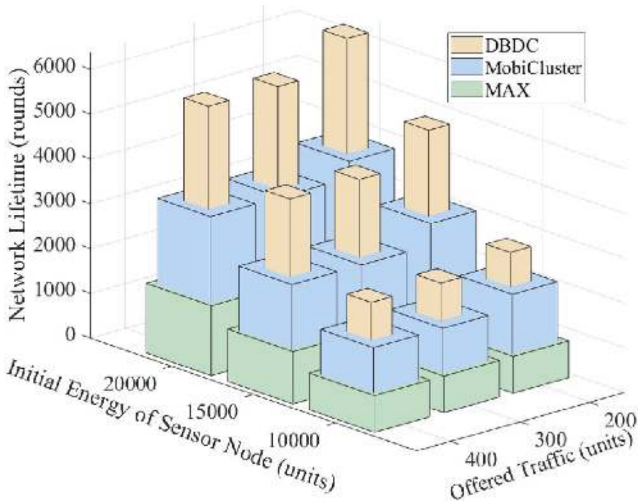
varies ranging from 100 to 300. Figs. 15(a) and 15(b) offer the data volumes 200 units and 400 units, respectively, in each round. The fairness index of data transmission among sensors is measured by Jain's Fairness Index, as shown in Equ. (14).

$$Fairness\ Index = \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n * \sum_{i=1}^{n} x_i^2}, \tag{14}$$

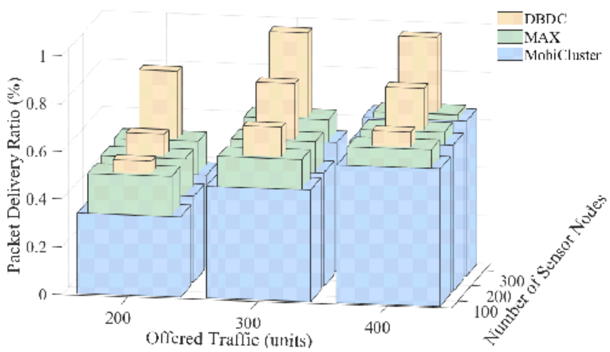
where x_i denotes the data volume transmitted by sensor node s_i. As shown in Fig. 15(a), the proposed *DBDC* achieves better performance than algorithms *MIN*, *VAR* and *MobiCluster* in all cases. The improvements of *DBDC*, as compared with *MIN*, *VAR* and *MobiCluster*, vary ranging from 35% to 47%, from 1% to 3% and from 173% to 193%, respectively. This occurs because that each sensor applying *DBDC* initially allocates slots in *GTS* to transmit data and reserves *STS* slots to its neighbors. In addition, each sensor applying *DBDC* further enlarges its power level to improve its own throughput. This helps sensors transmit the sensing data as fair as possible. Fig. 15(b) depicts similar performance result that *DBDC* outperforms *MIN*, *VAR* and *MobiCluster* in term of fairness index of transmission data. The performance improvements of *DBDC*, as compared with *MIN*, *VAR* and *MobiCluster*, vary ranging vary ranging from 63% to 69%, from 2% to 3% and from 274% to 332%, respectively.

Fig. 16 further compares the fairness indices of data transmission of three algorithms in scenario *SLD*. The number of deployed sensors varies ranging from 100 to 300. Figs. 16(a) and 16(b) offer the data volumes 200 units and 400 units, respectively, in each round. The improvements of *DBDC*, as compared with *MIN*, *VAR* and *MobiCluster*, vary ranging from 35% to 47%, from 1% to 3% and from 50% to 53%, respectively, as shown in Fig. 16(a). It is notable that the improvements of *DBDC* in Fig. 16(b) are larger than those in Fig. 16(a). This occurs because that the number of offered data to each sensor in Fig. 16(a) is smaller than that in Fig. 16(b). It is observed that some sensors having small data volume can successfully transmit their data in *GTS* and *STS*. This phenomenon can be found in *MobiCluster*. As a result, sensors in Fig. 16(a) have higher fairness index value, as compared with Fig. 16(b).

Fig. 17(a) compares the performances of *DBDC*, *MAX* and *MobiCluster* in terms of network lifetime. The initial energy of sensor node varies ranging from 10000 units to 20000 units. The offered traffic varies ranging from 200 units to 400 units. The *RCD* scenario is applied. In general, three algorithms have similar trend that the network lifetime is increased with the initial energy of sensor node but is decreased with the offered traffic. This occurs because that



(a) The comparisons of three algorithms in terms of the network lifetime of the WSNs using *RCD* scenario.



(b) The comparisons of three algorithms in terms of the packet delivery ratio using *RCD* scenario.

FIGURE 17. The comparisons of three algorithms in *RCD*scenario.

the higher initial energy can support more rounds of packet transmission. When the offered traffic is low and smaller than 300, the network lifetime is decreased with the offered traffic. This occurs because that more data transmitted from each sensor to the bus will consume more energy of each sensor, reducing the network lifetime. However, when the traffic is larger than 300, all the remaining *STS* and *GTS* slots have been used. As a result, the network lifetime keeps with a constant value even through the offered traffic is increased. In comparison, the proposed *DBDC* achieves better performance than *MobiCluster* and *MAX* in all cases. This occurs because *DBDC* initially adopts the lowest power level that can completely transmit data in *GTS* and *STS* slots.

Fig. 17(b) further compares the performances of *DBDC*, *MAX* and *MobiCluster* in terms of packet delivery ratio. The offered traffic varies ranging from 200 units to 400 units while the number of sensor nodes varies ranging from 100 to 300. The *RCD* scenario is applied. In general, three algorithms have similar trend that the packet delivery ratio is increased with the offered traffic when the offered traffic is low but keeps with a constant value when the offered traffic is larger than 300 units. This occurs because that the *STS* and *GTS* slots still have unused time slots when the offered traffic is low. However, when the offered traffic is more than 300 units, all *STS* and *GTS* slots have been used. In comparison, the proposed the proposed *DBDC* achieves better performance than *MobiCluster* and *MAX* in all cases. This occurs because sensors applying the proposed *DBDC* can obtain proper *STS* slots according to its traffic load. In addition, the power control mechanism also helps sensors increase the packet delivery ratio.

VI. CONCLUSION

Data collection is an important issue in wireless sensor networks. In recent years, most studies developed centralized algorithm to consider the controllable mobile sink for data collection from sensors, aiming to reduce the data forwarding load of static sensors. This paper develops distributed mechanism which considers the uncontrollable bus as mobile sink and investigates the slot scheduling and power adjusting mechanism, aiming to prolong the network lifetime while the goals of high throughput and low data lose rate can be achieved. A four-phase scheduling mechanism, called *DBDC*, is proposed, which aims to maximize the throughput and network lifetime. Applying the proposed *DBDC*, each sensor locally evaluates its data volume, negotiates with its neighbors, and schedule its *GTS* and *STS* slots in the way that all data can be transmitted to bus as more as possible while the network lifetime of each sensor can be balanced. Performance evaluations depicted that the proposed *DBDC* outperforms existing mechanisms in term of throughput, network lifetime and fairness index. The proposed *DBDC* generally outperforms the compared *MIN*, *VAR* and *MobiCluster*, in terms of network lifetime, throughput and traffic delivery ratio. The performance improvements of *DBDC* in term of throughput, as compared with *MIN*, *VAR* and *MobiCluster* in scenarios *RCD* and *SLD*, vary ranging from 29% to 251% and from 40% to 326%, respectively. In addition, the performances of *DBDC*, *MAX* and *MobiCluster* in terms of network lifetime, as compared with *MIN*, *VAR* and *MobiCluster* in scenarios *RCD* and *SLD*, vary ranging from 1% to 221% and from 1% to 303%, respectively.

REFERENCES

[1] C. I. Weng, C. Y. Chang, C. Y. Hsiao, C. T. Chang, and H. B. Chen, "On-supporting energy balanced k-barrier coverage in wireless sensor networks," *IEEE Access*, vol. 6, pp. 13261–13274, Jan. 2018.

[2] C. Y. Chang, Y.-W. Kuo, P. Xu, and H. Chen, "Monitoring quality guaranteed barrier coverage mechanism for traffic counting in wireless sensor networks," *IEEE Access*, vol. 6, pp. 30778–30792, May 2018.

[3] Z. Sun, X. Xing, B. Song, Y. Nie, and H. Shao, "Mobile intelligent computing in Internet of Things: An optimized data gathering method based on compressive sensing," *IEEE Access*, vol. 7, pp. 66110–66122, May 2019.

[4] S. Gao, H. Zhang, and S. Das, "Efficient data collection in wireless sensor networks with path-constrained mobile sinks," *IEEE Trans. Mobile Comput.*, vol. 10, no. 4, pp. 592–608, Apr. 2011.

[5] O. D. Incel, A. Ghosh, B. Krishnamachari, and K. Chintalapudi, "Fast data collection in tree-based wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 11, no. 1, pp. 86–99, Jan. 2012.

[6] S. Chen, M. Huang, S. Tang, and Y. Wang, "Capacity of data collection in arbitrary wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 1, pp. 52–60, Jan. 2012.

[7] M. Zhao and Y. Yang, "Bounded relay hop mobile data gathering in wireless sensor networks," *IEEE Trans. Comput.*, vol. 61, no. 2, pp. 265–277, Feb. 2012.
[8] S. Ji, R. Beyah, and Z. Cai, "Snapshot and continuous data collection in probabilistic wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 13, no. 3, pp. 626–637, Mar. 2014.
[9] D. Kim, R. N. Uma, B. H. Abay, W. Wu, W. Wang, and A. O. Tokuta, "Minimum latency multiple data MULE trajectory planning in wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 13, no. 4, pp. 838–851, Apr. 2014.
[10] R. Velmani and B. Kaarthick, "An efficient cluster-tree based data collection scheme for large mobile wireless sensor networks," *IEEE Sensors J.*, vol. 15, no. 4, pp. 2377–2390, Apr. 2015.
[11] A. Mehrabi and K. Kim, "Maximizing data collection throughput on a path in energy harvesting sensor networks using a mobile sink," *IEEE Trans. Mobile Comput.*, vol. 15, no. 3, pp. 690–704, Mar. 2016.
[12] W. Wen, S. Zhao, C. Shang, and C.-Y. Chang, "EAPC: Energy-aware path construction for data collection using mobile sink in wireless sensor networks," *IEEE Sensors J.*, vol. 18, no. 2, pp. 890–901, Jan. 2018.
[13] H. Salarian, K.-W. Chin, and F. Naghdy, "An energy-efficient mobile-sink path selection strategy for wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 63, no. 5, pp. 2407–2419, Jun. 2014.
[14] A. W. Khan, A. H. Abdullah, M. A. Razzaque, and J. I. Bangash, "VGDRA: A virtual grid-based dynamic routes adjustment scheme for mobile sink-based wireless sensor networks," *IEEE Sensors J.*, vol. 15, no. 1, pp. 526–534, Jan. 2015.
[15] L.-M. K. Ang, J. K. P. Seng, and A. M. Zungeru, "Optimizing energy consumption for big data collection in large-scale wireless sensor networks with mobile collectors," *IEEE Syst. J.*, vol. 12, no. 1, pp. 616–626, Mar. 2018.
[16] R. Deng, S. He, and J. Chen, "An online algorithm for data collection by multiple sinks in wireless-sensor networks," *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 1, pp. 93–104, Jun. 2018.
[17] M. Ma, Y. Yang, and M. Zhao, "Tour planning for mobile data-gathering mechanisms in wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 62, no. 4, pp. 1472–1483, May 2013.
[18] X. He, S. Liu, G. Yang, and N. Xiong, "Achieving efficient data collection in heterogeneous sensing WSNs," *IEEE Access*, vol. 6, pp. 63187–63199, Oct. 2018.
[19] M. Di Francesco, S. K. Das, and G. Anastasi, "Data collection in wireless sensor networks with mobile elements: A survey," *ACM Trans. Sensor Netw.*, vol. 8, no. 1, pp. 1–31, Aug. 2011.
[20] C. Konstantopoulos, G. Pantziou, D. Gavalas, A. Mpitziopoulos, and B. Mamalis, "A rendezvous-based approach enabling energy-efficient sensory data collection with mobile sinks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 5, pp. 809–817, May 2012.
[21] M. Ma and Y. Yang, "Data gathering in wireless sensor networks with mobile collectors," in *Proc. IEEE Int. Symp. Parallel Distrib. Process.*, Apr. 2008, pp. 1–9.
[22] R. Sugihara and R. K. Gupta, "Optimal speed control of mobile node for data collection in sensor networks," *IEEE Trans. Mobile Comput.*, vol. 9, no. 1, pp. 127–139, Jan. 2010.
[23] A. A. Somasundara, A. Ramamoorthy, and M. B. Srivastava, "Mobile element scheduling with dynamic deadlines," *IEEE Trans. Mobile Comput.*, vol. 6, no. 4, pp. 395–410, Apr. 2007.
[24] K. Almi'ani, A. Viglas, and L. Libman, "Energy-efficient data gathering with tour length-constrained mobile elements in wireless sensor networks," in *Proc. IEEE Local Comput. Netw. Conf.*, Oct. 2010, pp. 582–589.
[25] X. Ren, W. Liang, and W. Xu, "Data collection maximization in renewable sensor networks via time-slot scheduling," *IEEE Trans. Comput.*, vol. 64, no. 7, pp. 1870–1883, Jul. 2015.
[26] C. Wu, Y. Liu, F. Wu, W. Fan, and B. Tang, "Graph-based data gathering scheme in WSNs with a mobility-constrained mobile sink," *IEEE Access.*, vol. 5, pp. 19463–19477, Aug. 2017.
[27] K. Shah, M. Di Francesco, G. Anastasi, and M. Kumar, "A framework for resource-aware data accumulation in sparse wireless sensor networks," *Comput. Commun.*, vol. 34, no. 17, pp. 2094–2103, Nov. 2011.
[28] A. Chakrabarti, A. Sabharwal, and B. Aazhang, "Communication power optimization in a sensor network with a path-constrained mobile observer," *ACM Trans. Sensor Netw.*, vol. 2, no. 3, pp. 297–324, Aug. 2006.

CHIH-YUNG CHANG (M'05) received the Ph.D. degree in computer science and information engineering from National Central University, Taiwan, in 1995. He is currently a Full Professor with the Department of Computer Science and Information Engineering, Tamkang University, New Taipei, Taiwan. His current research interests include the Internet of Things, wireless sensor networks, artificial Intelligence, and deep learning. He has served as an Associate Guest Editor for several SCI-indexed journals, including the *International Journal of Ad Hoc and Ubiquitous Computing* (IJAHUC), from 2011 to 2014, the *International Journal of Distributed Sensor Networks* (IJDSN), from 2012 to 2014, the IET *Communications*, in 2011, *Telecommunication Systems* (TS), in 2010, the *Journal of Information Science and Engineering* (JISE), in 2008, and the *Journal of Internet Technology* (JIT), from 2004 to 2008.

CHUNG-CHIH LIN received the B.S. and M.S. degrees from National Central University, Taiwan, in 1998 and 2002, respectively, all in computer science and information engineering. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Information Engineering, Tamkang University, New Taipei, Taiwan. He is currently a Lecturer with the Department of Computer Science and Information Engineering, Tamkang University. His current research interests include wireless sensor networks and the Internet of Things.

CUIJUAN SHANG received the B.S. degree in biomedical engineering from Zhengzhou University, Zhengzhou, China, in 2008, and the M.S. degree in biomedical engineering from Shandong University, Jinan, China, in 2011. She is currently pursuing the Ph.D. degree with the Department of Computer Science and Information Engineering, Tamkang University, New Taipei, Taiwan. She is currently a Lecturer with the School of Computer and Information Engineering, Chuzhou University, Chuzhou, Anhui, China. Her current research interests include wireless sensor networks and healthcare.

I-HSIUNG CHANG received the Ph.D. degree in education from the National Chengchi University, Taiwan, in 2012. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Information Engineering, Tamkang University, Taiwan. His current research interests include the joint of education and information technologies in the AI-IoT, robot, and big data.

DIPTENDU SINHA ROY received the Ph.D. degree in engineering from the Birla Institute of Technology, Mesra, India, in 2010. He had served the Department of Computer Science and Engineering, National Institute of Science and Technology, Berhampur, India. In 2016, he joined the Department of Computer Science and Engineering, National Institute of Technology (NIT) Meghalaya, India, as an Associate Professor, where he has been the Chair of the Department of Computer Science and Engineering, since 2017. His current research interests include software reliability, distributed and cloud computing, and the IoT, specifically applications of artificial intelligence/machine learning for smart integrated systems. Dr. Roy is a member of the IEEE Computer Society.